Politecnico di Torino

Politecnico di Torino

Master's Degree in
Physics of Complex Systems
October 2024

Portfolio Optimization with Artificial Intelligence: A Neural Network Approach to Maximizing Risk-Adjusted Returns

Master thesis done in collaboration with Intesa Sanpaolo

Supervisors:
Prof. Luca Dall'Asta
Dott. Andrea Barral
Dott. Alessandro Sabatino

Candidate:
Gerardo Castagno

Thank you for this divine moment, where I coronate 5 years of sweat, tears and anxiety. I want to thank my parents for believing in me when I told them I wanted to move out to Turin, almost 1000 km away from home; Luca, for always being by my side during these five long years, enduring my mood swings and constant complaining; every single person who voted for me in the students' elections, allowing me to represent them and have access to the highest academic offices. Thank you. And do you know who I wanna thank? I wanna thank me. For believing in me and doing what I thought I could not do, and I want to say to myself in front of all you beautiful people: go on girl with your bad self. You did that.

Acknowledgements

I would like to thank Prof. Dall'Asta for helping me find this thesis opportunity; Andrea and Alessandro for their guidance and support throughout this work; Michele, Edoardo and Marco from the IMI Corporate & Investment Banking division of Intesa Sanpaolo for the interesting discussions and their useful advises.

Abstract

This master thesis focuses on developing an AI model designed to generate profitable investment strategies that outperform market benchmarks. Specifically, the model constructs a portfolio of 500 US equities, determining optimal positions and allocations for each stock. The goal is to maximize returns while minimizing risk and transaction costs. The decision to focus on portfolio optimization, rather than individual assets, stems from the ability to reduce risk through diversification. This is further enhanced by introducing an additional input parameter, λ, which accounts for the investor's risk aversion.

Building on the work of Zhang, Zohren, and Roberts (2020) [1], the model leverages multiple data sources to create meaningful features. A careful process of feature selection and hyperparameter tuning was undertaken to identify the most effective configuration. The model itself is a custom neural network implemented in TensorFlow, with four input types: stock returns time series, financials time series, risk aversion λ, and previous day allocation. It minimizes a custom loss function inspired by Modern Portfolio Theory (MPT) [2], balancing the trade-offs between maximizing returns, minimizing risk based on λ, and keeping transaction costs low.

The strategies generated by the model were benchmarked against the market and validated using statistical significance tests. Ultimately, the model successfully reduces systemic market risk and tailors investment strategies to suit the unique needs of individual investors.

Contents

List of Figures

Nomenclature

Roman Symbols

$\mathcal{L}$	Loss function
c	Transaction cost
L_t	Cumulative log return at time t
l_t	Log return at time t
l_t^{β}	β-adjusted log return at time t
$l_t^{\text{S\&P 500}}$	S&P 500 log return at time t
p^*	Penalty coefficient
R^{ann}	Annualized linear return
R_t	Cumulative linear return at time t
U	Utility function
W_t	Portfolio wealth at time t
$\mathbf{r}_t$	Equity (log) returns vector at time t
$\mathbf{w}_t$	Strategy weights vector at time t
$\mathbf{w}_t^{\text{S\&P 500}}$	Weights $\propto$ market cap at time t

Greek Symbols

λ	Risk-aversion parameter

σ^{ann} Annualized portfolio risk

Σ_t Returns covariance matrix at time t

σ_t Portfolio risk at time t

Other Symbols

^GSPC S&P 500 trading symbol

Acronyms / Abbreviations

AI Artificial Intelligence

ARQ As Reported Quarterly

Capex Capital Expenditures

DepAmor Depreciation & Amortization

FCFE Free Cash Flow to Equity

LSTM Long-Short Term Memory

MDD Max Drawdown

MPT Modern Portfolio Theory

MVU Mean-Variance Utility

NB Net Borrowing

NCFO Net Cash Flow from Operations

NI Net Income

NYSE New York Stock Exchange

P&L Profit & Loss

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

TTM Trailing Twelve Months

WC Working Capital

Chapter 1

Introduction

Before diving into the contents of this thesis, it may be useful to answer one question: *"Why should people invest their money?"*. We can identify two main reasons for which investing money is a smart option:

Inflation: every year we are faced with an increase in the cost of life and, consequently, a loss in our economic power [3]. Rather than devaluing our money by keeping it in a bank account, investing allows to tackle inflation by generating additional revenue.

Market growth: studies show that the economy tends to grow [4]. Under this evidence, we can exploit this growth to further enhance our gains by dedicating some of our money to financial investments.

Investing, however, comes with its own risks: it should thus be done consciously and methodologically. We can identify two main risk components of an investment: *systematic* risk and *unsystematic* (or idiosyncratic) risk.

Systematic risk is associated to macroeconomic events that affect the whole market (e.g. Covid, the 2008 crisis, etc...). Unsystematic risk, on the other hand, is the risk associated to the specific asset, thus being unique of each company [5].

When it comes to the reduction of unsystematic risk, many roads can be followed. In our work, we will be exploiting the mechanism of *diversification*.

For what concerns the systematic market risk, instead, we will be tackling its effect with long-short portfolios. We will indeed exploit a "betting against the market" philosophy, capable of compensating the effect of macroeconomic crashes and preserving our returns (section 1.2 for further details).

1.1 Portfolio Optimization

With the introduction of Modern Portfolio Theory (MPT) [2], portfolio optimization has gained a sturdier and more advanced mathematical basis to arise from. The strength of reasoning in terms of portfolios rather than single assets, stems from the ability of portfolios to reduce the investment's risk through *diversification*.

As a matter of fact, Markowitz's pioneering intuition was able to formalize the fact that, when managing a portfolio, one did not have to choose whether to increase the returns or reduce the risk, but could rather do both. Doing so, however, requires the portfolio to be *diversified*, i.e. it should be composed of assets characterized by low *covariance* [2].

By definition, given N assets $n_1, n_2, \ldots, n_N$, we define a portfolio as the pair $(\mathbf{n}, \mathbf{w})$ with w_i being the weight assigned to the i-th asset.

Starting from that, an *optimal portfolio* is one that maximizes the returns for a given level of risk (or, conversely, that minimizes the risk for a given level of returns). An equivalent way of saying it is that an optimal portfolio maximizes *risk-adjusted returns*. Portfolios that satisfy these conditions are also said to be *efficient* [2].

Achieving an efficient portfolio is not a trivial task. When it comes to trading, in most cases the investor will have a limited amount of capital, thus rendering the portfolio optimization a *constrained* problem.

Many classical theories have been developed but the rise of AI has opened a new chapter in the world of finance, with new alternatives for *algorithmic trading*. These new tools have been exploited in this work to develop a custom neural network model inspired from Zhang, Zohren, and Roberts (2020) [1]. The results have then been compared and validated with respect to a market benchmark: the *S&P 500 index* (^GSPC).

1.2 Basics on Finance

In order to fully grasp the work described in chapter 2, a brief introduction to key financial concepts is necessary.

By definition, an equity is *"one of the equal parts into which the value of a company is divided"* [6]. Owning some equity of a company thus means that we possess a given amount of ownership in that specific company.

The trading of assets, in our case equities, is regulated by a specific entity: the *Stock-Exchange*. For each financial market there is one (or more) Stock-Exchange. During their opening hours, equities can be traded and their prices vary depending on many factors (investors' trust, new available information, new rules and regulations, company's financials release, etc...).

When trading, different *positions* can be taken by the investor. In the definition of our portfolio strategy, we focus on two fundamental ones:

Long position: it is the most trivial way of investing in a financial market. It means that we are actively *buying* the equity of a given company and we aim at selling it, at a future time, for an higher price thus generating a positive net return.

Short position: in this case we **never** actually own any equity. What we do instead is borrow someone else's equity and sell it for its current value. We then hope that the company's value decreases in such a way that, when closing the *short-position*, we can buy back the same amount of equity for a lower price and obtain a positive net return.

Not every market allows for *short selling* and each might introduce stricter regulations (e.g. in 2020 the Consob banned short selling on the Borsa di Milano for 3 months after the Covid market crash [7]). In our particular case, we will be considering only equities from the US market, where short selling is allowed without impactful limitations on our model [8].

