FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF

HIGHER EDUCATION

«NATIONAL RESEARCH UNIVERSITY

HIGHER SCHOOL OF ECONOMICS»

Faculty of Computer Science

**Babynin Andrey Igorevich**

Graph Neural Networks for Stock Portfolio Optimization

*Master’s thesis*

for the degree "Machine Learning and High-Load Systems"

01.04.02 Applied Mathematics and Computer Science

Moscow, 2023

## TABLE OF CONTENT

[TABLE OF CONTENT 2](#_Toc135583832)

[ABSTRACT 3](#_Toc135583833)

[CHAPTER 1. INTRODUCTION 4](#_Toc135583834)

[CHAPTER 2. LITERATURE REVIEW 5](#_Toc135583835)

[CHAPTER 3. METHODOLOGY 7](#_Toc135583836)

[Section 3.1 Prioritized experience replay 8](#_Toc135583837)

[Section 3.2 Relation Graph Attention Networks 9](#_Toc135583838)

[CHAPTER 4. MODEL ARCHITECTURE 11](#_Toc135583839)

[Section 4.1 End to end portfolio construction 11](#_Toc135583840)

[Section. 4.1.1 Prediction layer 11](#_Toc135583841)

[Section 4.1.2 Optimization layer 11](#_Toc135583842)

[CHAPTER 5. EXPERIMENTS ON REAL MARKET DATA 11](#_Toc135583843)

[Section 5.1 Collecting the data 11](#_Toc135583844)

[Section 5.2 Benchmark 12](#_Toc135583845)

[Section 5.2 Features generation 12](#_Toc135583846)

[Section 5.3 Graph generation 13](#_Toc135583847)

[Section 5.4 Training process 15](#_Toc135583848)

[Section 5.5 Experiments 15](#_Toc135583849)

[Section 5.6 Performance comparison 15](#_Toc135583850)

[CHAPTER 5. CONCLUSION 16](#_Toc135583851)

[REFERENCES 17](#_Toc135583852)

[Appendix A. ETF Description 18](#_Toc135583853)

[Appendix B. Technical indicators 21](#_Toc135583854)

[Appendix C. Graphs generation 22](#_Toc135583855)

## 

## ABSTRACT

The work is devoted to stock portfolio optimization based on graph neural networks. The work uses the end-to-end optimization approach. As an additional measure to ensure the robustness of the algorithm, Prioritized Experience Replay (PER) is used.

## CHAPTER 1. INTRODUCTION

Since its inception, stock market analysis has revolved around two fundamental aspects: stock forecasting and stock portfolio allocation. The accurate prediction of future stock values simplifies the task of portfolio allocation, while relying solely on past performance and abandoning the prediction task heightens the significance of stock allocation under uncertainty. Researchers have predominantly focused on either one of these problems, thereby limiting the scope and degrees of freedom involved. However, in this paper, I adopt an alternative approach known as end-to-end optimization, which integrates prediction and optimization tasks into a cohesive framework. Еhis study aims to contribute to the advancement of stock portfolio optimization, paving the way for more effective portfolio allocation strategies.

To accomplish this, I employ a comprehensive set of ETF tickers, encompassing a wide range of assets, as the target universe. The prediction aspect of my work utilizes Relational Graph Attention Models. Consequently, my research is structured into several distinct sections.

The first section comprises a comprehensive literature review, which contextualizes the current study within the existing body of knowledge. In the second section, I elaborate on the methodology behind end-to-end optimization, elucidating the concept of prioritized experience replay employed to enhance the robustness of the model. Furthermore, I provide an overall explanation of the Relational Graph Attention Networks employed within the prediction layers.

The third section provides an overview of the model architecture, elucidating the structural components of both the prediction and optimization layers. Moving on to the fourth section, I delve into the details of data collection, feature generation, the training process, and the conducted experiments. Additionally, in this section, I present a comparative analysis of the experimental results.

Concluding the study, the final section outlines the main findings derived from the research and offers insights into potential directions for future work.

