ML Robo Advisory: A Comparative Study between Classical Portfolio Theory and Machine Learning Techniques for Active Stock Portfolio Management

by

Abdussamad Usman

University of St Andrews

Supervisor: Prof. Tom Kelsey

Word Count: 14,227

15 August 2023

Declaration of Authorship

This thesis is submitted to the Department of Computer Science at the University of St Andrews as part requirement for the degree of MSc. Artificial Intelligence. I certify that this thesis has been written by me, and that it is the record of work carried out by me, and that it has not been submitted in any previous application for a higher degree.

Abdussamad Usman
ID: 180023172

Acknowledgments

I want to extend my heartfelt thanks to Professor Tom Kelsey for his guidance and motivation during this project. Additionally, I would like to express my gratitude to my family and friends for their unwavering support throughout my degree.

Sic itur ad astra. -Virgil

Abstract

Throughout history, financial markets have facilitated the transfer of resources and wealth for both retail and institutional investors. Despite substantial global technological advancements, the complexity of financial markets remains typical. There is a continuous growth of financial products in markets. Portfolio management is becoming more complex for both retail and institutional investors, and the effectiveness of classical techniques for portfolio management in today's markets is questioned. However, the expanding function of artificial intelligence in finance presents an opportunity to simplify this process and potentially increase returns. Utilising machine learning techniques to automate portfolio creation, selection, and management processes may prove advantageous. This paper explores the efficacy of portfolio construction and management using machine learning techniques. We compare the performance of the models across several error metrics. A total of 17 model variants were tested from 8 classes of models. These include both classical and Machine learning models. The results show that machine learning models are able to generate statistical alpha and in some cases outperform classical portfolio theory models.

Contents

List of Figures

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

API Application Programming Interface

CAGR Compound Annual Growth Rate

CLA Critical Line Algorithm

CNN Convolutional Neural Network

CPU Central Processing Unit

DRL Deep Reinforcement Learning

EMA Exponential Moving Average

GPU Graphics Processing Unit

HRP Hierarchical Risk Parity

IVWP Inverse Volatility Weighted Portfolio

ML Machine Learning

MPT Modern Portfolio Theory

MVP Minimum Variance Portfolio

OHLC Open, High, Low, Close

OOP Object-Oriented Programming

S&P500 Standard & Poor's 500

SEC Securities and Exchange Commission

VWAP Volume-weighted Adjusted Price

XGBoost Extreme Gradient Boosting

1. Introduction

Financial markets play an important role in the contemporary intricate economic environment as they serve as essential mechanisms for supporting and promoting economic growth in the world. They have served as a source of liquidity for businesses and entrepreneurs and as a type of investment for investors and lenders who have excess funds on which to get a return (Hayes, 2023).

Over the course of time, financial markets have undergone significant growth and diversification, encompassing a wide range of offerings. These include various components such as the Bond, Stock, Commodity, Derivatives, money markets, and numerous other sub-markets. The expansion and intricacy of the offerings in these markets are attributed to the advancements in technology, globalization, and the heightened demand for a wide range of financial products due to the rise in financial literacy. The current scale of stock markets serves as an illustrative case, whereby several stock exchanges are present throughout various nations. The United States possesses a total of thirteen stock exchanges, two of which are majorly traded (Morah, 2022). With the increased presence of the internet, financial technology (fintech) focused companies, and financial applications, more people have access to financial tools than ever before (U.S. Securities and Exchange Commission). The US Securities and Exchange Commission (SEC) mentioned that the presence of technology has narrowed down the disparity between institutional investors and retail investors. In the past, market participation was mainly done by institutions. Information about securities, including reports, secondary market data, and access to more complex securities was limited to institutions. Due to technology and the internet, retail investors now have access to similar information that institutions do (U.S. Securities and Exchange Commission). With the continuous rise in computing power, we have also seen tremendous growth in areas like artificial intelligence (AI). More industries are integrating with AI and computation in general.

However, the rise of innovation and technology in financial markets has also led to an increase in their complexity. Particularly retail investors started getting involved in more complex instruments such as derivatives, structured products, and algorithmic trading, which offer higher projected returns (Agnew & Mitchell, 2019). One major challenge of these advancements in financial technology is that it is increasingly becoming non-applicable to investors with no experience. The continuous scale of financial markets emphasizes the necessity for individuals engaged in these markets to effectively participate similarly to institutional investors. With the complexity of markets, it may be hard for retail investors to participate due to a lack of knowledge in these markets and other constraints. Investors engage in the process of constructing portfolios consisting of various assets that align with their specific investment objectives such as risk, returns, and length of investment. The term used to describe this procedure is portfolio management. Portfolio management serves as a vital area in finance where financial assets of different classes are strategically curated to fit a certain goal. This procedure is practiced by both retail and Institutional Investors.

1.1. Modern Portfolio Theory and Its Importance in Efficient Portfolio Management

Evidence of the practices of portfolio management and diversification of investments trace back to the time of Shakespeare (H. M. Markowitz, 1999). However, in the past, there was no comprehensive theory of investment that shaped the difference between efficient and inefficient portfolios or consideration of areas like risk correlation in investments. Diversification and portfolio management were based on intuition rather than empirical evidence.

Harry Markowitz, a highly renowned economist, proposed the theoretical framework known as modern portfolio theory (MPT) in 1952. MPT revolutionized the way people thought of portfolios. Historically, institutional and retail investors would consider their assets as discrete entities rather than as components of a broader investment portfolio (Amenc & Sourd, 2003a). The emergence of MPT greatly enhanced the execution of investing strategies by prompting investors to take into account the collective composition of their asset portfolios. MPT explained that the risk and return of a portfolio are not only determined by the individual characteristics of each asset but also by the interaction among each of the assets in the portfolio (Amenc & Sourd, 2003a).

In the paper "Portfolio Selection," H. M. Markowitz (1999) explains that the key objective of an investor is to maximize their return on investment and to concurrently minimize their risk. MPT emphasizes the importance of diversification as a way to reduce risk. H. M. Markowitz (1999) believes that by investing in a variety of uncorrelated assets, we can mitigate certain risks without lowering the expected returns of a portfolio (Amenc & Sourd, 2003a). This theory is closely related to the risk-return trade-off, which states that larger rewards are often linked with higher risks, prompting investors to strike an optimal balance.

1.2. Real-World Application and Motivation

Since the inception of MPT, several classical approaches to building portfolios based on foundational quantitative principles have been developed and used by Portfolio managers. With the advent of computer-aided finance, portfolio selection, optimization, and management have become pivotal research areas due to the speed and increase in complexity of markets. Computers have evolved over time to make more complex decisions on behalf of humans or to aid us in making decisions. In the past decade, significant progress has been made in the field of computational finance. Several hedge funds and quant traders now utilise cutting-edge techniques that make use of machine learning to gain an edge in the market.

There has also been a noticeable upward growth of organizations offering robo-advisory services to regular investors. These services claim to simplify all aspects of the portfolio management process through automation. Historically, the responsibility of portfolio management has been handled by human experience and traditional risk-based rules. It is also costly for ordinary investors to get a portfolio manager. Robo-advisors attempt to democratise investing and portfolio management by offering automated investment options at lower costs with above-market returns to consumers without financial knowledge or a financial advisor. This demonstrates the transformative potential of machine learning and Artificial Intelligence in general in the wealth management sector. Both large organizations and institutional investors may therefore find automation of the entire investment process, including the selection of appropriate investments and allocation of the size of each investment to be very useful.

Given this shift from human-driven classical techniques to algorithmic machine learning-based techniques, It becomes critically important to investigate the efficacy of these techniques and their benefits and drawbacks. In the world today, we see multiple cases in which AI outperforms human intelligence in terms of speed and efficiency as some of the models are able to uncover intricate patterns and relationships within vast datasets that are considered impossible for humans to do manually.

An important question arises: Can machine learning methods perform better than classical portfolio construction techniques in creating active portfolios, or is there an inherent human wisdom that machines cannot replicate? The objective of this paper is to critically study and compare the efficiency of classic risk-based approaches with machine learning-based approaches in constructing well-diversified portfolios on long-term horizons. The study also seeks to evaluate the performance of the models on the same metrics and use the empirical findings to construct an automated portfolio construction pipeline.

1.3. Research Objectives

This project has been split into primary objectives and extension tasks. The primary objectives were attempted first before working on the extension tasks.

Primary Objectives:

1. Research and exploratory analysis of classic and machine learning techniques
2. Review the literature
3. Build the system design and data pipeline for a robo-advisor
4. Create a set of portfolios using both classical and machine-learning techniques
5. Analyze the different portfolio management techniques and rebalancing

Extension Tasks:

1. Continuous model retraining and optimization
2. Creating a user-friendly data dashboard for assessing model performance
3. Exploring advanced techniques: Paper Replication (Convolutional Neural Networks, Reinforcement learning)

1.4. The Advent of Robo-Advisory and Automated Investment

Robo-advisors are quantitative, algorithmically driven, automated investment tools that handle the whole investment process for users with little to no human supervision (Frankenfield, 2023). The goal of a Robo-advisory system is to replicate having a human advisor who would pick out suitable investment portfolios in terms of asset allocation and diversification based on a user's investment goals (Agnew & Mitchell, 2019). They are a recent advancement in the fintech industry and mostly cater for users with little to no financial knowledge or users who cannot afford the luxury of real investment advisors.

Automated investment also spans corporate financial companies. Hedge funds that utilise quantitative strategies design algorithms to automatically invest in markets. Some companies utilise AI to identify patterns in data that could help their investment process (Grobys et al., 2022). There has also been an increase in the number of companies that utilise AI in their investment process.

1.5. What Defines a 'Good' Portfolio?

According to Markowitz, the construction of a portfolio involves a two-stage process. The initial stage, also known as the determination of the efficient frontier, involves identifying the set of portfolios that provide the greatest expected return given the selected degree of risk. The subsequent step involves the identification of the most suitable portfolio from the efficient frontier that aligns with the investor's risk tolerance (H. Markowitz, 1952; H. M. Markowitz, 1999). Experts in the financial domain have carried out extensive research on building efficient portfolios. A well-crafted portfolio is said to be one that beats the benchmark but also minimises risk and is well-diversified. Each investor has their primary investment methodology or goal. In portfolio management, the selection of these strategies would depend on the investment objective of the investor, their risk tolerance, and their confidence in the reasoning for the proposed approach. Investors usually have the following goals in mind:

1. Sector diversification
2. Risk Diversification
3. Return Maximization
4. Ethical or Social Considerations

Section 2 starts with a literature review of relevant papers and then delves into the different models evaluated and the methodology behind them. I offer a fully automated pipeline for continuous training and delivery of models and mimicking market structures. I finalise the paper with a comprehensive analysis of results that compare several classes of models and offer my views. iii

2. Literature Review

Jain and Jain, 2019 aim to research the difference in performance between traditional risk-based portfolio construction and machine learning techniques for portfolio construction. The paper provides an extensive literature review of other papers related to both traditional risk-based strategies and the possibility of machine learning-based strategies. This study examines the issues associated with the traditional mean-variance optimization method and the benefits of introducing risk diversification into portfolio creation and management using machine learning techniques. The paper then examines multiple methods from the two groups (risk and machine-learning-based) and tests for performance across several rebalancing horizons. They are judged to see which method best minimizes portfolio variance and minimises out-of-sample Conditional Value at Risk(CVaR). In the risk-based approach, the author tests Minimum Variance Portfolio (MVP), Inverse Volatility Weighted Portfolio (IVWP), Equal Risk Contribution Portfolio (ERC), Maximum Diversification Portfolio (MDP), and Market-Capitalization-Weighted Portfolio (MCWP). The author also explores the use of the hierarchical risk parity (HRP) approach, which utilises graph theory and machine learning to build diversified portfolios. The author examines all the steps of the HRP pipeline including Clustering, Quasi Diagonalisation, and the Recursive Bisection. The study also found that the machine learning methods performed worse with longer rebalancing horizons. The paper provides a strong foundation for our research and gives insight into possible areas of exploration. A point to note was the data acquisition method of the paper. They used the top 10% members of multiple indices at the time of research which is prone to survivorship bias in the data. This means that the assets universe was specified and pre-known, which in a real-world scenario would not be true.

In the paper, Building Diversified Portfolios that Outperform Out-of-Sample, Marcos Lopez de Prado introduces the HRP approach to portfolio management for constructing diversified portfolios. HRP is a clustering approach that clusters similar behaving assets (López de Prado, 2016). The author proves that HRP performs better than other traditional methods such as MVP, critical line algorithm (CLA), and the equally-weighted portfolio. The author mentions the instability of these traditional methods and the failure to perform well out-of-sample. The author proves the stability of the HRP algorithm especially when it deals with changing correlation structures and both shocks that affect the individual asset and the market in general (common shocks). The paper offers a good backbone to the literature on portfolio construction for my research. Several papers reference this paper's approach. In this study, I will attempt to use this approach to construct a well-diversified portfolio and compare its performance to traditional techniques.

Schwendner et al. (2021) discuss the use of hierarchical clustering techniques for portfolio construction. The authors perform a detailed review of the HRP approach and its limitations. Backtests were run on the 57 different HRP-type asset allocation variations on a multi-asset futures investment universe. An empirical study was carried out in which a multi-asset universe of 17 liquid futures markets where backtested using the HRP approach. The authors found that most seriation strategies underperformed compared to the HRP approach. However static tree-based methods outperformed in comparison to

HRP Schwendner et al. (2021).

A paper written by (Millea & Edalat, 2022) uses deep reinforcement learning (DRL) as done in (Benhamou et al., 2020). However, this paper combines DRL and hierarchical clustering (HC) techniques. The authors argue that the flexibility of DRL and the reliability and efficiency of HC models can be combined to outperform traditional portfolio optimisation techniques. The authors analyse various portfolio optimisation techniques such as a deep learning approach, a risk parity approach, and mean-variance optimization.

The author uses the HC model for asset allocation and then uses a DRL agent to select the best model, thereby combining the power of the two models.

Jaeger et al. (2021) discuss the limitations of traditional portfolio construction techniques and propose the use of machine learning alternatives to construct the portfolios while making sure the AI technique remains explainable. The authors provide a literature review on studies of portfolio construction techniques ranging from machine learning, risk parity, and graph theory approaches. The paper examines Naive RP(Inverse Variance), and HRP approaches. The paper also discusses the use of adaptive serial risk parity as done by (Schwendner et al., 2021). In terms of the data, the study uses a multi-asset investment universe of equity indexes, commodities, and fixed-income assets. The authors had multiple evaluation metrics including Annualized Volatility and Return, Maximum Drawdown, Sharpe Ratio, and Conditional Value-at-Risk(CVaR). The results from the study show that the HRP method had better risk-adjusted returns compared to ERC.

Benhamou et al. (2020) explores the use of machine learning techniques such as DRL and compares them to the traditional approaches for portfolio creation and optimisation. The authors include a literature review of related work. They include papers with other applications of DRL in finance, as well as traditional optimization methods. For the traditional models, the author explores the Markowitz model, the MVP approach, and the risk parity portfolio approach. For the DRL approach, the paper models the portfolio allocation planning question as a dynamic control problem in which at each time step, the model has some market information and needs to decide the optimal portfolio allocation problem and evaluate the result with a delayed reward. The paper uses an Adversarial Policy Gradient approach as suggested by one of the authors to help train the model by intruding randomness during training, which in turn makes the model more robust. The DRL approach outperformed other methods in both the 2 and 5-year timeframe. The DRL approach showed a higher return on investment and Sharpe Ratio. The DRL model showed a continuous upward deviation from the risky asset, which suggests a consistent performance in contrast to other financial models.

W. Chen et al. (2021) proposes a new approach for portfolio construction using a Mean-Variance (MV) model and Improved Firefly Algorithm (IFA) with XGBoost. This approach involves first prediction of the stock price and then the portfolio selection. The XGBoost algorithm is used for price prediction. The stocks with the highest predicted returns are then allocated to the portfolio using the MV and IFA algorithms. The approach showed good performance with respect to the other methods, especially during a market decline. The author suggests further research and improvement of this approach to ensure its validity. The only concern regarding this paper is the prediction of the price of individual assets. Stock prices have been shown to be noisy and non-stationary as also shown by (Jiang et al., 2020). A good investigation will be to see the viability of using the XGBoost model to create several portfolios as it is a robust learner. It may also be worth it to attempt both using the model as a regressor and as a classifier.

