KU LEUVEN
FACULTY OF ECONOMICS AND BUSINESS

Reconciling Risk and Return: an Application Combining Momentum and Risk-based Strategies Using Machine Learning Techniques

Pieter Decat
Sebastiaan Jacobs

Thesis submitted to obtain the degree

MASTER OF BUSINESS (AND INFORMATION SYSTEMS) ENGINEERING
Data Science and Business Analytics

Promotor: Prof. Dr. Bart Baesens
Supervisor: Toon Vanderschueren

Academic year: 2021-2022

Reconciling Risk and Return: an Application Combining Momentum and Risk-based Strategies Using Machine Learning Techniques

May 18, 2022

Pieter Decat & Sebastiaan Jacobs

Abstract

In bear markets, momentum strategies have traditionally outperformed variance-minimizing strategies from a returns perspective thanks to their faster recovery, but have come at the cost of larger drawdowns, called momentum crashes. Momentum-based investment strategies have proven to be a persistent way of outperforming market indices across different time horizons and market environments. Simultaneously, research on risk-minimizing strategies has explored new ways of capturing investors' perceived risk to replace traditional measures like volatility. The iVaR framework minimizes perceived risk, calculated as the average drawdown over the investment horizon. We propose a novel approach that integrates the return-maximizing momentum strategy into the risk-minimizing iVaR framework in order to reconcile both strategies. We do this by using a broad range of machine learning (ML) models to classify assets as winners/losers in a momentum portfolio. This portfolio is then integrated into the iVaR optimization framework, either by requiring a minimum exposure towards momentum winners in the resulting portfolio (*constraint-based approach*), or by adding a second term to the objective function that minimizes the drawdown relative to the momentum portfolio (*objective-based approach*). We validate our methodology empirically and find that both the constraint- and objective-based approaches of our mixed strategy portfolios outperform not only the market, but also single-strategy momentum and iVaR portfolios. This is true across the majority of ML models, with LSTM giving the best results. Our findings indicate that a mixed strategy of momentum and iVaR successfully reconciles both goals of minimizing drawdowns and maximizing returns, and that machine learning improves momentum portfolio performance.

Contents

Chapter 1

Introduction

Trading off risk and return is every investor's bread and butter. Although these two measures are overall positively correlated (Drummen & Zimmermann, 1992), finding a way to simultaneously optimize for both measures (minimizing risk while maximizing return) would lead to a superior overall effect. Traditionally, Sharpe ratios have been the default yardstick used to evaluate the risk-reward trade-off. However, the measure is prone to manipulation by over-promising or under-achieving fund managers (Goetzmann et al., 2007) and cannot tell up from down (Sortino & Price, 1994), to name only two of its shortcomings. Innovative ways of modelling risk that take human perceptions of risk into account solve one side of the equation. Ensuring maximal returns while doing so solves the other.

In portfolio optimization, several factors can be used to generate investment strategies, including momentum, value, size, quality and minimum-volatility. On one end of the spectrum, momentum strategies usually result in high returns, but come at the cost of high variance (risk). More specifically, the risk of so-called momentum crashes, which usually occur in market panic states and are therefore contemporaneous with market rebounds, tend to wipe out much of the accumulated gain. Momentum strategies' fatal collapse during the Great Financial Crisis (GFC) are a testament to their strong inherent risk during downturns (Daniel & Moskowitz, 2013). However, their strong outperformance during periods of post-crisis recovery are tantalizing, with the recent recovery in markets starting from April 2020 providing a textbook example (Guobužaitė & Teresienė, 2021). On the other end of the spectrum, minimum-volatility strategies provide low risk, but fall far behind momentum portfolios, especially when markets are recovering, as the post-covid market recovery has illustrated (Wigglesworth, 2021).

Therefore, capturing the strong returns that momentum strategies provide while allowing for a human-centric measure of risk forces us to move beyond the generic factor investment strategies. iVaR portfolio optimization, developed by Belgian fintech company InvestSuite [1], draws its inspiration from minimum-volatility strategies. However, instead of optimizing a Sharpe ratio, it will aim to minimize perceived risk, measured as the cumulative difference between the high-water mark of the portfolio and its current position.

Both momentum portfolio optimization and iVaR could in their own way be called naive strategies. On the one hand, momentum-based strategies are considered naive as they look to outperform by exploiting market anomalies, without taking into account any form of risk. On the other hand, iVaR is naive in the sense that the algorithm selects a combination of assets that will yield a minimization of the iVaR measure, without taking into account the return characteristics of this combination.

This thesis proposes to combine both approaches, with the aim of optimizing the risk-reward trade-off in a non-traditional way. Instead of optimizing Sharpe ratios by tweaking one of the factor investing strategies, the aim will be to dynamically tilt between the risk-minimizing iVaR framework, which allows to capture perceived risk, and a returns-maximizing momentum strategy, based on the difference in performance between the two strategies. Furthermore, this thesis will argue for the use of Machine Learning (ML) techniques to construct momentum portfolios, in contrast to the traditional approach in which momentum portfolios were constructed based on an incrementally improving combination of model features. We will empirically evaluate the performance of ML techniques in comparison to traditional linear techniques, both in terms of standalone classification performance, but also their performance implemented in a backtested portfolio.

The COVID-19 crisis brought the world to a halt in a way not seen before, and with that also made investors run for the exit, causing a significant crash in all major global markets. Soon after, the world again found itself in crisis with Russia going to war against Ukraine and prices continuing to rise due to a combination of central bank policy and global supply chain strains. These crises provide an interesting and relevant way to apply our research findings and demonstrate how our method is able to reduce a momentum crash through iVaR's drawdown minimization.

First, chapter 2 will give an overview of the existing literature on (i) momentum portfolio strategies, (ii) iVaR and (iii) machine learning for portfolio optimization. Then, the research questions will be outlined in chapter 3. Thirdly, a discussion will follow in chapter 4 on the data and methodology that are used for generating the optimal momentum portfolio, implementing it in the iVaR framework and backtesting the results. Results and discussion are detailed in chapter 5. Lastly, we draw conclusions and provide suggestions for future research in chapter 6.

[1] https://www.investsuite.com/

Chapter 2

Literature review

The past decades of financial literature on portfolio optimization have focused on different aspects of both return and risk characteristics. Research has been centered around how to construct portfolios that minimize risk, maximize returns, or optimize for the ratio of both.

On the one hand, an essential challenge in the last century of financial literature has been to understand, correctly measure and then minimize risk. The notion of "risk" of a financial asset started in 1952, when Markowitz (1952) introduced the variance of an asset's return as the measure of risk in his Modern Portfolio Theory (MPT). Since Markowitz, many alternatives for risk measurement have been introduced, such as Value at Risk (VaR), Conditional Value at Risk (CVaR) (Artzner et al., 1999), Conditional Drawdown at Risk (CDaR) (Chekhlov et al., 2003) and other measures of drawdown. InvestSuite introduced their own measure of risk, called iVaR, which minimizes what investors actually perceive as risk, namely long and deep drawdowns.

Next to risk minimization, returns maximization is the second essential driver of portfolio optimization. One of the seminal strategies in the literature is momentum, which has been recognized as the most persistent market anomaly. It gives the opportunity to consistently make profits based on the mispricing of assets. However, it appears that there has been much less academic literature focused on include momentum as a determining factor in constructing portfolios to improve the financial performance of minimum-variance portfolios. For that reason, the challenge within the current academic landscape is to boost investor returns, possibly using a momentum-related factor, without making large concessions in the perceived risk of an investor, as measured by iVaR.

This literature review will dig deeper into (i) optimal portfolio construction, (ii) momentum-based investment strategies and (iii) measuring portfolio risk, specifically using iVaR.

Finally, a fourth subsection will look at how machine learning techniques can be used to address this remaining gap in the academic literature that combines the return optimization of momentum strategies with the risk optimization of iVaR in constructing an optimal portfolio.

2.1 Selecting the optimal portfolio

The question of how to allocate wealth to a specific set of assets is one that has been widely researched over a long period of time. Two of the factors that are most recognized for determining the asset allocation within a portfolio, are expected return and risk. Markowitz (1952) developed the idea that one can create portfolios by selecting those assets that optimize the trade-off between risk and return, which is considered to be the foundation of Modern Portfolio Theory (MPT). Markowitz introduced the concept of an efficient frontier, which consists of portfolios of assets that either minimize variance for a given level of expected returns, or maximize expected returns for a given level of variance.

A large variety of studies building on these original insights concerning MPT have been conducted since the 2000s (see Appendix A). However, the central idea of MPT has remained over time: to construct portfolios that optimize the risk and return preferences of specific investors.

2.2 Momentum: the most persistent market anomaly

"Momentum is an amazing thing when it is working in your favour." (Simon Mignolet)

Momentum of financial assets is the phenomenon where asset prices that have risen in a well-defined historical period are likely to keep rising, whereas asset prices that have fallen are likely to keep falling. From a mathematical perspective, this means that returns of financial assets, when observed within a well-defined time frame, are autocorrelated, as a stock's past returns are expected to be related to its future returns.

The existence of momentum contradicts the Efficient Market Hypothesis (EMH) (Fama, 1960), which states that all relevant information should be reflected in the share price and that only new information should move asset prices, implying that historic price movement should not impact future returns. Therefore, working from the assumption that markets are efficient, momentum is said to be a market anomaly that could be attributed to the irrationality of market participants, which is in sharp contrast with the condition that all market participants are rational in efficient markets.

Momentum in the context of the prices of financial assets is known as one of the most persistent market anomalies that has been proven to exist across many different asset classes, geographies, and time horizons (Hurst et al., 2017; Baz et al., 2015). Momentum-based investment strategies for financial assets are generally known to take advantage of the sign and (possibly) magnitude of historical developments of these asset prices, based on the observation that a recent trend in these prices is likely to continue within a given time frame.

2.2.1 A brief history of momentum

Despite the availability of data on share prices, the profitability of momentum-based investment strategies was not covered in academic research until the '90s. Jegadeesh & Titman (1993) were the first to demonstrate the profitability of momentum strategies for a 3- to 12-month holding period between 1965 and 1989, even when accounting for systematic risk.

Since then, many different momentum-based investment strategies have been evaluated across different markets and time periods. For example, Rouwenhorst (1998) confirmed this notion of momentum across an internationally diversified equity portfolio for the medium term. Asness et al. (2013) confirm previous findings by demonstrating consistent momentum return premia across 8 markets and asset classes. Cao et al. (2020) confirm the existence of momentum in the Chinese stock markets by empirically proving the existence of behavioural biases in the historical performance of share prices in terms of the distinction between "winners" and "losers", and extreme and mediocre-performing stocks.

Jegadeesh & Titman (2001) evaluated the possible explanations for the above-mentioned profitability of momentum strategies as documented by Jegadeesh & Titman (1993). In doing so, they address the issues raised by Conrad & Kaul (1998) who argue that momentum exists in the cross-sectional differences between the expected returns of assets rather than patterns in the time-series data of returns. Hong & Stein (1999) attempted to explain the existence of momentum by arguing that if there exists any type of under-reaction due to the slow diffusion of new information amongst investors in financial assets, there must be an over-reaction in the longer run as trend-chasers take advantage of this arbitrage opportunity.

In the literature, two sets of momentum-based strategies are considered, namely cross-sectional momentum strategies and time-series momentum strategies (Lim et al., 2019). The first are based on exploiting the relative price developments of winning vs. losing assets by buying these assets that experienced a relatively stronger positive price development ("winners"), whilst selling those assets that experienced a relatively weaker price development ("losers") (Jegadeesh & Titman, 1993). This can be done by dividing a group of assets in terms of their performance, and then selecting a specific proportion of these assets to be in the group of winners or losers, e.g., by taking the top and bottom $n\%$ respectively. The second set of strategies is based on the idea that a trend in the price of an asset is likely to continue if it complies with specific conditions (Moskowitz et al., 2012). In contrast with the cross-sectional momentum-based strategies, times-series momentum-based strategies do not build on the idea of dividing a group of assets into "winners" and "losers", but rather look for continuation of the price development of a certain asset in a certain direction.

Cooper et al. (2004) were the first to look at momentum from a time-series angle, when they argued that the profits of momentum strategies are related to the state of the market. Related to this paper, Wang & Xu (2010) have proven that recent market volatility in combination with the market state predict momentum profits within a time-series context. Moskowitz et al. (2012) introduced time-series momentum (TSM), a new type of momentum which looks at an individual asset's past performance. They find that across a range of asset classes, there is a clear persistence in prices from a 1- to 12-month period in the 25 years prior to the study.

Recently, variations on the traditional cross-sectional momentum and TSM investment strategies have proven to be profitable. For example, Papailias et al. (2021) introduce a new type of momentum-based investment strategy called return signal momentum, which comes down to basing investment decisions on the probability of the signs of past returns. They show that this strategy outperforms time-series momentum and other benchmark strategies in terms of higher returns, lower drawdowns and higher Sharpe ratios. Moreover, they argue that predicting the sign of return forecasts, which comes down to a binary prediction, could be less prone to overfitting as opposed to predicting a continuous return forecast. In a different study, Baltas & Kosowski (2020) show that the implementation of a novel strategy which involves pairwise signed correlations leads to a more efficient volatility estimation and price trend detection. This, in turn, results in a reduction in portfolio turnover of about one third without a significant loss in expected returns.

2.2.2 Why momentum matters

In two recent studies conducted to evaluate the predictive relevance of large sets of firm characteristics to asset pricing, momentum is shown to be one of the only remaining factors that holds nontrivial predictive power. As shown by Green et al. (2017), only 2 of the 94 firm characteristics in Fama-MacBeth regressions, which are used to estimate parameters for asset pricing models such as CAPM, have been independent determinants of U.S. monthly stock returns between 2003 and 2014. Freyberger et al. (2020) further demonstrate that only 9 to 16 out of 62 firm characteristics provide incremental information for expected returns. For both studies, momentum is one of them.

In the context of the use of machine learning (ML) in asset pricing, momentum is also one of the main drivers of return predictions. For example, using deep feed-forward neural networks based on a set of firm characteristics to predict the cross-section of stock returns can offer attractive risk-adjusted returns compared to a linear benchmark. Of the 68 firm characteristics examined in a study, momentum appears to be among the main drivers of return predictions (Messmer, 2017). In their study concerning the use of ML within the context of asset pricing, Gu et al. (2020) demonstrate that momentum, together with liquidity and volatility, is one of the dominant predictive signals to all machine learning methods. Han (2021) further builds on this idea that momentum is a strong predictive factor when ML is applied to make predictive models based on firm characteristics. He does this by introducing a novel cross-section predictive model based on ML, which he proves to outperform off-the-shelf ML models. In contrast with other studies that use a single moment strategy for a range of assets, Han develops a Deep Momentum model that yields an investing strategy at the individual stock level based on the nonlinear information from existing asset-related features.

Simple linear regressions and non-linear models used for factor prediction in asset pricing both suggest that momentum as a market anomaly is as relevant as ever. Recent applications of ML in the field of asset pricing show promising results for the combination of momentum-based investment strategies and ML, which we will expand on at a later stage.

2.2.3 Momentum of groups of financial assets

There is an important distinction between the momentum of an individual asset vs. the momentum of a group of assets. According to Moskowitz & Grinblatt (1999), much of the excess returns of individual assets selected based on momentum-based strategies are attributable to industry momentum. They demonstrate that strategies based on this industry momentum appear to be highly profitable. Industry momentum refers to the classification of assets according to their industry and the observation of the momentum effect at an industry level rather than the individual level. Building on this observation, O'Neal (2000) confirmed the industry momentum effect by applying momentum-based investment strategies on top-performing sector mutual funds over the period of 1989 to 1999. Andreu et al. (2013) demonstrate the existence of industry momentum by making the cross-section of the winners and losers within the space of industry ETFs. Li et al. (2019) confirm the possibility of extracting industry momentum by testing a value-weighted long-short strategy based on ETF momentum.

Moreover, by investing in a group of assets rather than in individual assets, it seems reasonable to believe that we can reach a higher level of diversification, thereby improving our risk-adjusted returns. ETFs are financial instruments that approximate these groups of assets, and are often grouped based on constituent characteristics, such as industry, region or asset type.

2.2.4 Limitations of current momentum-based investment strategies

The literature identifies 3 major types of drawbacks, namely momentum crashes, dependency of market state, and market cross-section dispersion negatively correlated to subsequent momentum premium.

Firstly, even though momentum-based investment strategies appear to be very promising when it comes to the design of portfolios that optimize (risk-adjusted) returns, momentum-based portfolios can suffer from major drawbacks that are known as momentum crashes. Moskowitz et al. (2012) describe these so-called momentum crashes as infrequent and persistent strings of negative returns that follow market declines when market volatility is high. They show that over a sample of US equities from 1927 to 2013, momentum crashes are a key and robust feature of momentum-based strategies. They argue that when poor market conditions ameliorate, following a multi-year market drawdown and in periods of high volatility, the losing assets are being shorted in large volumes, resulting in a momentum crash. Finally, as they observe that these crashes are predictable by means of bear market indicators and *ex-ante* volatility measures, they propose a dynamically weighted momentum strategy that doubles that Sharpe ratio of the static momentum portfolio.

Secondly, Cooper et al. (2004) find that the profitability of moment-based investment strategies depends significantly on the state of the market. More specifically, momentum profits appear to be higher post positive market returns as opposed to post negative market returns, making momentum-based strategies less fit in periods of market downturn.

Thirdly, Stivers & Sun (2010) found that recent market cross-section dispersion in stock returns is negatively related to the subsequent momentum premium. As the name suggests, cross-section dispersion in stock returns is a measure for the dispersion of the return of a particular share from the consensus represented by the market return. Being negatively correlated to the subsequent momentum premium, this suggests that the momentum premium could be considered as being procyclical, in line with the findings of Cooper et al. (2004).

The results of Moskowitz et al. (2012) show that the traditional momentum-based strategies can be significantly improved by considering the major and persistent drawbacks of momentum portfolios. As most momentum-based strategies covered in the research mentioned so far do not take into account the minimization of risk, we suggest that there can be promising results when we attempt to combine the principles of momentum-based investment strategies with risk-optimization strategies such as iVaR, which will be discussed extensively in the next part.

2.3 iVaR: a stress-less alternative to traditional risk measures

"Using volatility as a measure of risk is nuts. Risk to us is the permanent loss of capital [. . .]." (Charlie Munger)

In 1952, Nobel Prize winner Harry Markowitz laid the foundation for the field of research of portfolio optimization with his MPT. As a conception of risk, he used the concept of variance defined as "the average squared deviation of Y from its expected value" (Markowitz, 1952, p. 80). In other words, his original measure of risk represented the dispersion of a particular asset relative to the mean. However, this measure of risk fails to take into account the direction of the dispersion, since volatility as a measure is based on the concept of symmetry. Indeed, consistently negative returns are not considered as risky, while volatile positive returns are. Markowitz later acknowledged the shortcoming of his risk measure and introduced semi-variance as an alternative, coming to the realisation that "semi-variance is the more plausible measure of risk" (Markowitz, 1971, p. 374). The concept of semi-variance recognized the importance of downside risk over upside risk and is a specialized case of the lower partial moment (LPM) (Nawrocki, 1999). The integration of downside into risk measures was further developed by JP Morgan in its 1994 RiskMetrics™ document, describing Value at Risk (VaR) mathematically.