Short positions are particularly useful as their *"betting against the market"* nature allows to reduce the systemic market risk and helps in the portfolio diversification [9, 10].

When devising a good investment strategy, one should also take into account the presence of *transaction costs*. These can be composed of a multitude of factors, from broker fees to the *bid-ask spread* (the loss one would have it they were to sell the asset immediately after buying it) [11].

Another useful notion is the one of *market capitalization* (market cap for short). It can be computed as the product between the number of available shares and the current market value of each share [12]. Clearly, it represents an indicator of the value and importance of a company.

In order to benchmark our results, we need some sort of metric that measures the market's behaviour. A financial *index* groups a specific number of companies and tracks them in order to create a metric for the market's performance [13]. In this work we adopt the S&P 500 index, composed of the 500 top leading US companies. Each company is weighted proportionally to its market cap [14] and the index has obtained an annualized return of 13.39 % in the last 5 years [15].

Chapter 2

Developing the AI Model

2.1 Introduction to the Model Architecture

Taking inspiration from [1], we created an AI model that could generate long-short allocations for 500 US equities that maximized risk-adjusted returns and reduced transaction costs. More specifically, our aim was to try and *improve* the S&P 500 allocation strategy that suffered from some limitations.

As a matter of fact, being the S&P 500 a capitalization-weighted index [14], it only allows for *long positions* and does not have much control over the *risk*. By devising a model that balances the trade-offs between maximizing returns, minimizing risk based on λ, and keeping transaction costs low, all whilst keeping some similarity to the S&P 500 index, we are able to create more *efficient* strategies.

For the underlying neural network we implemented an architecture based on Long-Short Term Memory (LSTM) cells. LSTMs, an improved version of Recurrent Neural Networks (RNN), are particularly good when handling *historic time series* and are capable of solving the vanishing or exploding gradient problem of classical RNNs [16].

In order to extract meaningful information from the data, the LSTM requires its input to be a *time series* tensor of consistent shape. For instance, given n_{samples} data points of LAG periods historic series with n_{features} fea-

tures, the input tensor $\mathcal{T}_{\text{input}}$ would be:

$$\mathcal{T}_{\text{input}} \in \mathbb{R}^{n_{\text{samples}} \times LAG \times n_{\text{features}}}$$

The reason for immediately settling with an LSTM-architecture is due to the evidence of better results obtained by LSTM, when compared to other neural network architectures and classical methods [1, 17].

The model itself exploits 4 different data sources (the feature engineering process will be detailed in subsection 2.5.1):

Equity returns time series: contains daily data on equities' returns and market cap. It is the fundamental input of our model and is characterized by a LAG history of 30 days.

Financials time series: contains quarterly data regarding the financial situation of the considered companies. It is characterized by a LAG history of 8 quarters (2 years).

Risk-aversion λ: it is an additional parameter that allows to tune the generated strategy with respect to the desired level of risk-aversion. Higher λ values will correspond to strategies in which the minimization of risk is more important than the maximization of returns and vice versa.

Previous day allocation: in order to mimize the transaction costs, each generated allocation will also depend on the one generated at the previous time step. This allows to penalize excessive changes in the portfolio's composition and keep the transaction costs minimal.

In order to handle the 4 different inputs and have them correctly interact with the *custom loss function* (described in section 2.3), standard pre-existing models were not enough. A *custom sub-classed model* has thus been coded and implemented in TensorFlow for achieving the following results. Further informations about the model architecture will be disclosed in section 2.2.

2.2 Model Architecture

As introduced in section 2.1, carrying out this work required the creation of a complex neural network architecture based on LSTM (Figure 2.1). Starting from 4 input layers, 2 different LSTM pipelines have been implemented.

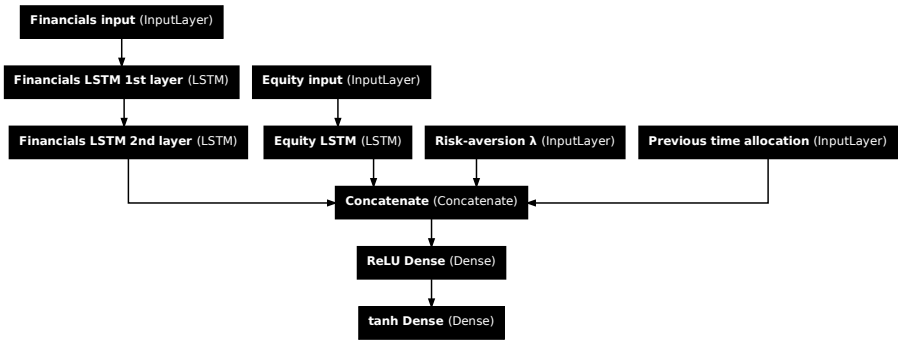


Fig. 2.1 Complete plot of the final model architecture

The reason behind this choice is due to the fact that equity time series and financial time series were created on data with different temporal scales (daily vs quarterly). This rendered impossible the use of a single LSTM pipeline as there would be a clear mismatch in the LAG history of the two inputs.

The output of the two LSTM pipelines is then concatenated with the remaining two input: the risk-aversion λ and the previous day allocation.

The output of the concatenation layer is then passed through an 8-neurons fully-connected (dense) layer with ReLU activation [18, 19]. The ReLU activation function is a piecewise linear activation function that is thus capable of capturing complex relations in the data without suffering the problems of saturation or low sensibility, common of other non-linear activation functions [18]. It is a key component of our neural network as it is exactly here that the model understands how the risk-aversion λ and the previous day allocation affect the custom loss and thus the allocation strategy.

Finally, the output of the ReLU dense layer is passed through another fully-connected layer with 500 neurons and *tanh* activation. This layer is used to generate the final allocation vector (apart from the norm-1 normalization).

The 500 neurons are needed for output reshaping, as we aim at optimizing portfolios of 500 assets and thus require the output for each day to be a vector of 500 weights, while the *tanh* activation function, unlike ReLU, allows to generate positive and negative weights for long/short positions.

2.3 Custom Utility Function

The *tanh* output described in section 2.2 produces *raw weights* which are not yet correctly normalized. Given that we want our long/short positions to allocate the totality of our capital, we must impose a norm-1 normalization condition on the weights vector. Moreover, in order for the model to generate allocations that satisfy our requests, we must implement a suitable loss function.

To obtain the required results, common regression loss functions (e.g. mean squared error) were not enough. Our model needed to minimize a custom loss function (or, conversely, maximize a custom utility function) that contained each of the elements of our request. Inspired from Modern Portfolio Theory's Mean-Variance Utility (MVU) [2] and Zhang et al. (2021) [17], the following *utility function* has been implemented:

$$U_t = \mathbf{w}_t^T \mathbf{r}_{t+1} - \frac{\lambda}{2}\sqrt{\mathbf{w}_t^T \Sigma_{t+1} \mathbf{w}_t} - p^* \left[1 - \frac{\mathbf{w}_t \cdot \mathbf{w}_t^{\text{S\&P 500}}}{\|\mathbf{w}_t\|_2 \|\mathbf{w}_t^{\text{S\&P 500}}\|_2}\right] - c\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$$

$$\text{with } \|\mathbf{w}_t\|_1 = \sum_{i=1}^{500} |w_{i,t}| = 1$$

The subscript t has been added in order to better explain the structure of the utility function. In reality, the utility U is evaluated on the returns of 10 future days, keeping the strategy allocation $\mathbf{w}_t$ constant. This allowed to obtain a more accurate metric by averaging over time and reducing the chaos's influence. Regardless of that, the utility function is composed of 4 main building blocks:

Portfolio returns $\mathbf{w}_t^T \mathbf{r}_{t+1}$: simply the product between the strategy weights at time t, $\mathbf{w}_t$, and the equity future returns, $\mathbf{r}_{t+1}$. Positive portfolio

returns are crucial for a successful strategy, thus the model tries to maximize them.

Portfolio risk $\frac{\lambda}{2}\sqrt{\mathbf{w}_t^T \Sigma_{t+1} \mathbf{w}_t}$**:** with Σ_{t+1} being the returns covariance matrix, this term is the product between the input risk-aversion λ and the portfolio's standard deviation. The $\frac{1}{2}$ is inherited by MPT [2]. The higher the λ, the more important is the minimization of the portfolio risk.