Jiang et al. (2020) discuss a unique and uncommon way to predict price trends. The authors start by discussing the complex challenge of predicting stock prices due to the non-stationarity of stock prices. They propose a novel approach that utilises images of raw price data which includes the OHLC candles, volume and moving average lines. These were fed into a convolutional neural network to predict the trend-based pattern of the stock. Rather than predicting price trends as a numerical pattern, they predict the trend from the actual image of the stock prices. The authors explain that convolutional neural networks (CNN) are designed specifically to efficiently process images and extract predictive features, which then eliminates the need for manual feature engineering as you would need with price-based prediction approaches. The authors highlight the need for techniques that strike a balance between flexibility and tractability. On the one hand, they want a method that is able to identify potentially complex predictive patterns, and on the other hand, a method that is also constrained and manageable.

In order to assess the efficacy of the methodology, the authors perform a series of experiments on the CNN approach, employing various combinations of strategies, datasets, and hyperparameters. The authors also test the approach using different hyperparameters such as the market data window. The authors vary between the use of 5, 20, and 60 days as input choices for the model, represented as images with candlestick data. The authors also factor in trading costs, which demonstrates the robustness of the experimentation.

Jiang et al. (2020) found that the predictive patterns identified by the CNN were highly robust and that the approach outperformed traditional price trend signals such as momentum and short-term reversal. The authors also explored the possibilities of transfer learning on the CNN and found that when the CNN model was estimated from daily US stock data and carried to other international markets, the model was still able to predict the patterns effectively.

Aithal et al. (2023) also provided an extensive study on portfolio construction and management using machine learning techniques. They discuss various approaches that can be used to optimize portfolio selection and improve investment performance in comparison with manual portfolio analysis techniques. The authors also emphasise the value of computer-aided finance in reshaping portfolio management. The authors start by critiquing the challenges of the manual techniques and investigating the use of several machine learning techniques such as genetic algorithms to construct portfolios.

The authors explore the use of machine learning techniques such as genetic algorithms for optimising the portfolio. They present a novel approach to grouping genetic algorithms for optimizing a group trading strategy portfolio by classifying stocks into distinct categories, including banking stocks, IT stocks, and health stocks. Genetic algorithms are a type of nature-inspired algorithm that iterates over a population of possibilities to find the optimal solution. Subsequently, a genetic algorithm is employed to optimise the portfolio within each respective category. The results were shown to have enhanced portfolio performance. Aithal et al. (2023) compare the performance of the genetic-based optimisation technique with a global MVP, which is a traditional technique. Their results indicate that the portfolio construction with genetic-based optimisation outperformed the global MVP method.

3. Methodology and Exploratory Analysis

3.1. Data Acquisition

Data Source and Asset Universe (Subset of the S&P 500) The asset universe serves as a crucial step in building a portfolio management tool. It serves as the container from which possible investments for the portfolio will be picked. During the exploration of possible assets, several constraints were considered. To construct an ideal asset universe from which the portfolio will pick possible suitable assets, there was a need for reliable data availability and quality. Given that the reliability of data is important, I chose to go with the S&P 500, which is an index that tracks the top 500 largest companies in the US stock exchanges(Standard & Poor's). The use of stock price data in this study was driven by its relative accessibility in comparison to other forms of financial market data. The availability of historical stock price data made it a suitable asset class for consideration as a universe.

3.1.1. Data Provider

The data was sourced from a single provider. I had to ensure that all data used came from a single provider to avoid discrepancies in different data providers. Financial Modelling Prep, a data provider, was able to provide several data endpoints that were vital for the project. Multiple variables were pulled from the Application Programming Interface(API). The required endpoints were:

1. S&P historical changes
2. S&P current constituents
3. Asset symbol changes Asset full historical prices (date, open, high, low, close, volume, volume-weighted adjusted price [VWAP], change over time)
4. US treasury yield rates

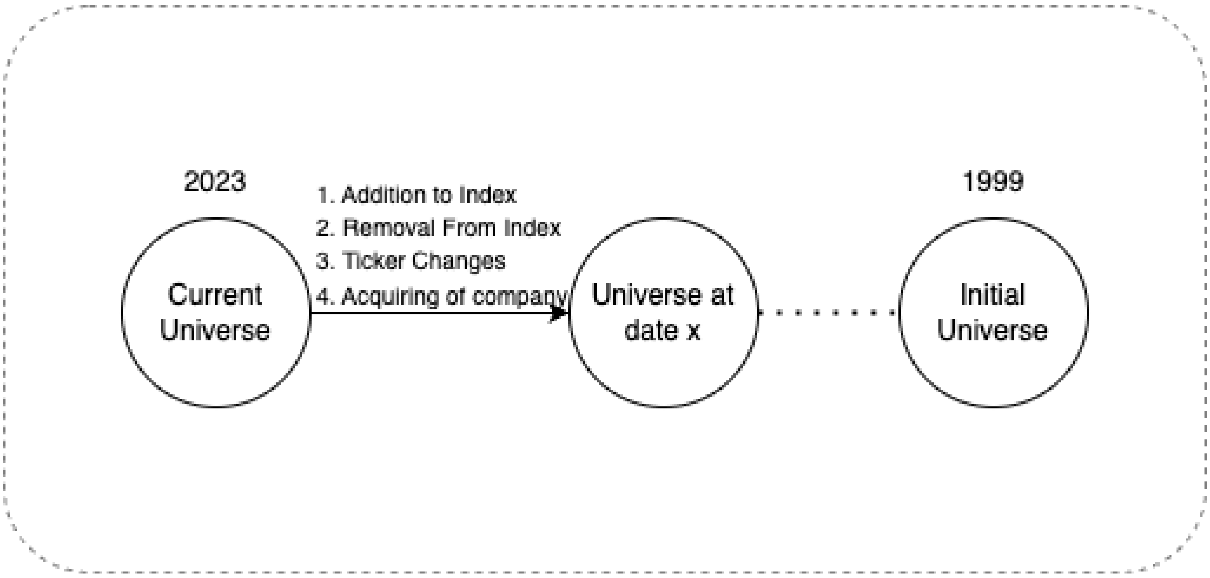

Figure 3.1.: Walk-back flow for universe construction

3.1.2. Construction of Index Walkback Script

The objective of the walk-back script in the universe construction is to construct an asset universe that mimics a real-world scenario in which the asset universe is not fixed and is unknown for future periods. To achieve this, I track the S&P 500 index over time and use this as the constituents of the universe. An issue faced was that most data providers do not give index membership with respect to time. I constructed a walk-back script that took the current constituents and then used the historical changes to identify the S&P constituents for every single date. This was important in order to have accurate S&P asset membership by date, which would give a consistent universe to track. If changes occur in the universe these are then reflected until the earliest, most accurate universe available. In this case, I have a variable universe that changes based on assets added into and removed from the S&P 500. There are three scenarios in which there are changes to the S&P 500:

1. When an asset in the index ceases to meet the indexes requirements
2. When an asset in the index is acquired by other companies
3. When there is a symbol change for an asset in the index
4. When an asset ceased to exist in the market

These were all modelled into the walkback algorithm to be able to have a consistent early universe. Once the walk-back was constructed, I noticed that the universe was fairly consistent throughout the script with its average bounds being between +10% to -10% of the original index size.

```
symbol change from:DD to DOW on  2017-09-01
 deleting(Inception of Stock) DD  on 2017-09-01 Size 491
 deleting(Inception of Stock) DOW  on 2017-09-01 Size 490
AddingWFM  on 2017-08-29 Size 489
 deleting(Inception of Stock) IQV  on 2017-08-28 Size 490
AddingAN  on 2017-08-08 Size 489
deletingBHF  on 2017-08-08 Size 490
symbol change from:ANDV to TSO on  2017-08-06
```

Figure 3.2.: Extract from the walkback script(Universe Construction) showing the change in assets in the universe

3.1.3. Data Bias Mitigation

Survivorship Bias

I faced an issue when it came to defining the asset universe (basket). Several data providers did not provide historical price details of delisted stocks and their membership in indexes. The data provider I found had the S&P 500 constituents and the constituent changes by date. If I had excluded them and used the data as is, this would have affected the quality of the results. Survivorship bias is when a researcher picks assets that are still listed and does not consider the ones that were delisted. A scenario might have been that a stock performed well and could have been in our portfolio but ended up being delisted while still being part of the portfolio due to it not meeting the index criteria. The possible loss should hence be captured. I found a financial data provider that was able to provide historical delisted stocks. I track the historical price data for delisted stocks, which is quite complex, and include it in the modelling to mimic real market conditions.

Strategy Look Ahead Bias

Look ahead bias occurs when you introduce information that is not available at a particular point in the past as part of a backtest. For example, suppose a strategy is being backtested from the year 2005 to 2020. Including data from 2008 to 2005 is look-ahead bias. This is a common error that tends to overstate the quality of results due to the availability of future data during backtesting. In this project, this was avoided by ensuring future data is not leaked at any point during the training of the models or during data engineering.

3.2. Backtester Construction & Data Engineering

3.2.1. Risk, and Return - Definitions & Relationships

A few key principles which are essential to this project need to be established. In this project, I refer to risk as volatility which is measured by the variance of the asset. The returns are calculated as the ratio between the current day's adjusted close price and the previous day's adjusted close price. The volatility is calculated using an exponential moving average of which the square of the daily return(variance) is used with a decay(alpha) factor. The formulas are given as

$$\text{Vol}_{\text{current}} = (1-\alpha) \times \text{Vol}_{\text{previous}} + \alpha \times (\text{Return})^2 \tag{3.1}$$

$$\alpha = 1 - \exp\left(\frac{\log(0.5)}{\text{halflife}}\right) \tag{3.2}$$

where
α = decay factor
$halflife$ = The rate of weight decay(set as 20 days)

Predictability of Stock Prices - Random Walk

In this project, I use machine learning models to predict a signal of the possible price movement of an asset. In stock prices, and in portfolio management in general, It is said that past performance is not really indicative of future performance. Stock prices have been seen to move in a random walk, which means, that the expected value of a price given its history is equal to the current price. This means the prices are fully independent and not correlated (Malkiel, 2003).

Harvey (1989) equates the prediction of stock prices to using the rearview mirror in a car to drive forward. The author however states that it is not impossible but the signals are noisy and it is difficult. Some authors believe forecasting prices does indeed work. Mallikarjuna and Rao (2019) performed an evaluation of forecasting methods for forecasting stock market returns on different market indices. The objective of their study is to help with better risk management and portfolio diversification through forecasting. The authors found that different markets may have ideal forecasting techniques which may be due to the non-similar behaviour of different markets.

Although the noisiness and unpredictability of stock price data were taken into account during the construction of the methods. I do not inherently attempt to use the exact predicted prices for trading or weight distribution but I use the general trend as a signal. For example, in the XGBoost regression strategy, if a stock has a positive predicted return, I invest in it regardless of the magnitude of the return.

3.2.2. Trading Frequency and Timing

The study focuses on long-term price movements and makes no attempt to target intraday price movements. The strategies I implement are in no way expected to perform on shorter timeframes. To keep the potential portfolio turnover realistic, the strategies trade weekly and I make use of daily price data.

The portfolio backtests were run from the earliest day on which data is available for assets in the universe. The backtest ran for a period of 25 years with the start date of the backtest being 1998 and the end date is 2023.

3.2.3. Portfolio Constraints

The constraints in this project were:

1. No short selling: The strategy was configured to only take long positions. The strategies can only buy and not short-sell of borrowed assets.

2. Sticking to stock asset class: An investment portfolio can span across different asset classes. Due to the availability of data and increased complexity, I am only using stock data and treasury yield data. Future versions could include other asset classes.

3.2.4. Portfolio Weighting Methods: Equal- vs. Signal-Weighted Portfolios

In this paper, I further group the approaches for allocating capital throughout a portfolio by two significant types of weighting that are commonly used in portfolio management. Both options provide their distinct advantages and also have their possible drawbacks.

The signal-weighted portfolio operates on the premise that the allocation of weights to each asset within the portfolio will be determined proportionally based on a specific signal such as momentum, valuation, or other fundamental or technical indicators (Russell, 2020). Signal-weighted portfolios can exploit market inefficiencies by systematically over and underweighting the assets based on the positive or negative signal that is gotten. Examples include the market cap-weighted portfolios and the volatility-weighted portfolios. The market cap and volatility serve as signals for weighing the assets.

An equal-weighted portfolio is a simple approach to portfolio weighting. As the name suggests, capital is equally distributed across all assets in the portfolio. In the model sense, the asset weights are all equal. This technique promotes neutrality and the reduction of concentration risk and overreliance on single assets within the portfolio. This opposes the signal-weighted portfolio, as assets with lower market caps, which are potentially riskier, are assigned the same weights as assets regarded to have greater stability.

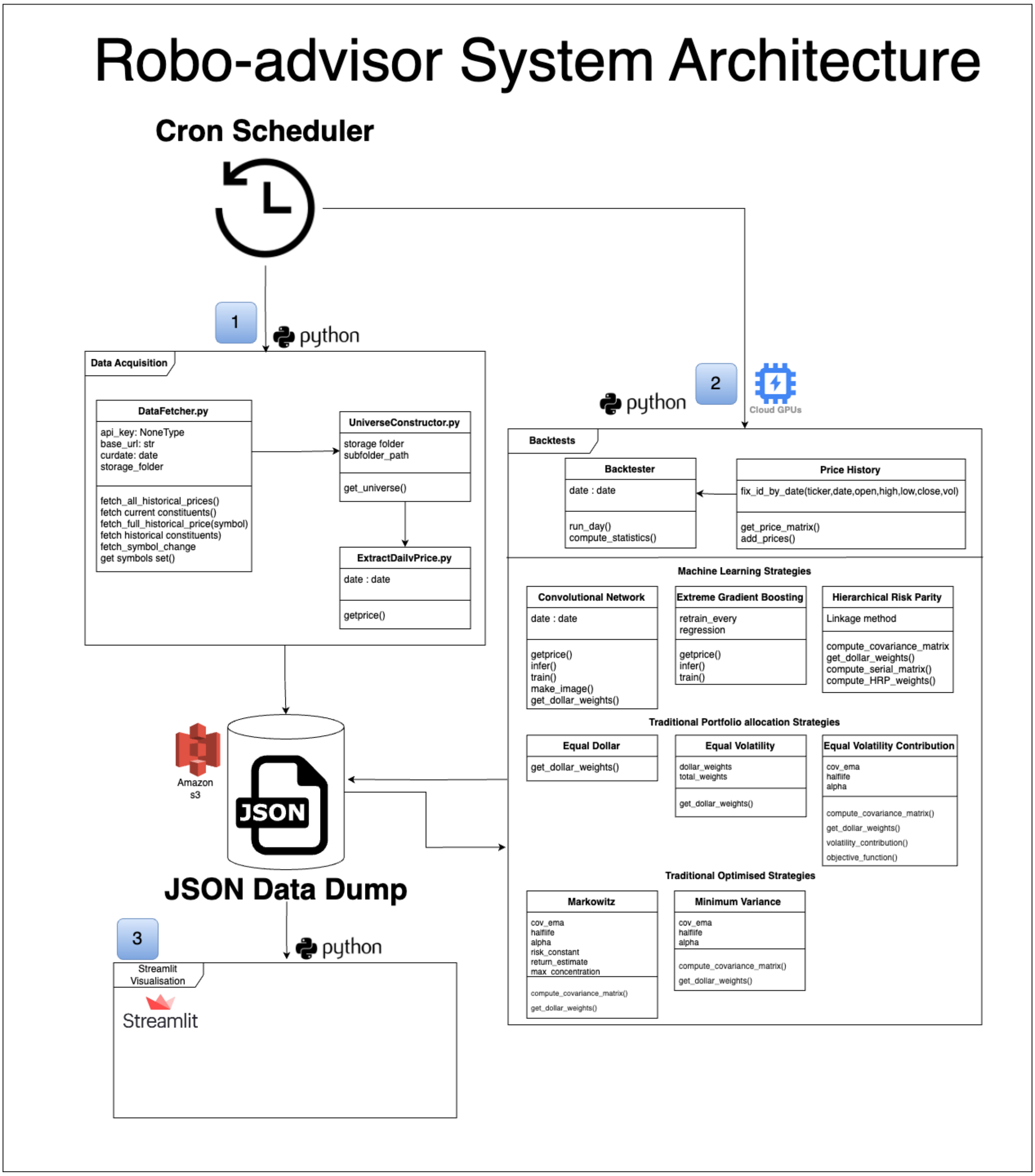

Figure 3.3.: System Architecture of the Robo-Advisor

3.3. System Architecture - Design Decisions, Deployment, Code Modularity, and File Formats

Initial data exploration was done on a Jupyter notebook to give insights into the data structure, help prepare the data pipeline, and test the strategies on smaller scales before moving to object-oriented scripts. Python was chosen as the programming language for the application because it is better adapted for an Object-oriented programming(OOP) approach and has exceptional availability and support for ML and numerical libraries. OOP also allows for code abstraction and reuse. Some complex classes are shared between models, which allows for better code design.

