NYU Tandon School of Engineering

Capstone Final Report
BoA Portfolio Construction in QWIM

Dongxin Huang, Hsiang-Chen Lee, Runfang Cao

Submitted in Fulfillment of The Requirements
for The Master's in Financial Engineering Summer 2023 Capstone Project
at NYU Tandon

NYU Capstone Advisor: Francisco Rubio
BoA Capstone Advisor: Cristian Homescu

Date: August 13, 2023

1 Introduction and Motivation

In the realm of investment management, the construction of a robust and efficient portfolio is a quintessential task for both individual and institutional investors. The ultimate goal is to allocate assets so that the portfolio attains the highest possible return for a given level of risk or the lowest risk for a given level of return. Since the seminal work of Harry Markowitz in 1952, which laid the foundation of Modern Portfolio Theory (MPT), various models and frameworks have been developed and embraced by practitioners and scholars [1].

One such enhancement to the classical MPT is Merton's portfolio problem [2], a continuous-time model that addresses portfolio selection with intertemporal consumption. This framework considers an investor's consumption patterns and their impact on the portfolio optimization process[2]. Moreover, it offers a dynamic approach, allowing investors to adapt to market changes efficiently. In the current landscape of financial markets, characterized by rapid changes and increasing complexity, Merton's model provides a more sophisticated and relevant approach to portfolio construction.

However, critical to the portfolio optimization process are the estimations of expected returns and the covariance matrix of asset returns, both notorious for being fraught with estimation errors. This study, therefore, endeavors to address these issues by employing efficient methodologies for the estimation of these critical parameters. In particular, we initially used the Bayes-Stein estimator [3] for calculating expected returns, which mitigates the impact of estimation errors by shrinking the extreme values. Pivoting from our initial approach, this study also leverages a multifactor regression model for predicting long-term returns, and validated both methods using the Information Coefficient (IC) and Information Ratio (IR). Diving deeper into the covariance matrix, we evaluated three models: the Fama-French 5-factor model [4], the double shrinkage model incorporating both L1 and L2 regularization [5], and the Ledoit-Wolf model [6]. The mathematical intricacies of these models will be thoroughly discussed in the implementation section. With these insights, we applied the mean-variance optimization model. The final section examines the models' performance using various measures such as annualized return, Sharpe ratio, maximum drawdown, omega ratio, Calmar ratio, Sortino ratio, and annual turnover. While all three covariance models can be instrumental in portfolio optimization, the double shrinkage model offers a slight edge in certain scenarios.

Besides the traditional portfolio optimization strategies, we incorporate reinforcement learning techniques to estimate the portfolio weights of 31 indices. Utilizing data from 2000 to 2023 obtained from the Bloomberg Terminal, our approach aims to capture the intricate relationships among these indices and adapt to changes over time. Machine learning techniques have proven to be remarkably adept at uncovering complex patterns in data, and their application in estimating the portfolio weights is anticipated to enhance the robustness of the portfolio construction process. This research represents a sophisticated approach to portfolio construction, combining classic economic theory with cutting-edge statistical and machine-learning methodologies. We hope that our findings will contribute valuable insights and practical guidance for both the academic and investment management communities.

2 Literature Review

2.1 Expected Return Prediction

Jorion, P. ("Bayes-Stein Estimation for Portfolio Analysis," 1986) [3]

In the field of portfolio analysis, the Bayes-Stein estimation emerges to address the uncertainty linked to parameter values that could lead to suboptimal portfolio decisions. Traditional estimation techniques in finance have, somewhat surprisingly, often overlooked the critical principle that security risk should be evaluated within the broader context of a portfolio. Such oversight can result in the inadequate use of sample means to estimate expected returns, thereby neglecting valuable information from other series. Jorion underscores this lacuna and introduces the application of shrinkage estimation to portfolio problems. Earlier applications of shrinkage estimators in finance have been predominantly ad hoc, lacking a comprehensive theoretical foundation. The Bayes-Stein approach seeks to remedy this by presenting a rationale for these estimators. Notably, when compared to the conventional sample mean, the Bayes-Stein method offers marked improvements, especially when portfolios comprise more than two assets, reducing estimation errors and potentially delivering enhanced portfolio performance.