This measure still has several flaws (InvestSuite, 2019). First of all, portfolio risk according to Value at Risk can increase with diversification, which illustrates the counter-intuitive nature of the measure [1]. Secondly, VaR requires a certain confidence level α to be set. This choice has a significant effect on the result, which consequently makes the optimization less robust. Lastly, Value at Risk fails to consider all potential losses beyond the threshold value, also called the tail risk.

Expected shortfall, also called Conditional Value at Risk (CVaR) or Tail VaR, mitigates this last shortcoming of regular VaR since it calculates the tail mean of the loss distribution. In contrast to VaR, CVaR is a coherent risk measure as defined by Artzner et al. (1999). Chekhlov et al. (2003) further build upon CVaR by introducing conditional drawdown-at-risk (CDaR), which is centered around the concept of drawdown. A drawdown is a peak-to-trough decline, expressed as the percentage between the peak and the subsequent trough. Conditional drawdown is the mean of all these drawdowns, or cumulative losses, above a certain threshold. Other drawdown-based measures of risk include Maximum Drawdown (MDD), Conditional Drawdown (CDD), and Conditional Expected Drawdown (CED) (Möller, 2018).

[1] Imagine a bond trader bound by a VaR limit of x euros with a confidence level c and a horizon T. If the trader buys bonds from an increasing number of different issuers or, in other words, diversifies her portfolio, the probability that one of the bond issuers defaults will at some point be greater than $1 - c$, thus resulting in a VaR > 0. However, a completely undiversified portfolio of bonds bought from only one single issuer will result in a VaR of 0 (Rau-Bredow, 2020).

Although these measures are mathematically sound and coherent, they still fail to capture what investors intuitively perceive as investment risk. iVaR, introduced by the financial technology firm InvestSuite, explicitly aims to address this problem by minimizing what an investor actually perceives as risk. This perceived risk is made up of three factors: (i) the magnitude of losses, (ii) the frequency of losses, and (iii) the time it takes to recover to pre-downturn levels.

Traditional volatility- or correlation-based portfolio construction uses historical returns, generated as the relative difference of prices at times t and $t-1$, to calculate expected return and risk (based on the covariance matrix). These two aspects are then optimized in combination by using measures such as the Sharpe ratio, which expresses the additional amount of return that an investor receives per unit of increase in risk (Sharpe, 1966), the Sortino ratio, which corrects for the shortcomings of the Sharpe ratio (Sortino & Price, 1994), or the Calmar ratio, which is a modification of the Sterling ratio (Young, 1991). iVaR on the other hand uses the possible paths of an instrument in order to include the time aspect of drawdowns. Mathematically, iVaR will calculate the integral of all drawdowns over the investment horizon (light pink in Figure 2.1). Based on this, it will minimize the time-weighted average distance between a set of simulated investment portfolios and their running maxima. Intuitively, this will result in a time series that rises as smoothly as possible without sharp falls.

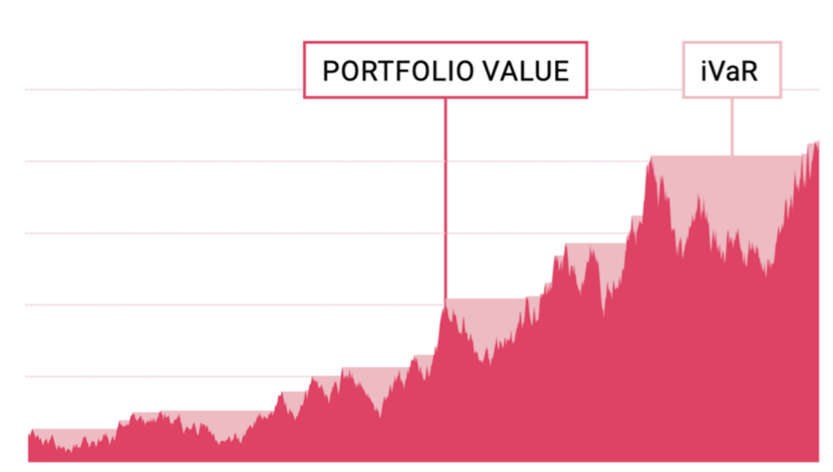


Figure 2.1: iVaR as an average expected drawdown (%) (InvestSuite, 2019)

It is important to note that iVaR does not optimize for maximal returns, which is usually the case for a portfolio optimization algorithm or solver. Rather, iVaR assumes that by minimizing losses at any given point in time, overall returns will be higher in the long run because of the compounding effect. Just like the momentum strategy, it could be argued that this is a naive approach. Essentially, the idea behind iVaR is not that far from Warren Buffet's rule 1: "Never lose money." Indeed, if there is a strong downturn in a traditional portfolio, higher returns will be needed to compensate for this crash, whereas a portfolio optimized on iVaR that can avoid (a larger part of) this downturn needs lower returns to reach the same final position, since it compounds on a larger basis. Also, if an instrument had higher returns over the past period, on average it has lower drawdowns in the future. This suggests a link between iVaR and momentum effects that shows promise for further examination.

Ex-ante, the optimization problem that minimizes iVaR can be formulated as minimizing the residuals with regard to monotonic growth. Traditionally, portfolio optimization makes use of a return space, from which expected returns $\mathbb{E}(R)$ and a covariance matrix $\sum$ are derived, in order to maximize the Sharpe ratio based on a certain technique for recursive convex optimization. However, this only considers the end state of the portfolio, regardless of the intermediate path and thus the size, frequency and duration of losses, which are essential elements to be included in a measure of perceived risk. Therefore, iVaR uses a market generator to generate tensors from historical data, which are pre-estimated simulated trajectories of the instruments. For an N-dimensional universe of instruments, we have $\mathbf{w}$ as the vector of weights $w_i, i \in \{1, ..., N\}$. The minimum drawdown portfolio w^* is the solution to the following linear optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbb{E}(\xi_{\mathbf{a}}(\mathbf{w})) \\ \text{s.t.} \quad & \xi_{\mathbf{a,t}} = \mathbf{m_{a,t}} - \mathbf{wS}_t \\ & \mathbf{m_{a,t}} \geq \mathbf{m_{a,t-1}} \\ & \mathbf{m_{a,t}} \geq \mathbf{wS}_t \\ & \mathbf{wI}^N = 1 \\ & \mathbf{w} \geq 0, \end{aligned} \tag{2.1}$$

where we minimize the expected drawdown ξ as a function of portfolio weights $\mathbf{w}$. The drawdown ξ is a non-linear function of the portfolio path $\mathbf{P_t} = \mathbf{wS}_t$, $\xi_{\mathbf{t}} = \max(\max_{t_i<t}(\mathbf{P_{t_i}}) - \mathbf{P_t}, 0)$, but can hence be written as a linear problem by instrument variable $\mathbf{m_t}$ which denotes the monotonic growth of the portfolio value $\mathbf{m}_t \geq \mathbf{wS}_t$.

2.4 Machine learning in portfolio optimization: raising the linear bar

"It is hard to believe that something as complex as 21st-century finance could be grasped by something as simple as inverting a covariance matrix." (De Prado, 2018)

Traditionally, linear methods have dominated the financial literature. By extension, in the area of empirical asset pricing and portfolio optimization, linear regression-based strategies have historically been put forward as the standard. However, recent research in various subdomains and applications provides strong evidence in favour of the use of Machine Learning (ML), falling broadly into three use-case categories: *(i)* constructing new and superior features, *(ii)* improving predictive performance, and *(iii)* designing problem-specific models.

2.4.1 Constructing new and superior features

The use of ML proves useful on two fronts when it comes to features.

Firstly, it allows to use novel data sources that were once off-bounds due to their vast scale that could not be captured with traditional financial models. For example, Ke et al. (2019) use Dow Jones Newswire articles, Manela & Moreira (2017) Wall Street Journal front-page articles, and Obaid & Pukthuanthong (2022) news photos, which are then fed to ML models.

Secondly, even using standard price data, ML has the potential for both automated feature extraction and feature selection. Instead of manual construction, selection and combination of features that form the basis trading rules for portfolio optimization, ML could generate these rules directly from the data.

Early research into the use of machine learning in finance done by Takeuchi & Lee (2013) confirms this, finding an improved variant of the momentum effect in individual stock prices. Messmer (2017) also finds that momentum appears to be among the main drivers of return predictions in the context of the use of ML in asset pricing. Likewise, Gu et al. (2020) demonstrate that momentum is one of the dominant predictive signals to ML methods.

2.4.2 Improving predictive performance

A second use-case for ML is the outperformance of non-linear methods in terms of predictive performance in recent research (Gu et al., 2020). One approach is to identify from a large set of features those variables that contain the highest information about the cross-section of future returns and then feed these into a predictive model. Moritz & Zimmermann (2016) and Kelly et al. (2019) both use ML to identify these variables, and subsequently use them to construct portfolios based on these variables, achieving high model accuracy. On top of its ability to detect features with strong predictive power, it is also able to capture non-linearities between variables. Oh (2019) demonstrates the importance of interaction between features in a prediction model and how this effect can be used to improve model performance.

2.4.3 Designing problem-specific models

Lastly, ML models can be adjusted or designed specifically for the intricacies of specific problems in finance. In the field of momentum investing, ML can play a role in mitigating an inherent shortcoming of momentum strategies, namely the strong risk of occasional large crashes. As Barroso & Santa-Clara (2015) point out: "In 1932, the Winners-Minus-Losers (WML) strategy delivered a -91.59% return in just two months. In 2009, momentum experienced a crash of -73.42% in three months."

Lim et al. (2019) devise a hybrid class of models that integrate volatility scaling with the use of deep learning models. They find that an LSTM model generates the best Sharpe ratios through their ability for trend estimation. This hints at the potential of some ML models to perform strongly during downturns.

2.4.4 Overview of ML techniques

An overview of the literature on the ML techniques used in our research can be found in Appendix B.

Chapter 3

Research questions

Our research will attempt to optimize risk and reward simultaneously in a novel approach, by combining the use of momentum strategies that aim to maximize returns, with iVaR optimization that aims to minimize variance.

Our proposed approach aims to innovate on three fronts, around which our research questions are focused. Firstly, it is based on a more accurate way of capturing investor risk perception than traditional portfolio optimization techniques such as minimum variance, by building on the iVaR risk measure. Secondly, our approach explores the potential of dynamically shifting between a return-maximizing strategy (momentum) and a risk-minimizing strategy (iVaR). Thirdly, we use machine learning techniques instead of traditional linear techniques for the construction of momentum portfolios. The sections below dive deeper into the latter two questions.

3.1 Momentum & iVaR

The second gap in the literature that our research explores is that of combining momentum, a return-maximization strategy, with iVaR, a risk-minimization strategy. More specifically, our thesis will explore whether and how it is possible to dynamically combine or shift between both strategies depending on the market status. Is there a way to shift the weight of the portfolio optimization strategy towards risk-reducing iVaR during a market crash and afterwards transition to a momentum strategy when markets recover? Our research explores two different possibilities for introducing a momentum effect in the iVaR optimization framework.

The first, objective-based method will explore the path of adding a second term to the objective function (beside the usual iVaR minimization objective function) that contains a momentum portfolio as a benchmark. The traditional iVaR objective term will optimize by minimizing the average *absolute* drawdown, being the top-to-bottom loss of the portfolio. The second term will also minimize a drawdown, but this time it is the drawdown *relative* to a benchmark portfolio that can be passed on, in our case a momentum portfolio. Because the drawdown is relative (and thus changing over time based on the performance of the two portfolios relative to each other), essentially a dynamic weight is allocated to both objective function terms (and thus both strategies), which gets larger (smaller) as the difference between the iVaR and momentum portfolio gets larger (smaller). Our hypothesis is that in periods of large drawdown, because the dynamically allocated weight of the iVaR strategy will increase, it will allow the overall portfolio to be more resistant to significant drawdowns. During bull market periods, the momentum strategy will gain a more significant weight, allowing for the overall portfolio to ride this wave of strong positive returns. Given these two counterbalancing effects, we expect a combined iVaR + momentum portfolio to relatively outperform on risk/reward trade-off measures.

The second, constraint-based way of combining iVaR and momentum is to set an additional constraint on the iVaR optimizer, which specifies a minimum exposure to a specific set of assets in the resulting portfolio at each point in time, where we specify that these assets need to come from the set of winners derived from the momentum strategy. This will ensure that, while optimizing fully for iVaR as the objective term in the objective function, we still ensure a minimum representation of momentum assets in the final portfolio at each point in time.

3.2 Machine learning for momentum

Our research explores how machine learning can play a role in constructing momentum portfolios. Traditionally, they are constructed using specific momentum indicators, such as the past performance of assets relative to others, or absolute returns of individual assets over the past 1 to 12 months (Moskowitz et al., 2012).

However, with ample data available, there could be signals for positive forward returns in the asset data that are not (fully) captured by *ex-ante* indicators, but that could be identified by machine learning techniques that are better suited to predict on large datasets, with interaction effects or non-linear dependencies. For example, recurrent neural networks are able to discover patterns in time-series data in the form of informative deep features, without needing to specify indicators *ex-ante*. More concretely, we will explore the use of machine learning techniques that autonomously identify indicators in the available data that prove to be good predictors for (high) positive forward returns.

Recent literature in the field including Gu et al. (2020) and Choi & Renelle (2019) suggests that the use of various ML techniques for portfolio construction can improve predictive power and as a consequence portfolio performance.

Chapter 4

Data and methodology

4.1 Overview

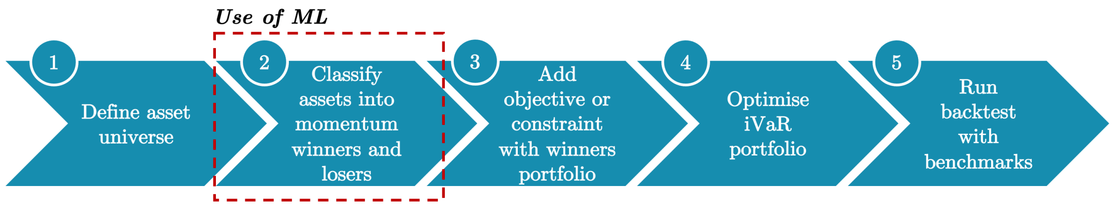


Figure 4.1: Methodology: from data selection to back-testing

In the sections below, a more detailed description will be given on the different steps in the portfolio optimization pipeline. A high-level overview of the methodology is given in Figure 4.1.

The first section (*step 1*) lays out the data used for our backtests. In the second section (*step 2*), more detail is given on the models used to classify assets into momentum "winners" and "losers", including an explanation on the application of the 8 classification techniques used in our research. We will also elaborate on how we optimized the hyperparameters for the different models, and which evaluation metrics we used to assess the quality of these models. The third section (*step 3 & 4*) outlines two approaches we used to integrate the momentum portfolio into the iVaR framework. In the fourth and final section (*step 5*), we will explain the procedures applied for evaluating both model classification and portfolio performance, as well as how we used InvestSuite's backtest constructor to run backtests and compare these results against market and pure strategy benchmarks.

4.2 Data

4.2.1 Data selection, cleaning and transformation

In terms of the financial assets that we will include in our universe to build and test our models, we will use a list of 103 European ETFs as provided by InvestSuite. The full list can be found in Appendix D.2. More detail on the selection process, as well as the cleaning and feature engineering, can be found in Appendix D.1.

4.2.2 Splitting the data into training and test data

Also note that we split the dataset in terms of time instead of training, testing and deploying the model on the exact same time dimension and splitting the dataset in terms of ETFs. The reason is that we need out-of-time testing for our model in order to evaluate the true accuracy measures for 2 different reasons. Firstly, if we went with the latter split in the dataset, our model would be prone to picking up the general trends in the market, classifying the majority of assets as "winners" in a bull period and "losers" in a bear period to attain a high accuracy. Secondly, as our models are based on looking at the cross-section of ETF returns within specific time periods, a large chunk of the information included in the model would be lost if we were to take the cross-section over a significantly smaller set of assets.

As we will run our model to calculate the momentum portfolio and iVaR optimization at multiple points in time during the backtest, we will repeat this split at each of these points in time.

4.2.3 Balancing the training data

Given the nature of the data, in most periods the proportion of financial assets having positive returns over 30 days is relatively large compared to the proportion with negative returns. Therefore, measures need to be taken to account for an imbalanced data set. To resolve this, we will work with cross-sectional momentum, *i.e.*, taking the relative winners in a given asset universe within a given time period. This allows us to make the training set balanced by defining the "winners" as being all assets in the 50th percentile when we order all assets based on their returns in each time period. As such, we split the training data set in 2 parts that contain an equal number of observations. An alternative strategy would be to over-sample from the observations defined as absolute winners. However, this approach would lead to data loss.

4.3 Classification

In the next step, a momentum portfolio is created that can be fed into the iVaR optimizer. Momentum portfolios are traditionally implemented as a cross-sectional trading strategy among individual stocks (Jegadeesh & Titman, 1993; Asness, 1994) or long-only equity portfolios (Moskowitz & Grinblatt, 1999; Lewellen, 2002). In our backtests, we will employ a similar cross-sectional strategy, in which the 50th percentile of stocks will be labeled as ”winners”, whilst the remaining assets will be labeled as “losers”. Gupta & Kelly (2019) demonstrate that this strategy that buys the recent top-performing factors performs significantly above traditional stock momentum.

4.3.1 Classification methodology

Our methods will cross-sectionally classify ETFs in the European ETF index into winners and non-winners using various machine learning techniques, ranging from more standard methods such as decision trees to more recent and complex techniques including recurrent neural networks.

In general, we will use a vector of return features $X_t^{(i)}$ over a specific time period to predict a target variable:

$$y_t^{(i)} = f\left(X_t^{(i)}\right),$$

where $y_t^{(i)}$ can be either the predicted return of asset i in period t or the probability that asset i is considered a “winner” in period t. The function f will depend on the model that we use to make this prediction, all of which will be discussed in the subsections below.

Given the fact that, as we move ahead in time with fixed start date of Dec 31, 2013 in our observed period, more data becomes available to train and test our model on. This allows for additional return features that look 1 period further back in every iteration, hence increasing the number of features and potentially predictive power as well. However, this would lead to relatively large models that far outgrow the 12-months boundary for which different research have found empirically that momentum-based strategies seem to outperform. Moreover, the generalisability of the results that are found for a specific time dimensions to different time dimensions is limited. Therefore, we will keep a fixed, limited number of time horizons starting from the observation time on which returns features are constructed. In other words, for each date in our backtest we will look at the previous n months to build and test our model.

4.3.2 Prediction methodology

Another way of classifying assets into "winners" and "losers" is to do it indirectly through a time-series prediction step. For this method, we will use Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models (with details on the implementation of the model below).