The application was written in Python. It was divided into modules that each perform different operations. Figure 3.3 shows the system architecture of the application and how each module communicates with all the others. I have a data acquisition module for sourcing the data from the data provider, local raw data storage and engineered data folders for local consumption by the model, and a module for the backtesters containing the main backtest script that runs each individual strategy. Each strategy was designed to have its own class and to utilize the backtester class for running the backtests. The backtester class collects the strategy type and the parameters for the strategy and passes it to the appropriate strategy class to get the model weights. Once the weights from the models are sent to the backtester class, they are then used for trading. The diagram in figure 3.4 shows how the daily backtests are run. A daily cronjob is programmed to run a shell script, which runs all the Python modules, including all the data fetching and backtesting scripts at 9:30 PM BST. This is after the US markets close and guarantees prices are available for the current day.

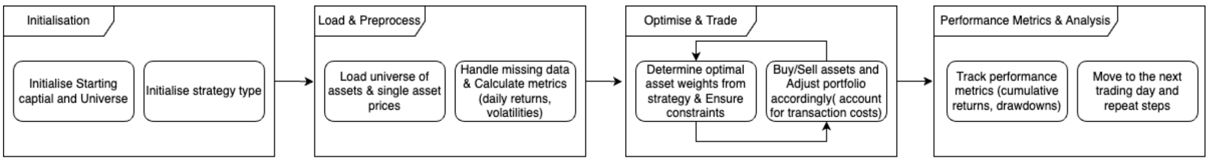

Figure 3.4.: Backtester Life Cycle

3.4. Additional Steps Taken

Due to the computational load of the application scripts, a few additional steps were taken to ensure the system performed optimally.

1. Parallel execution of shell commands: I was able to cut down the duration of the backtests significantly by parallelly executing shell commands to run on different processes at the same time rather than sequentially. This better utilised the CPU. The only scripts that had to be run sequentially were the data-fetching scripts. This was to ensure that the API provider does not rate limit the endpoints.

2. Using a GPU: I utilised 2 V100 GPUs, which were used with Pytorch CUDA to speed up the training process for the machine learning models. The CNN model, for example, would take more than 2 days to train without the GPUs.

3. Gurobi Optimiser: I started by using the Scipy optimiser for some of the models. It became too slow, so I had to find a faster solver that would be able to perform the same function. I found a quadratic solver called Gurobi that was able to drastically reduce optimisation time.

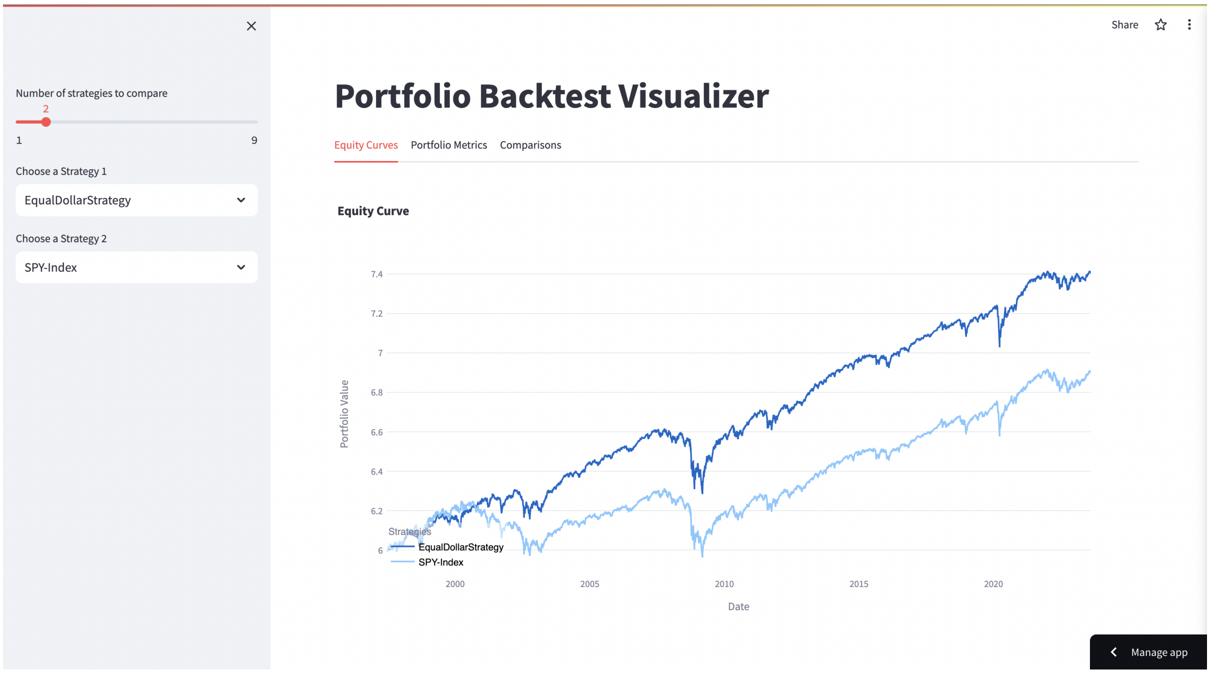

Figure 3.5.: Application Dashboard for viewing results accessible at (Link)

3.5. Trading Transaction Costs and Capital

To ensure the accuracy of the strategies, transaction costs were incorporated into buying and selling of assets in the portfolio. In real life, these scenarios are taken into account as they could build up over time and affect returns if not properly modelled. In this scenario, I assumed trading costs to be set as 0.1% of each transaction. This is assumed to handle broker costs and stamp duty costs. All trading positions are filled at VWAP, which serves as a benchmark of the average price a stock has been traded in a day (Fernando, 2023). I start with a balance of $1,000,000 in cash. When the backtest begins, the balance is left in cash, and when there are new positions to be filled, the cash amount needed to satisfy the weights is then subtracted from the cash balance, including the trading fees.

3.6. Strategies

I have grouped the strategies into three classes: baseline strategies, optimised classic strategies, and machine learning strategies. The traditional strategies are strictly rule-/formula-based. They trade based on a simple set of rules. The traditional optimised portfolios are typically rule-based, however, they are optimising for a goal. The machine learning methods hope to identify signals within the data, which combine with traditional portfolio weightings to form a portfolio. In each subsection, I explain the structure of each strategy and the functions necessary to achieve the strategies. I start by explaining the basis of the strategies and then explain the formulas behind the strategies. I then explain my rationale behind choosing the strategy.

3.6.1. Equal Volatility Strategy

The Equal Volatility Strategy is a portfolio construction technique that aims to distribute capital across different assets based on their volatility. The objective is to equalize the contribution to the overall portfolio volatility from each asset. In the portfolio, you would have assets with varying volatility of which stocks with higher volatility would be given a lower contribution and assets with lower volatility would be given a higher contribution to the portfolio due to the weight being the inverse of the volatility.

Equal Volatility Algorithm Breakdown

1. Volatility Calculation: The volatility is usually calculated as the standard deviation of asset return. I adjust dollar weights proportional to the inverse volatility: In this case, the strategy calculates the weight of each asset as the inverse of its volatility.

$$\omega_i = \frac{1}{\mathrm{vol}_i} \tag{3.3}$$

$$\mathrm{vol} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(R_i - \mu\right)^2} \tag{3.4}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{3.5}$$

where:

ω_i = weight of asset

Vol = Volatility of asset

μ = Periodical mean return

While the Exponential Moving Average(EMA) is calculated as:

$$EMA_t = (1-\alpha)\times EMA_{t-1} + \alpha \times x_t \tag{3.6}$$

x_t = Value at time t

α = decay factor

2. Normalization of Weights: The weights gotten from the calculation of each asset are then normalized to sum up to 1, which ensures that the total portfolio allocation remains as a whole.

$$\omega_i = \frac{\omega_i}{\sum_{i=1}^{N}\omega_i} \tag{3.7}$$

Strategy Rationale

The rationale behind this strategy is focused on risk management (reducing risk). Given that the weights are assigned as the inverse of the volatility, assets with higher risk are given less weighting in the portfolio. The approach also helps in reducing the impact of single-asset volatility, which means the portfolio is more resistant to market shocks and crashes. This strategy is also equally weighted in the sense that all stocks are invested in no matter how small the investment is. Nothing is left in cash. The strategy also assumes that past volatility is indicative of future volatility, which is not the case. It also does not take into account the correlations between assets, which means that the risk-adjusted return of the portfolio may not be fully optimized.

3.6.2. Equal Dollar Strategy

The Equal Dollar Strategy is a very simple approach to portfolio management in the sense that it focuses on simple diversification by evenly distributing the capital across all available assets in the universe as they become available. The weight of each asset then becomes the capital by the number of assets in the universe. Unlike other strategies that consider volatility, momentum, or other technical metrics, this strategy does not take into account the individual risk profiles of the assets but instead treats each asset the same. However, allocating equally ensures a level of diversification of the overall portfolio.

Equal Dollar Algorithm Breakdown

1. Determine the number of assets in the universe
2. Calculation of equal weights

$$\omega_i = \frac{1}{N} \tag{3.8}$$

   where:

   ω_i = weight of asset i

   N = Number of assets in the universe

3. Asset Capital Allocation

$$Ai = wi \bullet P \tag{3.9}$$

   where

   P = total portfolio value

   ω_i = weight of asset i

   Ai = Asset dollar investment

Strategy Rationale

This strategy avoids the complexities of modelling asset behaviour. It completely ignores volatility, correlation, and other metrics. For example, two assets with different risk profiles will receive the same weighting. The Equal Dollar Strategy is a benchmark for the allocation of our total capital across a range of assets if it were evenly distributed. Combining this strategy with the inclusion of the S&P 500 can be used as a benchmark for deciding the quality of other strategies.

3.6.3. Minimum Variance Portfolio Strategy

The Minimum Variance Portfolio(MVP) is the first type of optimised classic portfolio strategy. It is a portfolio optimisation approach that aims to construct a portfolio of assets with the lowest possible variance. The variance can be defined as a measure of how much the asset returns deviate from the mean portfolio return. The primary objective of this approach is to minimise variance within the portfolio, thereby offering the most stable returns and less exposure to volatility. I calculate the asset weights that minimize the portfolio's variance.

Minimum Variance Breakdown

1. Computing Covariance Matrix: Quantification of Asset Relationships Using Covariance of Asset Returns

   The construction of a covariance matrix plays a pivotal role in this strategy. The covariance matrix serves as a measure of how much the returns of two assets exhibit a relationship or correlation with each other. The higher the positive covariance, the higher the chance of the assets moving in the same direction. The covariance can be zero or negative. The former means there is no relationship, and the latter means there is an opposite relationship between the assets. If the relationship between assets is known, the correlation of its constituents can be minimised in the overall portfolio, which in turn reduces risk. The covariance matrix is calculated by:

$$\sum_{i,j} = \alpha \cdot \left(R_i \cdot R_j\right) + (1-\alpha) \cdot \sum_{i,j} \tag{3.10}$$

   where:

   α = Decay factor for the Exponential Moving Average (EMA)

   R_i = Returns for asset i

   R_j = Returns for asset j

Defining the Objective Function

The variance of a portfolio represents the total amount of risk associated with the portfolio and is calculated as the square of the standard deviation of the portfolio. The computation of portfolio variance takes into account the combined impact of covariances among all possible pairs of assets Jain and Jain (2019). The formula is given as

$$\sigma^2_{\text{objective}} = \omega^T \sum \omega \tag{3.11}$$

where
$\alpha^2_{portfolio}$ = portfolio variance
ω = Vector of asset weights

Model Constraints

The constraints set for the portfolio include

- *sum of weights should be* 1: $\sum \omega = 1$
- Weights should be within the bounds of $(0 \leq \omega \leq 0.05)$. Weights are non-negative since short selling is not allowed and should not be more than 5% of the portfolio. The upper bound is customisable

Optimization - Minimisation of Portfolio Variance

To minimise the portfolio weights, I use a minimise optimiser from the scipy.optimize library. However, due to the complex nature of the optimization process, I decided to switch to a more robust quadratic solver, Gurobi, which is very efficient for solving linear and quadratic constraints to optimise for an increase in optimisation speed.

Strategy Rationale

The goal of the MVP strategy is to reduce the overall susceptibility of the portfolio to market volatility. By minimising variance, the portfolio is more risk-averse and will handle volatile markets more efficiently. However, the strategy focuses on optimising variance and doesn't take into account expected returns, which may lead to suboptimal risk-reward tradeoffs.

3.6.4. Equal Volatility Contribution

The equal volatility contribution strategy is another type of optimisation strategy that aims to allocate assets in such a way that each asset has an equal risk contribution to the entire portfolio's volatility. This is handled through an optimisation process that balances the risk contributions to achieve better risk-adjusted returns. It is also referred to as risk parity.

Equal Volatility Model Breakdown

$$\text{VolContribution}_{\,i} = \omega_i \cdot \sum_{i=1}^{N} \omega_j \cdot \text{Cov}\left(r_i, r_j\right) \tag{3.12}$$

where
ω_i = Asset weight
$r_{i,j}$ = Asset Return
$Cov(r_i, r_j)$ = Covariance of ri and rj
N = no of Assets in portfolio

Defining the Objective Function

The function set in the model minimises the sum of squared differences between the contribution of each asset to the average overall contribution and the portfolio volatility.

$$\sum_{i=1}^{N} \left(\text{Volatility Contribution}_{\,i} - \frac{1}{N} \sum_{j=1}^{N} \text{Volatility Contribution}_{\,j} \right)^2 \tag{3.13}$$

Constraints

The only constraint set for this model is that the sum of the portfolio weights must be equal to 1. This is a step for normalisation.

sum of weights should be 1: $\sum \omega = 1$

Strategy Rationale

The Equal Volatility Contribution deters any single asset from overcontributing to the entire portfolio's risk, which is in contrast to the mean-variance portfolio, which may overcontribute. Every single asset contributes the same amount of risk, which means that the more risky assets will be adjusted lower in terms of weight and vice versa. This strategy could potentially provide better risk-adjusted returns.

3.6.5. Markowitz Strategy

The Markowitz Strategy, also known as the Modern Portfolio Theory (MPT) is another type of mathematical optimisation strategy that attempts to maximise the expected return of a given risk level or minimise risk for a given expected return. This strategy is considered a foundational method for building portfolios. It uses the concept of efficient frontier which is introduced by (H. Markowitz, 1952; H. M. Markowitz, 1999).

Markowitz Model Breakdown

1. **Defining the Risk and Returns**

   The risk is the standard deviation of the portfolio, and the return is the first derivative of an asset's price with respect to time.

   Expected return: $\mu_p = \omega^T \mu$

   Portfolio Variance: $\sigma_p^2 = \omega^T \sum \omega$

   where

   μ_p = Portfolio return

   ω = Vector of asset weights

2. **Defining the Objective Function**

   The objective function which is a combination of the expected return and the portfolio variance is given as:

   Objective $= \omega^T r - \lambda \omega^T C \omega$

   where

   $\omega^T r$ = Expected return of the portfolio

   $\omega^T C \omega$ = Portfolio Variance

   λ = The risk aversion parameter

Constraints

- Weights should be within the bounds of $\left(0 \leq \omega_i \leq \text{max\_concentration}\right)$
- Target return or risk level

Optimization

The Gurobi solver library was used to optimise the function.

The Parameterisation of the Model

The Markowitz Model was parameterized to allow for investigation into the strategy's performance with the following metrics:

1. Return estimate
2. Risk constant
3. Volatility weighted
4. Max concentration

Machine Learning Models

This section describes the machine learning-based methods that were used in this research. For each model, we explain the strategy design, the rationale behind the strategy and the mathematical explanation of the process.