2.2 Covariances, correlations, and volatilities: estimation, modeling, and analysis

De Nard and Zhao ("Using, Taming or Avoiding the Factor Zoo? A Double-Shrinkage Estimator for Covariance Matrices," 2021) [5]

Existing factor models struggle to model the covariance matrix for a large number of stocks and factors. Therefore, the authors introduce a new covariance matrix estimator that first shrinks the factor model coefficients and then applies nonlinear shrinkage to the residuals and factors. The estimator blends a regularized factor structure with conditional heteroscedasticity of residuals and factors and displays relatively good performance against various competitors. They show that for the proposed double shrinkage estimator, it is enough to use only the market factor or the most important latent factor(s). Thus there is no need for laboriously taking into account the factor zoo.

2.3 Optimization Models

Al Janabi ("Is optimum always optimal? A revisit of the mean-variance method under nonlinear measures of dependence and non-normal liquidity constraints," 2021) [7]

The authors are seeking to improve and generalize the classical Markowitz mean-variance approach to portfolio optimization. The author introduces a model that uses Liquidity-Adjusted Value-at-Risk (LVaR) as the risk measure. LVaR is an enhancement of the traditional Value-at-Risk (VaR), which incorporates the impact of liquidity on the portfolio's risk. This is an important modification as liquidity risk can be a significant factor, especially in stressed market conditions (Note stressed market conditions refer to situations in which the financial markets experience significant disruptions or extreme volatility. During stressed market conditions, there is often a sharp decline in asset prices, increased trading volumes, and higher levels of uncertainty among investors). Another modification the author makes to the traditional framework is by replacing the linear correlation measure, which is typically used, with Kendall's tau, a multivariate non-linear dependence measure. This is significant because linear correlations might not always accurately capture the relationships between asset returns, especially in times of market stress or when the relationships are inherently non-linear. For optimization, the model considers "multiple credible operational and budget constraints". By incorporating operational and budget constraints, the model becomes more realistic and practical, as these are crucial considerations for actual portfolio management. Lastly, the model is tested on a diversified large portfolio of international stock markets, including both developed and emerging economies, and commodities, which complies with the dataset we have so far.

Zhang et al. ("Deep Learning for Portfolio Optimization," 2020) [8]

The authors of the study employ deep learning models to directly optimize the Sharpe ratio of portfolios. They introduce a framework that eliminates the need for conventional forecasting procedures and enables the optimization of portfolio weights by updating model parameters. The Sharpe ratio optimization is accomplished using gradient ascent. The authors specifically utilized Long Short-Term Memory (LSTM) to adjust portfolio weights with Sharpe ratio maximization.

Peralta and Zareei ("A network approach to portfolio selection," 2016) [9]

The key of this paper is its demonstration of an inverse relationship between an asset's importance within a financial network and its optimal weighting in the context of the Markowitz portfolio theory. It accomplishes this by providing clear definitions, propositions, and corollaries that elucidate the link between the optimal weights of a portfolio and the centrality of its assets. A key point made is that securities occupying a central role within the financial network tend to carry higher systematic risk, as indicated by the positive correlation with the beta-CAPM. In conclusion, this paper creates a bridge between Markowitz's portfolio theory and network theory, illustrating the propensity to assign higher weights to low-central stocks in the pursuit of efficient portfolios. As such, after controlling for individual performance (measured either by Sharpe ratios or return volatility), optimal portfolio weights for high-influence securities in a correlation-based network are likely to be adjusted downward. This research also explores the potential of network-based investment strategies to enhance portfolio performance, examining both in-sample and out-of-sample data. The authors present the

ρ-dependent strategy and compare its performance against the simplistic 1/N rule and two Markowitz-based policies. The ρ-dependent strategy consistently exhibits higher portfolio Sharpe ratios and lower portfolio variance relative to the well-known benchmarks, a result that holds across a range of portfolio configurations, timeframes, and markets, even when transaction costs are factored in.