## CHAPTER 2. LITERATURE REVIEW

*Believe me, no. I thank my fortune for it—*

*My ventures are not in one bottom trusted,*

*Nor to one place, nor is my whole estate*

*Upon the fortune of this present year.*

*Therefore my merchandise makes me not sad.*

**Merchant of Venice, Act I, Scene 1**

The question of wealth allocation has captivated the minds of individuals since time immemorial, predating the advent of stock exchanges. However, the academic significance of this question can be attributed to the pioneering work of Harry Markowitz (Markowitz, 1952), who laid the foundation of modern portfolio theory. Markowitz's approach relied on the assumption of investor rationality and employed mean-variance optimization to determine portfolio weights. Subsequently, the development of the Black-Litterman model (Black & Litterman, 1992) relaxed the assumption of homogeneous beliefs among investors, further advancing the field of portfolio optimization.

Concurrently, researchers also directed their attention towards the challenge of stock forecasting, exploring two primary approaches: fundamental analysis and technical analysis. Fundamental analysis, exemplified by the influential 3-factor model (Eugene F. Fama, 1993). seeks to establish a relationship between fundamental factors and stock performance. On the other hand, technical analysis investigates patterns and trends in historical price and volume data. Numerous studies have investigated the efficacy of technical indicators in stock forecasting. For instance (Vargas, et al., 2018) combined news texts and technical indicators as inputs to an LSTM model, while (Huang, et al., 2021) utilized 22 years of S&P companies' quarterly financial data in a Feed-forward Neural Network.

The emergence of graph neural networks has also garnered attention in the realm of stock forecasting. In a notable attempt to predict stock prices (Chen & Wei, 2018) employed a graph representation of shareholders' ownership. They encoded individual stock features using Recurrent Neural Networks (RNNs) and leveraged the representations of neighboring nodes to forecast stock price performance using Convolutional Graph Networks (CNNs). Another study (Matsunaga, et al., 2019) expanded the universe of relations beyond shareholder types, incorporating "supplier," "customer," and "partner" relations. They introduced a dot product of node embeddings in the computations of convolutional layers to capture the dynamic nature of stock co-dependence over time (Temporal Graph Convolution).

These diverse approaches and methodologies highlight the multi-faceted nature of stock forecasting and wealth allocation. The fusion of portfolio optimization techniques, fundamental analysis, technical analysis, and the burgeoning field of graph neural networks showcases the ongoing quest for improved accuracy and effectiveness in forecasting stock performance.

## CHAPTER 3. METHODOLOGY

In this chapter I outline main assumptions of end-to-end optimization process. Following the previous work on this theme (Costa & Iyengar, 2022) I use approach to portfolio optimization. This approach postulates that the combine two tasks: predicting future returns and optimizing weights into a single model. This done in the following fashion:

* At the first stage, the predict future returns.

I assume the following structure of the problem. In order to make the training of model more stable a set of past predictions is also calculated.

Secondly, the optimization problem requires both the point prediction and the prediction errors as inputs to quantify and control the model risk. In the optimization layer, we calculate a nominal loss function, that is composed of Mean Squared errors of past predictions minus possible portfolio future return. At this stage we got weights that are later used in task loss that is our ultimate goal to optimize. It can be one of widely known financial metrics such as Sharpe Ratio, etc. represents features variables.

As a mathematical problem, I try to optimize the following function:

* represents target variables,
* represent weights that are chosen during the optimization process.
* represents realized returns used to indicate

The goal is to find weights of a neural network that helps to minimize the task loss function subject to a nominal cost function :

(1)

Nominal loss function:

(2)

Here, represents the size of the past prediction sample, and denotes the prediction window. The prediction window signifies the duration over which return is predicted. The rationale behind employing a longer prediction window is to make predictions less stochastic as over longer time horizon stocks exhibit more stable behavior.