3.6.6. Hierarchical Risk Parity Strategy (Clustering)

The Hierarchical Risk Parity (HRP) approach, proposed by Marcos Lopez de Prado, is a cutting-edge portfolio optimisation technique that aims to construct a portfolio by considering hierarchical relationships between the assets. This strategy was initially made for choosing between portfolios, but several papers have attempted to use this for asset portfolio management. It starts by addressing each stock as a cluster. For example, if our universe is made up of 300 stocks, then they are all addressed as 300 single clusters). We then calculate the distance matrix between each of the assets. We then build a tree called a dendrogram by continuously linking clusters with small distances. Clusters that are then closest to each other are then reclustered until we end up with a final set of clusters, which will determine the risk targets for the portfolio

The HRP strategy is expected to be explicitly variance-reducing. It diversifies risk by allocating more weight to assets that are less correlated.

HRPModel Breakdown

Calculation of the Covariance Matrix

The covariance matrix is calculated using the exponential moving average (EMA). The formula is as follows:

$$\text{Cov}_{i,j} = (1-\alpha)\times\text{Cov}_{i,j} + \alpha\times\left(r_i \times r_j\right)$$

where
α - decay factor of the EMA
$r_{i,j}$ - returns of asset I and j

Calculation of the Distance Matrix

To calculate the distance matrix the covariance matrix is used. Its purpose is to give accurate distances between assets and their correlations and volatilities.

$$\text{dist}_{i,j} = \sqrt{\frac{1-\text{Corr}_{i,j}}{2}} \tag{3.14}$$

Hierarchical Clustering

Hierarchical clustering is then applied to the distance matrix to create clusters. Within those clusters are assets with similar correlations. The clusters are initialised by assigning each stock as its own singular cluster and then reclustering for assets with smaller distances. In this case, we test out several linkage methods that are the ward, single, and average. The linkage methods were parameterised to test out the efficiency of the model using different linkage methods. The frequency of retraining was also parameterised. It was set as 252 days, which is equivalent to the average number of trading days in a year.

Weight Allocation (Computation of Weights)

Once the clusters are recursively divided into subclusters and we have a final list of clusters, the weights are calculated as the inverse of the variance of each cluster. The calculation of the allocation factor is as follows:

$$\text{Allocation factor } = 1 - \frac{\text{left cluster variance}}{\text{left cluster variance } + \text{ right cluster variance}} \tag{3.15}$$

$$\text{weights } = \text{ left\_alloc\_factor } * (1 - \text{ right\_alloc\_factor }) \tag{3.16}$$

Strategy Rationale

The HRP strategy is grounded under the belief that there are hierarchical structures in asset returns. In that case, the HRP method takes into account the relationships between assets and their respective volatilities. The strategy offers diversification in the sense that the strategy is well-diversified across different identified clusters. It is also worth noting that there is more diversification between the clusters than within the clusters. The HRP strategy also offers stability in the sense that it does not rely on the estimation of expected returns like the other methods, which may lead to estimation errors. Speaking about stability and reliability, Covariance is noisier than correlation matrices as they are more sensitive and can vary in scale depending on the data. If we are to base the portfolio weights on covariance matrices, we will end up with noisy estimates.

3.6.7. XGBoost-Based Strategy

The XGBoost Model, a popular gradient-boosting framework, was used as a strategy. The XG Boost model is well known for its versatility in use cases. XGBoost, which stands for extreme gradient boosting, is a tree-based model that works by iteratively adding multiple weak learners (decision trees) to a model in which the new trees are made to correct the prediction errors of the previous tree (T. Chen & Guestrin, 2016). Two key features of the XGBoost model are its scalability and its ability to handle overfitting through the use of regularisation in its loss function. In this case, we use the model to predict assets with the highest likelihood of increasing in value in price. This strategy focuses on predicting short-term price movements and using the scale of the prediction as compared to the actual value to assign the weights of the portfolio. This strategy dynamically assigns weights to portfolio assets in a value-weighted manner.

This strategy was achieved by first inferring the future return of the asset using the model and taking the normalised predictions as the weights for the portfolio. Negative return predictions were then assigned a 0 weighting in the portfolio.

XGBoost Model Breakdown

Feature Engineering

Given that the machine learning-based strategies require training data to compute predictions and the predictions are set to a constant window, the features of each of the assets were constructed using a sliding window of the previous price returns. The sliding window of features is represented as

$$X_i = \left(\frac{P_{i,1}}{P_{i,0}} - 1, \frac{P_{i,2}}{P_{i,1}} - 1, \dots \frac{P_{i,N}}{P_{i,N}} - 1 \right) \tag{3.17}$$

where

P = the price of a single asset i at time j

N = input window size

In this case, I specified the sliding window as 15 trading days and the prediction horizon as 5 trading days.

Label Construction

In terms of the regression-based XGBoost model, the label for each asset is then the future return over the given horizon of the asset.

$$Y_i = \frac{P_{i,\text{ future}}}{P_{i,\text{ present}}} - 1 \tag{3.18}$$

In terms of the classification-based XGBoost model, the label for each asset is a binary value indicating if the asset returns are positively or negatively predicted

$$Y_{\text{label}} = \begin{cases} 1 & \text{if } \frac{P_{i,\text{future}}}{P_{i,\text{present}}} - 1 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

Model Training and Portfolio Weight Allocation

As XGBoost can be used for both regression and classification tasks, two types of strategies were attempted with this model. The regression works by predicting continuous feature returns, while the classification works by predicting whether the return will be negative or positive. For the regression-based model, the weights are allocated by making them directly proportional to the predicted returns. For the binary classification, the top 10% of assets with the highest predicted probability of positive returns are equally weighted.

Strategy Rationale

XGBoost is a good strategy for use on stock price data as it adapts to different market conditions and can capture non-linear relationships between past and future asset behaviour that may not be able to be captured by regular linear models. The XGBoost model is also robust against overfitting through the use of regularisation techniques (**chen_xgboost_2016**). In the analysis section, I'll be benchmarking the XGBoost portfolios against the equal dollar portfolio due to exposure to the same market factors.

3.6.8. Convolutional Neural Network(CNN) Approach

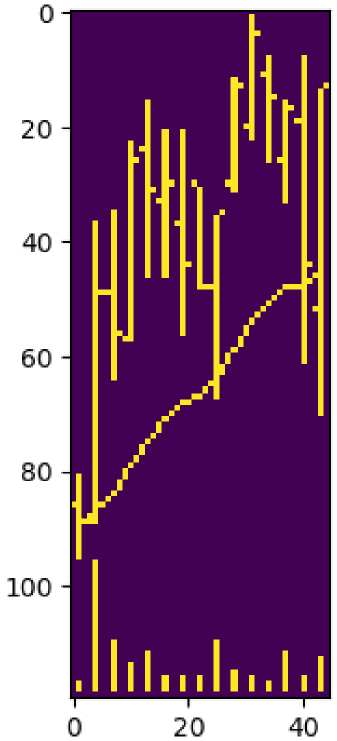

Figure 3.6.: Input image to the CNN, mimicking a price chart.

Data Transformations and Calculations

Bresenham line drawing: We use the Bresenham line drawing algorithm, extracted from Wikipedia to construct the pixel lines for the image as done in the research paper by (Jiang et al., 2020).

Data Split: The data was split 70:30 for a train and validation script respectively.

Early Stopping: The paper mentioned the implementation of early stopping as it was found that the models can overfit easily. I implemented early stopping to prevent overfitting and allow for better generalisation. In the case where model loss does not improve, we break, save the model, and use it as the best model for prediction as a form of regularisation technique.

Data Normalisation: The input features were scaled by dividing them by the opening price, which allows the model to focus on the relative changes. Price series are non-stationary, so trying to predict exact prices is not very fruitful. Returns, however, are better because they are a derivative of prices.

Calculation of EMA: The EMA was calculated as
$EMA_t = (1 - ema\_const)\ EMA_{t-1} + ema\_const\ Price_t$ where
ema_const = The smoothing constant which was set as
EMA_{t-1} = EMA at previous timestep
EMA_t = EMA at current timestep
$Price_t$ = price at current timestep

Model Caching: Given that neural networks are non-deterministic at every instantiation and that training from the initial date every day is memory intensive, I save the model state for each year and call the previous model states for previous years and only retrain for the current year.

```
Restored model from models/StockCNN_aa1d13938e5a1d9e5fe4dec7f4afe339_2002-06-19.pt.
2003-01-02
CAGR: 5.02%
Sharpe Ratio: 0.45
Volatility: 12.91%
Restored model from models/StockCNN_aa1d13938e5a1d9e5fe4dec7f4afe339_2003-06-19.pt.
2004-01-02
CAGR: 7.98%
Sharpe Ratio: 0.66
Volatility: 12.93%
Restored model from models/StockCNN_aa1d13938e5a1d9e5fe4dec7f4afe339_2004-06-21.pt.
2005-01-03
CAGR: 8.89%
Sharpe Ratio: 0.75
Volatility: 12.39%
```

Figure 3.7.: Example of Model Caching during training

CNN Architecture

The replication of the study was done by picking the neural network architecture with the highest performance. The three-block architecture consisted of two convolutional blocks with 1 convolutional and max-pooling layer in each block, followed by a fully connected output layer. I then utilise LeakyRelu activation, as the paper mentions. The output block consists of a flatten layer, a dropout layer for regularization, and a linear layer.

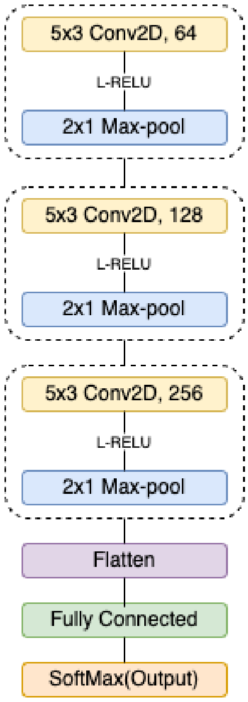

Figure 3.8.: Architecture of Convolutional Neural Network

CNN Model Training Process

I have a function called train_model, which handles the training process. It includes the splitting of data into training and validation sets. I also defined the optimiser as Adam, as done in the paper, and used cross-entropy loss.

Portfolio Strategy Implementation

I implemented several trading strategies using the CNN network. Once the model trains after each year, I used the prediction of the prices to determine the weights of each strategy I implemented. I have an infer method that is called by all the strategies to get the price prediction confidence information for each asset, and that information is then used by the strategy to define the weights. The strategies implemented are

1. **Equal Weight Percentage Strategy**

   This strategy involves investing equally across the top 10% of stocks predicted to go up. It starts by finding all the assets with positively predicted confidence and sorting them by prediction strength from the highest to the lowest. The top 10% of assets are taken to invest in. The capital is then equally allocated across the top 10%. This is the closest strategy to the decile portfolios used in the paper by Jiang et al. (2020).

2. **Equal Weight Positive Strategy**

   This strategy is similar to the Equal weight Percentage Strategy. However, in this case, once we find all the assets with positive predicted confidence, we assign equal weights across all the positive predicted assets. This is also similar to the equal dollar strategy. However, we are constraining our universe to be tighter. Only assets which are inferred by the CNN model to go up in price are invested in.

3. **Sigmoid Strategy**

   For the Sigmoid Strategy, we are using the Sigmoid function to scale the asset weight based on the prediction's deviation from the mean. We first find the stocks for which the predicted probability of a positive return is greater than the predicted probability of a negative return. Then we calculate the mean and standard deviation and assign to each stock and weight it proportional to the sigmoid of (yhat[1] - mean) / σ.

4. **Market Indicator Strategy**

   For the market indicator strategy, we determine the overall exposure ratio based on the number of overall positive predictions. In that sense, we scale your total market exposure by the percentage of stocks for which the model is expressing y_hat[1] > y_hat[0]. This approach will assign higher weights to assets with a higher predicted probability of a positive return while also considering the relative comparisons of these predicted probabilities with those of other assets.

Model Evaluation

In order to ensure an accurate evaluation of the models, we have evaluated all models across different performance metrics and also evaluating different groups of models accordingly. We have machine learning models, which have metrics such as loss and accuracy. However, the traditional rule-based strategies lack those metrics. In that case, we need to use similar metrics that allow us to compare the efficiency of both. The models were critiqued using metrics that are available to both classes of models. However, they are also critiqued within their classes to understand relative performance.

Performance Metrics

The performance metrics have been grouped into general performance metrics and those specific to the model class.In terms of the machine learning models, They are judged on an additional set of metrics used to analyse the performance of models in that class.

General Performance Metrics

The compound annual growth rate is the mean annual rate of

1. **CAGR (Compound Annual Growth Rate):**

$$\text{CAGR } = \left(\frac{F}{P}\right)^{\frac{1}{N}} \tag{3.20}$$

   where

   F = Final portfolio value

   P = Initial portfolio value

   N = Number of years

2. **Sharpe Ratio**

$$\text{Sharpe ratio } = \frac{\bar{r} - r_f}{\sigma} \tag{3.21}$$

   where

   $\bar{r}$ = Average portfolio return

   r_f= Risk-free rate (assuming 0

   σ = Portfolio standard deviation (volatility)

3. **Annualized Portfolio Volatility**

We also analyse the portfolio's yearly volatility, as this is a good measure of the risk of the portfolio in the year. This information is crucial as higher volatility means the returns can be spread out over a large range of values.

$$\text{average return } = \frac{\sum_{i=1}^{n} \text{ dailyreturns }_i}{n} \times 252 \tag{3.22}$$

where

N = number of daily returns

252 = Average number of trading days in a year

$$\text{volatility } = \sqrt{\frac{\sum_{i=1}^{n} \left(\text{ daily returns }_i - \text{ mean }^{i}_{\text{(daily returns}} \right)^2}{n}} \tag{3.23}$$

Machine Learning Performance Metrics

4. **Coefficient of determination $-R^2$ (R-squared)**

   The R^2 is a measure of the proportion that indicates the fit of the model to the data. The R^2 serves as a goodness of fit measure between the model and the dependent variable (Frost, 2023). This was used in the regression-based models to identify prediction quality. The formula is given as:

$$s_{\text{res}} = \sum (\text{actuals} - \text{predicteds})^2 \tag{3.24}$$

$$ss_{\text{targets}} = \sum \text{actuals}^2 \tag{3.25}$$

$$R^2 = 1 - \frac{s_{\text{res}}}{ss_{\text{targets}}} \tag{3.26}$$

   where

   s_{res} is the sum of squared residuals, and

   ss_{targets} is the sum of squared targets.

5. **Classification Accuracy**

   This is used for the classification models to find out how accurate they are from the ground truth labels of which In this case is if the asset was predicted to go up or down. This was used in the classification-based models to identify prediction quality. The formula for accuracy is given as:

$$accuracy = \frac{totalcorrectpredictions}{totalpredictions} \tag{3.27}$$

Model Results

I will start by investigating the portfolio performance of each class of portfolio and comparing it to the baseline within its class. For the rule-based strategies, this will be the equal dollar strategy.

Classical Rule-based Portfolios

Table 3.1 shows the results of the classical rule-based portfolios. The results show that over the 25-year period, the Equal Dollar Portfolio achieved an average CAGR of 14.24% and an average volatility of 19.18% yearly. The portfolio averagely has better Sharpe ratios than the SPY benchmark. I see large drawdowns during the market downturns in 2003 and 2009, which represented the dot com and housing market crashes, respectively. This portfolio fully replicates our universe, which is the S&P 500, but with different weightings. In the S&P 500, stocks are weighted based on market cap rather than equally. When the Equal Dollar Portfolio is compared with the S&P 500 index, we see that the equal dollar outperforms in terms of returns while also attaining the same level of diversification as the index.

The equal volatility backtest has a 13.804% average return and 17.50% average volatility over the 25-year period. The results show the equal volatility backtest to be less volatile than the equal dollar portfolio. The equal volatility portfolio comes with a slightly lower average return than the two other portfolios, which means that its risk reduction may have affected the possibility of returns. The equal volatility strategy attempts to equalise the volatility in the portfolio, thereby ensuring assets with higher risk do not affect the portfolio. The results show the portfolio to averagely be a less volatile portfolio than the equal dollar.

The equal volatility contribution portfolio performs slightly better than the equal dollar and the equal volatility in terms of CAGR and volatility, achieving 13.73% and 17.5% average, respectively.