Moody et al. ("Learning to trade via direct reinforcement," 2001) [10]

The study introduces Direct Reinforcement as a novel approach to optimizing portfolios, asset allocations, and trading systems. Unlike traditional methods that rely on building forecasting models or estimating value functions through algorithms like TD-learning and Q-learning, this approach views investment decision-making as a stochastic control problem. It employs an adaptive algorithm called Recurrent Reinforcement Learning (RRL) to directly discover investment policies, eliminating the need for complex models. The authors argue that the Direct Reinforcement approach allows for a more straightforward problem representation and brings significant advantages in efficiency. They also demonstrate that RRL can be used to optimize risk-adjusted investment returns, including the differential Sharpe ratio, taking into consideration transaction costs. Through extensive simulations with real financial data, the study concludes that the RRL-based Direct Reinforcement strategies outperform those utilizing Q-Learning.

Molina ("Stock Trading with Recurrent Reinforcement Learning (RRL)," 2016) [11]

The study introduces a relatively new approach to financial trading, which uses machine learning algorithms to predict the fluctuation of asset prices. An optimal and rational trader would buy an asset before the price rises and sell it before its value declines. An algorithm of recurrent reinforcement learning (RRL) with a gradient ascent is designed to maximize a reward function known as Sharpe's ratio. By choosing optimal parameters, the author attempts to take advantage of asset price changes. When the Sharpe ratio is maximized, the positions at each time step are predicted by iteratively updating parameters. The approaches in the study, such as gradient ascent and the reinforcement learning method, can also be used for asset allocations.

3 Implementation

3.1 Expected Return Models

3.1.1 Bayes-Stein Shrinkage Estimator

In the pursuit of enhancing our portfolio optimization strategy, we implemented the Bayes-Stein estimator, a shrinkage estimator [3]. The essence of this method lies in its ability to produce better return estimates by moving the sample mean returns towards a prior [3]. For our implementation, the prior was a constant mean return for all assets. The rationale behind this approach is to counteract the estimation error found in sample means, especially when dealing with a limited sample size or volatile return series. The estimator shrinks the sample mean of the returns towards a constant vector, which reduces the estimation error.

The estimator is given by:

$$\hat{\boldsymbol{\mu}}_{BS} = (1-\omega)\overline{\boldsymbol{Y}} + \omega \boldsymbol{y}_0 \mathbf{1} \tag{1}$$

where $\overline{\boldsymbol{Y}}$ is the sample mean vector, $\boldsymbol{y}_0$ is a scalar constant, $\mathbf{1}$ is a vector of ones, and ω is the shrinkage intensity. The shrinkage intensity is determined by a function of the total sample size, the number of assets, and the dispersion of the mean returns from the prior. This ensures that the shrinkage is adaptive to the structure of the data.

ω is given by:

$$\omega = \frac{\lambda}{\lambda + T} \tag{2}$$

with

$$\lambda = \frac{(N+2)(T-1)}{(\boldsymbol{Y} - \boldsymbol{y}_0 \mathbf{1})^T \boldsymbol{S}^{-1} (\boldsymbol{Y} - \boldsymbol{y}_0 \mathbf{1})(T-N-2)} \tag{3}$$

$\boldsymbol{S}$ is the sample covariance matrix, N is the number of assets, and T is the number of time periods.

Recognizing the time-varying nature of asset returns, we applied the Bayes-Stein estimator within a rolling window framework. This allowed us to dynamically update our return estimates as new data became available. The chosen window length was approximately six months, equivalent to 20 trading days (or 21 trading days per month times six). For each window, we recalculated the Bayes-Stein estimates and stored them for further use.

3.1.2 Regression-Based Prediction

The primary objective of this method is to predict the expected returns of assets using a linear regression model. This model integrates various predictive factors to forecast future returns.

Given the set of observed returns y, the aim is to model them as a linear combination of various predictors (or factors), plus an error term. Mathematically, this is expressed as:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{4}$$

Where y is the observed return, X_1, X_2, and X_3 are the predictive factors, $\beta_0, \beta_1, \beta_2$, and β_3 are the coefficients, and ϵ is the error term.