The difference with the classification method is that features are not pre-constructed. Daily price data is fed directly into the model, with a fixed look-back horizon to ensure trainability and consistency of the model. Batches of price data of the same length, along with the target variable (the n-day forward price) are passed on to the model for training. The predicted n-day forward price can then be used to calculate the n-day forward return, after which a percentile of highest returns can be classified as winners.

4.3.3 Implementation of ML techniques

An detailed overview of the implementation steps for each of the ML models used in our research is given in Appendix C.

4.3.4 Evaluation of model classification accuracy

To gain insights into how our models perform in terms of predicting future "winners", we will use both simple classification accuracy measures such as sensitivity, as well as more nuanced measures such as Area Under the ROC-curve (AUC). Explanations of these measures are detailed in Appendix E. Note that we will use the AUC as the primary measure to evaluate the classification accuracy of our model as this is both scale- and threshold-invariant, hence making it more consistent to compare results across different models and hyperparameter settings.

These accuracy measures will be used to heuristically tweak our model as to maximize the predictive power of our models, which evidently leads to higher momentum returns. To be more specific, we will use the AUC as the "scoring" method in the hyperparamter optimization, which is used to evaluate the performances across different iterations using different sets of hyperparameters. The specifics of this optimization procedure will be discussed after the following subsection.

4.3.5 Hyperparameter optimization

The classification accuracy of machine learning models can sometimes depend significantly on the choice of hyperparamters. Therefore allocating some computational time for optimizing model hyperparamers can pay off. However, the focus of our research is not on solely optimizing the classification models, but rather finding an effective balance between high predictive power with reasonably low computational time. As we will rebuild our model at each iteration in our backtest, we do not want to use computationally expensive methods such as grid search (systematically sampling hyperparameters in a grid). Instead, we will use random search to find a set of hyperparameters that can significantly improve the classifications accuracy, where we let the number of iterations depend on the time needed to run a random search for each model.

We will use the Area Under the ROC-Curve (AUC) to score the different sets of hyperparameters. As such, the model with the set of hyperparemeters that attains the highest AUC will be used as the final model at each specific point in time during the backtest.

In Table 4.1 an overview is given of the hyperparameters and ranges on which they are tuned for the different models.

Model	Hyperparemeter	Range
Linear reg	*No hyperparameters tuned*	*n.a.*
Logistic reg	*No hyperparameters tuned*	*n.a.*
Decision Tree	Max depth	2, 3 ... 20
	Min samples split	*2, 3 ... 20*
	Min samples leaf	*1, 2 ... 10*
	Min weight fraction leaf	*0.1, 0.2 ... 0.5*
	Max features	*0.1, 0.2 ... 0.9*
	Max leaf nodes	*2, 3 ... #ETFs*
	Min impurity decrease	*0.01, 0.02, ... 1*
Random Forest	No. estimators	*0, 50, 100 ... 1000*
	Max depth	*2, 3 ... 20*
	Min samples split	*2, 3 ... 20*
	Min samples leaf	*1, 2 ... 10*
	Min weight fraction leaf	*0.1, 0.2 ... 0.5*
	Max features	*0.1, 0.2 ... 0.9*
	Max leaf nodes	*2, 3 ... #ETFs*
	Min impurity decrease	*0.05, 0.1, ... 1*
Naive Bayes	*No hyperparameters tuned*	*n.a.*
SVM	C	*0.1, 1, 10, 100*
	Gamma	*scale, auto*
	Kernel	*rbf, poly, sigmoid*
GBC	No. estimators	*0, 50, 100 ... 1000*
	Max depth	*2, 3 ... 20*
	Min samples split	*2, 3 ... 20*
	Min samples leaf	*1, 2 ... 10*
	Min weight fraction leaf	*0.1, 0.2 ... 0.5*

Max features	*0.1, 0.2 ... 0.9*
Max leaf nodes	*2, 3 ... #ETFs*
Min impurity decrease	*0.05, 0.1, ... 1*
Learning rate	*0.01, 0.1, 1, 10, 100*

Table 4.1: Hyperparameters and ranges for different classification techniques

4.4 Combining momentum with iVaR optimization

A final step in our methodology is to integrate the predicted classification of assets in the iVaR optimization procedure. We identified several different possibilities for introducing a momentum effect in the iVaR optimization framework, based on a simplified representation of the iVaR optimization problem given in Equation 2.1.

4.4.1 Constraint-based approach

In our research, a first approach we have explored is that of adding a constraint to the iVaR optimizer that specifies the minimum exposure to the class "winners" in the momentum strategy. In this constraint-based approach, each iVaR-optimized portfolio will have a minimum exposure to what we have predicted to be winners in a previous step, making it more likely that the assets included in the portfolio will have a relatively better return in the following period under the momentum assumption. This would result in the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbb{E}(\xi_{\mathbf{a}}(\mathbf{w})) \\ \text{s.t.} \quad & \xi_{\mathbf{a,t}} = \mathbf{m_{a,t}} - \mathbf{w}\mathbf{S}_t \\ & \mathbf{m_{a,t}} \geq \mathbf{m_{a,t-1}} \\ & \mathbf{m_{a,t}} \geq \mathbf{w}\mathbf{S}_t \\ & \mathbf{w}\mathbf{I}^N = 1 \\ & \mathbf{w}\boldsymbol{\Omega} \geq \omega \\ & \mathbf{w} \geq 0, \end{aligned} \tag{4.1}$$

where $\boldsymbol{\Omega}$ is a binary vector of assets assigned 1 for momentum winners and 0 otherwise, and ω a threshold value that can be tuned.

4.4.2 Objective-based approach

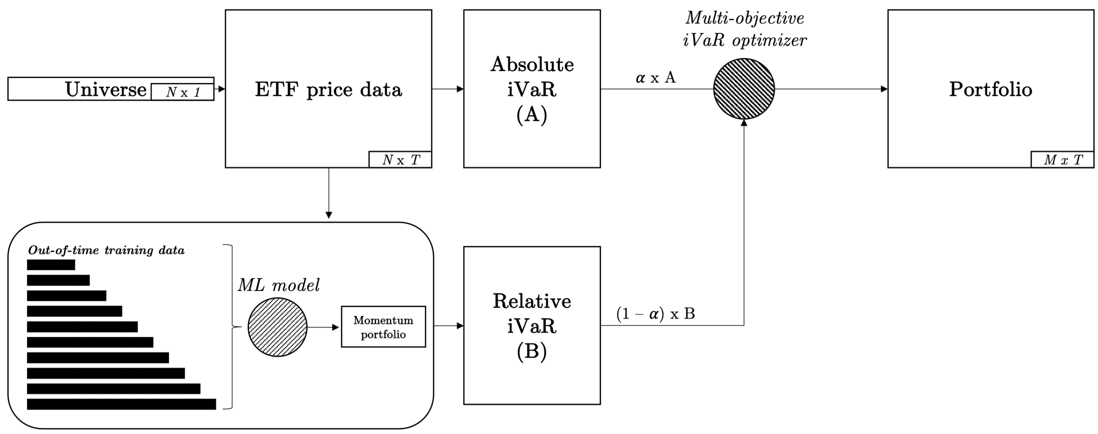


Figure 4.2: Multi-objective iVaR optimization

The second approach is that of multi-objective optimization, in which objective terms are weighed against each other on a Pareto front. In this objective-based approach, a momentum portfolio is added as a benchmark in a second term in the objective function for which the iVaR framework will optimize. A fixed *a-priori* weight is allocated to each of the two objective function terms (iVaR and the momentum portfolio benchmark), but the final weight will be dynamic based on the relative difference between the iVaR and momentum portfolio. For example, when the momentum portfolio is outperforming, the difference with the iVaR portfolio will be increasingly positive, and therefore so will the weight of the momentum portfolio. Equally, during a momentum crash, the difference will be negative, hence the momentum portfolio will have a smaller weight. Figure 4.2 shows the workflow, in which price data over a time horizon T is taken for a universe of N ETF assets. These are sampled point-of-time and trained on a machine learning model, giving a momentum portfolio for each time t in T. The absolute iVaR term is derived from drawdowns on the iVaR portfolio calculated on the ETF price data, the relative iVaR term is derived from the relative drawdowns compared to the momentum portfolio. These are then combined in the multi-objective iVaR optimizer, giving a portfolio of M assets at each time t in the horizon T.

This would result in optimization problem 4.2:

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \lambda \mathbb{E}(\xi_{\mathbf{a}}(\mathbf{w})) + (1-\lambda)\mathbb{E}(\xi_{\mathbf{r}}(\mathbf{w})) \\
\text{s.t.} \quad & \xi_{\mathbf{a,t}} = \mathbf{m}_{\mathbf{a,t}} - \mathbf{w}\mathbf{S}_t \\
& \mathbf{m}_{a,t} \geq \mathbf{m}_{\mathbf{a,t-1}} \\
& \mathbf{m}_{a,t} \geq \mathbf{w}\mathbf{S}_t \\
& \xi_{\mathbf{r,t}} = \mathbf{m}_{\mathbf{r,t}} - (\mathbf{w}\mathbf{S}_t - \mathbf{b}\mathbf{S}_t) \\
& \mathbf{m}_{r,t} \geq \mathbf{m}_{\mathbf{r,t-1}} \\
& \mathbf{m}_{r,t} \geq \mathbf{w}\mathbf{S}_t - \mathbf{b}\mathbf{S}_t \\
& \mathbf{w}\mathbf{I}^N = 1 \\
& \mathbf{w} \geq 0,
\end{aligned}
\tag{4.2}
$$

where λ is a set weight to the absolute objective term and b the momentum portfolio.

For both methods, we will pass on a dataframe containing all investible assets in a given period with a classification label for each. Depending on the classification model we used, we will either end up with exactly 50% of assets being classified as winners (if we can take the cross-section of predicted returns or predicted probabilities), or with a less balanced distribution of winners and losers. We will then specify a threshold for the minimum exposure to assets in the winners class. Note that this minimum constraint is expressed as an exposure to assets with the winner class, not the absolute number of winning assets to be included in the iVaR portfolio.

As mentioned before, we will run the iVaR optimization procedure iteratively at each given moment in our backtest. This means that, in order to have a result that is consistent with the momentum assumption, namely that the observation of these momentum effects only hold true to specific time periods, the classification exercise is also repeated each time we repeat the iVaR optimiziation procedure.

4.5 Evaluation

In order to evaluate the overall portfolio performance of the momentum portfolio generated by the ML model that is then integrated in the iVaR framework, we need a statistical method that allows us to compare several portfolios.

4.5.1 Relative evaluation of models

Our analysis leads to a set of models based on a variety of machine learning techniques in the form of time-series data that includes performance, risk and diversification measures.

We will use the Hansen et al. (2011) Model Confidence Set (MCS) approach to determine which models under- or outperform relative to others, giving a statistically founded method of assessing:

(i) How models that combine a momentum and iVaR strategy perform relative to the market, or simple momentum portfolio and iVaR-optimized portfolios in terms of risk and return measures;
(ii) How different momentum portfolios created with machine learning techniques, implemented in an iVaR strategy, perform relative to each other in terms of risk and return measures.

Appendix F gives a more detailed, mathematical explanation of how Hansen's MCS works. Essentially, it gives a set of included models, which comprise all statistically significantly better models based on the chosen measure, and a set of excluded models which do not significantly outperform other models in the set of all models.

4.5.2 Evaluation of portfolio returns

Backtest constructor

We will use InvestSuite's backtest constructor to backtest our investment strategies over the period of 31-12-2013 until now. In this backtest, a set of portfolio optimizer objects will be passed, that will be optimized at each given point in time during the backtest. Corresponding to these portfolio optimizer objects, we will pass on the full set of investible assets available at that point in time, classified in predicted winner and loser classes. As such, the full algorithm as elaborated on in the methdology section will be repeated with the information that we had available at the point in time we would have made the investment decision.

The backtest constructor optimizes the iVaR portfolio at each point in time, and returns the resulting portfolio for each timestamp. Next, we can calculate the portfolio value by passing on a start portfolio of, for example, EUR 1 million.

Benchmarks: iVaR portfolio returns, index and momentum-only portfolio

We can compare the combined momentum and iVaR portfolio with an index that represents a diversified ETF portfolio (which would correspond to a well-diversified ETF that invests in all equities available in the ETF univserse), the iVaR portfolio that does not take into account momentum, and a momentum-only portfolio that represents an equally weighted portfolio consisting of all assets that have been classified as "winners" in the construction of the combined portfolio. We will visualize these different portfolios in 3 different plot types. One that gives the portfolio value over time, a second that displays yearly portfolio returns as a bar chart, and a third that displays the drawdown occurrence frequencies per drawdown percentile.

For the index benchmark, we will use the returns of the iShares Core MSCI Europe UCITS ETF EUR. As the name suggests, this ETF replicates the index returns of the MSCI Europe, which consists of 429 stocks with European focus, making it a good proxy for the overall universe in which most of InvestSuite's (European) clients are interested in.

Chapter 5

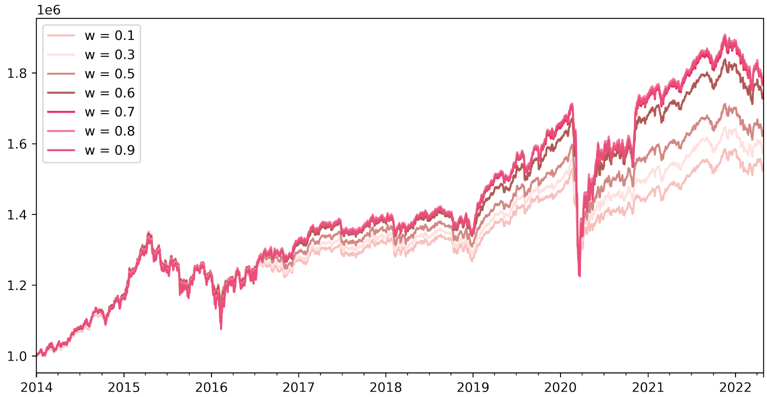
Results and discussion

Our results show that combining a momentum strategy with the iVaR framework is an effective investment strategy. This will be demonstrated based on our research findings in the following sequence. First, we will compare the objective-based and constraint-based approach of integrating the momentum strategy in the iVaR framework. Then, we will compare the ML models outlined in the methodology, to see which of these models will be most effective in predicting momentum portfolios. Lastly, we will evaluate the performance of the best performing iVaR + momentum portfolio relative to several benchmarks, to assess if our research methodology is able to outperform market and pure strategy benchmarks.

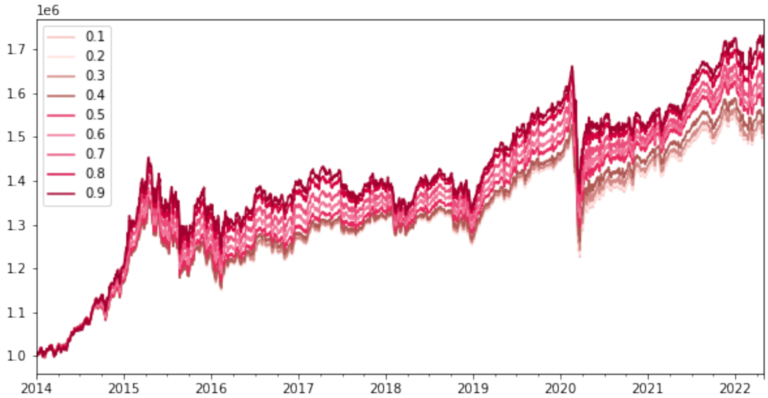
5.1 Comparison of constraint-based and objective-based method

When using the objective-based method, in which we add a second, relative term to the objective function, we also specify a weight for that additional term (and give the remaining weight to the iVaR term). Therefore, with weight λ assigned to the relative drawdown term, we get $\lambda\mathbb{E}(\xi_{\mathbf{a}}(\mathbf{w})) + (1-\lambda)\mathbb{E}(\xi_{\mathbf{r}}(\mathbf{w}))$. As can be seen in Figure 5.1, the higher the weight λ to the relative term (that contains the momentum effect), the stronger the portfolio index performance overall and especially during bull runs, but also the deeper the drawdown, most visibly during the COVID-related market crash in 2020. This weight λ can be adjusted manually depending on the risk preference of a specific investor, but it can also be tuned to an optimal level with the best risk-return trade-off.

Similarly for the constraint-based method, we can specify the minimum proportion of winners to be included in the iVaR-optimized portfolio. In picking such a higher proportion, we will get closer to the pure momentum-based portfolio, and on the flip side,



(a) Objective-based approach: different replication weights λ



(b) Constraint-based approach: different minimum winners proportions

Figure 5.1: Sensitivity analysis of replication weights / min winners proportion on index portfolio value for objective- and constraint-based approach

we will get closer to the pure iVaR portfolio in setting this minimum winner proportion lower.

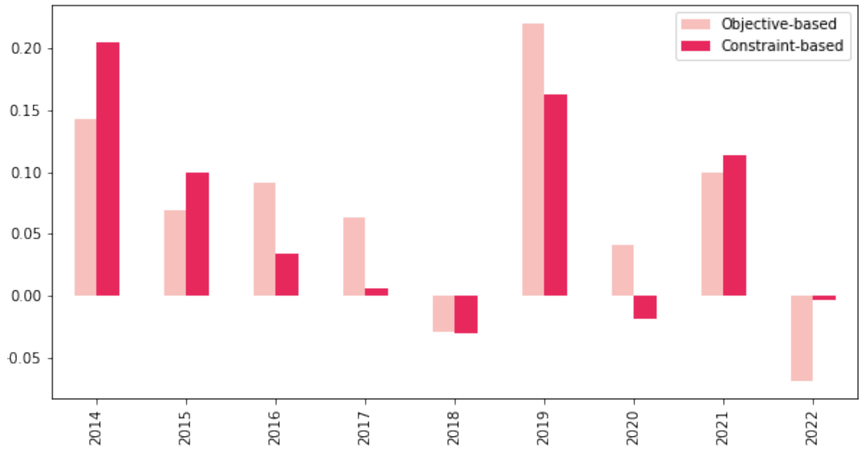
Before we can compare the results of both the constraint-based and objective-based methods, we first need to decide on the parameters for each of the methods. As in both cases we are looking to optimize the trade-off between minimizing drawdown in downturn periods and maximizing the returns when the market experiences a serious uptick, we will first look at which parameters will optimize each of the methods for this trade-off.

As shown in Figure 5.1 (a), a replication weight of 0.8 appears to be giving us the best results in terms of this trade-off. This means that in the objective-based method, we will use a weighting of 0.8 for the term that optimizes for the relative performance of the momentum benchmark, whilst a remaining weight of 0.2 will be give to the iVaR term. For the constraint-based method, Figure 5.1 (b) illustrates that a winners proportion of 0.8 gives us the best trade-off for the minimum number of winners to be included in our iVaR optimized portfolio. This means that the iVaR optimizer will be constrained to picking assets in such a way that at least 80% of the total exposure will be towards assets that have been classified as winners.