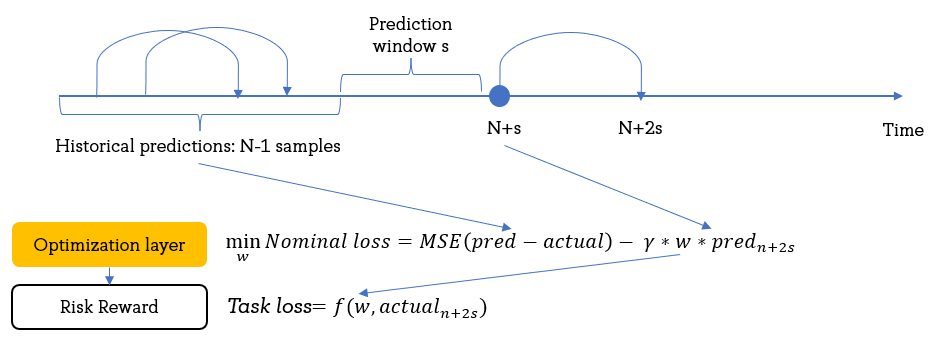
In this study, two target functions are employed: the Maximum Return and the Sharpe Ratio.

Figure 1: Schema of calculating *nominal* and *task losses*

### Section 3.1 Prioritized experience replay

Similar to the approach adopted in the original study (Costa & Iyengar, 2022), I incorporated a set of previous predictions to account for model risk. However, instead of assigning weights to these predictions based on their distribution and deviation from the mean, I opted to employ the concept of Prioritized Experience Replay (PER), which has gained popularity in the field of Reinforcement Learning (Schaul, et al., 2016). The fundamental principle behind PER is to train an agent using the specific cases where the agent obtains suboptimal results. By storing these "exceptional" cases in a buffer and retrieving them during training, the agent is expected to learn more efficiently.

In a broader context, PER addresses three key aspects of agent learning:

1. Efficiency: By prioritizing crucial and informative experiences, the agent can enhance its learning effectiveness and potentially reduce the overall number of experiences required for training.
2. Speed: Focusing on the most valuable experiences enables the agent to learn at an accelerated pace.
3. Performance: Empirical findings have demonstrated that agents employing PER often achieve superior performance across a range of tasks.

In my specific case, I chose to employ PER to store the indices of time periods when the losses were most substantial, indicating notable challenges in portfolio optimization. By employing this approach, I anticipate mitigating the need to model the distribution of portfolio returns, which may exhibit instability over time. Moreover, this approach does not introduce any additional learnable parameters, thus maintaining a simple model structure and reducing the susceptibility to overfitting.

### Section 3.2 Relation Graph Attention Networks

In contrast to conventional approaches, I opted to utilize Graph Neural Networks (GNN) for the prediction layer in my study. GNNs are a specialized type of neural network explicitly designed to process and analyze data structured as graphs. This approach has garnered significant attention in recent years owing to its capability to effectively model relationships in irregular data. Unlike Convolutional Neural Networks (CNNs), which are predominantly suited for Euclidean data, GNNs excel in handling non-Euclidean data, such as social networks, molecular structures, citation networks, and recommendation systems.

The emergence of GNNs took a leap forward in 2018 when graph models with attention mechanisms were introduced (Veličković, et al., 2018). This breakthrough opened new avenues for enhanced graph-based modeling. Subsequently, Relational Graph Attention Networks (R-GAT) (Busbridge, et al., 2019) emerged as the next stage in the development of neural networks. R-GATs introduced the concept of multiple relations within a graph, enabling more comprehensive modeling capabilities. By incorporating attention mechanisms across relations, R-GATs excel in capturing interactions among the nodes at multiple levels.

The input to an R-GAT is a graph, defined as a set of nodes and edges. Each node and edge in the graph is associated with a feature vector. These vectors serve as the initial representations for nodes and edges. The key component of R-GATs (and GATs in general) is the attention mechanism. For each node, the model computes an attention score for each of its neighbors. This score determines how much influence the neighbor's features should have on the node's new feature representation. In R-GATs, the attention score is also dependent on the edge features, allowing it to incorporate relational reasoning. Once the attention scores are computed, the model updates each node's features by taking a weighted sum of its neighbors' features, where the weights are the attention scores. Just like in the original GAT, R-GATs often use multiple attention heads. Each head learns to pay attention to different aspects of the neighbors' features, and the outputs from all heads are concatenated to form the final updated node features.