X_0**: Bayes-Stein Forecast** shown in 3.1.1

X_1**: Rolling Mean of Log Returns** This factor calculates the mean of log returns over a rolling window. It's computed as:

$$X_1 = \frac{1}{n} \sum_{i=1}^{n} \log(1 + r_i) \tag{5}$$

Where r_i are the raw returns and n is the length of the rolling window.

X_2**: Relative Position of Log Price**

This factor assesses where the current log return stands relative to its recent range. Computationally, it's given by:

$$X_2 = \frac{\log(1 + p_t) - \min(\log(1 + p_i))}{\max(\log(1 + p_i)) - \min(\log(1 + p_i))} \tag{6}$$

Where p_t is the current raw price and min and max functions give the minimum and maximum log prices over a rolling window, respectively.

The predictive power of these expected return models is assessed through the Information Coefficient (IC), which measures the correlation between the predicted and actual returns. A higher magnitude of IC suggests a stronger predictive ability. The consistency is also evaluated using the Information Ratio (IR), a standardized measure adjusting for the IC's volatility.

3.1.3 Numerical Results

Figure 1 illustrates the cumulative historical IC of the Bayes-Stein estimator. Its mean IC is 0.0165, and the annualized IR is 0.7275. The Bayes-Stein estimator is used for long-term predictions of returns, but its performance is not good enough. Therefore, we utilize a multifactor regression model for return prediction.

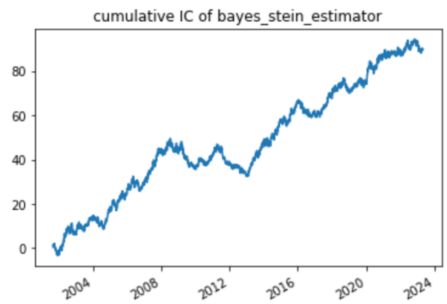

Figure 1: cum IC of Bayes-Stein estimator

Figure 2 illustrates the cumulative historical IC of X_1. X_1 is the 3-day rolling mean of log returns, which is a short-term reversal factor. Its mean IC is -0.0613, and the annualized IR is -2.6313. Figure 3 illustrates the cumulative historical IC of X_2. X_2 is the 10-day relative position of log price, which is a short-term reversal factor. Its mean IC is -0.0455, and the annualized IR is -2.2005.

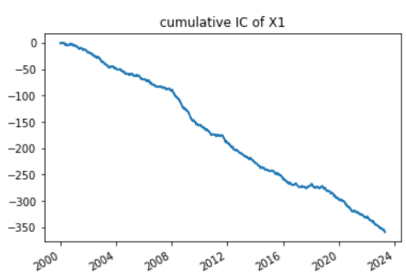

Figure 2: cum IC of X_1

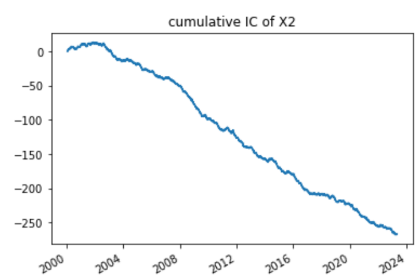

Figure 3: cum IC of X_2

Figure 4 illustrates the cumulative historical IC of $reg_forecasts$. Its mean IC is 0.0671, and the annualized IR is 3.0814. From the perspective of ICIR, reg_forecasts provide strong predictions for expected returns.

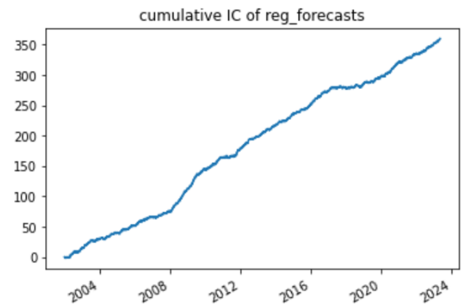

Figure 4: cum IC of reg forecasts

3.2 Asset Covariance Matrix Models

3.2.1 Fama-French 5 Model

The Fama-French 5-factor model (FF5) is an expansion of the previously developed 3-factor model by Eugene Fama and Kenneth French [4]. The original 3-factor model was introduced by Fama and French in the 1990s to explain anomalies in the well-known Capital Asset Pricing Model (CAPM) which relies on market beta alone to predict returns. Building on this, in 2015, Fama and French introduced two more factors to account for patterns in the cross-section of expected stock returns that were not captured by their original model. The Fama-French 5-factor model expands on the 3-factor model by adding two additional factors to account for profitability and investment [4].