In terms of the Sharpe Ratio, the equal volatility strategy shows a higher average Sharpe than the two other strategies with a Sharpe of 0.81, suggesting that it offers better risk-adjusted returns. Both the equal volatility and equal volatility contribution portfolios exhibit lower average risk than the equal dollar strategy, which shows that the objective of distributing risk across assets has an overall effect on the portfolio. The equal volatility however underperforms and this may be due to lower beta and beta generally has positive returns. The equal volatility strategy by nature is a more concentrated portfolio compared to the equal dollar which is least concentrated. Figure 3.9 shows the distribution of asset weights between the Equal Dollar and the Equal volatility. The weights for the Equal volatility are more sparsely distributed across the assets.

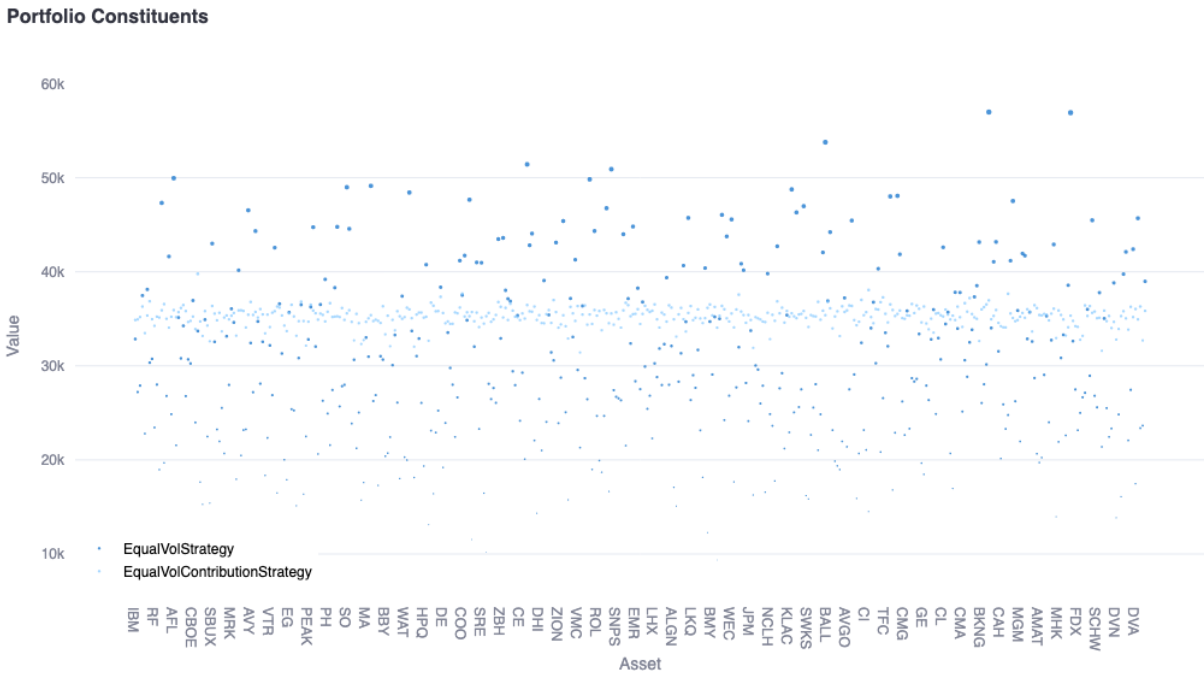

Figure 3.9.: Portfolio Constituents weight distribution for Equal Dollar and Equal Volatility

In terms of the equity curves, we see the three portfolios recording their lowest returns in 2003 and 2009 which indicate the market-wide downturns from the dot com and housing market crashes. The performance across all models shows a consistent upward trend from 2018 to 2023.

Plot 3.10 also shows the comparison of the models to the S&P 500 index. We see a statistical outperformance of all models compared to the index in terms of portfolio value. The equal volatility strategy underperforms as compared to the equal dollar and the equal volatility contribution. The equal volatility contribution strategy appears to offer the best blend between risk and return in that it provides slightly higher returns than the Equal Dollar strategy while maintaining similar average risk profiles. Jain and Jain (2019) found that the Equal Risk Contribution portfolio, which in my case is the Equal Volatility Contribution portfolio performed superior in most cases when the objective was to maximize the out-of-sample diversification ratio and minimize the Herfindahl index. However, the differences in the average performance of these strategies over time are minor, which may suggest that the general market conditions are a large factor in the portfolios. This may be the case because the universe used to create the portfolios is the same as the underlying index used to create it(S&P 500 Index). We typically expect these strategies to perform well in earlier years of the backtests as most trades done during those times were done manually with no technical alpha.

Figure 3.10.: Equity Curves of Classic Portfolios and the SPY index

Table 3.1.: Yearly Backtest Results of Classic Rule-based Portfolios

	CAGR			**Sharpe Ratio**			**Volatility**		
Year	Equal Dollar	Equal Volatlity	Equal Volatility Contr	Equal Dollar	Equal Volatility	Equal Volatility Contr	Equal Dollar	Equal Volatlity	Equal Volatility Contr
1998	16.91%	26.10%	17.22%	1	1.47	1.01	17.07%	16.70%	16.70%
1999	20.13%	22.07%	20.30%	1.13	1.29	1.13	17.70%	16.61%	16.61%
2000	16.04%	14.54%	16.14%	0.98	0.94	0.98	16.67%	15.71%	15.71%
2001	17.67%	17.33%	17.73%	1.02	1.05	1.03	17.39%	16.43%	16.43%
2002	14.89%	14.37%	14.94%	0.88	0.91	0.89	17.52%	16.28%	16.28%
2003	10.65%	10.23%	10.68%	0.63	0.64	0.63	18.99%	17.56%	17.56%
2004	13.60%	12.63%	13.62%	0.78	0.78	0.78	18.55%	17.16%	17.16%
2005	14.48%	13.44%	14.50%	0.85	0.85	0.85	17.75%	16.42%	16.42%
2006	14.44%	13.27%	14.46%	0.88	0.87	0.88	17.08%	15.82%	15.82%
2007	14.70%	13.69%	14.71%	0.92	0.92	0.92	16.51%	15.28%	15.28%
2008	13.58%	12.76%	13.59%	0.86	0.86	0.86	16.49%	15.29%	15.29%
2009	9.07%	8.52%	9.08%	0.53	0.53	0.53	20.42%	18.60%	18.60%
2010	11.29%	10.16%	11.30%	0.6	0.6	0.6	21.69%	19.33%	19.33%
2011	12.06%	10.86%	12.07%	0.64	0.63	0.64	21.52%	19.19%	19.19%
2012	11.45%	10.50%	11.46%	0.61	0.61	0.61	21.77%	19.44%	19.44%
2013	12.10%	11.09%	12.12%	0.64	0.65	0.64	21.34%	19.05%	19.05%
2014	13.26%	12.19%	13.27%	0.7	0.71	0.7	20.89%	18.67%	18.67%
2015	13.55%	12.54%	13.56%	0.72	0.74	0.72	20.47%	18.32%	18.32%
2016	12.78%	11.88%	12.78%	0.7	0.71	0.7	20.23%	18.16%	18.16%
2017	13.11%	12.15%	13.12%	0.72	0.73	0.72	19.97%	17.93%	17.93%

2018	13.57%	12.62%	13.57%	0.75	0.77	0.75	19.54%	17.55%	17.55%
2019	12.56%	11.70%	12.57%	0.71	0.72	0.71	19.37%	17.43%	17.43%
2020	13.37%	12.49%	13.38%	0.75	0.77	0.75	19.11%	17.21%	17.21%
2021	13.42%	12.51%	13.43%	0.72	0.73	0.72	20.29%	18.39%	18.39%
2022	14.16%	13.21%	14.17%	0.76	0.77	0.76	20.06%	18.18%	18.18%
2023	13.09%	12.25%	13.10%	0.71	0.72	0.71	20.19%	18.31%	18.31%
Average	13.69	13.27	13.73	0.78	0.81	0.78	19.18	17.5	17.5

1. **Goal-Optimised Portfolios (Markowitz and Minimum Variance)**

   The goal-optimised portfolios consist of the Markowitz and Minimum Variance Portfolios, in which there are constraints that have to be satisfied. Appendix A.4 shows the full results of all the variants of the Markowitz portfolio. It was found that the best performing Markowitz model was that with a risk constant of 1, max concentration of 1, and non-volatility weighted. This is then further compared with the Minimum Variance Portfolio in Table 3.2 below. The results show the better performance of the minimum variance strategy in both the CAGR and risk-adjusted returns (Sharpe Ratio). It is slightly more volatile than the Markowitz portfolio, but the difference is negligible. It is considered a better strategy, as it has a noticeably better Sharpe and CAGR. Minimum Variance is more concentrated than Markowitz, but in this case, performed better. Generally, the Markowitz model is good at picking out diversifiers. It will crash during crashes as seen in both significant crashes in the market, but in terms of normal market behaviour, where correlations tend to be decently stable, it will perform well. An advantage of the Minimum variance strategy is that it exploits the diagonals in the covariance matrix.

Figure 3.11.: Equity Curve of Goal-optimised Portfolios

Table 3.2.: Yearly Backtest Results of Goal-Optimised Portfolios

Year	CAGR		Sharpe Ratio		Volatility	
	Minimum Variance	Best Markowitz	Minimum Variance	Best Markowitz	Minimum Variance	Best Markowitz
1998	33.10%	39.96%	2.06	2.44	14.33%	14.16%
1999	24.00%	23.96%	1.53	1.64	14.84%	13.67%
2000	10.53%	10.57%	0.77	0.8	14.36%	13.66%
2001	10.30%	10.26%	0.74	0.75	14.78%	14.30%
2002	11.65%	12.15%	0.85	0.9	14.29%	13.90%
2003	13.99%	9.62%	0.9	0.69	16.06%	14.88%
2004	16.05%	13.03%	1.05	0.93	15.29%	14.32%
2005	16.41%	14.01%	1.12	1.03	14.56%	13.66%
2006	15.35%	13.31%	1.09	1.01	14.02%	13.21%
2007	15.88%	13.97%	1.16	1.09	13.51%	12.77%
2008	14.56%	12.58%	1.09	1	13.32%	12.69%
2009	8.69%	7.94%	0.61	0.58	15.61%	15.13%
2010	8.77%	8.35%	0.61	0.6	15.90%	15.40%
2011	9.38%	8.76%	0.65	0.63	15.61%	15.14%
2012	10.00%	9.38%	0.69	0.67	15.49%	15.06%
2013	10.26%	9.65%	0.72	0.7	15.14%	14.73%
2014	11.14%	10.46%	0.79	0.76	14.85%	14.45%
2015	11.31%	10.77%	0.81	0.79	14.58%	14.19%
2016	10.78%	10.05%	0.78	0.75	14.52%	14.15%
2017	10.93%	10.33%	0.79	0.77	14.40%	14.01%
2018	11.23%	10.67%	0.83	0.81	14.11%	13.73%
2019	10.46%	9.89%	0.78	0.76	14.03%	13.66%
2020	11.06%	10.51%	0.83	0.81	13.91%	13.56%
2021	11.58%	10.80%	0.81	0.77	15.04%	14.69%
2022	11.89%	11.25%	0.83	0.81	14.89%	14.55%
2023	11.09%	10.45%	0.78	0.75	14.93%	14.61%
Average	13.09	12.41	0.91	0.89	14.71	14.16

2. **Machine Learning Portfolios**

   In this section, we analyse and compare the results of the machine learning-based portfolios. The portfolios, which consist of the stock image trained Convolutional Neural Network, the Hierarchical Risk Parity Portfolio (HRP), and the XGBoost Model all showed interesting results. For this class of portfolios, we have the general metrics, which we use to analyse the performance, but we also have supplementary metrics in terms of the learning metrics which are specific to machine learning models, which we computed to provide further understanding into the model's performance.

   While working on the machine learning models, we tested out different hyperparameters of the models to find which of them performed best. Appendix A.5 to A.10 shows the full results of all the different configurations of the tested models results from the machine learning models. Table 3.3 contains the best models from each category and compares them.

From the results, it can be seen that the XGBoost Model shows a higher average CAGR than the HRP and CNN models over the years. Despite HRP having the lowest average CAGR, the strategy has the highest Sharpe Ratio. This can be attributed to its main objective of diversifying risk based on the hierarchical relationships between assets. HRP has a better Sharpe because it is not as concentrated as CNN and XGBoost.

The CNN model also performs relatively well. From the CNN model's accuracy, we see that the model has a higher accuracy than the Naive benchmark which shows that the strategy is able to generate signal. The issue with the XGBoost strategy is that it has significant drawdowns during market crisis times. In general, all models show out performance to the index which is similar results to what the individual papers achieved.

Figure 3.12.: Equity curves of Best Performing Machine Learning Strategies

Table 3.3.: Yearly Backtest Results of Machine Learning Portfolios

	CAGR			**Sharpe Ratio**			**Volatility**		
Year	XG Boost (Classification)	CNN (equal percent)	HRP (Average)	XG Boost (Classification)	CNN (equal percent)	HRP (Average)	XG Boost (Classification)	CNN (equal percent)	HRP (Average)
1998	NaN	NaN	28.58%	NaN	NaN	1.7	NaN	NaN	15.47%
1999	9.66%	1.93%	20.79%	0.66	0.21	1.34	15.90%	13.16%	14.99%
2000	16.39%	10.83%	12.27%	0.96	0.76	0.88	17.30%	14.99%	14.32%
2001	17.63%	19.22%	16.19%	0.86	1.1	1.07	21.47%	17.42%	15.02%
2002	19.45%	16.58%	13.64%	0.87	0.93	0.94	23.63%	18.27%	14.75%
2003	15.14%	12.20%	9.99%	0.67	0.67	0.68	25.89%	20.15%	15.79%
2004	18.27%	14.92%	12.33%	0.8	0.81	0.83	24.98%	19.60%	15.39%
2005	20.24%	17.74%	13.10%	0.9	0.97	0.91	23.71%	18.78%	14.75%
2006	20.44%	17.48%	12.66%	0.94	0.98	0.91	22.64%	18.12%	14.25%
2007	20.71%	17.34%	13.15%	0.98	1	0.97	21.77%	17.55%	13.76%
2008	19.22%	15.25%	12.35%	0.93	0.9	0.92	21.37%	17.53%	13.78%
2009	13.76%	10.11%	8.14%	0.63	0.56	0.55	25.82%	21.34%	16.62%
2010	18.44%	12.68%	9.29%	0.74	0.64	0.61	28.35%	22.61%	17.02%
2011	18.58%	13.87%	9.86%	0.75	0.69	0.64	27.85%	22.40%	16.88%
2012	17.67%	13.38%	9.82%	0.72	0.67	0.63	27.81%	22.67%	17.07%
2013	18.01%	13.90%	10.39%	0.75	0.7	0.68	27.15%	22.22%	16.72%
2014	18.78%	15.06%	11.51%	0.78	0.75	0.75	26.47%	21.74%	16.41%
2015	18.63%	15.42%	11.89%	0.79	0.78	0.78	25.87%	21.30%	16.12%
2016	17.23%	14.59%	11.33%	0.75	0.75	0.75	25.47%	21.09%	16.05%
2017	18.17%	15.46%	11.46%	0.79	0.8	0.76	25.07%	20.84%	15.87%
2018	18.57%	16.09%	11.85%	0.82	0.83	0.8	24.52%	20.41%	15.53%
2019	17.29%	15.13%	11.03%	0.78	0.8	0.76	24.23%	20.27%	15.46%
2020	17.99%	16.30%	11.76%	0.81	0.85	0.81	23.87%	20.05%	15.26%
2021	19.06%	16.32%	11.87%	0.82	0.82	0.77	25.09%	21.14%	16.41%
2022	19.74%	17.07%	12.49%	0.85	0.86	0.81	24.80%	20.95%	16.23%
2023	18.46%	15.96%	11.68%	0.8	0.81	0.76	24.96%	21.15%	16.33%
Average	17.9	14.59	12.67	0.81	0.7856	0.85	24.24	19.83	15.63

Table 3.4.: Classification Accuracy ML Portfolios as compared to the Naive Baseline

Classification Models' Accuracy				
	Naive Baseline Benchmark	Model Accuracy	x1 coefficient (compared to S&P 500 index)	coefficient of intercept
CNN Model	0.3198533356	0.356775974	-0.0774	0.0006
XGBoost Model(Classification)	0.4560600673	0.4684257799	-0.0836	0.0006

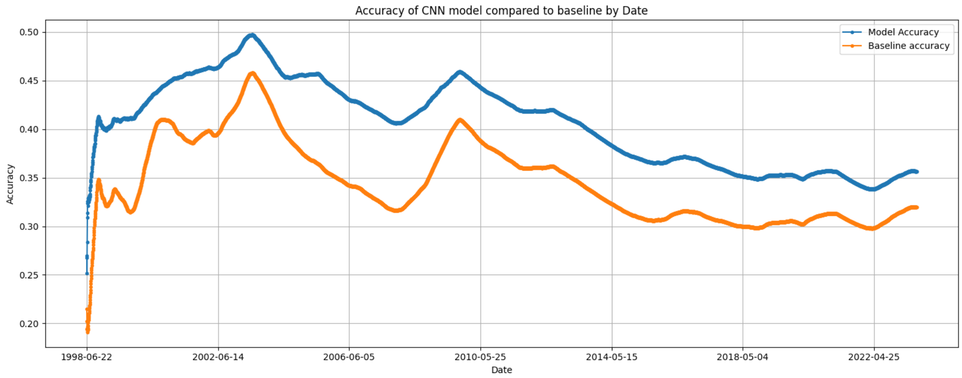

Figure 3.13.: Diagram comparing the accuracy of the CNN model over time to the accuracy of always predicting one class as a baseline

Portfolio Performance with priority to lowest predicted performing assets

In this section, I stress test the techniques in which we test out the efficiency of the models if I were to take the opposite direction to what was predicted by the model. In this case, we are taking the assets that were predicted to have negative returns and investing in them to see if the performance still remains. This will also identify which of the signals are valuable and how much the machine learning models contribute to generating a good portfolio. Figure 3.14 illustrates the distinction between an XGBoost strategy that selects assets with negative predicted returns and the conventional positive return model. We clearly note that with negative weights, the model continuously loses money which proves that the signal in the inference is significant.