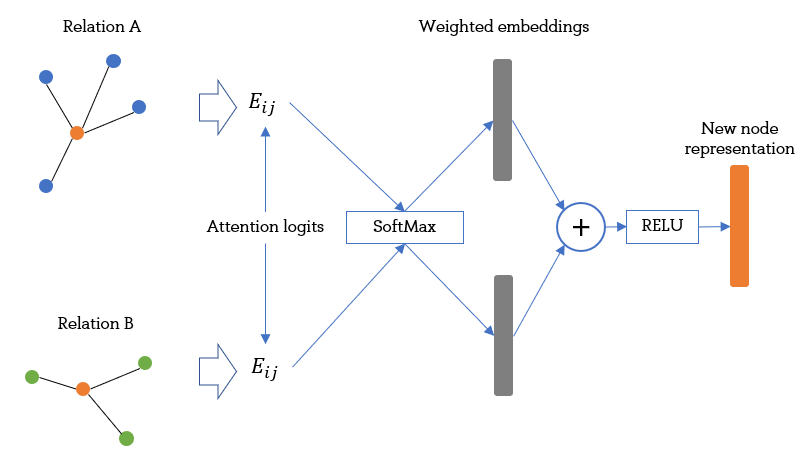
There are two types of normalizing attention logits: Within-Relation Graph Attention (WIRGAT) and Across-Relation Graph Attention (ARGAT). The latter averages attention logits irrespective of the relation type, while the former averages only within each Relation type. In this study, I use both to compare their relative performance.

Figure 2. Schematic work of RGAT.

Attention logits are computed for each relation type with the help of both query and key kernels:

Then the additive attention is applied, since we want to take into account edge features as well, we add then to the attention logits)

After that, we reweight attention weights using *across-relation attention mechanism*:

After that, the new node embedding is obtained through propagation:

In case of multiple-head attention, we aggregate individual embeddings by concatenation.

## CHAPTER 4. MODEL ARCHITECTURE

### Section 4.1 End to end portfolio construction

The algorithm introduced in this study aims to leverage past prediction errors to gain insights into the local landscape of time series data. To achieve this objective, the algorithm follows a systematic process. First, N-1 samples are obtained, predictions are made, and the corresponding errors are calculated. Subsequently, we take a time gap of prediction window and the algorithm predicts the desired day K based on the accumulated information. The specific steps of this algorithm are described in Algorithm 1.

**Algorithm 1. Calculation of portfolio performance (without PER)**

**for** *each day*  **do**

**for** each sample in training sample **do**

predict future return

calculate the error

aggregate with previous errors to the sum

Predict future return for

Calculate Mean Squared Error of past prediction errors

Find optimal weights

Use optimal weights to calculate Risk-Reward

Back-propagate

It is important to highlight that there are two versions of this algorithm: one with the inclusion of Prioritized Experience Replay (PER) and one without. In our perspective, this inclusion enhances the robustness of the algorithm, as consecutive samples drawn without the intervention of PER may exhibit high similarity, potentially leading to a form of local overfitting.

Another idea to explore the effectiveness of the optimization layer by comparing it with a model-free approach as opposed to the established model-based one. Model-free approach is the one without the optimization layer. We take the outputs of the prediction layer and consider them to be the logits of weights. By taking a SoftMax, we obtain weights that are included in the risk reward function. The difference is depicted in Figure 3 and Figure 4.

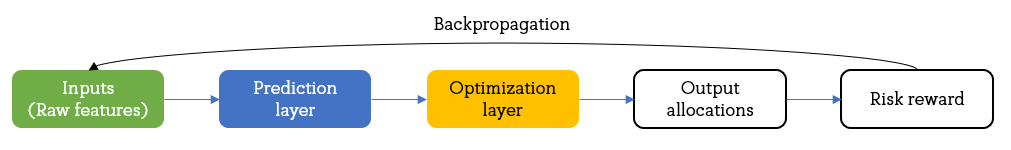
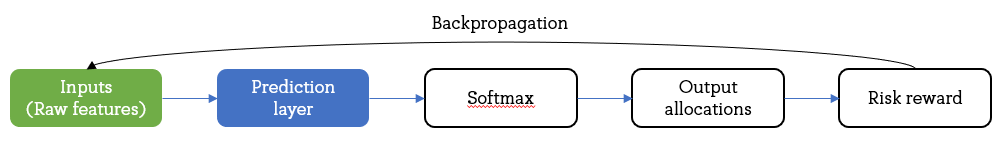