The factors are [4]:

1. Market excess return
2. Size (SMB: Small Minus Big)
3. Value (HML: High Minus Low)
4. Profitability (RMW: Robust Minus Weak)
5. Investment (CMA: Conservative Minus Aggressive)

Given the matrix of asset returns R, assets are filtered based on the condition:

$$N_{\text{missing}} > \frac{1}{2} \times \text{length}(R)$$

For each asset, the excess returns over the risk-free rate, denoted as $R - R_f$ where R_f represents the risk-free rate, were regressed against the Fama-French Five-Factor matrix FF_5. The regression model is represented as:

$$R_i - R_f = \alpha + \beta_1 \times \text{Factor}_1 + \beta_2 \times \text{Factor}_2 + \cdots + \beta_5 \times \text{Factor}_5 + \epsilon$$

Where R_i is the return of asset i; α is the intercept; β_j denotes the exposure (beta value) of asset i to Factor j; ϵ is the residual risk or idiosyncratic risk

The covariance matrix, denoted as Σ, of the assets is constructed using the equation:

$$\Sigma = \beta \times \Sigma_{FF_5} \times \beta^T + D$$

Where β is the matrix of factor exposures (betas) of the assets to the Fama-French factors; Σ_{FF_5} is the covariance matrix of the Fama-French factors; D is a diagonal matrix with variances, denoted as ϵ^2, of the residuals from the regressions;

3.2.2 Double-Shrinkage Model

This model is an extensive study of the fama-french 5-factor model [5]. Given the matrix of asset returns R, assets are filtered based on the condition:

$$N_{\text{missing}} > \frac{1}{2} \times \text{length}(R)$$

For each asset, the excess returns over the risk-free rate, denoted as y, where $y = R - R_f$ and R_f represents the risk-free rate, are regressed against the Fama-French Five-Factor matrix X (denoted as FF_5) using ElasticNet regression:

$$y = X\beta + \epsilon$$

Where β denotes the vector of coefficients representing the exposure (beta values) of the asset to the Fama-French factors; ϵ is the residual error;

ElasticNet regression uses the objective:

$$\min_{\beta} \left(\frac{1}{2n} ||y - X\beta||_2^2 + \alpha\rho||\beta||_1 + \frac{\alpha(1-\rho)}{2} ||\beta||_2^2 \right)$$

Where:

α is the penalty term. ρ balances between L1 and L2 regularization

The covariance matrix, denoted as Σ, of the assets is constructed using the equation:

$$\Sigma = \beta \times \Sigma_{FF_5} \times \beta^T + D$$

Where β is the matrix of factor exposures (betas) of the assets to the Fama-French factors. Σ_{FF_5} is the covariance matrix of the Fama-French factors. D is a diagonal matrix with variances, denoted as ϵ^2, of the residuals from the regressions.

For both of the models, the correlation matrix, denoted as C, derived from the covariance matrix Σ, is represented as:

$$C_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \times \Sigma_{jj}}}$$

Where C_{ij} is the correlation between assets i and j, and Σ_{ii} and Σ_{jj} represent the variances of assets i and j respectively. For the thresholding and filtering of correlated pairs, the condition is:

$$|C_{ij}| > 0.7$$

This means that assets i and j are considered highly correlated if the condition is met.

3.2.3 Ledoit-wolf Shrinkage Model

The ledoit-wolf shrinkage model is a classic model used in the financial industry [6]. The Ledoit-Wolf shrinkage estimator is given by:

$$\Sigma_{\text{LW}} = \delta F + (1 - \delta) S \tag{7}$$

where S is the sample covariance matrix, F is the target matrix, and δ is the shrinkage constant between 0 and 1.