Figure 3.14.: Diagram showing Equity Curve of XGBoost model based on Negative vs Positive return signal

Portfolio Performance of Machine Learning vs. Classical Models

In this section, we address the paper's primary objective, which is to compare machine learning models to traditional models for portfolio generation. From a model-building perspective, it is essential to determine if machine learning portfolios generate an advantage (alpha) over traditional portfolios. I would like to determine whether the machine learning portfolios outperform the traditional portfolios. I begin by selecting and comparing the top models from both the machine learning and classical model categories using all available metrics.

Table 3.5.: Comparison of Best Performing Portfolios from the Machine Learning and Classical Methods

	CAGR			**Sharpe Ratio**			**Volatility**		
	Classic Rule Based	Classic Optimised	Machine Learning	Classic Rule Based	Classic Optimised	Machine Learning	Classic Rule Based	Classic Optimised	Machine Learning
Year	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)
1998	17.22%	33.10%	NaN	1.01	2.06	NaN	16.70%	14.33%	NaN
1999	20.30%	24.00%	9.66%	1.13	1.53	0.66	16.61%	14.84%	15.90%
2000	16.14%	10.53%	16.39%	0.98	0.77	0.96	15.71%	14.36%	17.30%
2001	17.73%	10.30%	17.63%	1.03	0.74	0.86	16.43%	14.78%	21.47%
2002	14.94%	11.65%	19.45%	0.89	0.85	0.87	16.28%	14.29%	23.63%
2003	10.68%	13.99%	15.14%	0.63	0.9	0.67	17.56%	16.06%	25.89%
2004	13.62%	16.05%	18.27%	0.78	1.05	0.8	17.16%	15.29%	24.98%
2005	14.50%	16.41%	20.24%	0.85	1.12	0.9	16.42%	14.56%	23.71%
2006	14.46%	15.35%	20.44%	0.88	1.09	0.94	15.82%	14.02%	22.64%
2007	14.71%	15.88%	20.71%	0.92	1.16	0.98	15.28%	13.51%	21.77%
2008	13.59%	14.56%	19.22%	0.86	1.09	0.93	15.29%	13.32%	21.37%
2009	9.08%	8.69%	13.76%	0.53	0.61	0.63	18.60%	15.61%	25.82%
2010	11.30%	8.77%	18.44%	0.6	0.61	0.74	19.33%	15.90%	28.35%
2011	12.07%	9.38%	18.58%	0.64	0.65	0.75	19.19%	15.61%	27.85%
2012	11.46%	10.00%	17.67%	0.61	0.69	0.72	19.44%	15.49%	27.81%
2013	12.12%	10.26%	18.01%	0.64	0.72	0.75	19.05%	15.14%	27.15%
2014	13.27%	11.14%	18.78%	0.7	0.79	0.78	18.67%	14.85%	26.47%
2015	13.56%	11.31%	18.63%	0.72	0.81	0.79	18.32%	14.58%	25.87%
2016	12.78%	10.78%	17.23%	0.7	0.78	0.75	18.16%	14.52%	25.47%
2017	13.12%	10.93%	18.17%	0.72	0.79	0.79	17.93%	14.40%	25.07%
2018	13.57%	11.23%	18.57%	0.75	0.83	0.82	17.55%	14.11%	24.52%
2019	12.57%	10.46%	17.29%	0.71	0.78	0.78	17.43%	14.03%	24.23%
2020	13.38%	11.06%	17.99%	0.75	0.83	0.81	17.21%	13.91%	23.87%
2021	13.43%	11.58%	19.06%	0.72	0.81	0.82	18.39%	15.04%	25.09%
2022	14.17%	11.89%	19.74%	0.76	0.83	0.85	18.18%	14.89%	24.80%
2023	13.10%	11.09%	18.46%	0.71	0.78	0.8	18.31%	14.93%	24.96%
Average	13.73	13.09	17.9	0.78	0.91	0.81	17.5	14.71	24.24

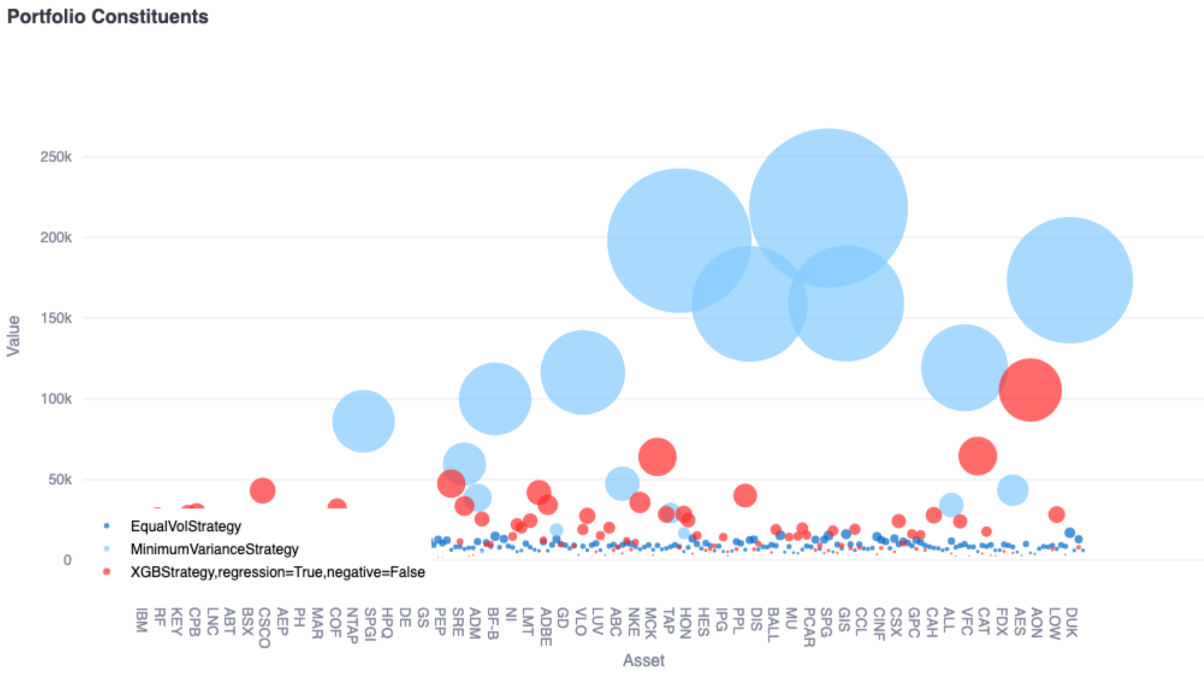

Figure 3.15.: Diagram showing the portfolio constituent and weights of the there classes of strategies on 4th of June,2003

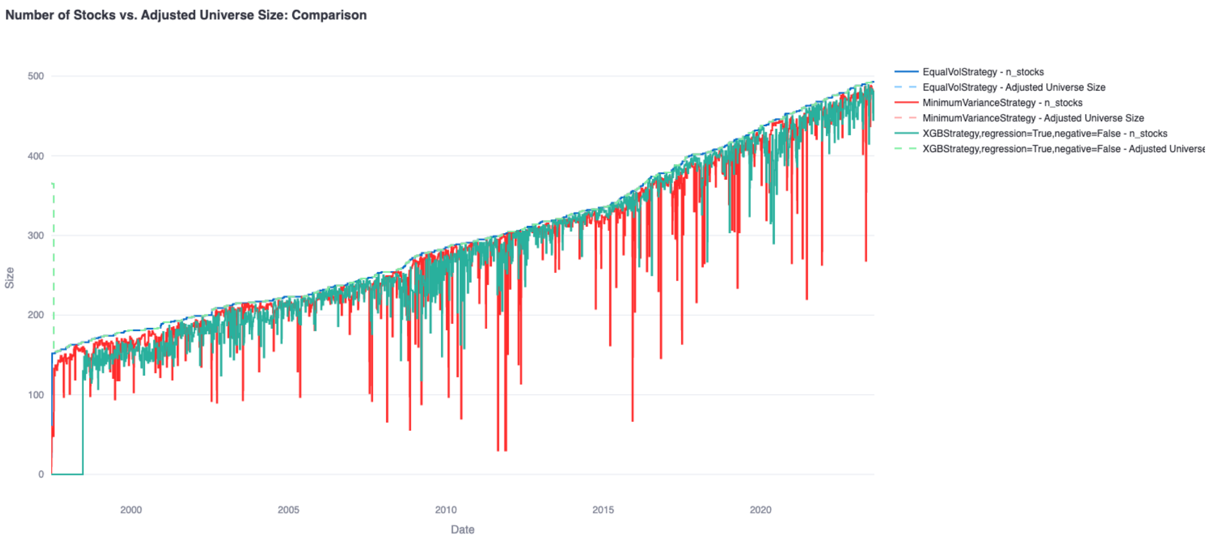

Figure 3.16.: Comparison of Number of Assets invested in over time as compared to the assets in the universe

From the results, we see that the XGBoost model outperforms all other classical models in terms of CAGR. However, due to the XGBoost strategy being a concentrated strategy, it does become more volatile as compared to the traditional approaches. The Minimum Variance strategy concentrates on a few assets in the portfolio with the XGBoost behaving similarly, but slightly less concentrated.
Figure 3.16 shows the comparison of the number of assets invested by the portfolios as compared to the universe size. Equal volatility is a diversified strategy because all assets are invested in the same

proportion. However, the number of assets invested in Minimum Variance and XGBoost models differs. Clearly, during certain time periods, the minimal variance strategy has significantly fewer assets than the other two strategies. The XGBoost is less volatile in comparison to the universe size and the Minimum Variance.

Future Work & Robustness

The objective of this study was to present and investigate the potential of and to develop a proof of concept for constructing intelligent investment portfolios aided by machine learning. The potential for further research is vast. Further studies could be conducted in areas such as proposing a new approach involving including fundamental analysis data, such as sentiment analysis of news sources, and integrating advanced language models to analyze firm income statements. In this case, the income statements will be converted into a suitable format that can be utilized as training data for the models. Another consideration would be to build hybrid techniques that integrate both machine learning and classical portfolio theory approaches for portfolio management. These may offer stronger signals and better metrics.

In this paper, only equities and cash were evaluated as asset classes in the universe. The addition of additional asset classes to create a more diversified portfolio will also be a consideration. This could be accomplished by obtaining real-time information on other asset classes, such as derivatives (futures and options). We could find a method to combine them with a short-term US Treasury yield, as hedge funds are less likely to leave capital in the bank in real-world scenarios.

As the end goal is to also build an end-to-end Robo-Advsior, it will be beneficial to build a customer-facing application that uses machine learning and investors' responses to help users select what investment portfolio works best for them.

Conclusion

The intersection of portfolio management with machine learning is growing as a significant field of study. As markets become more complex in their offering and advancements in computational finance continue, research into this domain is of importance. In this paper, I attempt to perform a comparative analysis of Classical Portfolio theory approaches and Machine learning techniques in the context of Building Portfolios. I investigate the possibility of inference-led portfolio management and the possible combination of Machine learning and Classical techniques. I develop a pipeline for automated daily multi-strategy backtesting and a frontend application for viewing daily results.

The results of the models demonstrate that machine learning strategies generate statistical alpha(advantage) relative to the S&P 500 index and other classical portfolio strategies. The best-performing machine learning model is the classification-based XGBoost Model, which, despite exhibiting greater volatility, generated greater returns over the examined time frame.

I also replicate and validate the CNN approach which is an out of the box approach proposed by Jiang et al. (2020). It performed similar but slightly below the XGBoost model in which the equal percentage CNN strategy was able to provide a stable return while providing similar risk to the Equal volatility and Equal Volatility Contribution Models.

Machine learning models prove to be a viable way of finding patterns and generating statistical alpha for long term portfolio management. In a real world scenario, combining the alphas from multiple approaches is very likely.

Bibliography

Agnew, J., & Mitchell, O. S. (2019). *The disruptive impact of fintech on retirement systems*. Oxford University Press.

Aithal, P. K., Geetha, M., U, D., Savitha, B., & Menon, P. (2023). Real-time portfolio management system utilizing machine learning techniques. *IEEE Access*, *11*, 32595–32608. https://doi.org/10.1109/access.2023.3263260

Amenc, N., & Sourd, V. L. (2003a). *Portfolio theory and performance analysis*. Wiley.

Amenc, N., & Sourd, V. L. (2003b). *Portfolio theory and performance analysis*. Wiley.

Benhamou, E., Saltiel, D., Ungari, S., & Mukhopadhyay, A. (2020). Bridging the gap between markowitz planning and deep reinforcement learning. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3702112

Chen, T., & Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, 106943. https://doi.org/10.1016/j.asoc.2020.106943

Fernando, J. (2023). Volume-weighted average price (vwap): Definition and calculation. https://www.investopedia.com/terms/v/vwap.asp#:~:text=The%20volume%2Dweighted%20average%20price%20(VWAP)%20is%20a%20technical,on%20both%20volume%20and%20price.