Distance from S&P 500 $p^* \left(1 - \frac{\mathbf{w}_t \cdot \mathbf{w}_t^{\text{S\&P 500}}}{\|\mathbf{w}_t\|_2 \|\mathbf{w}_t^{\text{S\&P 500}}\|_2}\right)$**:** it is the cosine distance between the model's allocation, $\mathbf{w}_t$, and an S&P 500-like allocation (i.e. $\propto$ market cap), $\mathbf{w}_t^{\text{S\&P 500}}$. The norm-2 normalization ensures that the distance lies in $[0,2]$. p^* is the penalty coefficient, and has been chosen in such a way that the model is capable of differentiating itself with respect to the S&P 500 index without generating completely arbitrary strategies.

Portfolio transaction costs $c\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$**:** with c being the single asset transaction cost. By computing the norm-1 difference between the allocation of consecutive days, we are able to determine how much the allocation has changed for each asset. By multiplying for the single asset transaction cost, we can subtract them to the portfolio returns and *penalize* excessive changes across consecutive days. Lacking the bid-ask spread for each single equity, a *reasonable* single value has been adopted: we assume $c = 10^{-4}$, which corresponds to a transaction cost of 0.01 \$ for an equity of value 100 \$.

The maximization of the utility function has been carried out with the Adam optimizer, a stochastic gradient descent algorithm [20, 21] that minimizes the custom loss $\mathcal{L} = -U$.

2.4 Training Process and Allocation Creation

The model uses data from different sources (see subsection 2.5.1 for further details), in the period 01/01/2010 - 27/02/2024. The available data has then been splitted in 2 sets:

- Train + validation set from 01/01/2010 to 31/12/2021 ($\approx$ 85%)
- Test set from 01/01/2022 to 27/02/2024 ($\approx$ 15 %)

These sets have then been used to train the model (subsection 2.4.1), determine the optimal feature and hyperparameter configuration (section 2.5) and finally to evaluate the performance of the model and compare it to the market benchmark (section 3.1).

2.4.1 Training of the Model

In this subsection, we will focus on the general training procedure without entering into the specific features given to the model. A more in-depth analysis of the feature engineering process will be carried out in subsection 2.5.1.

As already mentioned in section 2.1, the equity and financial data have been used as time series with, respectively, LAG histories of 30 days and 8 quarters. The remaining 2 inputs are the λ risk-aversion parameter and the previous day allocation.

The decision to use λ as an actual parameter is linked to the possibility of creating a *single model*, that would learn how to behave for many λ values, and then use it to visualize the generated strategies for different risk-aversions. An alternative way would have been to keep λ as an hyperparameter of the loss function but that would be highly impractical as it would require the training of a different model for every value of λ.

In order to make the model learn how to behave for different values of λ and, most importantly, to recognize the effect of many different previous day allocations, we generated artificial data for the training phase of the model:

Risk-aversion $\boldsymbol{\lambda}$: for each training day, a different value of λ has been given. More specifically, the values have been sampled randomly from a uniform distributions in order to cover most values.

Previous day allocation: similarly to what has been done for λ, we don't want the model to overfit on a specific allocation. We thus sampled (and correctly normalized) random long-short allocations from a Gaussian

distribution. This has been done since evidence from previous models (that did not track previous day allocations) showed that the generated long-short allocations were indeed normally distributed.

The training process has then been carried out with a learning rate of 2×10^{-5} and an early stopping regularization, in order to avoid overfitting. The final model was obtained after ~ 160 epochs. Below, a plot of the training and validation losses can be found for reference.

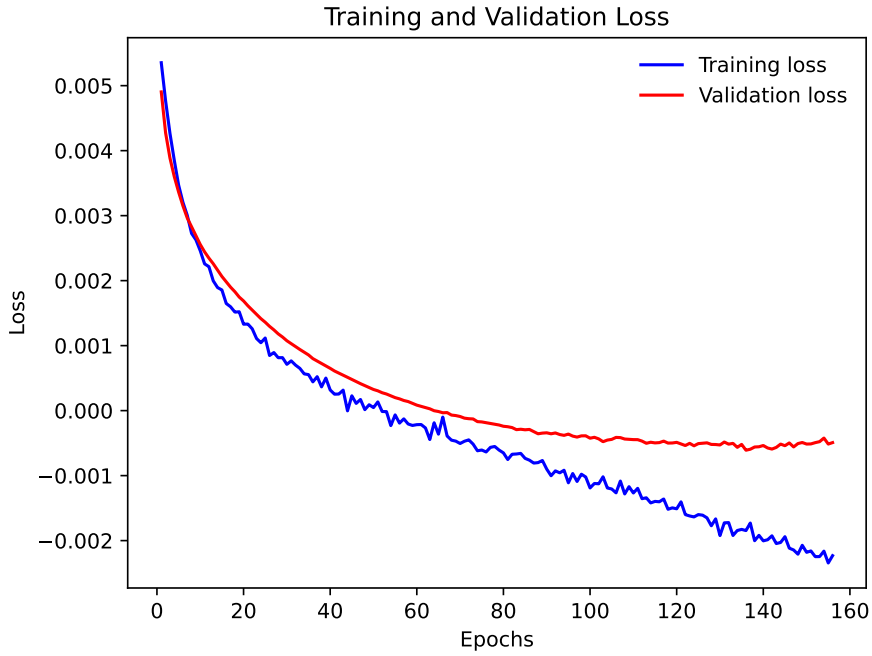


Fig. 2.2 A plot of the training and validation loss against training epochs

2.4.2 Generating Test Allocations

Given the peculiar architecture that aims at minimizing transaction costs, the generation of the weights allocation has to be carried out with a rolling-forward loop (Figure 2.3).

To *"jump-start"* the strategy generation process, we must first train a second model. This second model will be totally identical to the one described above but **will not** minimize the transaction costs. It will only be employed

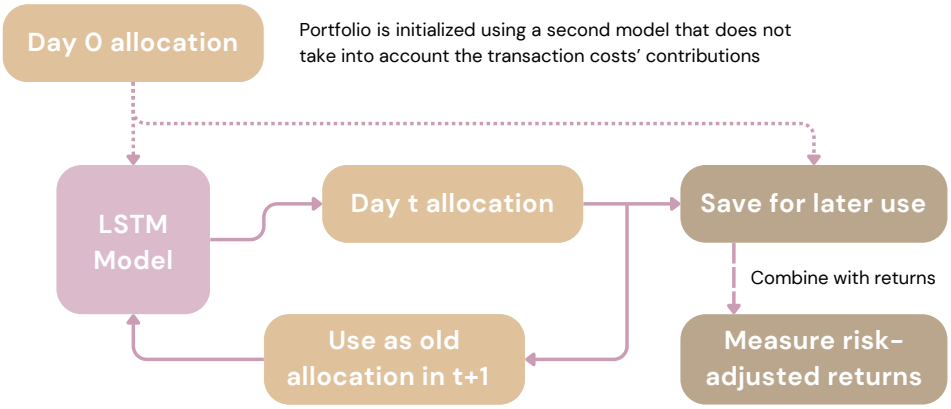


Fig. 2.3 A schematic representation of the rolling-forward loop

once, to generate the *"day 0 allocation"* corresponding to our first trading day (for which we don't have a previous allocation and thus whatever allocation we start with will amount to $\approx$ the same transaction costs).

This *"day 0 allocation"* is then passed to the actual LSTM model to determine the day 1 allocation. From the day 1 allocation, the model becomes *self-sufficient*: each day t allocation will be reused as previous day allocation for the generation of the day $t+1$ allocation until the test data ends.

By collecting all these allocations (including the *day 0 allocation*), we have obtained the model strategy for the test period. By combining with equity returns, we are then able to measure the risk-adjusted returns and benchmark our model against the market (chapter 3).

Analysis of the Test Allocations

After generating the strategy, a thorough analysis of the obtained strategy has been carried out.

By looking at Figure 2.4, it is apparent how the model is capable of distancing itself from the benchmark S&P 500 allocation. Moreover, we can definitely see that the model is capable of exploiting and suggesting short positions on some assets.