The sample covariance matrix S of a matrix R of asset returns is:

$$S_{ij} = \frac{1}{T-1} \sum_{t=1}^{T} (r_{ti} - \bar{r}_i)(r_{tj} - \bar{r}_j) \tag{8}$$

The constant correlation covariance matrix C is described as:

$$C_{ij} = \rho \sigma_i \sigma_j \tag{9}$$

where ρ represents the average pairwise correlation, and σ_i and σ_j are the standard deviations of assets i and j respectively.

The shrinkage covariance matrix Σ_{shrink} is defined as:

$$\Sigma_{\text{shrink}} = \delta C + (1 - \delta) S \tag{10}$$

with S being the sample covariance matrix, C the constant correlation covariance matrix, and δ as the shrinkage parameter between 0 and 1.

3.3 Optimization Model

The described function and its mathematical representation align with the Mean-Variance Optimization (MVO) approach to portfolio management. MVO is a quantitative tool developed by Harry Markowitz in the 1950s, which focuses on optimizing a portfolio's expected return for a given level of risk (as measured by variance or standard deviation) [1].

$$\mathbf{m}^T \mathbf{w}$$

where $\mathbf{m}$ is the vector of expected returns for each asset and $\mathbf{w}$ is the vector of weights of assets in the portfolio.

$$\mathbf{w}^T \Sigma \mathbf{w}$$

where Σ is the covariance matrix of returns. This term calculates the expected variance (or volatility) of the portfolio, considering both the variances of individual assets and the covariances between pairs of assets.

$$U(\mathbf{w}) = \mathbf{m}^T \mathbf{w} - \lambda \mathbf{w}^T \Sigma \mathbf{w}$$

This captures the trade-off between reward and risk. The function aims to maximize expected return while penalizing increased risk. The parameter λ denotes the risk aversion of the investor. A higher λ means the investor is more risk-averse and is more willing to sacrifice expected return to achieve lower risk.

Constraints on the weights ensure

1. $\sum_{i=1}^{n} w_i = 1$ (The portfolio is fully invested.)
2. $w_i \geq 0$ (Preventing short sales.)
3. $w_i \leq w_{max}$ (Ensuring diversification and limiting concentration risk.)

The goal in MVO is to find the set of weights $\mathbf{w}$ that provide the maximum utility. By adjusting the risk aversion λ or other parameters, one can trace out the efficient frontier, which is a graph showing the highest expected return available for each level of portfolio risk [1].

3.4 Reinforcement Learning with Gradient Ascent

The two reinforcement learning models with gradient ascent [10][11] in this section are used separately for portfolio weights optimization and for creating a strategy with a high Sharpe ratio for trading.

3.4.1 Reinforcement Learning with Gradient Ascent for Portfolio Weights Optimization

Unlike the methods mentioned above, the reinforcement model used in this section simplifies the steps in the traditional strategies. The model circumvents the requirements for anticipating expected returns, calculating the covariance matrix, and traditional objective function optimization such as mean-variance optimization, minimum variance optimization, and the Black-Litterman model. The model's output is the optimized portfolio weights which are updated through the process of the Sharpe ratio maximization. Below, the framework is introduced, and an exploration of how the Sharpe ratio can be optimized by gradient ascent is explained. The model algorithm and the detail of the functionality of each component in the method are also mentioned.

Objective Function

The Sharpe ratio is an important metric for evaluating the risk-adjusted return of a portfolio. It is defined as the ratio of the portfolio's expected return to its volatility.

Gradient Ascent

Gradient ascent is an optimization technique used to find the local maximum of a function. It works by iteratively moving in the direction of the steepest ascent or the direction of the positive gradient. For the goal of optimizing a portfolio using the Sharpe ratio, gradient ascent can be applied to find the portfolio weights that maximize the Sharpe ratio. By computing the gradient of the Sharpe ratio with respect to the portfolio weights and adjusting the weights in the direction where the Sharpe ratio increases, the Sharpe ratio can be converged to a solution that provides the highest risk-adjusted return.

$$w_{new} = w_{old} + \alpha \frac{\partial S}{\partial w} \tag{11}$$

where w is the portfolio's weights, S is the Sharpe ratio, and α is the learning rate.