Frankenfield, J. (2023). What is a robo-advisor? https://www.investopedia.com/terms/r/roboadvisor-roboadviser.asp

Frost, J. (2023). How to interpret r-squared in regression analysis. https://statisticsbyjim.com/regression/interpret-r-squared-regression/

Grobys, K., Kolari, J. W., & Niang, J. (2022). Man versus machine: On artificial intelligence and hedge funds performance. *Applied Economics*, *54*(40), 4632–4646. https://doi.org/10.1080/00036846.2022.2032585

Harvey, C. R. (1989). Forecasts of economic growth from the bond and stock markets. *Financial Analysts Journal*, *45*(5), 38–45. https://doi.org/10.2469/faj.v45.n5.38

Hayes, A. (2023). Financial markets: Role in the economy, importance, types, and examples. https://www.investopedia.com/terms/f/financial-market.asp

Jaeger, M., Krügel, S., Marinelli, D., Papenbrock, J., & Schwendner, P. (2021). Interpretable machine learning for diversified portfolio construction. *The Journal of Financial Data Science*, *3*(3), 31–51. https://doi.org/10.3905/jfds.2021.1.066

Jain, P., & Jain, S. (2019). Can machine learning-based portfolios outperform traditional risk-based portfolios? the need to account for covariance misspecification. *Risks*, *7*(3), 74. https://doi.org/10.3390/risks7030074

Jiang, J., Kelly, B. T., & Xiu, D. (2020). (re-)imag(in)ing price trends. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3756587

Korzeniewski, J. (2018). Efficient stock portfolio construction by means of clustering. *Acta Universitatis Lodziensis. Folia Oeconomica*, *1*(333). https://doi.org/10.18778/0208-6018.333.06

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59–69. https://doi.org/10.3905/jpm.2016.42.4.059

Malkiel, B. G. (2003). The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, *17*(1), 59–82. https://doi.org/10.1257/089533003321164958

Mallikarjuna, M., & Rao, R. P. (2019). Evaluation of forecasting methods from selected stock market returns. *Financial Innovation*, *5*(1), 40. https://doi.org/10.1186/s40854-019-0157-x

Markowitz, H. (1952). Portfolio selection*. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Markowitz, H. M. (1999). The early history of portfolio theory: 1600–1960. *Financial Analysts Journal*, *55*(4), 5–16. https://doi.org/10.2469/faj.v55.n4.2281

Millea, A., & Edalat, A. (2022). Using deep reinforcement learning with hierarchical risk parity for portfolio optimization. *International Journal of Financial Studies*, *11*(1), 10. https://doi.org/10.3390/ijfs11010010

Morah, C. (2022). What are all of the major us stock exchanges? https://www.investopedia.com/ask/answers/08/security-market-usa.asp

Poor's, S.
bibinitperiod. (n.d.). S&p 500®. https://www.spglobal.com/spdji/en/indices/equity/sp-500/

Russell, F. (2020). Smart beta: 2015 global survey findings from asset owners. https://www.ftserussell.com/research/smart-beta-2015-global-survey-findings-asset-owners

Schwendner, P., Papenbrock, J., Jaeger, M., & Krügel, S. (2021). Adaptive seriational risk parity and other extensions for heuristic portfolio construction using machine learning and graph theory. *The Journal of Financial Data Science*, *3*(4), 65–83. https://doi.org/10.3905/jfds.2021.1.078

Securities, U., & Commission, E. (n.d.). U.s. securities and exchange commission. https://www.sec.gov/news/studies/techrp97.htm

Wikipedia. (2023). Bresenham's line algorithm. https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Yang, Q., Hong, Z., Tian, R., Ye, T., & Zhang, L. (2020). Asset allocation via machine learning and applications to equity portfolio management. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3722621

A. Appendix

	CAGR	Sharpe Ratio	Volatility
1998	26.1	1.47	16.70%
1999	22.07	1.29	16.61%
2000	14.54%	0.94	15.71%
2001	17.33%	1.05	16.43%
2002	14.37%	0.91	16.28%
2003	10.23%	0.64	17.56%
2004	12.63%	0.78	17.16%
2005	13.44%	0.85	16.42%
2006	13.27%	0.87	15.82%
2007	13.69%	0.92	15.28%
2008	12.76%	0.86	15.29%
2009	8.52%	0.53	18.60%
2010	10.16%	0.6	19.33%
2011	10.86%	0.63	19.19%
2012	10.50%	0.61	19.44%
2013	11.09%	0.65	19.05%
2014	12.19%	0.71	18.67%
2015	12.54%	0.74	18.32%
2016	11.88%	0.71	18.16%
2017	12.15%	0.73	17.93%
2018	12.62%	0.77	17.55%
2019	11.70%	0.72	17.43%
2020	12.49%	0.77	17.21%
2021	12.51%	0.73	18.39%
2022	13.21%	0.77	18.18%
2023	12.25%	0.72	18.31%

	CAGR	Sharpe Ratio	Volatility
1998	17.22%	1.01	17.10%
1999	20.30%	1.13	17.70%
2000	16.14%	0.98	16.67%
2001	17.73%	1.03	17.39%
2002	14.94%	0.89	17.52%

2003	10.68%	0.63	18.99%
2004	13.62%	0.78	18.55%
2005	14.50%	0.85	17.75%
2006	14.46%	0.88	17.08%
2007	14.71%	0.92	16.51%
2008	13.59%	0.86	16.49%
2009	9.08%	0.53	20.42%
2010	11.30%	0.6	21.70%
2011	12.07%	0.64	21.52%
2012	11.46%	0.61	21.77%
2013	12.12%	0.64	21.34%
2014	13.27%	0.7	20.89%
2015	13.56%	0.72	20.47%
2016	12.78%	0.7	20.23%
2017	13.12%	0.72	19.97%
2018	13.57%	0.75	19.54%
2019	12.57%	0.71	19.37%
2020	13.38%	0.75	19.11%
2021	13.43%	0.72	20.29%
2022	14.17%	0.76	20.06%
2023	13.10%	0.71	20.19%

	CAGR	**Sharpe Ratio**	**Volatility**
1998	33.10%	2.06	14.33%
1999	24.00%	1.53	14.84%
2000	10.53%	0.77	14.36%
2001	10.30%	0.74	14.78%
2002	11.65%	0.85	14.29%
2003	13.99%	0.9	16.06%
2004	16.05%	1.05	15.29%
2005	16.41%	1.12	14.56%
2006	15.35%	1.09	14.02%
2007	15.88%	1.16	13.51%
2008	14.56%	1.09	13.32%
2009	8.69%	0.61	15.61%
2010	8.77%	0.61	15.90%
2011	9.38%	0.65	15.61%
2012	10.00%	0.69	15.49%
2013	10.26%	0.72	15.14%
2014	11.14%	0.79	14.85%
2015	11.31%	0.81	14.58%

2016	10.78%	0.78	14.52%
2017	10.93%	0.79	14.40%
2018	11.23%	0.83	14.11%
2019	10.46%	0.78	14.03%
2020	11.06%	0.83	13.91%
2021	11.58%	0.81	15.04%
2022	11.89%	0.83	14.89%
2023	11.09%	0.78	14.93%

Year	CAGR				Sharpe Ratio				Volatility			
	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=0.05’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=0.05’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=0.05’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=0.05’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=1’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=False’, ’max_concentration=0.05’	risk_constant=1’, ’return_estimate=0.000269’, ’vol_weighted=True’, ’max_concentration=0.05’
1998	39.96%	33.23%	32.91%	33.52%	2.44	2.19	2.16	2.16	14.16%	13.46%	13.56%	13.78%
1999	23.96%	16.97%	18.30%	16.55%	1.64	1.25	1.32	1.18	13.67%	13.23%	13.41%	13.88%
2000	10.57%	7.11%	8.61%	8.14%	0.8	0.58	0.68	0.64	13.66%	13.38%	13.37%	13.67%
2001	10.26%	7.93%	8.35%	7.98%	0.75	0.61	0.64	0.61	14.30%	14.09%	14.03%	14.25%
2002	12.15%	9.61%	8.92%	9.38%	0.9	0.74	0.69	0.72	13.90%	13.75%	13.75%	13.87%
2003	9.62%	7.50%	6.82%	8.19%	0.69	0.57	0.53	0.61	14.88%	14.55%	14.41%	14.68%
2004	13.03%	10.01%	9.02%	10.34%	0.93	0.75	0.69	0.77	14.32%	14.05%	13.89%	14.14%
2005	14.01%	11.25%	10.31%	11.46%	1.03	0.86	0.81	0.87	13.66%	13.42%	13.29%	13.51%
2006	13.31%	10.89%	9.80%	10.76%	1.01	0.86	0.79	0.85	13.21%	12.99%	12.86%	13.07%
2007	13.97%	11.91%	10.78%	11.62%	1.09	0.96	0.89	0.94	12.77%	12.55%	12.43%	12.61%
2008	12.58%	11.39%	10.43%	11.08%	1	0.93	0.87	0.9	12.69%	12.45%	12.37%	12.53%
2009	7.94%	6.76%	5.64%	6.52%	0.58	0.52	0.44	0.5	15.13%	14.64%	14.94%	14.70%
2010	8.35%	7.19%	6.25%	6.98%	0.6	0.54	0.48	0.53	15.40%	14.89%	15.10%	14.87%
2011	8.76%	7.66%	7.04%	7.46%	0.63	0.58	0.53	0.56	15.14%	14.66%	14.88%	14.72%
2012	9.38%	8.36%	7.48%	7.99%	0.67	0.62	0.56	0.6	15.06%	14.61%	14.83%	14.67%
2013	9.65%	8.74%	7.78%	8.30%	0.7	0.66	0.59	0.63	14.73%	14.29%	14.50%	14.35%
2014	10.46%	9.70%	8.84%	9.38%	0.76	0.73	0.67	0.71	14.45%	14.03%	14.24%	14.09%
2015	10.77%	9.78%	9.32%	9.77%	0.79	0.75	0.71	0.74	14.19%	13.81%	13.99%	13.86%
2016	10.05%	9.11%	8.80%	9.26%	0.75	0.7	0.68	0.71	14.15%	13.79%	13.96%	13.84%
2017	10.33%	9.47%	9.31%	9.71%	0.77	0.73	0.71	0.75	14.01%	13.68%	13.83%	13.72%
2018	10.67%	9.95%	9.79%	10.18%	0.81	0.78	0.76	0.79	13.73%	13.40%	13.55%	13.44%
2019	9.89%	9.28%	9.09%	9.50%	0.76	0.73	0.71	0.75	13.66%	13.34%	13.49%	13.40%
2020	10.51%	9.68%	9.86%	10.09%	0.81	0.77	0.77	0.79	13.56%	13.23%	13.33%	13.24%
2021	10.80%	9.87%	10.47%	10.73%	0.77	0.72	0.77	0.79	14.69%	14.45%	14.32%	14.27%
2022	11.25%	10.37%	10.80%	11.16%	0.81	0.76	0.8	0.82	14.55%	14.30%	14.18%	14.13%
2023	10.45%	9.79%	10.18%	10.52%	0.75	0.72	0.75	0.78	14.61%	14.34%	14.24%	14.19%
Average	12.41%	10.52%	10.19%	10.64%	0.89	0.79	0.77	0.8	14.20%	13.80%	13.90%	13.90%

	CAGR				Sharpe Ratio				Volatility			
	equal positive	**equal percent**	**sigmoid**	**market Indicator**	**equal positive**	**equal percent**	**sigmoid**	**market Indicator**	**equal positive**	**equal percent**	**sigmoid**	**market Indicator**
1998	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1999	6.44%	1.93%	6.33%	4.26%	0.55	0.21	0.54	0.47	12.92%	13.16%	12.87%	9.90%
2000	7.89%	10.83%	7.91%	6.38%	0.62	0.76	0.62	0.66	13.87%	14.99%	13.86%	10.12%
2001	12.97%	19.22%	14.39%	8.50%	0.86	1.1	0.93	0.84	15.70%	17.42%	15.77%	10.34%
2002	10.95%	16.58%	12.23%	7.60%	0.72	0.93	0.79	0.75	16.32%	18.27%	16.46%	10.57%
2003	7.94%	12.20%	9.12%	5.02%	0.51	0.67	0.57	0.45	18.13%	20.15%	18.31%	12.91%
2004	11.27%	14.92%	12.28%	7.98%	0.69	0.81	0.74	0.66	17.81%	19.60%	17.97%	12.93%
2005	12.79%	17.74%	14.03%	8.89%	0.79	0.97	0.85	0.75	17.10%	18.78%	17.24%	12.39%
2006	12.99%	17.48%	14.03%	9.24%	0.82	0.98	0.87	0.8	16.50%	18.12%	16.63%	11.98%
2007	13.39%	17.34%	14.31%	9.75%	0.87	1	0.91	0.86	15.96%	17.55%	16.09%	11.70%
2008	12.34%	15.25%	13.26%	9.26%	0.81	0.9	0.85	0.8	16.00%	17.53%	16.13%	12.00%
2009	7.85%	10.11%	8.60%	5.64%	0.48	0.56	0.51	0.42	20.10%	21.34%	20.27%	16.30%
2010	10.24%	12.68%	11.09%	7.59%	0.56	0.64	0.6	0.51	21.48%	22.61%	21.64%	17.42%
2011	11.17%	13.87%	12.07%	8.76%	0.6	0.69	0.64	0.57	21.32%	22.40%	21.47%	17.38%
2012	10.66%	13.38%	11.53%	8.37%	0.58	0.67	0.61	0.54	21.59%	22.67%	21.74%	17.84%
2013	11.34%	13.90%	12.14%	9.22%	0.61	0.7	0.64	0.59	21.17%	22.22%	21.32%	17.53%
2014	12.55%	15.06%	13.38%	10.36%	0.68	0.75	0.71	0.66	20.72%	21.74%	20.87%	17.20%
2015	12.87%	15.42%	13.66%	10.75%	0.7	0.78	0.73	0.69	20.31%	21.30%	20.45%	16.90%
2016	12.14%	14.59%	12.87%	10.16%	0.67	0.75	0.7	0.66	20.08%	21.09%	20.22%	16.80%
2017	12.49%	15.46%	13.24%	10.56%	0.69	0.8	0.72	0.69	19.82%	20.84%	19.97%	16.66%
2018	12.97%	16.09%	13.69%	11.12%	0.73	0.83	0.76	0.73	19.40%	20.41%	19.54%	16.32%
2019	12.00%	15.13%	12.71%	10.26%	0.69	0.8	0.72	0.68	19.24%	20.27%	19.39%	16.27%
2020	12.82%	16.30%	13.55%	11.06%	0.73	0.85	0.76	0.73	18.98%	20.05%	19.14%	16.10%
2021	12.88%	16.32%	13.62%	11.21%	0.7	0.82	0.73	0.69	20.18%	21.14%	20.34%	17.61%
2022	13.65%	17.07%	14.40%	12.03%	0.74	0.86	0.77	0.74	19.95%	20.95%	20.12%	17.46%
2023	12.60%	15.96%	13.35%	11.06%	0.69	0.81	0.72	0.68	20.08%	21.15%	20.27%	17.71%
Average	11.49	14.59	12.31	9	0.6836	0.7856	0.7196	0.6648	18.59	19.83	18.72	14.81

	CAGR		Sharpe Ratio		Volatility	
	Regression	**Classification**	**Regression**	**Classification**	**Regression**	**Classification**
1998	NaN	NaN	NaN	NaN	NaN	NaN
1999	13.15%	9.66%	0.93	0.66	14.47%	15.90%
2000	18.18%	16.39%	1.16	0.96	15.35%	17.30%
2001	16.93%	17.63%	0.85	0.86	21.05%	21.47%
2002	11.89%	19.45%	0.6	0.87	23.13%	23.63%
2003	10.18%	15.14%	0.51	0.67	25.45%	25.89%
2004	14.39%	18.27%	0.67	0.8	24.51%	24.98%
2005	15.79%	20.24%	0.75	0.9	23.29%	23.71%
2006	15.61%	20.44%	0.77	0.94	22.21%	22.64%
2007	16.09%	20.71%	0.81	0.98	21.34%	21.77%
2008	14.60%	19.22%	0.75	0.93	21.02%	21.37%
2009	10.81%	13.76%	0.53	0.63	25.89%	25.82%
2010	14.10%	18.44%	0.6	0.74	29.29%	28.35%
2011	15.19%	18.58%	0.64	0.75	28.72%	27.85%
2012	14.32%	17.67%	0.61	0.72	28.61%	27.81%
2013	14.83%	18.01%	0.64	0.75	27.91%	27.15%
2014	16.05%	18.78%	0.68	0.78	27.21%	26.47%
2015	16.23%	18.63%	0.7	0.79	26.57%	25.87%
2016	15.21%	17.23%	0.67	0.75	26.12%	25.47%
2017	15.81%	18.17%	0.7	0.79	25.75%	25.07%
2018	16.20%	18.57%	0.72	0.82	25.16%	24.52%
2019	15.26%	17.29%	0.7	0.78	24.82%	24.23%
2020	16.13%	17.99%	0.73	0.81	24.42%	23.87%
2021	16.71%	19.06%	0.73	0.82	25.62%	25.09%
2022	17.40%	19.74%	0.76	0.85	25.28%	24.80%
2023	16.44%	18.46%	0.73	0.8	25.32%	24.96%
Average	15.1	17.9	0.72	0.81	24.34	24.24