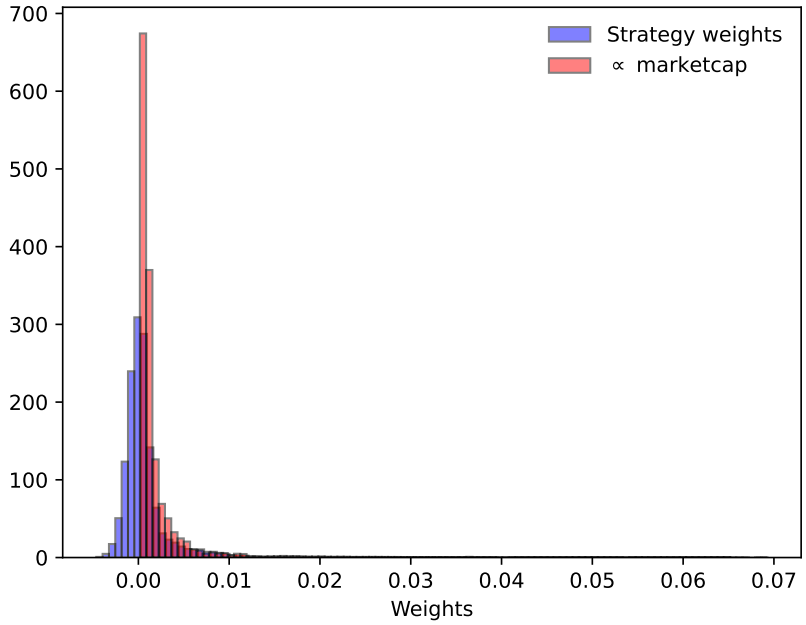


Fig. 2.4 A histogram with the strategy allocation and the benchmark $\propto$ market cap

Regardless of the apparent difference between the two strategies, a Wilcoxon test [22] has been applied to ensure the statistical significance of the results. The Wilcoxon test represents the *non-parametric* alternative to the paired t-test [23] and is thus very handy when the normality condition of the t-test is not satisfied (as in our case). As expected, the test confirmed the significance of the result with a p-value $= 0.0 < 0.05$.

The cosine-distance between consecutive days allocations has been computed and, on average, amounted to ≈ 0.043. This result was compared with the distance between random Gaussian allocations (average distance ≈ 1.0) with another Wilcoxon test. Once again, the statistical significance has been proven with a p-value $\approx 0.0 < 0.05$.

We were also interested in analyzing the distribution of the short-positions with respect to the market cap rank of the assets.

Figure 2.5 shows the total strategy allocation (long $+$ |short|) by market cap rank group, compared to the S&P 500 allocation. We can see how about 50 % of out portfolio is composed of the top-50 companies and is more or less

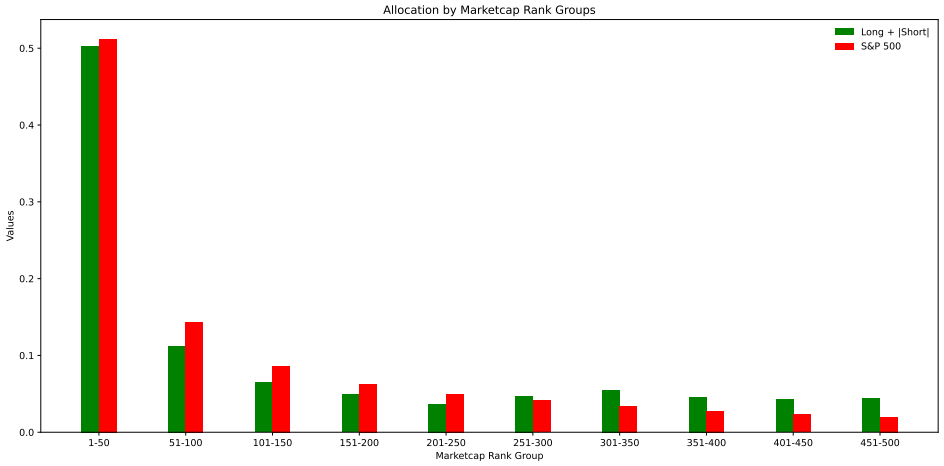


Fig. 2.5 A bar plot comparing the allocation of the strategies w.r.t. groups of 50 assets

in line with the S&P 500. Going down in market cap, instead, our strategy allocates more than the S&P 500 on the low-cap companies.

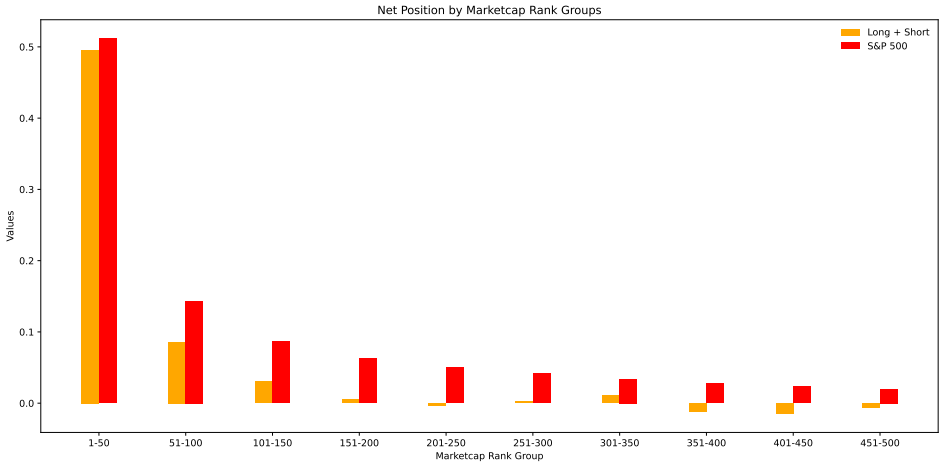


Fig. 2.6 A bar plot comparing the net position of the strategies w.r.t. groups of 50 assets

In Figure 2.6, on the other hand, we plotted the bars corresponding to the sum between (positive) long positions and (negative) short positions, compared to the (long-only) allocation of the S&P 500.

By comparing the two figures, we can determine that, as expected, the top equities by market cap are mostly characterized by *long-only positions*. Going down in the market cap rank, instead, more and more short positions

get introduced. This is in line with the expectation that high market cap companies would have better performance and thus their value would increase over time. It is also apparent how the greater allocation (compared to the S&P 500) on *low-cap companies* is due to the presence of short position that, as it will be shown in chapter 3, allow to better manage the *systemic market risk*.

2.5 Features and Hyperparameters

2.5.1 Feature Engineering

The raw data was obtained from 3 different data sources, acquired through the Nasdaq Data Link service:

Equity prices: a dataset containing the daily stock prices. From this dataset, the daily *close* price (i.e. the *split adjusted* value of the company's equity) and the daily *closeadj* price (i.e. the *split & dividend adjusted* value of the company's equity) have been obtained.

Financials: it contained data related to the balance sheet of the companies. We referred to two different reporting methodologies:

Trailing Twelve Months (TTM): given the publication date, it contained the company's performance in the previous 12 months. It was used to extract the market cap value as well as the *price* (adjusted by splits) of the company at that time.

Quarterly Reports (ARQ): gives information on the performance of the company in each quarter. From this, we obtained the current assets owned by the company, the current liabilities, the net income, the depreciation and amortization, the capital expenditures, the current debt, the revenues and the net cash flow from operations (NCFO).

Fund prices: contained daily values of financial indices. It was used to extract the data of the S&P 500 index (^GSPC).

Daily Equity Features

These daily features, with *LAG history* of 30 days, are made up of *log returns* and *marketcap ratio*.

Log(arithmic) returns, are obtained starting from the *closeadj* price. Being the *closeadj* price adjusted for splits and dividends, it gives a more accurate indicator of the stock's value for each company. Whilst we could just use it as a *raw* feature inside the model, we preferred to exploit it in a different way since the amount of training data is not large enough to ensure that the model actually understands the effect of this feature.