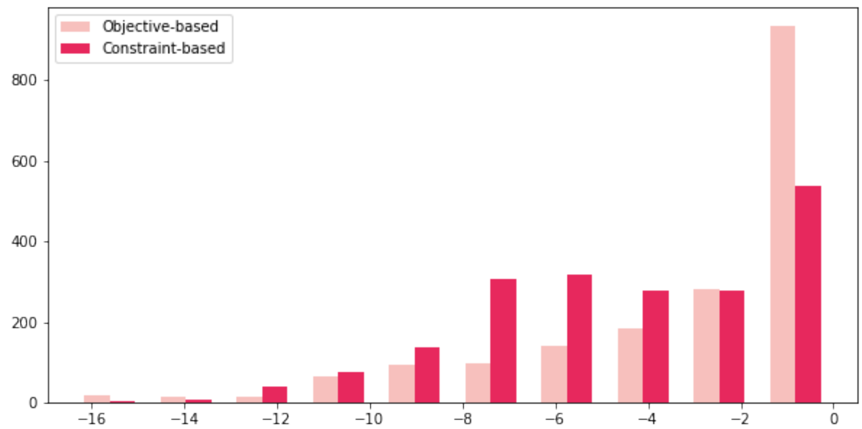
Figure 5.2 demonstrates the difference in performance of the objective- and constraint-based method. In summary, the objective-based approach more closely resembles the momentum strategy (due in part to a high replication weight of 0.8), whereas the constraint-based approach more closely follows the iVaR strategy. However, both strategies have important improvements compared to the single-strategy portfolios. Indeed, the objective-based approach is able to reduce the deepest drawdowns compared to a pure-momentum strategy, and the constraint-based approach delivers superior returns to the iVaR strategy, at similar drawdown.

We will continue to compare our results across the different methods and benchmarks using the objective-based method. However, Appendix G contains a full overview of both the objective- and constraint-based method for each of the models.

(a) indexed value



(b) yearly returns



(c) drawdowns

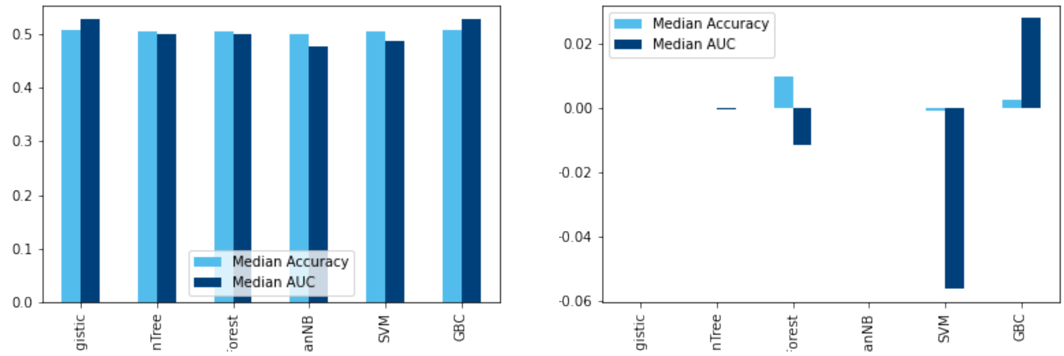
Figure 5.2: Comparison of constraint- and objective-based methods for GRU-based portfolio

5.2 Comparison of ML models

5.2.1 Stand-alone ML model performance

As detailed in the methodology in subsection 4.3.5, hyperparameter tuning through random search was done to optimize the ML models. We found that hyperparemter optimization did not consistently and significantly increase AUC and accuracy across models, as can be seen in Figure 5.3, even when we ran a high number of iterations of random search ($n = 100$). Therefore, we did not dig deeper into fine-tuning these hyperparemters.

The accuracy and AUC of our models are slightly above 50%. Although this might seem low for a typical machine learning model problem, it is actually a realistic result. Before balancing the dataset, accuracies were significantly higher, but this is simply due to the fact that the stock market has on average, over time and across assets, trended positively, especially when looking on a longer forward horizon (e.g. 30 days), and hence a model that predicts all winners would have an accuracy equal to the proportion of winners, which is (much) higher than 50%. However, this does not give a realistic representation of the actual predictive power of the model. Therefore, we balanced our dataset to have an equal number of momentum winners and losers. Furthermore, asset prices on the stock market follow the random walk hypothesis, meaning that asset price moves are purely random on an infinitesimal time horizon. The combination of both factors makes an accuracy of slightly above 50% realistic. As a result, the difference in accuracy and AUC across our models is relatively small.



(a) After hyperparemeter optimization

(b) Delta before vs. after optimization

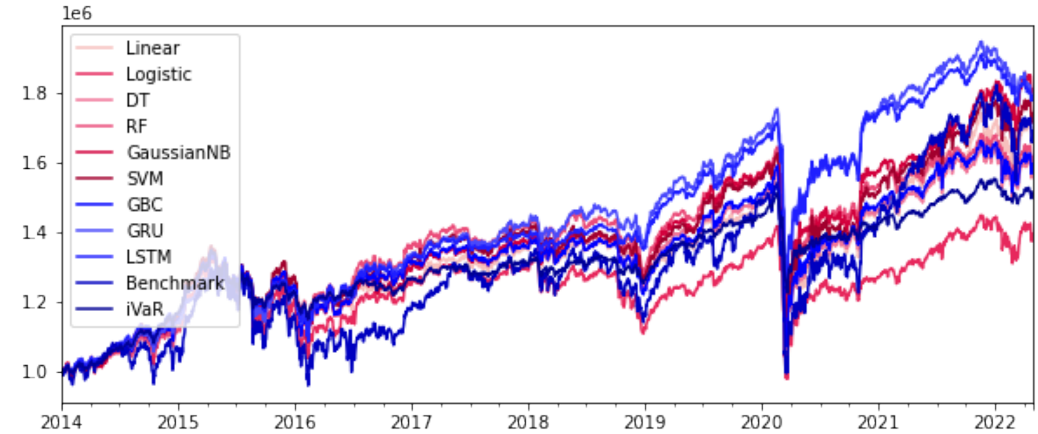
Figure 5.3: Median accuracies and AUCs

5.2.2 ML model backtest performance

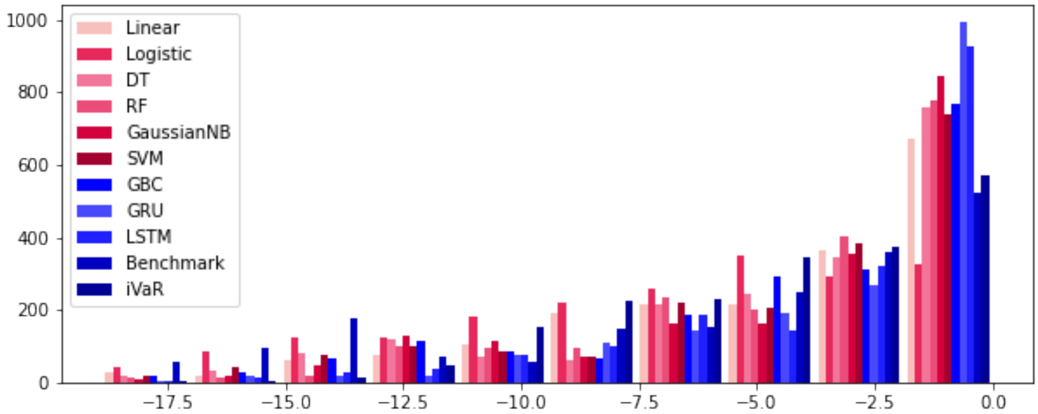
Although the accuracy and AUC of the different ML models is not strongly different, the resulting performance when implemented in the mixed iVaR + momentum strategy does strongly diverge. Figure 5.4 gives a visual overview of the performance of the different ML models in the iVaR + momentum strategy using the objective-based approach. Table 5.1 statistically evaluates the performance of this mixed strategy with different ML models against several key metrics, giving a more nuanced view of the different aspects of portfolio performance. In addition, using the Hansen Model Confidence Set (MCS) test, we can draw conclusions on the statistically significantly outperforming set of models.

At first glance, the average monthly return statistics of many of the mixed-strategy portfolios do not (significantly) beat the benchmark. However, financial performance should always be viewed from a risk-reward trade-off perspective. Indeed, several of the mixed-strategy portfolios were able to reach these similar returns but with significantly lower drawdowns, showing the strong positive impact of the iVaR framework on the mixed portfolio. This means the models manage to both reduce drawdowns, as well maximize returns during specific periods. Based on the latter two measures in Table 5.1, it further becomes clear that on a risk-return trade-off, the mixed-strategy portfolios using ML models significantly outperform a simple market benchmark.

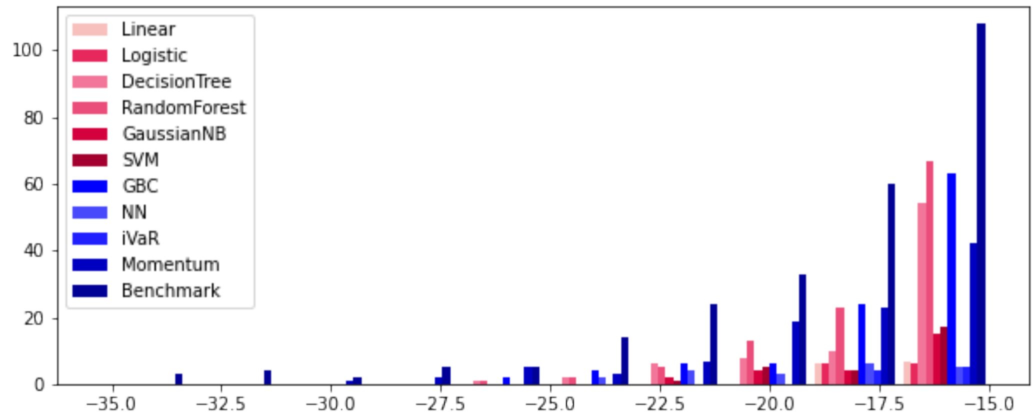
Another important finding confirming one of the hypotheses set out in the research questions and backed by recent literature, is that several (although not all) machine learning models clearly outperform the linear model. Despite minimal and seemingly insignificant differences in accuracy measures between models in terms of predicting winners to include in the momentum portfolio, the performance in backtests is markedly



(a) indexed value



(b) drawdowns



(c) drawdowns (with cutoff -15%) (2)

Figure 5.4: Comparison of different ML models portfolio performance

different and significant.

Finally, comparing individual models, the LSTM-based portfolio performs best when looking at minimizing drawdowns and maximizing the Calmar and Pain ratios.

Model	Returns	Max DD	Avg. DD	Calmar ratio	Pain ratio
Linear regression	0.005589*	0.026046	0.009998	1.367413*	6.528987*
Logistic regression	**0.003806**	0.028046	0.010908	1.259888*	6.117601*
Decision tree	0.005021*	0.024811	0.009567	1.668172*	10.600884*
Random forest	0.005218	0.025160	0.009606	1.511305*	9.394603*
Gaussian naive bayes	*0.006394**	0.024595	0.009564	1.433610*	8.526098*
Support vector machines	0.006005*	0.025400	0.009795	1.429352*	8.512710*
Gradient boosting	0.005092	0.024747	0.009488	1.648397*	10.549359*
GRU	0.006228*	0.021652	0.008585	1.590444*	8.707911*
LSTM	0.006189*	*0.019936**	*0.007837**	*1.931698**	*16.246320**
ETF benchmark	0.006024*	**0.039376**	**0.014756**	**1.224212**	**5.955694**

Table 5.1: Overview of average metric values over investment horizon for all ML models; Models in **bold** are worst performing, models in *italics* are best performing; The set of significantly outperforming models according to Hansen MCS is indicated with *.

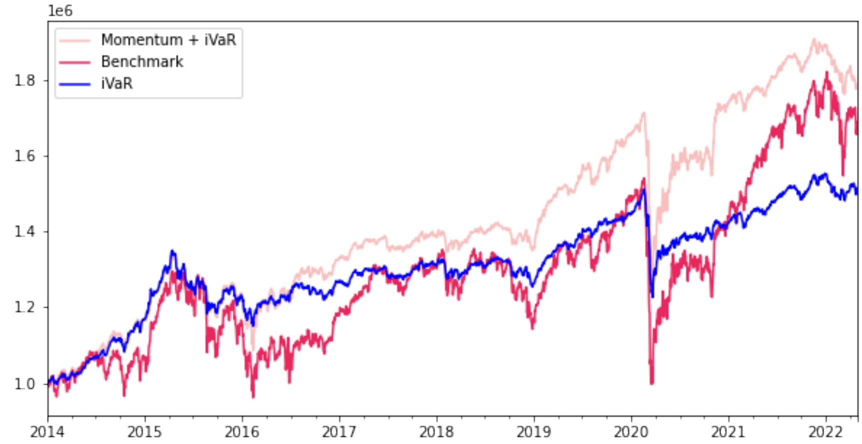
5.3 Comparison of portfolio optimization methods

The primary objective of the iVaR optimizer is to minimize the average accumulated drawdown over the investment horizon. In our backtests, the risk-minimizing behaviour of iVaR is visible throughout the horizon of the backtest and is most notable in periods where the market benchmark goes through a significant downturn, such as in the 2020 downturn caused by the COVID pandemic. As a result, in Figure 5.5 (c) we see that the iVaR model has significantly less episodes of deep drawdowns of 15% or more. This aligns with iVaR's goal of minimizing stress for investors. iVaR has less pronounced troughs, but also less pronounced peaks. It provides an overall stable increasing trend.

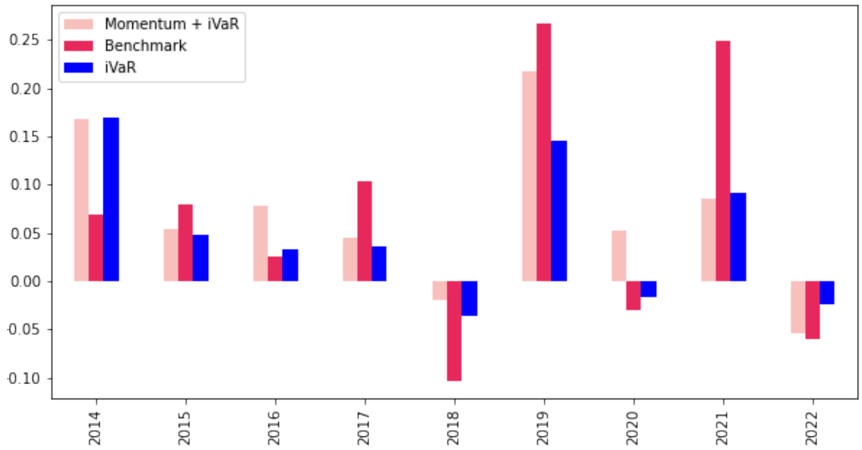
The simple momentum portfolio has opposite characteristics, with outsized bull runs but also strong troughs in downturns, in line with what the literature describes as momentum crashes (Daniel & Moskowitz, 2013).

Hansen's MCS test demonstrates that our iVaR + momentum combined portfolio significantly outperforms on all measures, being returns, maximum drawdown, average drawdown, Calmar ratio and pain ratio (depending on the ML used), since at least one of the mixed-strategy portfolios is included in the set of statistically significantly different models compared to the ETF benchmark.

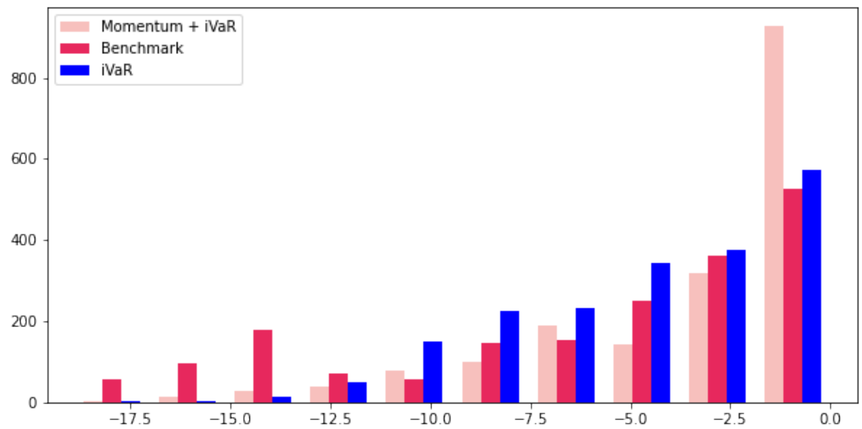
Individual backtests of each of the ML models discussed in section 5.2, compared with market and single strategy benchmarks, can be found in Appendix G.



(a) indexed value



(b) yearly returns



(c) drawdowns

Figure 5.5: LSTM-based ML portfolio vs benchmark portfolio performance

5.4 Discussion

The findings of the results in the sections above allow us to answer the research questions set out in chapter 3. Indeed, our research shows that our ML models improve predictive power compared to the linear model, that iVaR is a promising framework for capturing investor perception of risk by minimizing drawdown, and that integrating a momentum strategy into the iVaR framework further improves returns, in tandem with lower drawdowns.

Firstly, our results show that iVaR is successful at reducing drawdowns during a market downturn. Figure 5.5 demonstrates that during the periods 2015-2016, 2020 and the latest market crash in 2022, the pure iVaR portfolio has significantly less large drawdowns ($> 10\%$), which are a primary source of an investor's perceived risk.

Secondly, our research finds that several models outperform the linear model in backtests, as illustrated in Figure 5.4 and statistically underpinned in Table 5.1. This confirms that there is indeed a benefit to stepping away from traditional linear models for portfolio optimization, as suggested in recent financial literature.

Lastly, our results demonstrate that reconciling the momentum strategy with the iVaR framework successfully improves returns. Figure 5.5 shows that the momentum + iVaR portfolio generates strong returns in the period of 2016-2020, thanks to its ability to allocate a high weight to the momentum strategy. Conversely, in periods of downturn such as 2020, our mixed strategy is able to re-allocate more weight to the drawdown-reducing iVaR strategy. In summary, the mixed iVaR + momentum strategy achieves the goal of following the positive market trends in bull periods more than the traditional iVaR portfolio, whilst being able to keep the drawdowns fairly limited as opposed to the market and momentum benchmark.

Chapter 6

Conclusion, limitations and recommended further research

6.1 Conclusion

Momentum strategies are known to produce strong positive average returns and Sharpe ratios, with the momentum phenomenon appearing across diverse equity markets and over a wide range of asset classes. However, they also suffer from occasional crashes where the momentum effect is reversed and previous gains are largely offset (Daniel & Moskowitz, 2013).

Our research reconciles the investor objectives of maximizing returns and minimizing risk through an approach that integrates a momentum portfolio into the drawdown-minimizing iVaR framework. First, momentum portfolios are generated using various ML models. Then, they are used either as a constraint, such that a minimum threshold is set for the number of assets classified as momentum winners at any given time, or as a dual-objective, such that, besides minimizing absolute drawdown, it also minimizes for drawdown relative to the momentum portfolio. This essentially creates an optimization problem that dynamically allocates more weight to the relevant strategy depending on market conditions.