Figure 3: Computational schema of **model-based** approach

Figure 4: Computational schema of **model-free** approach



#### Section. 4.1.1 Prediction layer

#### Section 4.1.2 Optimization layer

The Python package CvxpyLayer (Agrawal, et al., 2019) is a seamlessly integrated tool within the widely used deep learning package Pytorch. By incorporating this package, users can construct a computational graph with efficient back-propagation capabilities. CvxpyLayer builds upon the foundation of the convex optimization package cvxpy, enabling the solution of optimization programs where the output of a layer depends on the solution of the previous layer.

Utilizing a convex optimization layer in a neural network can enhance interpretability, as it explicates the relationship between consecutive layers through the optimization program. When the convex optimization problem has a unique solution, the convex optimization layer functions similarly to a layer with a deterministic functional relationship. However, in cases where an analytical solution is either nonexistent or impractical, the convex optimization layer offers an elegant approach to encode such relationships.

In the context of the end-to-end model-based portfolio, we employ the convex optimization layer within the network to solve the weights optimization problem. By incorporating the convex optimization layer, we can effectively address optimization challenges and achieve more robust and tractable portfolio optimization within the overall model.

## CHAPTER 5. EXPERIMENTS ON MARKET DATA

### Section 5.1 Collecting the data

The decision to utilize ETFs as the target universe instead of individual stocks is based on two primary reasons. Firstly, it addresses the concern of survivorship bias prevalent in stock selection. In today's market, selecting stocks solely based on their performance introduces a risk of overfitting, as only the successful companies that have not been acquired or gone bankrupt are observable. Consequently, mitigating this bias requires additional measures, such as obtaining information on ceased tickers, which can be challenging. ETFs offer a solution to this issue by tracking the performance of indices and automatically adjusting their holdings to account for such cases.

The second reason pertains to the opportunity for diversification across multiple dimensions, including geography, asset classes, and investment styles, among others. Achieving a similar level of diversification with individual stocks would necessitate considering a wide range of tickers. However, due to limitations in data availability and computational costs associated with processing large graphs, this approach is not feasible. Consequently, ETFs emerge as ideal targets for experimentation due to their inherent ability to provide diversification benefits across various dimensions.

The data utilized in this study was obtained by gathering historical quotes from Yahoo Finance. A total of 20 exchange-traded funds (ETFs) were included in the analysis, with a comprehensive list of these ETFs provided in Appendix A. The selection of ETFs was based on several criteria, including their capitalization, asset class, geographical focus, and thematic attributes (such as sectors, dividends, or growth/value characteristics). In order to encompass a broad representation, ETFs from four major asset classes were chosen, namely Equity, Bonds, Commodities, and Real Estate. Furthermore, ETFs targeting various regions around the world, including the United States, Developed Markets, and Developing Markets, were also included, ensuring diversity in the sample.

For each ticker, detailed price information (open, close, high, low) and volume data were collected. As different ETFs were launched in different years, a subset of quotes encompassing their common intersection was utilized for the analysis. The dataset covers a time span ranging from 13th September 2004 to 13th April 2023, providing a comprehensive view of the market dynamics over the considered period.

### Section 5.2 Benchmark

For the benchmark I use simple average of ETF performances. Since the ETF universe described in the study includes several asset classes with vastly divergent risk-return profiles: stocks, bonds, real estate, commodities, it is not relevant to measure only against specific index like S&P 500 because it is applicable only to the part of the universe that is connected to US stock market.

### Section 5.2 Features generation

To capture relevant information from the ETF price-volume data, I derived a comprehensive set of features based on widely recognized technical indicators. These indicators included the Relative Strength Index (RSI), Moving Average Convergence-Divergence (MACD), Stochastic Oscillators, and several others. The specific selection of indicators and their respective parameters can be found in Appendix B, providing transparency and reproducibility to the analysis.