We therefore computed, for each day t, the logarithmic return l_t as:

$$l_t := \log\left(\frac{\text{closeadj}_t}{\text{closeadj}_{t-1}}\right)$$

This allowed us to use *dimensionless* features with 2 main advantages:

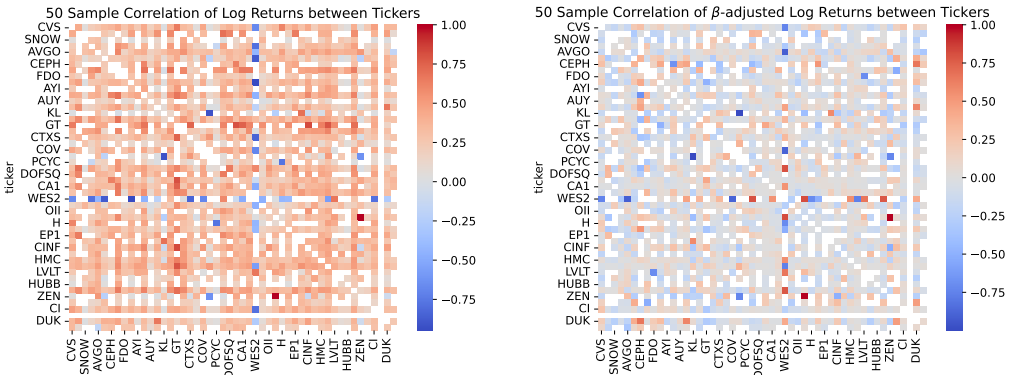
- Log returns are normally distributed (unlike linear returns that are log-normally distributed), thus making them more favourable for the model's training.
- They are summable and take into account the *compounding effect* (the cumulative log return among more days is easily computed as the sum of log returns).

A further improvement to log returns can be obtained by introducing the fund prices dataset. A similar calculation can in fact be carried out on the S&P 500 returns, thus obtaining the *index log returns* $l_t^{\text{S\&P 500}}$. We can then define the β-adjusted log returns l_t^{β} as:

$$l_t^{\beta} := l_t - l_t^{\text{S\&P 500}}$$

The reason for using β-adjusted log returns rather than plain and simple log returns is linked to two main factors:

- By removing the *"market's returns"*, we are removing a global (market) drift component. This allows to better focus on the actual single asset returns and be unaffected by the *systemic market risk*.
- Ideally, we would like our datapoints to be independent. This is pretty much always **not** the case, as financial assets tend to be correlated with each other. By implementing β-adjusted returns, however, we can significantly reduce the assets' correlation (Figure 2.7) and thus create better datapoints for the training of our model.



(a) Log returns

(b) β-adjusted log returns

Fig. 2.7 Comparison between the log return and β-adjusted log return correlations for a random sample of 50 assets (autocorrelation at lag 0 has been manually removed)

Marketcap ratio is the second daily feature given to the model. It basically represents the equity's S&P 500 allocation and is computed by combining data from the equity and financials (TTM) datasets. By extracting the TTM market cap, as well as the related issued stock *price* (split-adjusted), we are able to obtain its daily value using the daily, split-adjusted, *close* price from the equity dataset:

$$\text{marketcap}_t := \text{marketcap}_{\text{issue}} \times \left(\frac{\text{close}_t}{\text{price}_{\text{issue}}} \right)$$

After obtaining the daily market cap values, we were able to rank each equity and filter the top-500. From this, the *marketcap ratio* for the i-th stock

at time t was simply obtained as:

$$\text{marketcap\_ratio}_{i,t} := \frac{\text{marketcap}_{i,t}}{\sum_{n=1}^{500} \text{marketcap}_{n,t}}$$

Quarterly Financial Features

The feature engineering of quarterly financial features revolves around 3 main indicators of a company's performance:

Free Cash Flow to Equity (FCFE): is one of the indicators used in company valuation. It indicates the money available to shareholders after all expenses, reinvestment, and debt have been paid [24].

Revenues: is related to the income generated by *usual* sale operations (so it does not take into account extraordinary income sources such as the selling of a facility) [25]. As such, it allows to gain an insight on how well the sales of the given company are performing [26].

Net Cash Flow from Operations (NCFO): it is similar to the revenues but rather than being a measure related to sales effectiveness, it represents the amount of cash inflow/outflow related to usual sales operations (thus being a *liquidity* indicator) [26, 27].

While the revenues and NCFO are directly available from the financial statements of the companies, the FCFE must be computed separately. To do so, we need data related to:

- Current assets (what's owned by the company, e.g. what's stocked in the warehouse) and current liabilities (what the company owes to its suppliers, does not include bank debts). By subtracting current assets and current liabilities, we can compute the *working capital* (WC) [28].
- Net income (NI), the total revenue of the company after expenses [29].
- Depreciation and Amortization (DepAmor) are associated to the cost of an intangible asset, spread over its lifespan (amortization) and to the loss of value of tangible assets over time (depreciation) [30]. For

instance, an example of amortization could be the spreading of the cost for the renewal of a patent in the years prior to its renewal [30]. For the depreciation, instead, we can think of how a working machine loses value after each use.

- Capital expenditures (Capex) "are funds used by a company to acquire, upgrade, and maintain physical assets such as property, plants, buildings, technology, or equipment" [31].
- Net borrowing (NB) is obtained "by subtracting the amount of debt repaid in the year from the total debt borrowed during the year" [32].

From these elements, the FCFE can be computed as [32]:

$$\text{FCFE} = \text{NI} + \text{DepAmor} - \text{Capex} - \Delta\text{WC} + \text{NB}$$

Given these 3 indicators (FCFE, Revenues and NCFO), four different features have been constructed. The first 3 are related to the relative change of the indicators across quarters:

$$\text{change\_of\_indicator}_t = \frac{\text{indicator}_t - \text{indicator}_{t-1}}{|\text{indicator}_t|}$$

Given that using the relative change of the FCFE we lose the information on whether the FCFE is positive or negative (which we thought could be a strong indicator when doing company valuation), a fourth *categorical* feature has been added:

$$\text{sign\_of\_FCFE}_t = \text{sign}(\text{FCFE}_t)$$

As anticipated in section 2.1, these 4 features have been adopted to create times series with *LAG history* of 8 quarters (2 years).

2.5.2 Feature Subset Selection

After creating these features, it was mandatory to check whether they were actually improving the performance of the model. The daily equity features

(log returns and marketcap ratio) are the foundation of the model, thus checking whether they should be kept or not is pointless.

The process of *feature subset selection* is thus carried out only on the financial features (that indeed were added chronologically later in the process of creating the final model) and on an iteration of the model that was not yet taking into account the minimization of transaction costs.

The analysis made use of k-fold cross-validation [33, 34], an algorithm that allows to split the train+validation set into k elements and then perform the evaluation of the model k times by alternating which of the k elements is used as validation set.

A first round of analysis was carried on all the possible 16 feature configurations (including the one with no financial features) using a 3-fold cross-validation. The three loss values per configuration have then been averaged to identify the most promising models.

After this first evaluation process, the model with all 4 features appeared to be the best performing one.

In order to ensure the significance of this result, a further test was carried out. In particular, a 30-fold cross-validation has been performed on 2 models: the one with all 4 financial features and the one with none.

The choice of a 30-fold cross-validation allowed us to obtain a more accurate loss average and, most importantly, a distribution of losses over which a statistical significance test could be carried out.

After averaging, the model with 4 features showed an $\approx 73\%$ performance improvement with respect to the one without any financial feature.

This improvement has been validated with a paired t-test. The result of the test showed a p-value $\approx 0.0443 < 0.05$ thus confirming the statistical significance of the better result.

2.5.3 Hyperparameter Tuning

For the hyperparameter tuning, a similar process to the one described in subsection 2.5.2 has been employed. This time, however, we are unable to

carry out the cross-validation on *every* possible configuration, given that for the hyperparameters these are *infinite*.

We therefore had to settle with the introduction of a random search algorithm that could explore the phase space of hyperparameters and try out different models with 3-fold cross-validation.

Research shows that the random search algorithm is much more efficient than manual searches or grid search algorithms. It is indeed capable of finding models, with same or even better performances, with smaller computation times [35].