During several market crashes over the past decade, our backtests demonstrate the relevance of the iVaR optimization framework in reducing strong drawdowns that investors perceive as a significant risk. Also, it demonstrates the effect of momentum crashes and how they have a significant negative effect on the cumulative performance of momentum strategies. Conversely, our research also shows the renewed relevance of the momentum strategy that is able to rebound from market crashes, as is visible during the bull run starting mid-2020 and continuing throughout 2021. The fact that these two conflicting forces were prominently at play in global markets across asset classes in the past two years demonstrates the relevance of research that aims to strike a balance between these effects. Indeed, our research shows that integrating a momentum strategy into the iVaR framework further improves returns, in tandem with lower drawdowns. Additionally, ML models improve predictive power compared to the linear model.

6.2 Limitations and further research

Our research findings confirm the importance of exploring the combination of momentum and iVaR strategies to improve returns while minimizing for drawdowns. Further research could build on our findings by addressing the limitations in the scope of our methodology in order to further assess the profitability of the combined strategy. We recommend further work to focus on (i) exploring more ML models, (ii) examining different conceptual ways of integrating momentum portfolios into the iVaR framework, and (iii) using predictive techniques to steer the dynamic allocation of the weights between the momentum and iVaR strategies over time.

Firstly, our methodology allows future research to experiment with other ML techniques, and pass the resulting momentum portfolio on to the iVaR optimizer. Although our research covers a wide array of the most used ML techniques, demonstrating the broad validity of our methodology, it does not provide an exhaustive overview of classification and prediction techniques to construct momentum portfolios. For example, more recent and complex techniques like Convolutional Variational Autoencoders (CVAEs) could be used for prediction. Although the ML techniques covered in our research already demonstrate outperformance compared to linear models, unexplored models could attain even stronger performance.

Secondly, further research should be done into the different ways of integrating momentum strategies into the iVaR framework. Our research focuses in depth on two of the possible approaches. Other potential unexplored approaches could include adding a target return constraint in the iVaR optimizer, or restricting the original universe of assets to iVaR-optimize with, or exploring the interaction effect between between the two methods used in this research. Both the objective-based and constraint-based approach used in this research prove to be effective ways to integrate momentum in the iVaR framework. Exploring additional approaches could potentially allow for further performance improvement.

Lastly, future research should address in a more exhaustive way the aspect of switching dynamically between return-maximizing and risk-minimizing strategies. For example, Daniel & Moskowitz (2013) demonstrate that market crashes are predictable based on bear market indicators as well as ex ante volatility, which allows to create a dynamically weighted momentum portfolio that delivers promising Sharpe ratios across asset classes.

Future research could look into dynamically switching between the iVaR and momentum strategies by predicting market downturns using these bear market indicators.

Appendix A

Portfolio selection literature

During the 1960s, Sharpe (1964), Treynor (1962), Lintner (1965) and Mossin (1966) developed the Capital Asset Pricing Model (CAPM) based on the idea that not all types of risk that an asset assumes should be compensated, since some types of risk of an individual asset can be diversified away when held in a portfolio. The CAPM was the first coherent framework developed to describe the relationship between the expected return and risk that cannot be diversified away in a portfolio.

With the increase in available computing power, Elton et al. (1976) showed how an optimal portfolio can be selected based on a simple ranking mechanism, clarifying the characteristics that lead to specific assets being included in the optimal portfolio. Most of these ranking algorithms were based on the estimation of covariances between assets, which can usually be solved mathematically by inverting matrices.

As these models for optimizing portfolio selection were based on a large number of inputs that needed to be estimated, there was a need for new models to come up with these predictions. Index models, which are statistical models that use simple linear regression to describe asset returns based on one or multiple indices and the sensitivity of an asset to that index, were introduced to tackle this new issue. The first variation of these index models, the market model, was developed by Sharpe (1964). Shortly after Sharpe introduced the single-index (market) model, researchers introduced the multi-index model in an attempt to better estimate these portfolio optimization inputs:

$$R_{it} = \alpha_i + \sum \beta_{ij} I_{jt} + e_{it}, i = 1, ..., N,$$

where R_{it} is the return of stock I in period t, α_i is the expected return of security I, β_i is the sensitivity of stock I to index j, I_j is the jth index, J is the total number of indices employed and e_{it} is the unique risky return of security I in period t (Elton & Gruber, 1997).

Ross (1978) and Ingersoll (1987) were the first to reformulate the portfolio selection problem to multi-index models. As such, choosing the optimal portfolio can be reduced to choosing among portfolios in a multi-dimensional space where one dimension represents expected return, one dimension represents residual return in case securities can be mispriced, and the other dimensions each represent a different beta (Elton & Gruber, 1997). Elton & Gruber (1992) show that in the case where indices cannot be exactly replicated, securities exist that are theoretically mispriced, and indices are not necessarily normally distributed, each investor can form the optimal portfolio by investing in a riskless asset and choosing from a number of portfolios equal to the number of indices generating return plus one.

Appendix B

ML models literature

B.1 Decision trees

Decision trees (DT) are a way to acquire knowledge in the form of a set of discrete rules, ordered in a tree. Our implementation of this popular discriminatory model uses an optimized version of the CART algorithm, originally presented by Breiman et al. (1984). CART is a greedy algorithm based on the earlier C4.5 algorithm (Quinlan, 1996), which in turn was an improvement of the ID3 algorithm (Quinlan, 1979, 1986). We use CART for a classification tree that predicts relative (non-)winners based on a range of features derived from the ETF price data. The biggest benefit of DT is that they are transparent, returning a set of decision rules that give a full and unambiguous view on the classification process. However, DT are prone to over-fitting. Random Forests provide a solution to this problem.

B.2 Random forests

Breiman (2001) introduces Random Forests (RF), which consist of multiple randomized decision trees. RF are used as a meta estimator to improve predictive accuracy and avoid over-fitting. It averages over a specified number of DTs which are fitted on different sub-samples of the data. This makes RFs effectively ensemble methods that uses multiple simple classifiers (DTs) that would otherwise produce an unstable result across multiple sample, making the result more robust as the final prediction will be averaged out across the different DTs.

The downside main of RFs is that they lose the transparency of DTs, because it is more difficult to trace back the classification rules through the large amount of randomized

DTs.

B.3 Neural networks

The idea of mimicking the brain through Neural Networks (NN) started with the Rosenblatt Perceptron, made up of a single artificial neuron (Rosenblatt, 1958, 1961). Multilayer perceptrons are more advanced versions of this basic model that allow to incorporate non-linear effects. As the name suggests, several layers containing neurons can be combined in various compositions depending on the number of layers and the number of neurons per layer. Each neuron has a bias b, and each connection between two nodes has a weight W. Hidden layers use an activation function, such as $tanh(\cdot)$ or sigmoid.

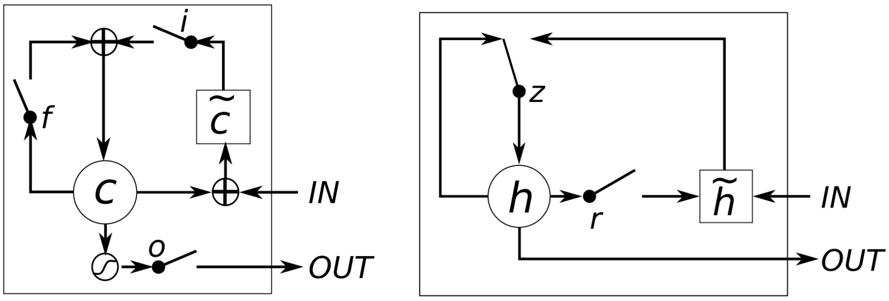
$$h_t^{(i)} = tanh\Big(W_h u_{t-\tau:t}^{(i)} + b_h\Big)$$

$$Z_t^{(i)} = g\Big(W_z u_t^{(i)} + b_z\Big)$$

The findings of Gu et al. (2020) suggest that Neural Networks (NN), more specifically 3-layer multi-layer perceptrons (MLP), result in the best out-of-sample predictive R^2. They find that machine learning forecasts of asset prices can significantly increase performance over leading linear methods because of their capacity to include non-linear predictor interactions. They test a wide array of ML methods ranging from boosted regression trees to various architectures of NNs. Lim et al. (2019) confirm the finding that neural networks can significantly improve benchmark linear strategies. Choi & Renelle (2019) combine DL clustering methods and recurrent NN and show a significantly enhanced Sharpe ratio.

B.4 Recurrent Neural Networks (RNN): LSTM and GRU

When using neural networks on time-series data, such as stock price data, storing information over longer times can help uncover patterns over time. RNN have short term memory, because they use persistent previous information in the current neural network. However, RNN can experience a loss in information, often referred to as the vanishing gradient problem (Hochreiter, 1998). After identifying the issue that storing information over extended time intervals via recurrent backpropagation takes a very long time (Hochreiter, 1991), Hochreiter & Schmidhuber (1997) proposed a gradient-based method to resolve this problem: Long Short Term Memory (LSTM). The LSTM model decides what information will be stored and what discarded using “gates”. In an LSTM model,



(a) LSTM (b) GRU

Figure B.1: Illustration of (a) LSTM and (b) gated recurrent unit gating mechanisms (Chung et al., 2014)

the recurrent weight matrix is replaced by an identify function in the carousel and controlled by a series of gates. Due to the model's ability to learn long term sequences of observations, LSTM has become a trending approach to time series forecasting.

The central unit in an LSTM is a memory cell, which gets input from a multiplicative output gate, and from another multiplicative input gate. The output of memory cell c_j at time t with internal state s_{c_j} is:

$$y^{c_j}(t) = y^{out_j}(t)h(s_{c_j}(t)),$$

where $s_{c_j}(0) = 0$ and $s_{c_j}(t) = s_{c_j}(t-1) + y^{in_j}(t)g(net_{c_j}(t))$ for $t > 0$.

The differentiable function g squashes net_{c_j}; the differentiable function h scales memory cell outputs computed from the internal state s_{c_j} (Hochreiter & Schmidhuber, 1997).

Another important set of RNN that use a type gating mechanism (similar to their LSTM counterpart) are GRU models. GRU models do not have an output gate and hence have fewer parameters (Cho et al., 2014). Chung et al. (2014) demonstrate the superiority of both types of gated units over a simple *tanh* unit. Relative performance of the LSTM and GRU model versions depend on the implementation and prediction task.

B.5 (Gaussian) Naive Bayes

Naive Bayes Classification (NBC) is a supervized learning algorithm used for classification, based on Bayes' Theorem, which in simplified notation posits that $P(class|data) = (P(data|class) * P(class))/P(data)$, where $P(class|data)$ is the probability of a class given the provided data. The priors for the class and the data can easily be estimated from a training dataset. However, calculating the conditional probability of an observation given a class would require a vast dataset containing enough observations for each individual combination of feature values in order to estimate the probability distribution, making it increasingly intractable as the number of features n increases (Rish et al., 2001).

Therefore, Naive Bayes relaxes the key assumption that each feature is dependent on all other features. As such, all conditionally dependent probabilities are dropped, leaving a simplified feature-independent calculation:

$$P(class|X_1, X_2, \ldots, X_n) = P(X_1|class) * P(X_2|class) * \ldots * P(X_n|class) * P(class)$$

Gaussian Naive Bayes is an extension of Naive Bayes that assumes that the input values follow a Guassian distribution. As such, it is an extension that allows for real-valued features of the input variables, as opposed to the probabilities in the model described above.

B.6 Support vector machines

The work of Boser et al. (1992) and Cortes & Vapnik (1995) laid the foundation for a supervized classification and regression algorithm that has evolved into what is now known as Support Vector Machnines (SVM). This class of algorithms is built around kernel functions, on which convex optimization is applied using statistical learning theory (Vapnik, 1999).

The essence of SVM is to find a hyperplane that best divides a dataset into two classes. This surface is called a maximal margin hyperplane, because the surface is chosen such that it is maximally distant from the two classes. Furthermore, SVM allow for a non-linear decision boundary in the original data space, by mapping the data into a higher dimensional feature space in which the decision surface will be linear (Cristianini & Ricci, 2008).

B.7 Gradient boosting classification

Gradient Boosting is an algorithm that optimizes a differential loss function, for which in each iteration the negative gradient of the deviance loss function is calculated, on which subsequently regression trees are fit. The algorithm works by fitting an additive model (ensemble) in a forward stage-wise manner. In each stage, a weak learner is introduced that compensates the shortcomings of existing weak learners: $F(X) = \alpha_1 h_1(X) + \alpha_2 h_2(X) + ... + \alpha_t h_t(X)$, where α is a real value and h is a weak model. In Gradient Boosting specifically, "shortcomings" are identified by gradients of the loss. An important benefit of gradient boosting is that it is effective at avoiding under- or overfitting (Bühlmann & Hothorn, 2007).

Appendix C

Implementation of ML models

C.1 Linear regression

Our linear regression model will predict the returns of a financial asset for a specific time period. After we have predicted the return, we can use the same algorithm to determine whether an asset is a "winner" as used in the feature generation phase of our data. As such, we will label a given observation as being a "winner" when it belongs to the 50th percentile of the full universe of assets that we use to build our model on.

This implementation method is chosen over a classification method (where a numerical value would be predicted with 1 for the winner and 0 for the loser category), because classification would give a larger error and logistic regression is better suited to a conversion from numerical to binary classification thanks to its target variable limits between 0 and 1.

C.2 Logistic regression

Logistic regression is a second linear method used to predict the classification of assets. Similar to how we used linear regression for binary classification, we will convert the target variable from class labels into the probability that a given asset will be labeled a "winner" or "loser". As such, we will use a cutoff value of 0.5 to determine whether the model predicts a given observation to be a "winner" after the logistic regression model has been applied.

C.3 Decision trees

We used sklearn's *DecisionTreeClassifier* to construct a decision tree based on a large set of hyperparameters. The decision tree classifier is a multi-class classification tool that is based on the more general decision tree, which is a non-parametric supervized learning method that predicts the value for a target variable by applying simple decision rules based on the features. For our specific application, we used the decision tree classifier as a binary classifier to predict whether each specific asset will be either a winner or a loser in the following time period.

Similar to what we did in the previous methods, the monthly time features will be used to predict the future class label. However, for decision trees, we will no longer predict the probability that a given observation belongs to a class, but rather we will directly classify all observations in a specific leaf node to its corresponding class. The probability estimates will still be used in the evaluation of our model accuracy, even though they will not be used in the actual classification of the assets.

Given the large set of hyperparameters that we could possibly tune to improve the classification accuracy of our model, we used sklearn's default parameters for some of the hyperparameters, whilst we applied hyperparameter optimization techniques to the most important ones, which will be discussed further in the next section.

C.4 Random forests

Very similar to DTs in terms of application and hyperparameter optimization, we used sklearn's *RandomForestClassifier* to construct a random forest model. As already discussed in the literature review, RFs are effectively ensemsble methods that combine multiple DTs that are each based on a specific sub-sample on the data. The final prediction of the RF is the average taken from the different DTs. For the binary classification variant, each decision tree predicts a class label and the RF will effectively take the most occurring class label amongst the predictions from the DTs as the final class label for the target variable.

We used defaults for some of the hyperparameters, whilst other hyperparameters that could potentially have a significant impact on the model quality were optimized using hyperparameter optimization discussed below.

Classification within the DTs will be done exactly the same as how we did this for decision trees. This means that, based on the time features, we will derive simple decision rules that will classify assets into one of two classes.

C.5 (Gaussian) Naive Bayes

Similar to the previous methods, (Gaussian) Naive Bayes will be used to predict the probability of observations belonging to each class based on a set of return features. An important note is that Naive Bayes assumes Independence between the features, which is a strong assumption that we can certainly not make in the case of our return features as this would be contradictory to our momentum assumption. Nevertheless, empirical studies have found that Naive Bayes can still be effective for classification when this independence assumption is violated Rish et al. (2001). As such, we will include Naive Bayes in our analysis and observe in a later section that indeed, the classification accuracy will be similar to these of other ML techniques.

C.6 Support vector machines

For SVMs, the methodology in terms of application and hyperparamter optimization will be almost identical to DT, RF and (Gaussian) NB.

C.7 Gradient boosting classification

For GBC, the methodology in terms of application and hyperparamter optimization will be almost identical to DT, RF, (Gaussian) NB, and SVMs.

C.8 Recurrent Neural Networks: LSTM and GRU

In our LSTM and GRU architecture, we use stock price data of individual stocks as input in order to predict a 30-day forward momentum return, which is then cross-sectionally classified into (non-)winners.

First, the time-series of all instruments are MinMax scaled and pre-processed into batches with a 360 days look-back period of independent variable train data, complemented with a 30-day forward price as the dependent variable. Then, we use an architecture with an input layer, 2 hidden layers, and a dense output layer with the same shape as the input layer, in order to get parallel predictions for each of the instruments that we feed into the model. We use Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions (Kingma & Ba, 2014), with a Mean Squared Error (MSE) loss function. Predicted prices are then re-scaled and converted to 30-day forward returns.

Appendix D

Data

D.1 Data selection and cleaning

The reason we use this specific list is that InvestSuite serves clients that are mainly looking to invest in a diversified set of equities that are relatively cheap to invest in in terms of costs. As such, since the large majority of their client base is European, they mainly look at European listed products, as the costs associated with these are lower as compared to non-European investible products. Note that these ETFs, despite being traded on the European markets only, can still cover non-European products. Also, these ETFs can include (a certain proportion of) non-equity products such as bonds or cash, for which we will add a constraint to our iVaR optimization for total exposure in a later stage. More specifically, we will instruct the optimizer to have at least 50% exposure to equities. This means that for each fund, we will also take into account the proportion of equity vs. non-equity products that the ETF holds.

Working with ETFs rather than individual assets gives us a high degree of diversification in our portfolio. As such, we will not worry about diversifying our portfolio and only focus on the risk-return characteristics of our portfolio as driven by momentum and iVaR-related metrics.

In this first step of data selection, we will simply load in the daily prices, adjusted for corporate events such as stock splits and mergers for the constituents of the ETFs. This data will be pulled from Morningstar and Thomson Reuters using InvestSuites's existing modules and licences.

D.1.1 Treatment of missing values

As our asset universe exists of more than 100 assets, we will choose to simply drop all assets for which we could not calculate at least one of the returns features as defined in the subsection below. As the data has already been cleaned to some extent by the data providers (Morningstar and Thomson Reuters), this will only lead to a very small reduction of our asset universe, even for longer time periods.

As the reason for data to be missing in our set is most likely the limited lifetime of some of our assets, we can be confident that removing such observations will not lead to removing observations with significant impact on the model.

This code will be applied at each iteration of model construction, meaning that as we move ahead in time, we expect more assets to be included in the model training, potentially leading to an increasing model quality.

D.1.2 Feature engineering

As discussed above, we are transforming the data for technical reasons: we need to add return features to the data set to base our predictive models on. In practice, this means that we will calculate the returns over a given set of time periods (e.g. past month, the month before that...) and add them to our data set. After replacing the existing daily returns by these monthly returns, we will end up with a data set consisting of n historical monthly returns where r represents the number months between the start date and the date at which we are observing the data.