Training Algorithm

The algorithm of gradient ascent is presented in the following steps. First, some parameters should be initialized. The initial weights for all assets are set randomly. The initial learning rate for gradient ascent equals the indicator learning rate, which is set at 0.02. It should be noted that the initial learning rate should be set at a reasonable rate. The advantage of a high learning rate is that it can allow the search process to speed up, but the risk of missing the maximum of the objective variable also exists. Setting a low learning rate can offset this risk but will result in a slow search process. Second, the searching iterations and the halting condition should be determined. The new weights will be calculated by gradient ascent, which will be used to calculate the new Sharpe ratio. If the new Sharpe ratio is larger than the max Sharpe ratio from the past iterations, the max Sharpe ratio will be replaced with the new Sharpe ratio. Otherwise, the learning rate will be halved, and implement the search algorithm again. The stopping condition is that the current learning rate is less than 0.1 times the indicator learning rate because using the low learning rate is ineffective, and the max Sharpe ratio may be found. The reason for using multiple learning rates in the algorithm is to speed up the optimization process. As the explanation mentioned above, at the beginning, the bigger learning rate is used to accelerate the process. If the relative biggest Sharpe ratio is found, the model will keep searching for the bigger Sharpe ratio using the same learning rate. Otherwise, the learning rate is halved.

Description of the Dataset

All thirty-one recommended assets listed in Appendix.A are used. The model will select some critical assets and unselect some assets by setting zero for their weights and then output the portfolio weights. All used assets have existed for more than 23 years and are diversified. A diversified portfolio can enhance the overall return per risk, and the idea of the strategy in this section is to have a system that delivers a good reward-to-risk ratio. The dataset ranges from 2000-01-01 to 2023-06-01 and contains daily observations. The model was retrained every two years to update the weights. The testing period is from 2018 to the mid of June 2023, including the most recent financial crisis due to COVID-19.

3.4.2 Reinforcement Learning with Gradient Ascent for Trading

Besides portfolio weights optimization, gradient ascent can be used in other situations, such as trading. A strategy with a high Sharpe ratio can be created by a reinforcement learning model with gradient ascent. Reinforcement learning can be employed to maximize the Sharpe ratio over a specific set of training data. By iteratively learning from the data and making decisions that aim to increase the Sharpe ratio, the reinforcement learning model can identify an investment strategy that achieves an optimal balance between risk and reward. This learned strategy is then evaluated on out-of-sample data, where the goal is to confirm that the strategy continues to exhibit a high Sharpe ratio. Reinforcement learning utilizes its ability to learn from complex and dynamic environments, such as the trading strategy in this section with multiple assets, which makes it a potentially powerful tool for portfolio optimization and risk-adjusted performance enhancement.

Objective Function

The objective function (reward function) for the trading strategy is also the Sharpe ratio. Also, assuming a risk-free rate of 0, the formula for computing the Sharpe ratio is simply the mean returns of the assets divided by the standard deviation of the returns.

$$S_T = \frac{A}{\sqrt{B - A^2}} \tag{12}$$

where $A = \frac{1}{T}\sum_{t=1}^{T} R_t$, and $B = \frac{1}{T}\sum_{t=1}^{T} R_t^2$

Trader Function

After determining the reward function, the trader function should be decided to know when to trade. The trader function F_t is the position at time t. The output of the trader function is the percentage between -1 and 1, which determines how many portions of the assets' position in the portfolio should be opened or closed. *theta* are the parameters that are optimized by using gradient ascent. $x_t = [1, r_{t-M}, ..., r_t, F_{t-1}]$ is the input vector, where r_t is the asset's prices difference at time t and $t-1$, and M is the number of r_t in x_t for calculating each F_t and also a historical window of size. Therefore, at each time step, the model utilizes its previous position along with a series of historical price changes to determine its next position.

$$F_t = \tanh(\theta^T x_t) \tag{13}$$

Returns

After knowing the positions at each time step, the portfolio's return can be calculated by the following formula [11]. The returns formula can be used for the Sharpe ratio formula and for calculating the portfolio's return.

$$R_t = F_{t-1} r_t - \delta |F_t - F_{t-1}| \tag{14}$$

where δ is the cost for a transaction at period t.