Our random search cross-validation has thus analyzed 20 different configurations, randomly selected from the following parameter grid:

```
param_grid = {
    'num_layers_eq': [1,2,3,4],
    'num_layers_fin': [1,2,3,4],
    'units_vector_eq': [1,2,4,8],
    'units_vector_fin': [1,2,4,8],
    'dense_units': [8,10,12],
    'dropout_rate': [0.0,0.2,0.4],
    'batch_normalization': [False, True]
}
```

For each LSTM pipeline (Figure 2.8), the model was able to use up to 4 LSTM layers. Each layer could independently be characterized by 1/2/4/8 neurons. If more than 1 layer were employed, a Dropout layer between the LSTM layers was placed. The Dropout layer is used to avoid overfitting by turning off a fraction of the neurons (determined by its rate) during training [36]. Moreover, the random search algorithm could also choose whether to add a normalization layer. Lastly, the number of neurons (8/10/12) of the ReLU dense layer was also tuned.

After analyzing 20 configuration with 3-fold cross-validation, the following appeared to be the best performing one:

- Equity LSTM pipeline: 1 layer with 1 neuron.

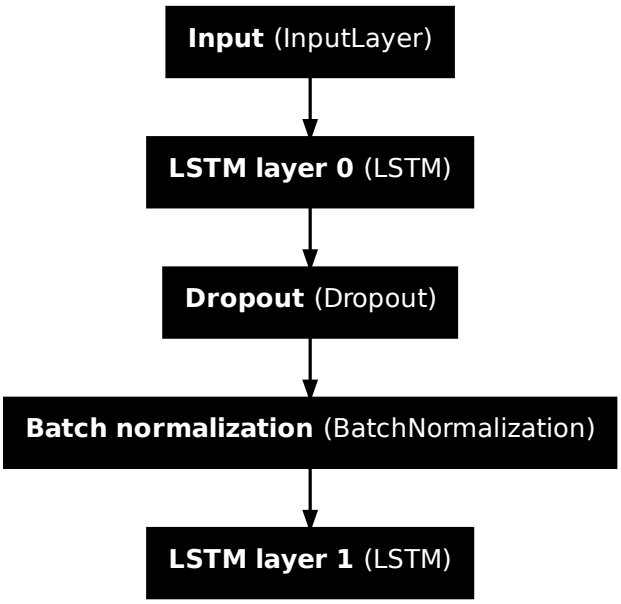


Fig. 2.8 Example of a 2 layer LSTM pipeline with Dropout and BatchNormalization layers

- Financial LSTM pipeline: 2 layers with 1 neuron (thus not followed by Dropout) and 2 neurons each.
- ReLU dense layer: 8 neurons.
- No BatchNormalization layer has been employed.

As the random search seemed to be suggesting that simpler models where better performing, the 30-fold cross-validation test has been carried out between the random-search selected model and the simplest possible model (1 layer with 1 neuron for each LSTM pipeline & 8 ReLU dense neurons).

After averaging, the random-search model showed an average loss $\approx -1.106 \times 10^{-3}$ while the simplest model achieved an average loss $\approx 5.925 \times 10^{-7}$. Moreover, a paired t-test has been performed on the distribution of losses and it returned a p-value $\approx 0.008 < 0.05$. This once again shows the validity of the process and ensures that these results are **not** due to randomness.

Chapter 3

Results and Future Developments

3.1 Profit & Loss and Efficient Frontier

As anticipated in chapter 1, in order to benchmark the strategy generated by our model, we've run comparisons with respect to the *S&P 500 index*.

The first way to see how our strategy behaves is by plotting the so-called *profit & loss* (P&L) curve. This basically consists of the plot of the portfolio cumulative returns against time, and thus allows us to visualize how our generated portfolio strategy behaves over the test period. For the sake of this plot, we move from logarithmic to linear results as their interpretation is much more straightforward (e.g. 0.2 means that we have a 20 % return).

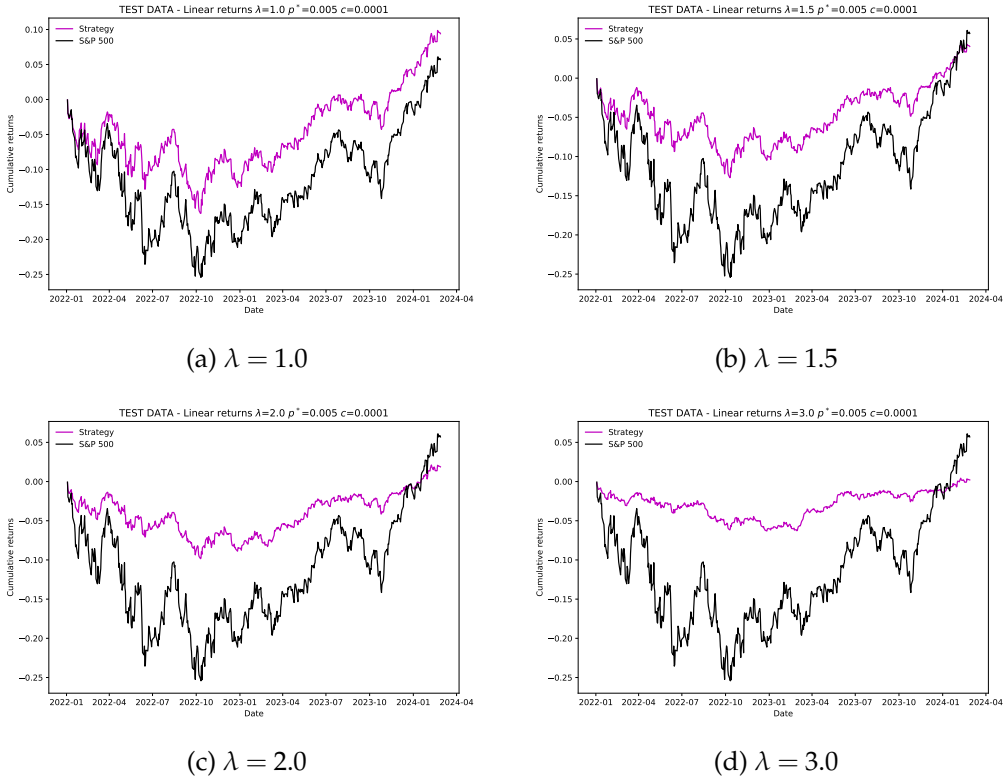
To move from logarithmic returns l_t to linear *cumulative* returns R_t, we first exploit the fact that log returns are *summable*. For the day T, the cumulative log return L_T is thus just:

$$L_T := \sum_{t \leq T} l_t$$

We can then move to cumulative linear returns R_t:

$$R_t := \exp(L_t) - 1$$

Figure 3.1 shows the P&L curves of the final strategy at different levels of risk-aversion λ. In the following, we will not consider $\lambda < 1$ as we expect that the majority of investors will have $\lambda \in [1, 10]$ [37]. Moreover, it should be noted that the strategies below already take into account the transaction costs, which have been subtracted by using the estimated coefficient $c = 10^{-4}$.



(a) $\lambda = 1.0$ (b) $\lambda = 1.5$

(c) $\lambda = 2.0$ (d) $\lambda = 3.0$

Fig. 3.1 Profit & loss curves for the generated strategy (blue line) and the S&P 500 index (black line) at varying values of λ risk-aversion

The first notable result is that our model actually works. Indeed, by changing the value of λ, we are capable of generating different strategies with different risk profiles. This allows to tune the generated strategy based on the investor's own risk-aversion. Secondly, it is apparent how the S&P 500 is characterized by a riskier strategy than the one generated by our model. Lastly, it should be noted that we are also capable of obtaining better returns than the S&P 500 in the case $\lambda = 1.0$, and landing very close to it with $\lambda = 1.5$.

While it's true that we can extract some useful information regarding the model's performance from these graphs, they are not very explicit. Given

that our model can generate strategies with different λ-s, a better approach would be to plot the ex-post *efficient frontier* (Figure 3.2). Doing so allows us to compare the overall S&P 500 strategy with ours, determining which of the two is *optimal*.