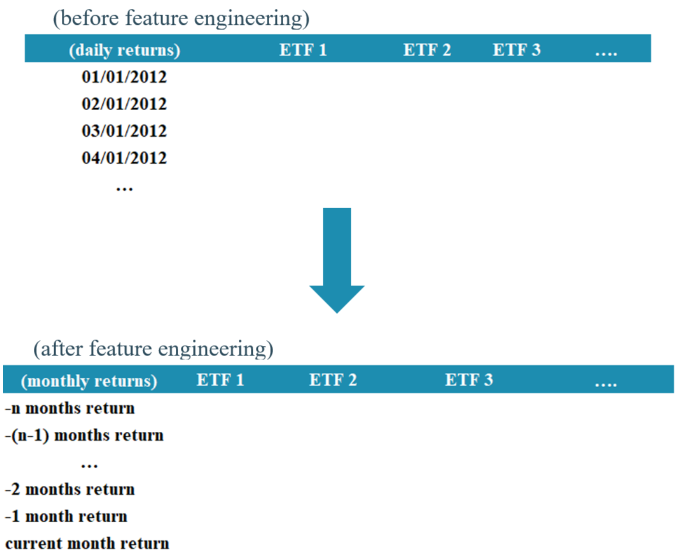


Figure D.1: Data set before and after feature engineering

We also ran experiments in which we added a broad range of non-pricing features, including all qualitative and quantitative asset-related characteristics such as region, trading volume, asset composition. However, as this did not improved our results in terms of prediction accuracy measures (AUC) significantly in early stages, we decided to keep the focus of our research limited to these time features. Moreover, restricting our results to time features only, we were able to single out the predictive power of returns in previous time periods for the return in the subsequent time period, hence keeping our focus to the central idea of momentum. Lastly, taking into account a larger number of features would significantly increase the computational requirements of classifying the momentum portfolio and running the corresponding backtests, as this step of predicting the binary classification occurs at every single step in both procedures.

D.2 ETF list

ISIN	ETF Name
LU0629460675	UBS(Lux)Fund Solutions - MSCI EMU Socially Responsible UCITS ETF(EUR)A-dis
LU1681040900	Amundi Index Solutions - Amundi Floating Rate USD Corporate ESG - UCITS ETF...
LU0274211480	Xtrackers DAX UCITS ETF 1C
LU1686830065	Lyxor Index Fund - Lyxor EuroMTS Covered Bond Aggregate UCITS ETF Acc
IE00B579F325	Invesco Physical Gold ETC
IE00B53QG562	iShares VII PLC -iShares Core MSCI EMU UCITS ETF EUR (Acc)
IE00B4M7GH52	iShares MSCI Poland UCITS ETF USD (Acc)
IE00BM67HR47	Xtrackers MSCI World Communication Services UCITS ETF 1C
IE00B8GKDB10	Vanguard FTSE All-World High Dividend Yield UCITS ETF USD Distributing
LU1190417599	Lyxor Smart Cash - UCITS ETF C-EUR

LU0908508731	Xtrackers II Global Government Bond UCITS ETF 5C
IE00BYXYYL56	iShares $ High Yield Corp Bond UCITS ETF USD (Acc)
DE000A0H0785	iShares Euro Government Bond Capped 1.5-10.5yr UCITS ETF (DE)
LU0524308870	Universal Invest Medium A Acc
IE00B3VWMM18	iShares VII PLC - iShares MSCI EMU Small Cap ETF EUR Acc
LU1681044720	Amundi Index Solutions - Amundi MSCI Switzerland UCITS ETF-C EUR
IE00BM67HS53	Xtrackers MSCI World Materials UCITS ETF 1C
IE00BQQP9H09	VanEck Morningstar US Sustainable Wide Moat UCITS ETF
IE00B4L5Y983	iShares Core MSCI World UCITS ETF USD (Acc)
LU0524307047	Universal Invest Low B Acc
IE00B3B8PX14	iShares Global Inflation Linked Govt Bond UCITS ETF USD (Acc)
IE00BTJRMP35	Xtrackers MSCI Emerging Markets UCITS ETF 1C
IE00BZ0PKT83	iShares Edge MSCI World Multifactor UCITS ETF USD (Acc)
IE00B3VWN393	iShares $ Treasury Bond 3-7yr UCITS ETF USD (Acc)
IE00BP3QZ601	iShares Edge MSCI World Quality Factor UCITS ETF USD (Acc)
IE00BF11F565	iShares Core â,¬ Corp Bond UCITS ETF EUR (Acc)
IE00B1TXHL60	iShares Listed Private Equity UCITS ETF USD (Dist)
IE00BF3N7094	iShares â,¬ High Yield Corp Bond UCITS ETF EUR (Acc)
IE00BZ0PKV06	iShares Edge MSCI Europe Multifactor UCITS ETF EUR (Acc)
IE00BYXPSP02	iShares $ Treasury Bond 1-3yr UCITS ETF USD (Acc)
LU0322253229	Xtrackers S&P Global Infrastructure Swap UCITS ETF 1C
IE00B48X4842	SPDRÂ® MSCI Emerging Markets Small Cap UCITS ETF
LU0460391732	Xtrackers Bloomberg Commodity ex-Agriculture & Livestock Swap UCITS ETF 2C USD
LU0290357507	Xtrackers II Eurozone Government Bond 15-30 UCITS ETF 1C
IE00B802KR88	SPDRÂ® S&P 500 Low Volatility UCITS ETF
IE00B44Z5B48	SPDRÂ® MSCI ACWI UCITS ETF
BE0146659926	Sivek - Global Medium Cap
FR0010754127	Amundi ETF Euro Inflation UCITS ETF DR
IE00B86MWN23	iShares Edge MSCI Europe Minimum Volatility UCITS ETF EUR (Acc)
IE00BFZPF546	iShares J.P. Morgan EM Local Govt Bond UCITS ETF USD (Acc)
IE00BYVJRP78	iShares MSCI EM SRI UCITS ETF USD (Acc)
BE0146661948	Sivek - Global Low Cap
BE0146932745	BNP Paribas B Strategy - Global Sustainable Conservative Classic-Capitalisation
BE0146934766	BNP Paribas B Strategy - Global Sustainable Defensive Classic-Capitalisation
LU1691909508	Lyxor Global Gender Equality (DR) UCITS ETF - C-USD
IE00B8FHGS14	iShares Edge MSCI World Minimum Volatility UCITS ETF USD (Acc)
IE00B3VWM098	iShares VII PLC - iShares MSCI USA Small Cap ETF USD Acc
BE0131576440	Belfius Fullinvest - Low C Acc
LU0290356954	Xtrackers II Eurozone Government Bond 3-5 UCITS ETF 1C
IE00B4K48X80	iShares Core MSCI Europe UCITS ETF EUR (Acc)
LU1291109616	BNP Paribas Easy Energy & Metals Enhanced Roll UCITS ETF EUR Cap
IE00BM67HK77	Xtrackers MSCI World Health Care UCITS ETF 1C
IE00BM67HL84	Xtrackers MSCI World Financials UCITS ETF 1C
IE00BFM6TB42	iShares Global Corp Bond UCITS ETF USD (Acc)
IE00B5BMR087	iShares Core S&P 500 UCITS ETF USD (Acc)
LU0524311072	Universal Invest High A Acc
LU0478205379	Xtrackers II EUR Corporate Bond UCITS ETF 1C
IE00B3VWN518	iShares VII PLC - iShares $ Treasury Bd 7-10y ETF USD Acc
LU0322250712	Xtrackers LPX Private Equity Swap UCITS ETF 1C
IE00BNH72088	SPDR Refinitiv Global Convertible Bond UCITS ETF
IE00BDGN9Z19	Xtrackers MSCI EMU ESG Screened UCITS ETF 1D
IE00BP3QZ825	iShares Edge MSCI World Momentum Factor UCITS ETF USD (Acc)
IE00BM67HT60	Xtrackers MSCI World Information Technology UCITS ETF 1C

IE00B8KGV557	iShares Edge MSCI EM Minimum Volatility UCITS ETF USD (Acc)
IE00BM67HV82	Xtrackers MSCI World Industrials UCITS ETF 1C
LU0290355717	Xtrackers II Eurozone Government Bond UCITS ETF 1C EUR
BE0146657904	Sivek - Global High Cap
LU0290356871	Xtrackers II Eurozone Government Bond 1-3 UCITS ETF 1C
LU0952581584	Xtrackers II Japan Government Bond UCITS ETF 1C
IE00BM67HP23	Xtrackers MSCI World Consumer Discretionary UCITS ETF 1C
BE0131577455	Belfius Fullinvest - Medium C Acc
LU0290357259	Xtrackers II Eurozone Government Bond 7-10 UCITS ETF 1C
LU1681038243	Amundi Index Solutions - Amundi Nasdaq-100 ETF-C EUR
LU0489337690	Xtrackers FTSE Developed Europe Real Estate UCITS ETF 1C
IE00BM67HQ30	Xtrackers MSCI World Utilities UCITS ETF 1C
IE00BZ036H21	Xtrackers USD Corporate Bond UCITS ETF 1D
IE00BD4DX952	Xtrackers ESG USD Emerging Markets Bond Quality Weighted UCITS ETF 1D
LU0514695690	Xtrackers MSCI China UCITS ETF 1C
IE00BYVJRR92	iShares MSCI USA SRI UCITS ETF USD (Acc)
IE00BM67HN09	Xtrackers MSCI World Consumer Staples UCITS ETF 1C
IE00B5L01S80	HSBC FTSE EPRA NAREIT Developed UCITS ETF
LU0274209740	Xtrackers MSCI Japan UCITS ETF 1C
IE00BZ0PKS76	iShares Edge MSCI USA Multifactor UCITS ETF USD (Acc)
BE0146936787	BNP Paribas B Strategy - Global Sustainable Neutral Classic-Capitalisation
BE0131578461	Belfius Fullinvest - High C Acc
IE00B52MJY50	iShares VII PLC - iShares Core MSCI Pac ex-Jpn ETF USD Acc
LU0290357176	Xtrackers II Eurozone Government Bond 5-7 UCITS ETF 1C
IE00BLNMYC90	Xtrackers S&P 500 Equal Weight UCITS ETF 1C
LU0838780707	Xtrackers FTSE 100 UCITS ETF 1C
IE00BL25JM42	Xtrackers MSCI World Value UCITS ETF 1C
IE00BYM11K57	UBS (Irl) Fund Solutions plc - MSCI ACWI SF UCITS ETF (hedged to EUR) A-acc
IE00BDFL4P12	iShares Diversified Commodity Swap UCITS ETF
IE00BDT6FP91	SPDR Refinitiv Global Convertible Bond EUR Hdg UCITS ETF (Acc)
IE00BF4RFH31	iShares MSCI World Small Cap UCITS ETF USD (Acc)
IE00BM67HM91	Xtrackers MSCI World Energy UCITS ETF 1C
IE00B2QWDY88	iShares MSCI Japan Small Cap UCITS ETF USD (Dist)
LU0378818131	Xtrackers II Global Government Bond UCITS ETF 1C - EUR Hedged
LU1602144229	Amundi MSCI World Climate Transition CTB - UCITS ETF DR - EUR-C
BE0163304539	BNP Paribas B Strategy - Global Sustainable Dynamic Classic-Capitalisation
IE00BYXYYK40	iShares J.P. Morgan $ EM Bond UCITS ETF USD (Acc)
IE00B44CND37	SPDRÂ® Bloomberg Barclays U.S. Treasury Bond UCITS ETF
LU0322253906	Xtrackers MSCI Europe Small Cap UCITS ETF 1C
IE00BYX2JD69	iShares MSCI World SRI UCITS ETF EUR (Acc)

Table D.1: Full list of European ETFs included in our research

Appendix E

Evaluation metrics

E.1 Sensitivity

The sensitivity (true positive rate (TPR), recall) is the proportion of winners that we are able to predict correctly using our classification models. As such, it is calculated as:

$$TPR = \frac{tp}{tp + fn}$$

E.2 Specificity

The specificity (true negative rate (TNR), selectivity) is the proportion of losers that we are able to predict correctly using our classification models. As such, it is calculated as:

$$TNR = \frac{tn}{tn + fp}$$

E.3 1-specificity

The 1-specificity (false positive rate (FPR)) is the proportion of winners that we did not classify correctly using our classification models. As such, it is calculated as:

$$FPR = \frac{fp}{fp + tn} = 1 - TNR$$

E.4 Area under the ROC-curve

The ROC-curve aggregates classification performance across multiple classification thresholds. It plots the sensitivity (true positive rate) against the false positive rate (1-specificity) for different levels of the classification threshold. The classification threshold is the level we use to classify given instances as “winners” vs “losers”, after we have predicted a probability using our classification models.

The Area under the ROC-curve (AUC) is a classification evaluation metric that gives the probability that a new instance that is an actual winner will be ranked higher than an actual loser. In practice, it is calculated as the area under the ROC-curve.

Appendix F

Hansen's Model Confidence Set

Hansen's MSC procedure can be used to determine the set of models, V^*, that consists of the best model(s) from a collection of models, V^0, based on a pre-specified criterion, with a given level of confidence (Hansen et al., 2011). The set of best models can contain several models in the case where they do not statistically differ from the best-in-class based on the chosen measure.

We define a loss function $d_{ij,t} = L_{i,t}L_{j,t}$ between models i and j from our set of models V^0 at some point in time t in our backtest. Alternative models are then ranked in terms of expected loss, so that model i is preferred to model j if $\mathbf{E}(d_{ij,t}) < 0$. We then define the set of superior models as:

$$V^* = \{i \in V^0 \text{ for } \mathbf{E}(d_{ij,t}) \leq 0 \text{ for all } j \in V^0\}.$$

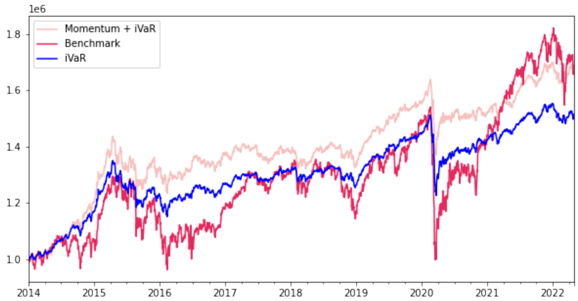
Therefore, Hansen's MSC is essentially a sequence of significance tests, where objects are compared against the null hypothesis $H_{0,V} : \mathbf{E}(d_{ij,t}) = 0$ for all $i, j \in V$ (Hansen et al., 2011).

In our research, we will evaluate models based on several criteria: returns, maximum and average drawdown, Calmar and pain ratio. For each of these criteria, the value is calculated for each of the portfolios at each month over the duration of the backtest, resulting in a list of values that can then be compared over time using Hansen's statistical test. The return statistic is calculated as the simple monthly percentage change in portfolio value. The maximum drawdown is calculated as the maximum of the percentage drawdown from the high watermark at that time. Equally, the average drawdown is calculated as the mean over time of the percentage drawdown from the high watermark. The Calmar ratio is calculated as the monthly return divided by the monthly maximum drawdown. The pain ratio is calculated as the monthly return divided by the monthly average drawdown.

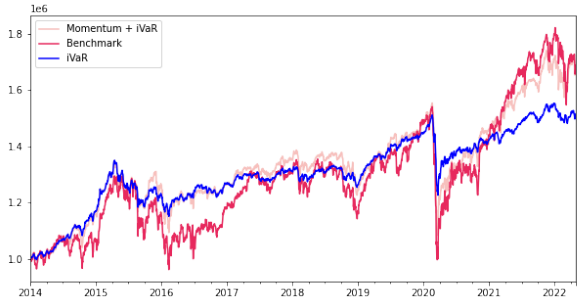
Appendix G

ML classification-based indexed portfolio returns, yearly returns and drawdowns

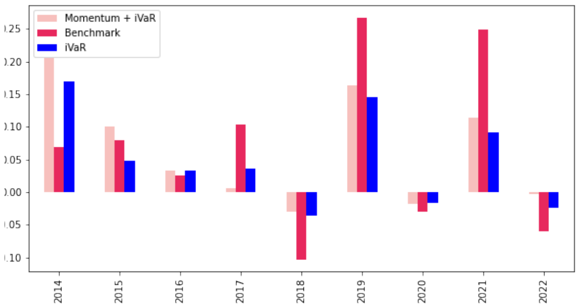
G.1 Linear model



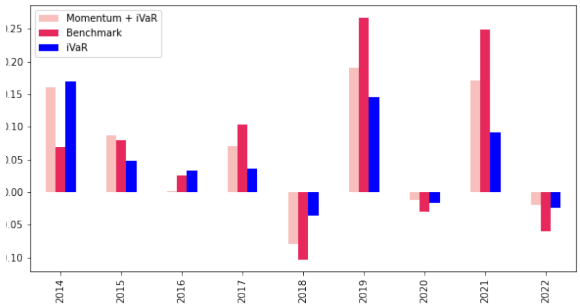
(a) Constraint-based: indexed value



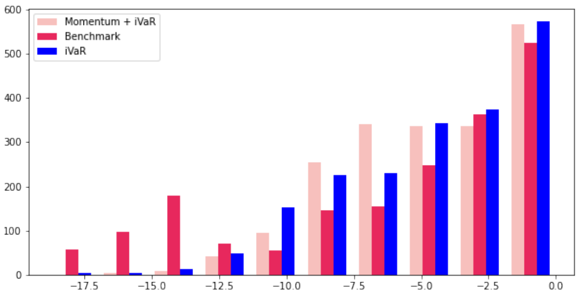
(b) Objective-based: indexed value



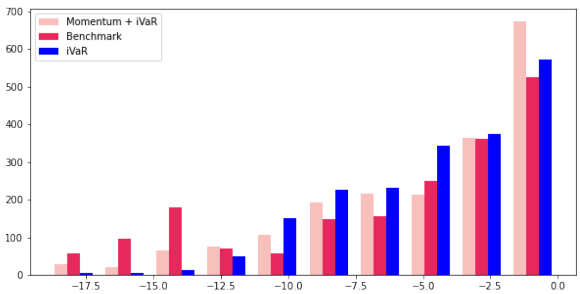
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