Additionally, I incorporated cumulative past returns for various time frames, namely [1, 3, 7, 14, 30, 40, 50, 60, 70, 80, 90] trading days. Notably, it is crucial to clarify that when referring to days, I specifically mean trading days rather than calendar days. This distinction is necessary to accurately capture the dynamics of the ETF market.

Overall, leveraging the "ta" Python library, I systematically generated 14 features for each node at every time period under consideration.

### Section 5.3 Graph generation

It is common to consider modified correlation table as a prime source for adjacency matrix. For example, In papers (Taylor & Cerbo, 2019), (Jaeger & Marinelli, 2022) it was proposed that *minimum spanning tree* *(MST)* and *maximum filtered planar tree (MFPL)* can better picture market structure than the representation obtained based on correlation matrix.

Therefore in the following approach was used:

* Gower distance metric as a distance to measure nodes connectivity:
* (3)

Then correlation is close to 1, the distance is approaching unity and then correlation is close to -1, the metric is approaching 1, that helps to alleviate the problem of different sign in correlation matrix.

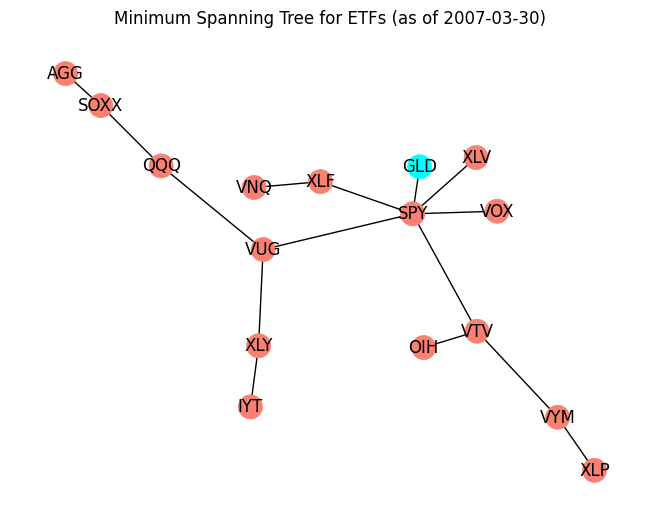
**

Figure X: Example of *minimum spanning tree* on ETFs 90-periods return correlation

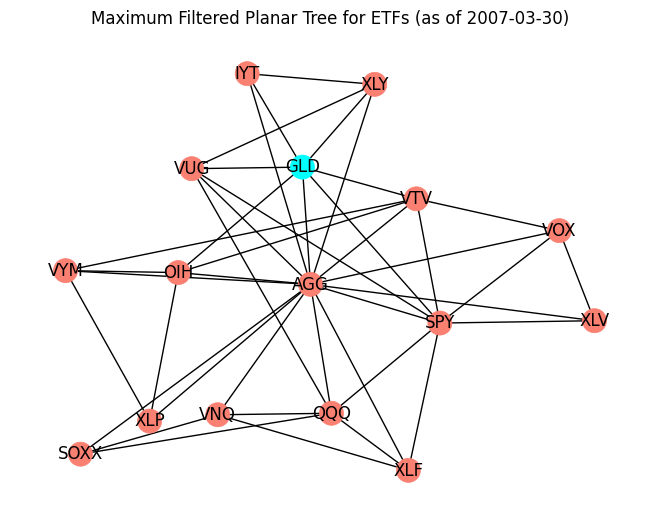
**

Figure X: Example of *maximum filterted planar tree* on ETFs 90-periods return correlation

In this work, both types of graphs and adjacency matrices are used. They are applied both to ETF return and ETF trading volume changes.

In addition to them another novel method has been developed. The first one is Relative Rotation Graph (RRG). These charts were developed by Julius de Kempenaer and have been available on Bloomberg terminals since 2011.[[1]](#footnote-1) The graph has two axes measuring relative momentum and relative strength respectifully. There are four quadrants on the RRG chart: Leading (strong relative strength and strong momentum), Weakening (strong relative strength but weakening momentum), Lagging (weak relative strength and weak momentum) and Improving (weak relative strength but improving momentum). The idea is to cluster stocks that appears to be in one quadrant, suspecting that similar economic forces lead to over and underperformance of the ETF relative to the benchmark.