In this plot, the annualized (linear) returns R^{ann} of the strategies are plotted against the annualized risk (standard deviation) σ^{ann} and, by varying λ, a curve can be obtained. The annualization is pretty straight-forward. Given that in a year there are ≈ 252 trading days, and given that our test data is composed of 540 trading days:

$$R^{\text{ann}} := R_{540} \times \frac{252}{540}$$
$$\sigma^{\text{ann}} := \sigma_{540} \times \sqrt{252}$$

As we can see from Figure 3.2, the point related to the S&P 500 can be projected on the curve generated by our model. This implies that our model is capable of generating a strategy that achieves the same returns of the S&P 500 with, however, a much lower risk. This means that, for any *rational* investor [2], the strategy generated by our model will be optimal with respect to the S&P 500.

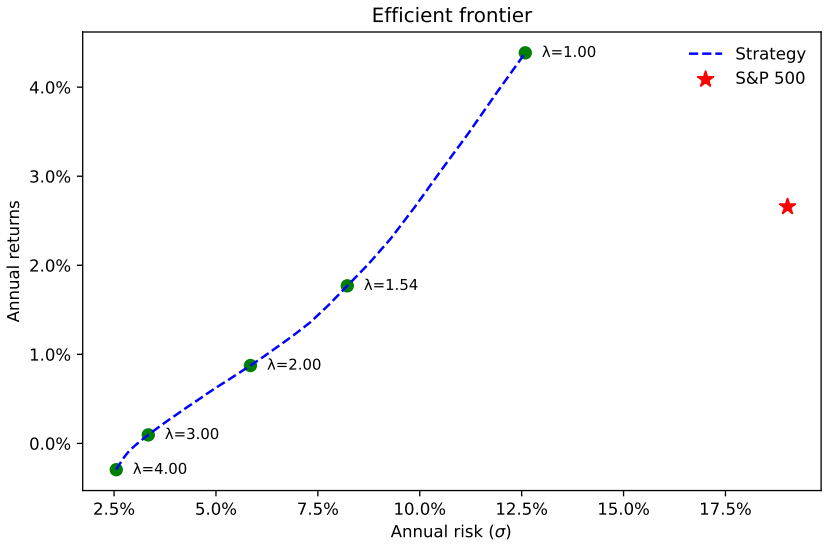


Fig. 3.2 *Ex-post* efficient frontier of the strategy compared to the S&P 500

To recap, our model is therefore capable of generating different strategies with varying risk profiles. With respect to the original work by Zhang, Zohren and Roberts [1], we are also capable of landing in a *specific point* of the efficient frontier by varying λ. Lastly, we are also taking into account the transaction costs, by minimizing them in the process of strategy generation, and by subtracting them to our returns when comparing our strategy to the market benchmark.

One should however beware of the fact that, even though this model considers transaction costs, it is **not** taking into account the short position *borrowing costs* [38]. Moreover, shorting an equity will give us an immediate positive cash inflow that could possibly benefit our returns thanks to *leverage* or the interest accrued on it.

3.1.1 Max Drawdown

To further analyze how our strategy has a better risk profile when compared to the S&P 500, we have computed another useful indicator:

Max Drawdown (MDD): *"the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained"* [39]. It is thus a measure of the risk of our portfolio. By defining the *wealth* at time t as $W_t := R_t + 1$ (i.e. the percentage of the initial capital that we own at time t), we can compute the MDD (which is a negative quantity being a *loss measure*) in the following way:

$$\begin{aligned} \text{MDD} &:= \frac{\text{Through Value} - \text{Peak Value}}{\text{Peak Value}} \\ &:= \min_t \left(\frac{W_t - \max_{t' \leq t}(W_{t'})}{\max_{t' \leq t}(W_{t'})} \right) \end{aligned}$$

In Figure 3.3 we can compare our strategy to the S&P 500, at varying values of λ risk-aversion. It is once again apparent how the strategy generated by our model has a much lower *"risk"* than the S&P 500 index. Starting from neutral risk-aversion ($\lambda = 1.0$) we gain ≈ 10 % points in the max draw-

down when compared to the S&P 500. Moreover, we are capable of further improving it by increasing the λ risk-aversion.

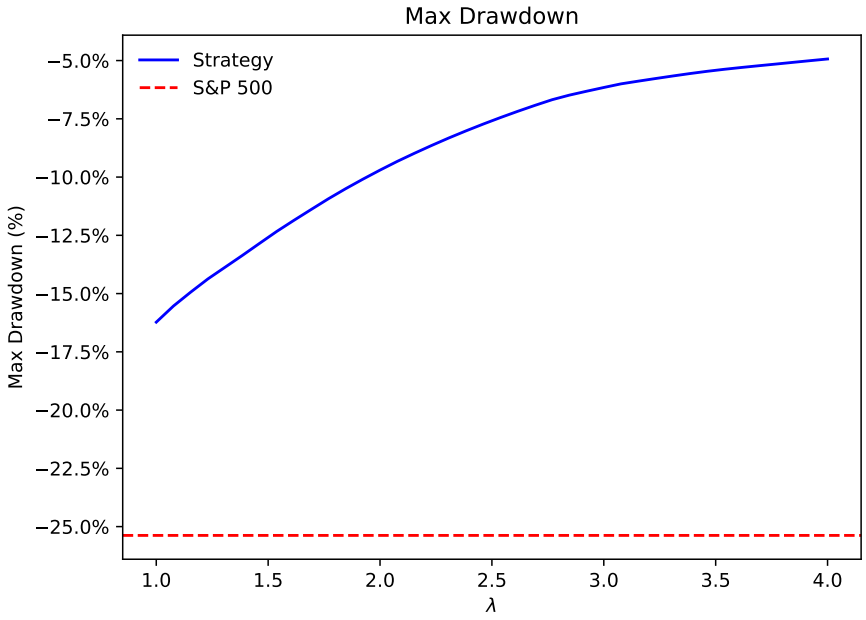


Fig. 3.3 Max drawdown of the strategy compared to the S&P 500

3.2 Future Developments

The work presented in this thesis was able to take the theoretical background from [1], and create a practical, more advanced implementation that is better qualified to be put in the production phase.

Regardless, there are still many additions that can be included into this model, in order for it to be better descriptive of reality and, possibly, improve its performance.

Firstly, we should mention that there are many more possibilities for feature engineering than the ones described in this thesis. Due to hardware limitations (RAM in particular) we were unable to further expand the used features (e.g. by adding the sign of NCFO) and introduce other data sources. For instance, data regarding institutional investors' portfolio imbalance,

sentiment analysis, macroeconomic data or historical short level, could be employed in future iterations of this work. It is thus likely that, with a stronger machine, better and more features could be engineered, further improving the performance of the model.

Moreover, as already mentioned in chapter 2, the proposed model uses a simplistic representation of trading costs:

- We are assuming a constant transaction cost among stocks and trading days. We know that in reality this is not the case as, by depending on the *bid-ask spread*, each asset will have a specific transaction cost that can fluctuate daily (or even intra-daily) [11].
- We are neglecting the short position borrowing costs. As of now, the cost of acquiring a short position corresponds to the one of a long position. Similarly, short position borrowing costs will depend on the asset and on the day; moreover, there will be interests depending on how long we keep that short position open [38].
- The positive cash inflow obtained from the short position could accrue interests or, more interestingly, be used as *leverage* to boost our capital and carry out more investments.

Addressing these idealities will definitely create a more accurate model, that takes into account the much more complex dynamics of financial markets.

Lastly, we should mention that the strongest limitation of this model, perhaps, is the limited amount of available data. In order to fix this problem, a possible solution could be to increase the data frequency and move towards *intra-daily* training. Given that the NYSE and Nasdaq trading hours go from 9:30 to 16:00 [40, 41], just by moving from daily to hourly we would get ≈ 6 times the number of datapoints. Of course, implementing *intra-daily* trading would require much more computational power than the one available from the consumer hardware employed in this work, thus making it unachievable for the sake of this thesis.

References

[1] Zihao Zhang, Stefan Zohren, and Stephen Roberts. "Deep Learning for Portfolio Optimization". In: *The Journal of Financial Data Science* 2.4 (Aug. 2020), pp. 8–20. ISSN: 2640-3943. DOI: 10.3905/jfds.2020.1.042. URL: http://dx.doi.org/10.3905/jfds.2020.1.042.