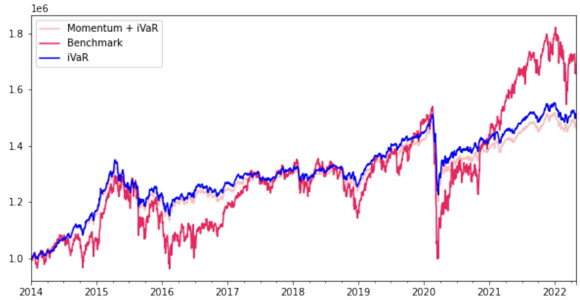
(e) Constraint-based: drawdowns



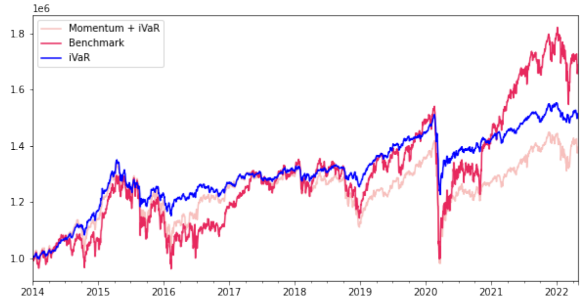
(f) Objective-based: drawdowns

Figure G.1: Linear model

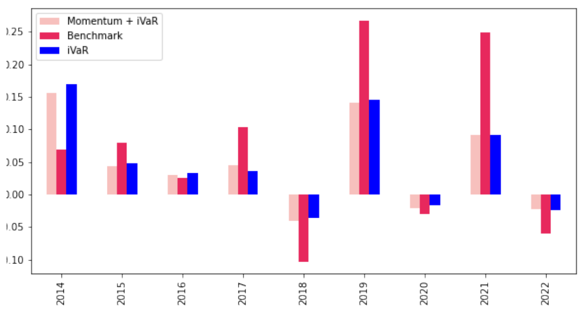
G.2 Logistic model



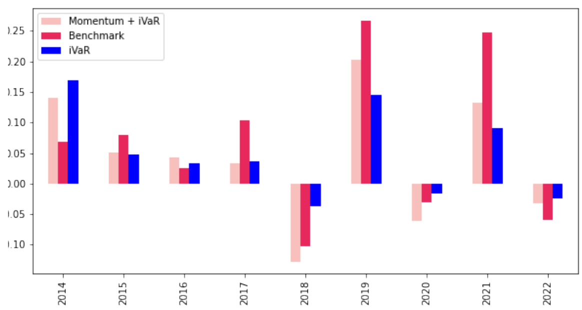
(a) Constraint-based: indexed value



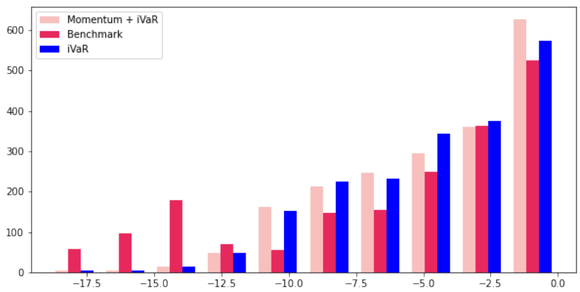
(b) Objective-based: indexed value



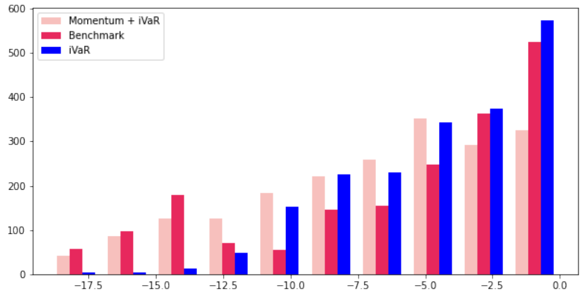
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



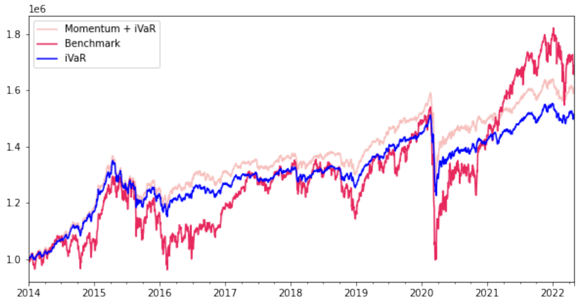
(e) Constraint-based: drawdowns



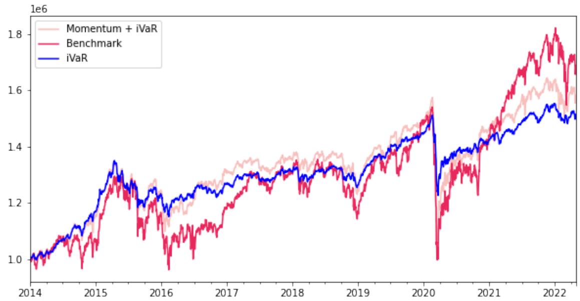
(f) Objective-based: drawdowns

Figure G.2: Logistic model

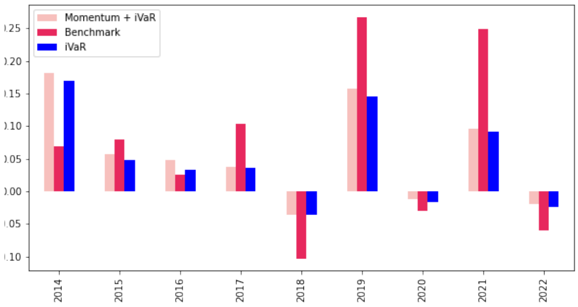
G.3 Decision tree model



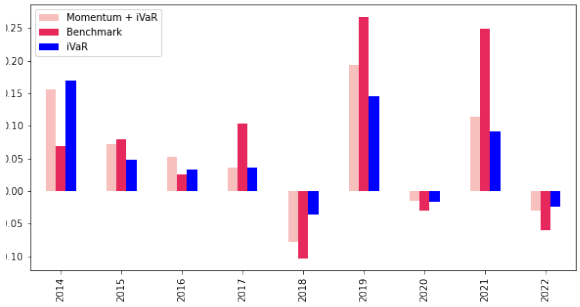
(a) Constraint-based: indexed value



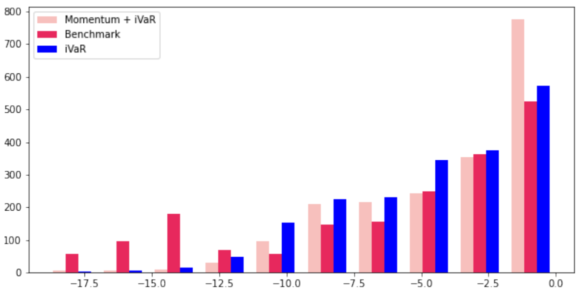
(b) Objective-based: indexed value



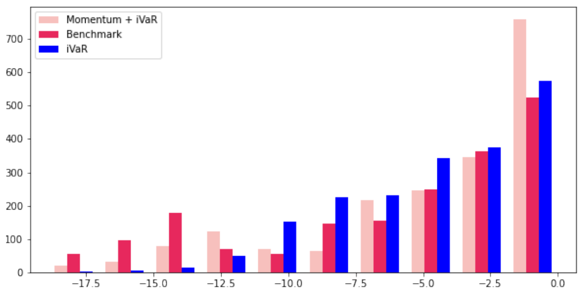
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



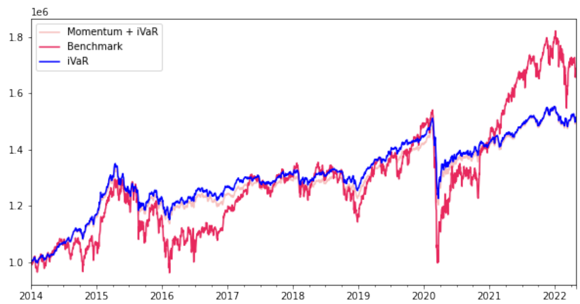
(e) Constraint-based: drawdowns



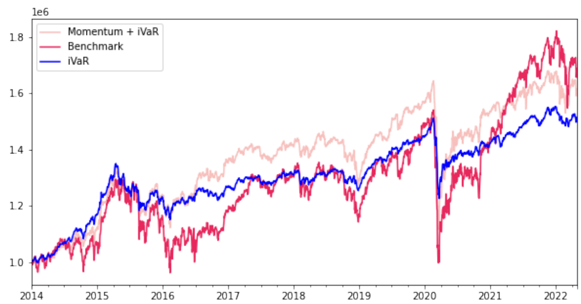
(f) Objective-based: drawdowns

Figure G.3: Decision Tree model

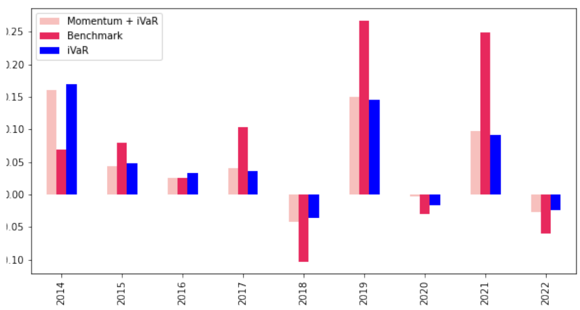
G.4 Random forest model



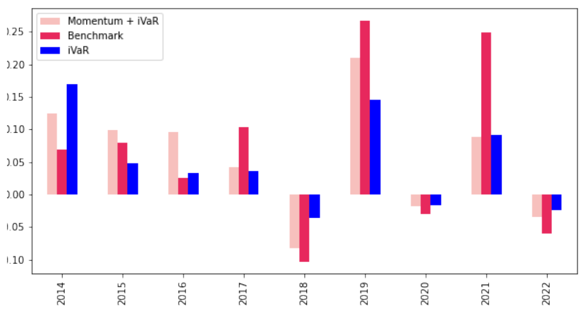
(a) Constraint-based: indexed value



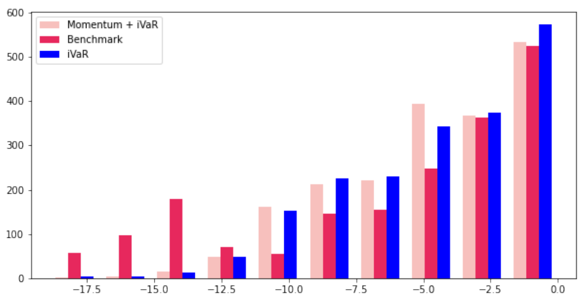
(b) Objective-based: indexed value



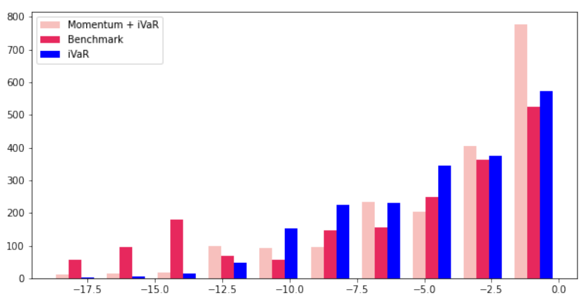
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



(e) Constraint-based: drawdowns

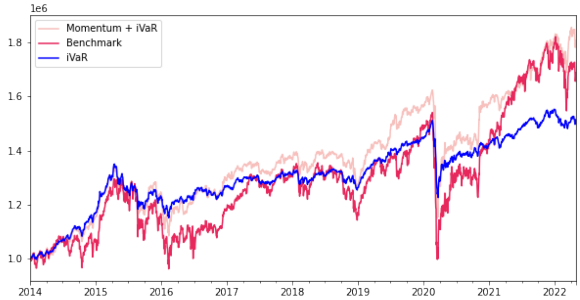


(f) Objective-based: drawdowns

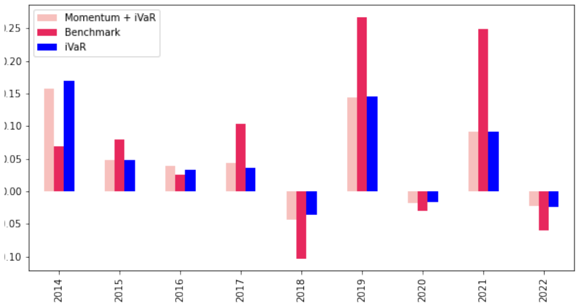
Figure G.4: Random Forest model

G.5 (Gaussian) NB model

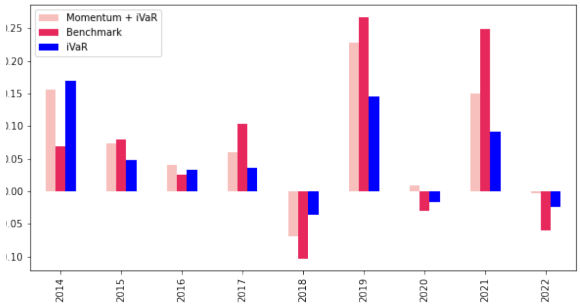
(a) Constraint-based: indexed value



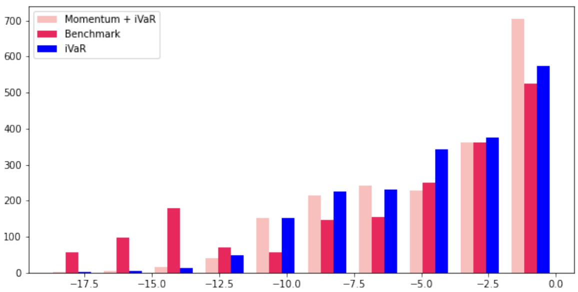
(b) Objective-based: indexed value



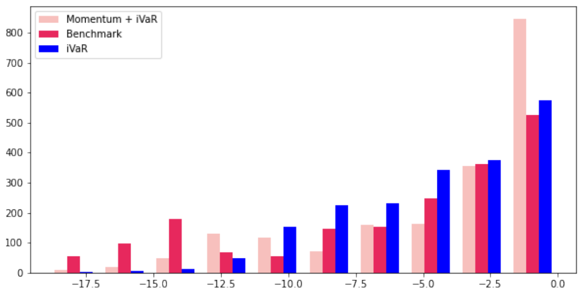
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



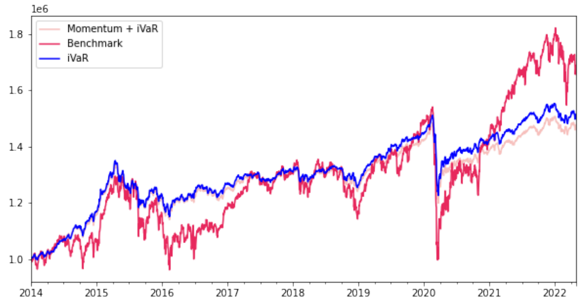
(e) Constraint-based: drawdowns



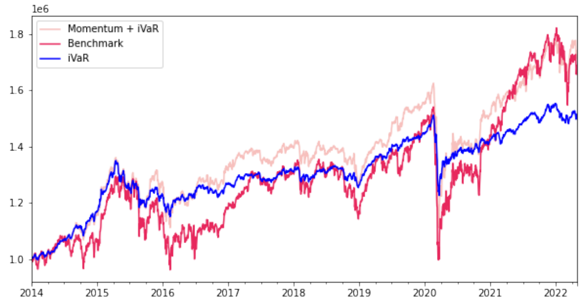
(f) Objective-based: drawdowns

Figure G.5: (Gaussian) NB model

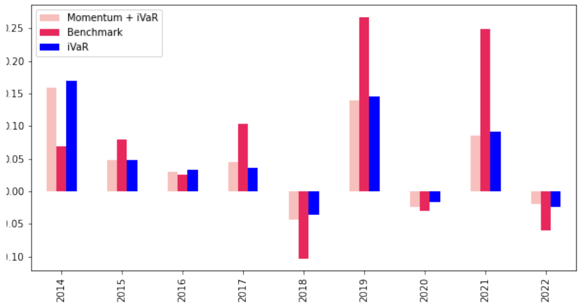
G.6 SVM model



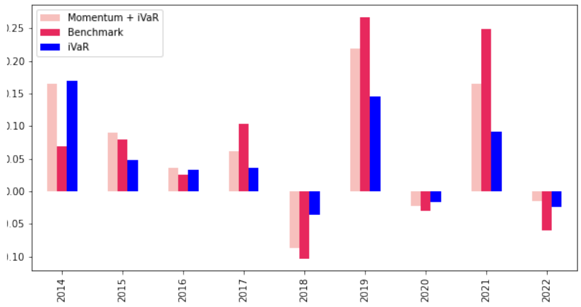
(a) Constraint-based: indexed value



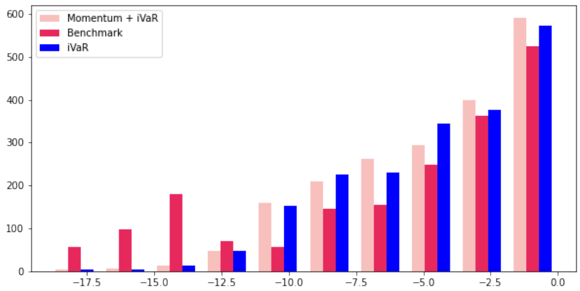
(b) Objective-based: indexed value



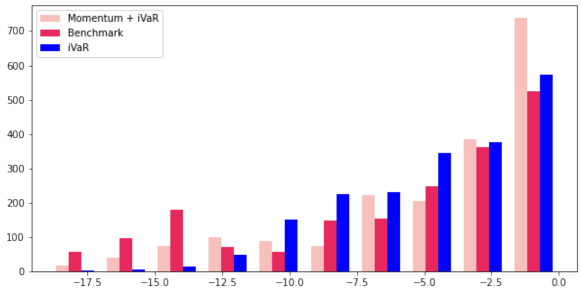
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



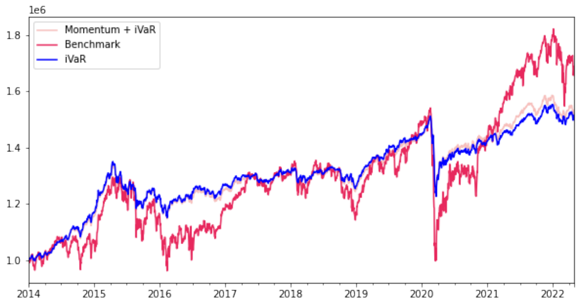
(e) Constraint-based: drawdowns



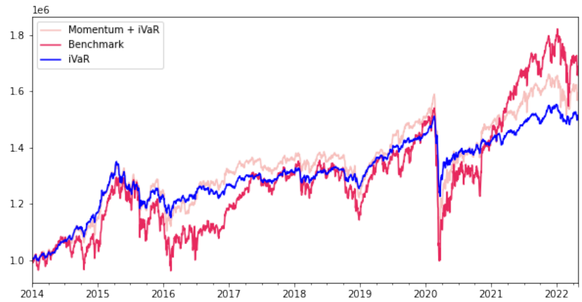
(f) Objective-based: drawdowns

Figure G.6: SVM model

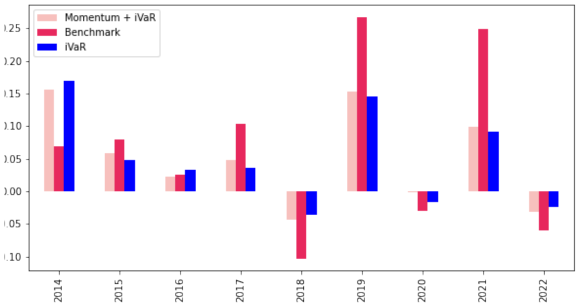
G.7 GBC model



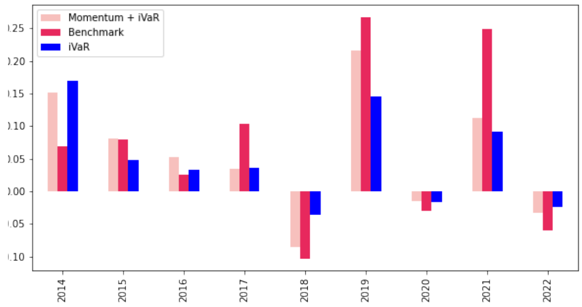
(a) Constraint-based: indexed value



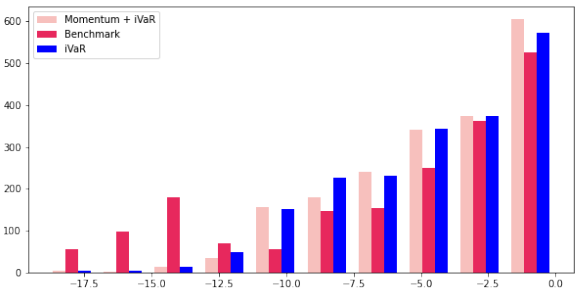
(b) Objective-based: indexed value



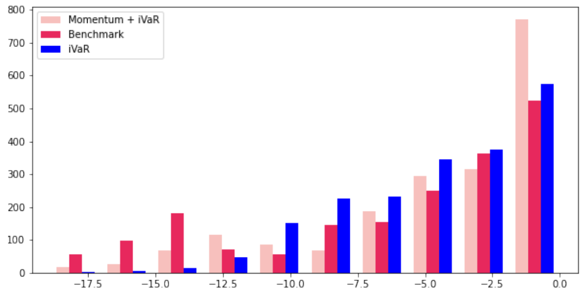
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