It is natural to cluster stocks by region, industry or secotre, as well, as by their capitalization. It was shown (Marcos de Prado) that clustering approach can help to achieve better portfolio diversification.

|  |  |  |
| --- | --- | --- |
| # | Variable | Type of graph |
| 1 | Daily Return | MST |
| 2 | Daily Return | MFPL |
| 3 | Daily Volume Change | MST |
| 4 | Daily Volume Change | MFPL |
| 5 | Daily Return | Clustering based on RRG |

Table 1: Description of Relation Graphs

The whole description of the methods to create graphs can be found in Appendix C.

### Section 5.4 Training process

### Section 5.5 Experiments

* With and without PER
* Different number of heads
* Different task loss (max return, sharpe ratio
* Different constraints on weights no constraints, 25%, 10%
* Different prediction window?
* Allow short (with restrictions (no more than 30%)

### Section 5.6 Performance comparison

## CHAPTER 5. CONCLUSION

## REFERENCES

Black, F. & Litterman, R., 1992. Global Portfolio Optimization. *Financial Analysts Journal*, Volume 48, pp. 28-43.

Busbridge, D., Sherburn, D., Cavallo, P. & Hammerla, N. Y., 2019. Relational Graph Attention Networks.

Chen W, J. M. Z. W. G. e. a., 2021. A novel graph convolutional feature based convolutional neural network for stock trend. *Information Sciences*, Volume 556, pp. 67-94.

Chen, Y. & Wei, Z., 2018. Incorporating Corporation Relationship via Graph Convolutional Neural Networks for Stock Price Prediction. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Costa, G. & Iyengar, G. N., 2022. Distributionally Robust End-to-End Portfolio Construction.

Eugene F. Fama, K. R. F., 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, Volume 33, pp. 3-56.

Fama, E. . F. & French, K. R., 1969. The Adjustment of Stock Prices to New Information.. *International Economic Review*, Volume 10, pp. 1-21.

Huang, Y., Capretz, L. F. & Ho, D., 2021. Machine Learning for Stock Prediction Based on Fundamental Analysis. *IEEE Symposium Series on Computational Intelligence (SSCI)*.

Markowitz, H., 1952. Portfolio Selection. *The Journal of Finance*, Volume 7, pp. 77-91.

Matsunaga, D., Suzumura, T. & Takahashi, T., 2019. Exploring Graph Neural Networks for Stock Market Predictions with Rolling Window Analysis. *arXiv*.

Schaul, T., Quan, J., Antonoglou, I. & Silver, D., 2016. Prioritized Experience Replay.

Vargas, M. R., Anjos, C. E. M. d., Bichara, G. L. G. & Evsukoff, A. G., 2018. Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles. *2018 International Joint Conference on Neural Networks (IJCNN)*.

Veličković, P. et al., 2018. Graph Attention Networks.