[2] Harry Markowitz. "PORTFOLIO SELECTION". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. DOI: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1952.tb01525.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x.

[3] ISTAT. *Inflazione*. URL: https://www.istat.it/tag/inflazione/ (visited on Oct. 5, 2024).

[4] Simon Smith Kuznets. "POPULATION AND ECONOMIC GROWTH". In: 2016. URL: https://api.semanticscholar.org/CorpusID:150690501.

[5] James Chen, Gordon Scott, and Katrina Munichiello. *Idiosyncratic Risk: Definition, Types, Examples, Ways To Minimize*. URL: https://www.investopedia.com/terms/i/idiosyncraticrisk.asp#toc-idiosyncratic-risk-vs-systematic-risk (visited on Oct. 11, 2024).

[6] Cambridge Dictionary. URL: https://dictionary.cambridge.org/dictionary/english/equity (visited on Sept. 17, 2024).

[7] La Stampa. *Borsa di Milano, Consob verso un nuovo divieto delle vendite allo scoperto fino a 90 giorni*. 2020. URL: https://www.lastampa.it/economia/2020/03/16/news/borsa-di-milano-consob-verso-un-nuovo-divieto-delle-vendite-allo-scoperto-fino-a-90-giorni-1.38601393/ (visited on Sept. 17, 2024).

[8] U.S. Securities and Exchange Commission. *Key Points About Regulation SHO*. URL: https://www.sec.gov/investor/pubs/regsho.htm (visited on Sept. 17, 2024).

[9] Gevorg Hunanyan. "The Systematic Risk of Short-Sale Restrictions". In: *Finance Educator: Courses* (2017). DOI: 10.2139/ssrn.2958122.

[10] Xiaohu Deng, Lei Gao, and Jeong-Bon Kim. "The Pathogen, Scapegoat, or a Miracle Drug? Short Selling and Stock Price Crash Risk". In: *SSRN Electronic Journal* (Jan. 2016). DOI: 10.2139/ssrn.2782559.

[11] Lucas Downey, Gordon Scott, and Yarilet Perez. *What Are Transaction Costs? Definition, How They Work, and Example*. URL: https://www.investopedia.com/terms/t/transactioncosts.asp (visited on Sept. 19, 2024).

[12] Jason Fernando, Samantha Silberstein, and Pete Rathburn. *Market Capitalization: What It Means for Investors*. URL: https://www.investopedia.com/terms/m/marketcapitalization.asp (visited on Sept. 18, 2024).

[13] Dan Caplinger. *What Is a Stock Market Index?* URL: https://www.fool.com/investing/stock-market/indexes/ (visited on Sept. 18, 2024).

[14] *SP U.S. Indices Methodology*. SP Global. 2024. URL: https://www.spglobal.com/spdji/en/documents/methodologies/methodology-sp-us-indices.pdf.

[15] SP Global. *SP 500 Overview*. URL: https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview (visited on Sept. 18, 2024).

[16] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[17] Chao Zhang et al. *A Universal End-to-End Approach to Portfolio Optimization via Deep Learning*. 2021. arXiv: 2111.09170 [q-fin.PM].

[18] Jason Brownlee. *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. 2020. URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ (visited on Sept. 20, 2024).

[19] Danqing Liu. *A Practical Guide to ReLU*. 2017. URL: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7 (visited on Sept. 20, 2024).

[20] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[21] TensorFlow. *TensorFlow: Adam Optimizer*. URL: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam (visited on Sept. 20, 2024).

[22] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987. URL: http://www.jstor.org/stable/3001968.

[23] J.H McDonald. *Handbook of Biological Statistics (3rd ed.)* Sparky House Publishing, Baltimore, Maryland, 2014, pp. 186–189. URL: https://www.biostathandbook.com/wilcoxonsignedrank.html.

[24] Will Kenton, Julius Mansa, and Yarilet Perez. *Free Cash Flow to Equity (FCFE) Formula and Example*. URL: https://www.investopedia.com/terms/f/freecashflowtoequity.asp (visited on Sept. 22, 2024).

[25] Adam Hayes, Julius Mansa, and Suzanne Kvilhaug. *Revenue Definition, Formula, Calculation, and Examples*. URL: https://www.investopedia.com/terms/r/revenue.asp (visited on Sept. 22, 2024).

[26] J.B. Maverick, Margaret James, and Ryan Eichler. *How Are Cash Flow and Revenue Different?* URL: https://www.investopedia.com/ask/answers/011315/what-difference-between-cash-flow-and-revenue.asp (visited on Sept. 22, 2024).

[27] Shobhit Seth, Natalya Yashina, and Suzanne Kvilhaug. *Cash Flow From Operating Activities (CFO) Defined, With Formulas*. URL: https://www.investopedia.com/terms/c/cash-flow-from-operating-activities.asp (visited on Sept. 22, 2024).

[28] Adam Hayes, David Kindness, and Yarilet Perez. *Liability: Definition, Types, Example, and Assets vs. Liabilities*. URL: https://www.investopedia.com/terms/l/liability.asp#toc-what-is-a-liability (visited on Sept. 22, 2024).

[29] Will Kenton, Khadija Khartit, and Kirsten Rohrs Schmitt. *Net Income (NI): Definition, Uses, and Formula*. URL: https://www.investopedia.com/terms/n/netincome.asp (visited on Sept. 22, 2024).

[30] Sean Ross, Gordon Scott, and Vikki Velasquez. *Amortization vs. Depreciation: What's the Difference?* URL: https://www.investopedia.com/ask/answers/06/amortizationvsdepreciation.asp (visited on Sept. 22, 2024).

[31] Jason Fernando, Margaret James, and Suzanne Kvilhaug. *Capital Expenditure (CapEx) Definition, Formula, and Examples*. URL: https://www.investopedia.com/terms/c/capitalexpenditure.asp (visited on Sept. 22, 2024).

[32] Brooke Tomasetti. *Free Cash Flow to Equity (FCFE)*. URL: https://www.carboncollective.co/sustainable-investing/free-cash-flow-to-equity (visited on Sept. 22, 2024).

[33] M. Stone. "Cross-Validatory Choice and Assessment of Statistical Predictions". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2 (Dec. 2018), pp. 111–133. ISSN: 0035-9246. DOI: 10.1111/j.2517-6161.1974.tb00994.x. eprint: https://academic.oup.com/jrsssb/article-pdf/36/2/111/49096683/jrsssb_36_2_111.pdf. URL: https://doi.org/10.1111/j.2517-6161.1974.tb00994.x.

[34] M. Stone. "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (Dec. 2018), pp. 44–47. ISSN: 0035-9246. DOI: 10.1111/j.2517-6161.1977.tb01603.x. eprint: https://academic.oup.com/jrsssb/article-pdf/39/1/44/49117086/jrsssb_39_1_44.pdf. URL: https://doi.org/10.1111/j.2517-6161.1977.tb01603.x.

[35] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *J. Mach. Learn. Res.* 13.null (Feb. 2012), pp. 281–305. ISSN: 1532-4435. URL: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[36] TensorFlow. *TensorFlow: Dropout*. URL: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout (visited on Sept. 23, 2024).

[37] Andrew Ang. *Asset Management: A Systematic Approach to Factor Investing*. Oxford University Press Inc, 2014. URL: https://books.google.de/books?id=e5yzAwAAQBAJ&lpg=PP1&hl=it&pg=PP1#v=onepage&q&f=false.

[38] Adam Hayes, Samantha Silberstein, and Yarilet Perez. *What Is a Stock Loan Fee (Borrow Fee)? Definition and Example*. URL: https://www.investopedia.com/terms/s/stock-loan-fee.asp (visited on Sept. 24, 2024).

[39] Adam Hayes, Gordon Scott, and Yarilet Perez. *Maximum Drawdown (MDD) Defined, With Formula for Calculation*. URL: https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp (visited on Oct. 1, 2024).

[40] NYSE. *NYSE: Holidays & Trading Hours*. URL: https://www.nyse.com/markets/hours-calendars (visited on Sept. 24, 2024).

[41] Nasdaq. *Nasdaq: Stock Market Holidays & Trading Hours*. URL: https://www.nasdaq.com/market-activity/stock-market-holiday-schedule (visited on Sept. 24, 2024).