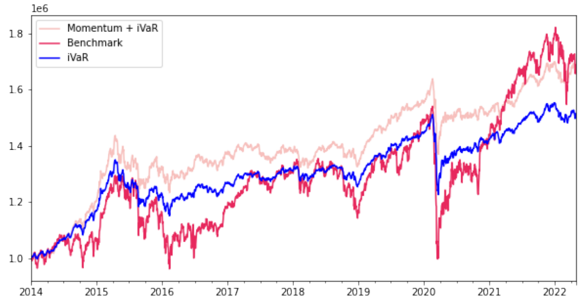
(e) Constraint-based: drawdowns



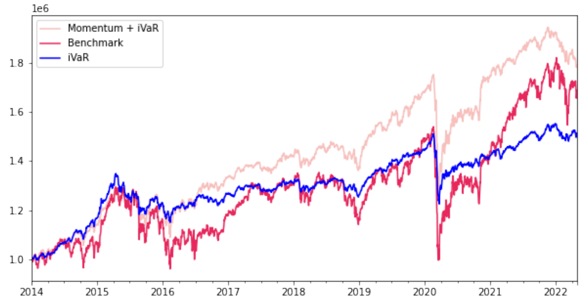
(f) Objective-based: drawdowns

Figure G.7: GBC model

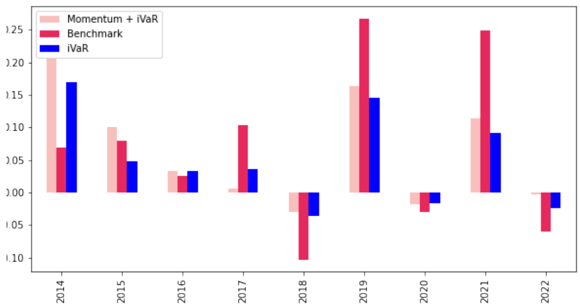
G.8 GRU model



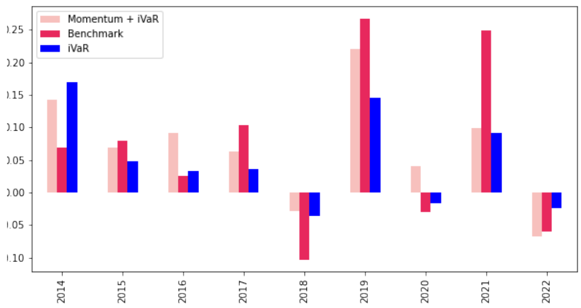
(a) Constraint-based: indexed value



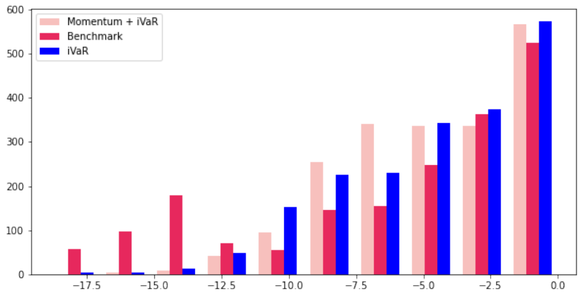
(b) Objective-based: indexed value



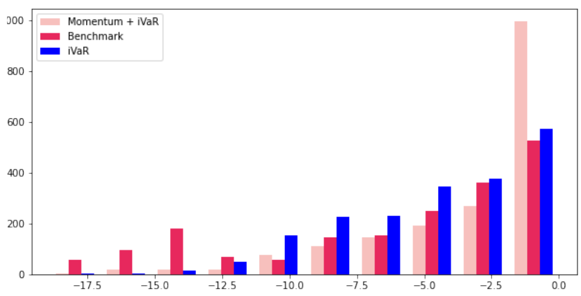
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



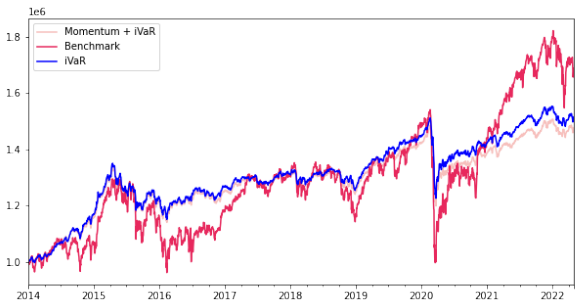
(e) Constraint-based: drawdowns



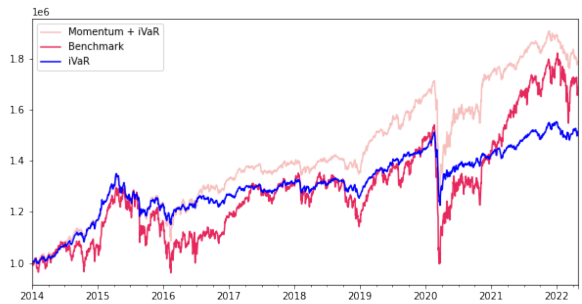
(f) Objective-based: drawdowns

Figure G.8: GRU model

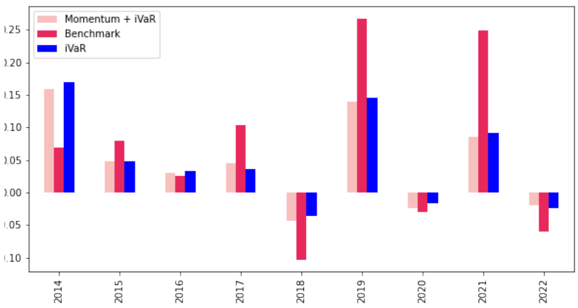
G.9 LSTM model



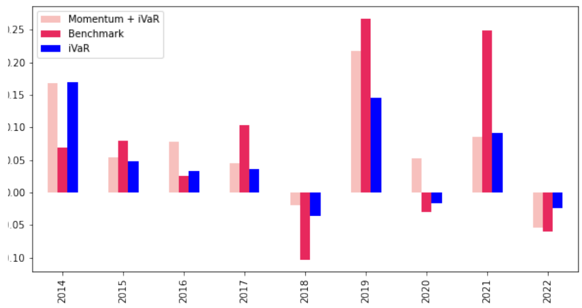
(a) Constraint-based: indexed value



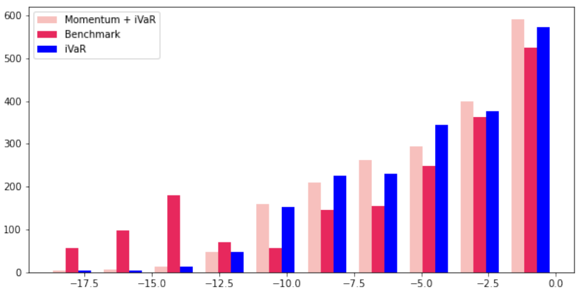
(b) Objective-based: indexed value



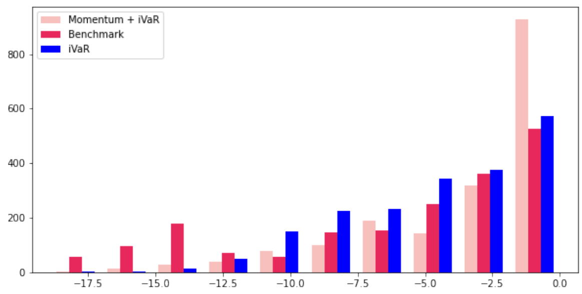
(c) Constraint-based: yearly returns



(d) Objective-based: yearly returns



(e) Constraint-based: drawdowns



(f) Objective-based: drawdowns

Figure G.9: LSTM model

References

Andreu, L., Swinkels, L., & Tjong-A-Tjoe, L. (2013). Can exchange traded funds be used to exploit industry and country momentum? *Financial Markets and Portfolio Management*, *27*(2), 127–148.

Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, *9*(3), 203–228. Retrieved from https://doi.org/10.1111/1467-9965.00068 doi: 10.1111/1467-9965.00068

Asness, C. S. (1994). Variables that explain stock returns: Simulated and empirical evidence. *Ph. D. Dissertation*. (Graduate School of Business, University of Chicago, Chicago, Ill.)

Asness, C. S., Moskowitz, T. J., & Pedersen, L. H. (2013). Value and momentum everywhere. *The Journal of Finance*, *68*(3), 929–985.

Baltas, N., & Kosowski, R. (2020). Demystifying time-series momentum strategies: Volatility estimators, trading rules and pairwise correlations. *Market Momentum: Theory and Practice", Wiley*.

Barroso, P., & Santa-Clara, P. (2015). Momentum has its moments. *Journal of Financial Economics*, *116*(1), 111–120.

Baz, J., Granger, N., Harvey, C. R., Le Roux, N., & Rattray, S. (2015). Dissecting investment strategies in the cross section and time series. *SSRN Electronic Journal*. (Available at SSRN 2695101)

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth.

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical science*, *22*(4), 477–505.

Cao, R., Horváth, L., Liu, Z., & Zhao, Y. (2020). A study of data-driven momentum and disposition effects in the chinese stock market by functional data analysis. *Review of Quantitative Finance and Accounting*, *54*(1), 335–358.

Chekhlov, A., Uryasev, S. P., & Zabarankin, M. (2003). Drawdown measure in portfolio optimization. *SSRN Electronic Journal*. Retrieved from https://doi.org/10.2139/ssrn.544742 doi: 10.2139/ssrn.544742

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Choi, S., & Renelle, T. (2019). Deep learning price momentum in us equities. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1-8.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Conrad, J., & Kaul, G. (1998). An anatomy of trading strategies. *The Review of Financial Studies*, *11*(3), 489–519.

Cooper, M. J., Gutierrez Jr, R. C., & Hameed, A. (2004). Market states and momentum. *The journal of Finance*, *59*(3), 1345–1365.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cristianini, N., & Ricci, E. (2008). Support vector machines. In M.-Y. Kao (Ed.), *Encyclopedia of algorithms* (pp. 928–932). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-0-387-30162-4_415 doi: 10.1007/978-0-387-30162-4_415

Daniel, K. D., & Moskowitz, T. J. (2013). Momentum crashes. *Swiss Finance Institute Research Paper*(13-61), 14–6.

De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.

Drummen, M., & Zimmermann, H. (1992). The structure of european stock returns. *Financial Analysts Journal*, *48*(4), 15–26.

Elton, E. J., & Gruber, M. J. (1992). Portfolio analysis with a nonnormal multi-index return-generating process. *Review of Quantitative Finance and Accounting*, *2*(1), 5–16.

Elton, E. J., & Gruber, M. J. (1997). Modern portfolio theory, 1950 to date. *Journal of banking & finance*, *21*(11-12), 1743–1759.

Elton, E. J., Gruber, M. J., & Padberg, M. W. (1976). Simple criteria for optimal portfolio selection. *The Journal of Finance*, *31*(5), 1341–1357.

Fama, E. F. (1960). Efficient market hypothesis. *Ph. D. Dissertation.* (University of Chicago Graduate School of Business)

Freyberger, J., Neuhierl, A., & Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, *33*(5), 2326–2377.

Goetzmann, W., Ingersoll, J., Spiegel, M., & Welch, I. (2007). Portfolio performance manipulation and manipulation-proof performance measures. *The Review of Financial Studies*, *20*(5), 1503–1546.

Green, J., Hand, J. R., & Zhang, X. F. (2017). The characteristics that provide independent information about average us monthly stock returns. *The Review of Financial Studies*, *30*(12), 4389–4436.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Guobužaitė, R., & Teresienė, D. (2021). Can economic factors improve momentum trading strategies? the case of managed futures during the covid-19 pandemic. *Economies*, *9*(2), 86.

Gupta, T., & Kelly, B. (2019). Factor momentum everywhere. *The Journal of Portfolio Management*, *45*(3), 13–36.

Han, C. (2021). Bimodal characteristic returns and predictability enhancement via machine learning. *Management Science*.

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, *91*(1).

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116.

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

Hong, H., & Stein, J. C. (1999). A unified theory of underreaction, momentum trading, and overreaction in asset markets. *The Journal of finance*, *54*(6), 2143–2184.

Hurst, B., Ooi, Y. H., & Pedersen, L. H. (2017). A century of evidence on trend-following investing. *The Journal of Portfolio Management*, *44*(1), 15–29.

Ingersoll, J. E. (1987). *Theory of financial decision making* (Vol. 3). Rowman & Littlefield.

InvestSuite. (2019). The human-centered, next-generation quantitative approach to constructing portfolios.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of finance*, *48*(1), 65–91.

Jegadeesh, N., & Titman, S. (2001). Profitability of momentum strategies: An evaluation of alternative explanations. *The Journal of finance*, *56*(2), 699–720.

Ke, Z. T., Kelly, B. T., & Xiu, D. (2019). *Predicting returns with text data* (Tech. Rep.). National Bureau of Economic Research.

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, *134*(3), 501–524.

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

Lewellen, J. (2002). Momentum and autocorrelation in stock returns. *The Review of Financial Studies*, *15*(2), 533–564.

Li, F. W., Teo, S. W. M., & Yang, C. C. (2019). ETF momentum. *Research Collection Lee Kong Chian School Of Business*.

Lim, B., Zohren, S., & Roberts, S. (2019). Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science*, *1*(4), 19–38.

Lintner, J. (1965, February). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, *47*(1), 13. Retrieved from https://doi.org/10.2307/1924119 doi: 10.2307/1924119

Manela, A., & Moreira, A. (2017). News implied volatility and disaster concerns. *Journal of Financial Economics*, *123*(1), 137–162.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91. Retrieved from http://www.jstor.org/stable/2975974

Markowitz, H. (1971). *Portfolio selection. efficient diversification of investments.* Yale University Press.

Messmer, M. (2017). Deep learning and the cross-section of expected returns. *SSRN Electronic Journal*. (Available at SSRN 3081555)

Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. *SSRN Electronic Journal*. (Available at SSRN 2740751)

Moskowitz, T. J., & Grinblatt, M. (1999). Do industries explain momentum? *The Journal of finance*, *54*(4), 1249–1290.

Moskowitz, T. J., Ooi, Y. H., & Pedersen, L. H. (2012). Time series momentum. *Journal of financial economics*, *104*(2), 228–250.

Mossin, J. (1966, October). Equilibrium in a capital asset market. *Econometrica*, *34*(4), 768. Retrieved from https://doi.org/10.2307/1910098 doi: 10.2307/1910098

Möller, P. M. (2018, November). Drawdown Measures And Return Moments. *International Journal of Theoretical and Applied Finance (IJTAF)*, *21*(07), 1-42. Retrieved from https://ideas.repec.org/a/wsi/ijtafx/v21y2018i07ns0219024918500425.html doi: 10.1142/S0219024918500425

Nawrocki, D. N. (1999, August). A brief history of downside risk measures. *The Journal of Investing*, *8*(3), 9–25. Retrieved from https://doi.org/10.3905/joi.1999.319365 doi: 10.3905/joi.1999.319365

Obaid, K., & Pukthuanthong, K. (2022). A picture is worth a thousand words: Measuring investor sentiment by combining machine learning and photos from news. *Journal of Financial Economics*, *144*(1), 273–297.

Oh, S. (2019). Feature interaction in terms of prediction performance. *Applied Sciences*, *9*(23), 5191.

O'Neal, E. S. (2000). Industry momentum and sector mutual funds. *Financial Analysts Journal*, *56*(4), 37–49.

Papailias, F., Liu, J., & Thomakos, D. D. (2021). Return signal momentum. *Journal of Banking & Finance*, *124*, 106063.

Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples. *Expert systems in the micro electronics age*.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Quinlan, J. R. (1996). Improved use of continuous attributes in c4. 5. *Journal of artificial intelligence research*, *4*, 77–90.

Rau-Bredow, H. (2020). Value at risk and diversification. *SSRN Electronic Journal*. Retrieved from https://doi.org/10.2139/ssrn.3711009 doi: 10.2139/ssrn.3711009

Rish, I., et al. (2001). An empirical study of the naive bayes classifier. In *Ijcai 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, pp. 41–46).

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms* (Tech. Rep.). Cornell Aeronautical Lab Inc Buffalo NY.

Ross, S. A. (1978). A simple approach to the valuation of risky streams. *Journal of business*, 453–475.

Rouwenhorst, K. G. (1998). International momentum strategies. *The journal of finance*, *53*(1), 267–284.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, *19*(3), 425–442.

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

Sortino, F. A., & Price, L. N. (1994, August). Performance measurement in a downside risk framework. *The Journal of Investing*, *3*(3), 59–64. Retrieved from https://doi.org/10.3905/joi.3.3.59 doi: 10.3905/joi.3.3.59

Stivers, C., & Sun, L. (2010). Cross-sectional return dispersion and time variation in value and momentum premiums. *Journal of Financial and Quantitative Analysis*, *45*(4), 987–1014.

Takeuchi, L., & Lee, Y.-Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. *Working paper, Stanford University*.

Treynor, J. L. (1962). Toward a theory of market value of risky assets. *SSRN Electronic Journal*. (Available at SSRN 628187)

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.

Wang, K. Q., & Xu, J. (2010). Time-varying momentum profitability. *SSRN Electronic Journal*. (Available at SSRN 1534325)

Wigglesworth, R. (2021, Mar). A fallen star of the investment world. *Financial Times*. Retrieved from https://www.ft.com/content/c5bc2880-07f0-4998-85c5-be4300a32427

Young, T. W. (1991). Calmar ratio: A smoother tool. *Futures*, *20*(1), 40.

FACULTY OF BUSINESS AND ECONOMICS
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be