	CAGR			Sharpe Ratio			Volatility		
	ward	**average**	**single**	**ward**	**average**	**single**	**ward**	**average**	**single**
1998	29.62%	28.58%	31.24%	1.73	1.7	1.85	15.61%	15.47%	15.26%
1999	21.44%	20.79%	21.18%	1.36	1.34	1.37	15.12%	14.99%	14.90%
2000	12.81%	12.27%	12.83%	0.91	0.88	0.92	14.40%	14.32%	14.27%
2001	15.84%	16.19%	15.19%	1.05	1.07	1.02	15.04%	15.02%	14.98%
2002	12.82%	13.64%	12.85%	0.89	0.94	0.9	14.75%	14.75%	14.73%
2003	9.15%	9.99%	9.42%	0.63	0.68	0.65	15.87%	15.79%	15.80%
2004	11.37%	12.33%	11.78%	0.77	0.83	0.8	15.48%	15.39%	15.40%
2005	12.19%	13.10%	12.70%	0.85	0.91	0.89	14.83%	14.75%	14.75%
2006	11.91%	12.66%	12.31%	0.86	0.91	0.89	14.32%	14.25%	12.95%
2007	12.60%	13.15%	12.95%	0.93	0.97	0.96	13.83%	13.76%	13.76%
2008	11.90%	12.35%	12.08%	0.88	0.92	0.9	13.86%	13.78%	13.78%
2009	7.75%	8.14%	7.98%	0.53	0.55	0.55	16.75%	16.62%	16.64%
2010	8.91%	9.29%	9.20%	0.59	0.61	0.6	17.12%	17.02%	17.05%

2011	9.49%	9.86%	9.79%	0.62	0.64	0.64	16.98%	16.88%	16.90%
2012	9.46%	9.82%	9.71%	0.61	0.63	0.63	17.18%	17.07%	17.10%
2013	10.06%	10.39%	10.31%	0.65	0.68	0.67	16.82%	16.72%	16.75%
2014	11.16%	11.51%	11.42%	0.72	0.75	0.74	16.51%	16.41%	16.44%
2015	11.55%	11.89%	11.82%	0.76	0.78	0.77	16.22%	16.12%	16.15%
2016	11.00%	11.33%	11.25%	0.73	0.75	0.74	16.14%	16.05%	16.07%
2017	11.19%	11.46%	11.41%	0.75	0.76	0.76	15.96%	15.87%	15.89%
2018	11.61%	11.85%	11.84%	0.78	0.8	0.8	15.62%	15.53%	15.56%
2019	10.83%	11.03%	11.08%	0.74	0.76	0.76	15.55%	15.46%	15.49%
2020	11.57%	11.76%	11.80%	0.79	0.81	0.81	15.35%	15.26%	15.29%
2021	11.67%	11.87%	11.88%	0.75	0.77	0.77	16.53%	16.41%	16.43%
2022	12.27%	12.49%	12.52%	0.79	0.81	0.81	16.34%	16.23%	16.25%
2023	11.49%	11.68%	11.68%	0.74	0.76	0.76	16.45%	16.33%	16.36%
Average	12.37	12.67	12.62	0.82	0.85	0.84	15.72	15.63	15.58

	CAGR			Sharpe Ratio			Volatility		
Year	Equal Dollar	Equal Volatlity	Equal Volatility Contr	Equal Dollar	Equal Volatlity	Equal Volatility Contr	Equal Dollar	Equal Volatlity	Equal Volatility Contr
1998	16.91%	26.10%	17.22%	1	1.47	1.01	17.07%	16.70%	16.70%
1999	20.13%	22.07%	20.30%	1.13	1.29	1.13	17.70%	16.61%	16.61%
2000	16.04%	14.54%	16.14%	0.98	0.94	0.98	16.67%	15.71%	15.71%
2001	17.67%	17.33%	17.73%	1.02	1.05	1.03	17.39%	16.43%	16.43%
2002	14.89%	14.37%	14.94%	0.88	0.91	0.89	17.52%	16.28%	16.28%
2003	10.65%	10.23%	10.68%	0.63	0.64	0.63	18.99%	17.56%	17.56%
2004	13.60%	12.63%	13.62%	0.78	0.78	0.78	18.55%	17.16%	17.16%
2005	14.48%	13.44%	14.50%	0.85	0.85	0.85	17.75%	16.42%	16.42%
2006	14.44%	13.27%	14.46%	0.88	0.87	0.88	17.08%	15.82%	15.82%
2007	14.70%	13.69%	14.71%	0.92	0.92	0.92	16.51%	15.28%	15.28%
2008	13.58%	12.76%	13.59%	0.86	0.86	0.86	16.49%	15.29%	15.29%
2009	9.07%	8.52%	9.08%	0.53	0.53	0.53	20.42%	18.60%	18.60%
2010	11.29%	10.16%	11.30%	0.6	0.6	0.6	21.69%	19.33%	19.33%
2011	12.06%	10.86%	12.07%	0.64	0.63	0.64	21.52%	19.19%	19.19%
2012	11.45%	10.50%	11.46%	0.61	0.61	0.61	21.77%	19.44%	19.44%
2013	12.10%	11.09%	12.12%	0.64	0.65	0.64	21.34%	19.05%	19.05%
2014	13.26%	12.19%	13.27%	0.7	0.71	0.7	20.89%	18.67%	18.67%
2015	13.55%	12.54%	13.56%	0.72	0.74	0.72	20.47%	18.32%	18.32%
2016	12.78%	11.88%	12.78%	0.7	0.71	0.7	20.23%	18.16%	18.16%
2017	13.11%	12.15%	13.12%	0.72	0.73	0.72	19.97%	17.93%	17.93%
2018	13.57%	12.62%	13.57%	0.75	0.77	0.75	19.54%	17.55%	17.55%
2019	12.56%	11.70%	12.57%	0.71	0.72	0.71	19.37%	17.43%	17.43%
2020	13.37%	12.49%	13.38%	0.75	0.77	0.75	19.11%	17.21%	17.21%
2021	13.42%	12.51%	13.43%	0.72	0.73	0.72	20.29%	18.39%	18.39%
2022	14.16%	13.21%	14.17%	0.76	0.77	0.76	20.06%	18.18%	18.18%
2023	13.09%	12.25%	13.10%	0.71	0.72	0.71	20.19%	18.31%	18.31%
Average	13.69	13.27	13.73	0.78	0.81	0.78	19.18	17.5	17.5

Year	CAGR		Sharpe Ratio		Volatility	
	Minimum Variance	Best Markowitz	Minimum Variance	Best Markowitz	Minimum Variance	Best Markowitz
1998	33.10%	39.96%	2.06	2.44	14.33%	14.16%
1999	24.00%	23.96%	1.53	1.64	14.84%	13.67%
2000	10.53%	10.57%	0.77	0.8	14.36%	13.66%
2001	10.30%	10.26%	0.74	0.75	14.78%	14.30%
2002	11.65%	12.15%	0.85	0.9	14.29%	13.90%
2003	13.99%	9.62%	0.9	0.69	16.06%	14.88%
2004	16.05%	13.03%	1.05	0.93	15.29%	14.32%
2005	16.41%	14.01%	1.12	1.03	14.56%	13.66%
2006	15.35%	13.31%	1.09	1.01	14.02%	13.21%
2007	15.88%	13.97%	1.16	1.09	13.51%	12.77%
2008	14.56%	12.58%	1.09	1	13.32%	12.69%
2009	8.69%	7.94%	0.61	0.58	15.61%	15.13%
2010	8.77%	8.35%	0.61	0.6	15.90%	15.40%
2011	9.38%	8.76%	0.65	0.63	15.61%	15.14%
2012	10.00%	9.38%	0.69	0.67	15.49%	15.06%
2013	10.26%	9.65%	0.72	0.7	15.14%	14.73%
2014	11.14%	10.46%	0.79	0.76	14.85%	14.45%
2015	11.31%	10.77%	0.81	0.79	14.58%	14.19%
2016	10.78%	10.05%	0.78	0.75	14.52%	14.15%
2017	10.93%	10.33%	0.79	0.77	14.40%	14.01%
2018	11.23%	10.67%	0.83	0.81	14.11%	13.73%
2019	10.46%	9.89%	0.78	0.76	14.03%	13.66%
2020	11.06%	10.51%	0.83	0.81	13.91%	13.56%
2021	11.58%	10.80%	0.81	0.77	15.04%	14.69%
2022	11.89%	11.25%	0.83	0.81	14.89%	14.55%
2023	11.09%	10.45%	0.78	0.75	14.93%	14.61%
Average	13.09	12.41	0.91	0.89	14.71	14.16

Year	CAGR			Sharpe Ratio			Volatility		
	XG Boost (Classification)	CNN (equal percent)	HRP (Average)	XG Boost (Classification)	CNN (equal percent)	HRP (Average)	XG Boost (Classification)	CNN (equal percent)	HRP (Average)
1998	NaN	NaN	28.58%	NaN	NaN	1.7	NaN	NaN	15.47%
1999	9.66%	1.93%	20.79%	0.66	0.21	1.34	15.90%	13.16%	14.99%
2000	16.39%	10.83%	12.27%	0.96	0.76	0.88	17.30%	14.99%	14.32%
2001	17.63%	19.22%	16.19%	0.86	1.1	1.07	21.47%	17.42%	15.02%
2002	19.45%	16.58%	13.64%	0.87	0.93	0.94	23.63%	18.27%	14.75%
2003	15.14%	12.20%	9.99%	0.67	0.67	0.68	25.89%	20.15%	15.79%
2004	18.27%	14.92%	12.33%	0.8	0.81	0.83	24.98%	19.60%	15.39%
2005	20.24%	17.74%	13.10%	0.9	0.97	0.91	23.71%	18.78%	14.75%
2006	20.44%	17.48%	12.66%	0.94	0.98	0.91	22.64%	18.12%	14.25%
2007	20.71%	17.34%	13.15%	0.98	1	0.97	21.77%	17.55%	13.76%
2008	19.22%	15.25%	12.35%	0.93	0.9	0.92	21.37%	17.53%	13.78%

2009	13.76%	10.11%	8.14%	0.63	0.56	0.55	25.82%	21.34%	16.62%
2010	18.44%	12.68%	9.29%	0.74	0.64	0.61	28.35%	22.61%	17.02%
2011	18.58%	13.87%	9.86%	0.75	0.69	0.64	27.85%	22.40%	16.88%
2012	17.67%	13.38%	9.82%	0.72	0.67	0.63	27.81%	22.67%	17.07%
2013	18.01%	13.90%	10.39%	0.75	0.7	0.68	27.15%	22.22%	16.72%
2014	18.78%	15.06%	11.51%	0.78	0.75	0.75	26.47%	21.74%	16.41%
2015	18.63%	15.42%	11.89%	0.79	0.78	0.78	25.87%	21.30%	16.12%
2016	17.23%	14.59%	11.33%	0.75	0.75	0.75	25.47%	21.09%	16.05%
2017	18.17%	15.46%	11.46%	0.79	0.8	0.76	25.07%	20.84%	15.87%
2018	18.57%	16.09%	11.85%	0.82	0.83	0.8	24.52%	20.41%	15.53%
2019	17.29%	15.13%	11.03%	0.78	0.8	0.76	24.23%	20.27%	15.46%
2020	17.99%	16.30%	11.76%	0.81	0.85	0.81	23.87%	20.05%	15.26%
2021	19.06%	16.32%	11.87%	0.82	0.82	0.77	25.09%	21.14%	16.41%
2022	19.74%	17.07%	12.49%	0.85	0.86	0.81	24.80%	20.95%	16.23%
2023	18.46%	15.96%	11.68%	0.8	0.81	0.76	24.96%	21.15%	16.33%
Average	17.9	14.59	12.67	0.81	0.7856	0.85	24.24	19.83	15.63

Classification Models accuracy				
	Naive Baseline Benchmark	Model Accuracy	x1 coefficient (compared to S&P 500 index)	coefficient of intercept
CNN Model	0.3198533356	0.356775974	-0.0774	0.0006
XGBoost Model(Classification)	0.4560600673	0.4684257799	-0.0836	0.0006

	CAGR			**Sharpe Ratio**			**Volatility**		
	Classic Rule Based	Classic Optimised	Machine Learning	Classic Rule Based	Classic Optimised	Machine Learning	Classic Rule Based	Classic Optimised	Machine Learning
Year	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)	Equal Volatility Contr	Minimum Variance	XG Boost (Classification)
1998	17.22%	33.10%	NaN	1.01	2.06	NaN	16.70%	14.33%	NaN
1999	20.30%	24.00%	9.66%	1.13	1.53	0.66	16.61%	14.84%	15.90%
2000	16.14%	10.53%	16.39%	0.98	0.77	0.96	15.71%	14.36%	17.30%
2001	17.73%	10.30%	17.63%	1.03	0.74	0.86	16.43%	14.78%	21.47%
2002	14.94%	11.65%	19.45%	0.89	0.85	0.87	16.28%	14.29%	23.63%
2003	10.68%	13.99%	15.14%	0.63	0.9	0.67	17.56%	16.06%	25.89%
2004	13.62%	16.05%	18.27%	0.78	1.05	0.8	17.16%	15.29%	24.98%
2005	14.50%	16.41%	20.24%	0.85	1.12	0.9	16.42%	14.56%	23.71%
2006	14.46%	15.35%	20.44%	0.88	1.09	0.94	15.82%	14.02%	22.64%
2007	14.71%	15.88%	20.71%	0.92	1.16	0.98	15.28%	13.51%	21.77%
2008	13.59%	14.56%	19.22%	0.86	1.09	0.93	15.29%	13.32%	21.37%
2009	9.08%	8.69%	13.76%	0.53	0.61	0.63	18.60%	15.61%	25.82%
2010	11.30%	8.77%	18.44%	0.6	0.61	0.74	19.33%	15.90%	28.35%

2011	12.07%	9.38%	18.58%	0.64	0.65	0.75	19.19%	15.61%	27.85%
2012	11.46%	10.00%	17.67%	0.61	0.69	0.72	19.44%	15.49%	27.81%
2013	12.12%	10.26%	18.01%	0.64	0.72	0.75	19.05%	15.14%	27.15%
2014	13.27%	11.14%	18.78%	0.7	0.79	0.78	18.67%	14.85%	26.47%
2015	13.56%	11.31%	18.63%	0.72	0.81	0.79	18.32%	14.58%	25.87%
2016	12.78%	10.78%	17.23%	0.7	0.78	0.75	18.16%	14.52%	25.47%
2017	13.12%	10.93%	18.17%	0.72	0.79	0.79	17.93%	14.40%	25.07%
2018	13.57%	11.23%	18.57%	0.75	0.83	0.82	17.55%	14.11%	24.52%
2019	12.57%	10.46%	17.29%	0.71	0.78	0.78	17.43%	14.03%	24.23%
2020	13.38%	11.06%	17.99%	0.75	0.83	0.81	17.21%	13.91%	23.87%
2021	13.43%	11.58%	19.06%	0.72	0.81	0.82	18.39%	15.04%	25.09%
2022	14.17%	11.89%	19.74%	0.76	0.83	0.85	18.18%	14.89%	24.80%
2023	13.10%	11.09%	18.46%	0.71	0.78	0.8	18.31%	14.93%	24.96%
Average	13.73	13.09	17.9	0.78	0.91	0.81	17.5	14.71	24.24

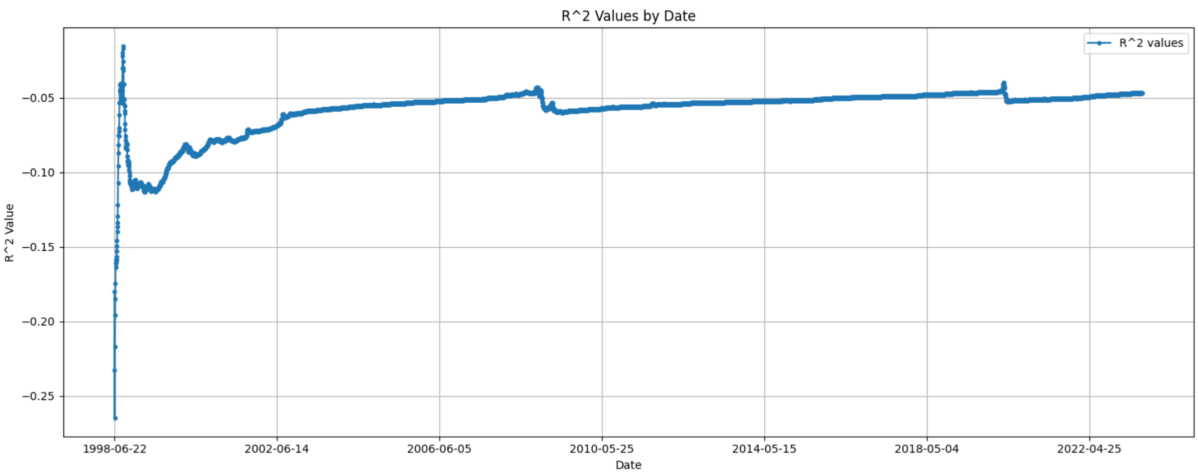

Figure A.1.: XGBoost model R^2 over time