## Appendix A. ETF Description

| **Ticker** | **Name** | **Capitalization (bn)[[2]](#footnote-2)** | **Description** | **Asset class** |
| --- | --- | --- | --- | --- |
| AGG | iShares Core U.S. Aggregate Bond ETF | 89.17 | Tracks an index of US investment-grade bonds. The market-weighted index includes Treasuries, agencies, CMBS, ABS and investment-grade corporates. | Bonds |
| BND | Vanguard Total Bond Market Index Fund |  | Tracks a broad, market-value-weighted index of US dollar-denominated, investment-grade, taxable, fixed-income securities with maturities of at least one year. | Bonds |
| BSV | Vanguard Short-Term Bond Index Fund | - | Tracks a market-weighted index of US-government bonds, investment-grade corporate and investment-grade international dollar-denominated bonds with maturities of 1-5 years. | Bonds |
| GLD | SPDR Gold Shares | - | Tracks the gold spot price, less expenses and liabilities, using gold bars held in London vaults. | Commodity |
| IEMG | iShares Core MSCI Emerging Markets ETF | - | Tracks a market-cap-weighted index of emerging-market firms, covering 99% of market capitalization. | Equity |
| IYT | iShares Transportation Average ETF | - | Tracks a broad-based, modified market-cap-weighted index of US stocks in the transportation industry. | Equity |
| OIH | VanEck Oil Services ETF | - | Tracks a market-cap-weighted index of 25 of the largest US-listed, publicly traded oil services companies. | Equity |
| QQQ | Invesco QQQ Trust | - | Tracks a modified-market-cap-weighted index of 100 NASDAQ-listed stocks. | Equity |
| SOXX | iShares Semiconductor ETF | - | Tracks a modified market-cap-weighted index of 30 US-listed semiconductor companies. | Equity |
| SPY | SPDR S&P 500 ETF Trust | - | Tracks a market cap-weighted index of US large- and mid-cap stocks selected by the S&P Committee. | Equity |
| VEA | Vanguard Developed Markets Index Fund | - | Passively managed to provide exposure to the developed markets ex-US equity space. It holds stocks of any market capitalization. | Equity |
| VOX | Vanguard Communication Services Index Fund | - | Seeks to track the performance of a benchmark index that measures. | Equity |
| VTV | Vanguard Value Index Fund | - | Tracks an index of large-cap stocks in the US. Holdings are selected and weighed based on five value factors. | Equity |
| VUG | Vanguard Growth Index Fund | - | Tracks an index of large-cap stocks in the US. Holdings are selected and weighed based on growth factors. | Equity |
| VWO | Vanguard Emerging Markets Stock Index Fund | - | Passively managed to provide exposure to the emerging markets equity space. It holds stocks of any market capitalization. | Equity |
| VYM | Vanguard High Dividend Yield Index Fund | - | Tracks the FTSE High Dividend Yield Index. The index selects high-dividend-paying US companies, excluding REITS, and weights them by market cap. | Equity |
| VNQ | Vanguard Real Estate Index Fund | - | Tracks a market-cap-weighted index of companies involved in the ownership and operation of real estate in the United States. | Real Estate |
| XLF | Financial Select Sector SPDR Fund | - | Tracks an index of S&P 500 financial stocks, weighted by market cap. | Equity |
| XLP | Consumer Staples Select Sector SPDR Fund | - | Tracks a market-cap-weighted index of consumer-staples stocks drawn from the S&P 500. | Equity |
| XLV | Health Care Select Sector SPDR Fund | - | Tracks health care stocks from within the S&P 500 Index, weighted by market cap. | Equity |
| XLY | Consumer Discretionary Select Sector SPDR Fund | - | Tracks a market-cap-weighted index of consumer-discretionary stocks drawn from the S&P 500. | Equity |

## Appendix B. Technical indicators

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Parameters Specification** |
| RSI | Momentum |  |
| MACD | Trend |  |
| Vortex Indicator | Trend |  |
| Stochastic Oscillator | Momentum |  |
| Williams Indicator | Momentum |  |
| Ulcer Index | Volatility |  |
|  |  |  |

## Appendix C. Graphs generation

Relative Rotation graph:

Algorithm[[3]](#footnote-3):

1. Prices are loaded and normalized relative to the first value of the period
2. The price of the stock relative to the benchmark is calculated
3. A rolling mean and standard deviation with a window size of 50 days is calculated for this relative price
4. The JdK RS ratio is then calculated as follows:100 + ((relative - rolling\_mean) / rolling\_std)
5. For the momentum, the price is calculated for each day relative to the price 10 days ago
6. It's going back to normal with a rolling mean of 50 days
7. I then calculate the JdK RS momentum analogously with:100 + ((momentum - momentum\_rolling\_mean) / momentum\_rolling\_std)
8. Finally, the RS ratio and RS momentum are smoothed with a rolling mean of 10 days

1. https://www.relativerotationgraphs.com/about/the-company [↑](#footnote-ref-1)
2. As of 17.05.2023 [↑](#footnote-ref-2)
3. https://msw.flxn.de/tool/sector/ [↑](#footnote-ref-3)