Gradient Ascent

The detail of gradient ascent is mentioned above in the last section. Gradient ascent is used to maximize the reward function, Sharpe ratio. In this section, the details of calculating the gradient is presented. In order to perform gradient ascent for maximizing the Sharpe ratio, it's necessary to compute the derivative of the Sharpe ratio with respect to the parameters being optimized, denoted by θ. The gradient of the Sharpe ratio provides the direction in which the parameters must be changed to increase the Sharpe ratio.

$$\theta_{new} = \theta_{old} + \alpha \frac{dS}{d\theta} \tag{15}$$

where S is the Sharpe ratio function, and α is the learning rate.

$$\frac{dS_T}{d\theta} = \sum_{t=1}^{T} (\frac{dS_T}{dA} \frac{dA}{dR_t} + \frac{dS_T}{dB} \frac{dB}{dR_t}) \cdot (\frac{dR_t}{dF_t} \frac{dF}{d\theta} + \frac{dR_t}{dF_{t-1}} \frac{dF_{t-1}}{d\theta}) \tag{16}$$

The complete steps for calculating the derivative and the partial derivatives mentioned above are presented by Molina [11].

Training

The exploration of the training algorithm is as follows. First, the positions F_t from time $t = M$ to $t = T$ are predicted. M is a historical window of size, and T is the number of all daily observations. Second, the gradient ascent function is implemented to calculate the gradient. θ are updated by gradient ascent. Also, the Sharpe ratio is updated. Third, repeat the first and second steps for many epochs until the convergence of the Sharpe ratio. The resulting Sharpe ratio can be plotted over each epoch to observe how it converges to a maximum, which is shown in Figure 13.

Testing

The recent COVID-19 pandemic led to significant declines and extreme volatility in global stock markets. These unprecedented market conditions, while challenging for investors, offer a unique opportunity to stress test investment methods and models. By evaluating performance during the crisis, it is possible to gain insight into how a model or strategy behaves under extreme circumstances.

Experimental Results

The reinforcement learning model greatly outperformed the benchmark method, which is shown in Figure 16 - Figure 19.

Description of the Dataset

The dataset used in the trading strategy is the same as that used for the portfolio weights optimization. Before training the model, each asset's portfolio weights from the strategy in the last section are used to calculate the returns, standard deviation, and Sharpe ratio.

3.5 Package Use

The datasets downloaded from Bloomberg Terminal for this portfolio construction project comprise daily financial indices. The majority of these indices have an extensive historical record, with data spanning 23 years. Financial indices generally exhibit more favorable statistical characteristics than individual stocks, consequently facilitating subsequent analyses and implementations. The lengthy time frame also ensures that various market regimes are adequately represented. The indices have been categorized based on their type and geographical focus. A comprehensive list of the datasets can be found in Appendix A.

The cvxpy is a Python library designed for defining and solving convex optimization problems. Through its syntax, users can model mathematical optimization problems in a manner consistent with standard mathematical expressions. Once these problems are set up, they can be solved using various solvers. This library is applied across various sectors, including operations research, finance, signal processing, and machine learning.

The fastparquet library in Python provides functionality to interact with the Parquet format, a columnar storage format. With fastparquet, users can read and write in the Parquet format and achieve compatibility with big data platforms such as Apache Spark and Apache Hive. The format's design offers certain data compression and performance characteristics that differ from other storage methods.

The empyrical library offers a collection of metrics and statistics commonly used for financial portfolios or stocks. This library allows for the calculation of various measures to evaluate portfolio performance and risk.

From the scikit-learn library, the LinearRegression module enables ordinary least squares linear regression, allowing for the determination of a linear relationship between variables. The ElasticNet module provides a linear regression model that combines both L1 and L2 regularization. This combination is used in scenarios where the dataset contains multiple correlated predictors.

Additionally, scikit-learn offers the LedoitWolf estimator, which computes a regularized covariance matrix using the Ledoit-Wolf shrinkage method. The shrinkage process adjusts the sample covariance matrix towards a structured estimator, with the goal of reducing estimation error. This method of covariance matrix estimation is employed in fields where the number of variables may exceed the number of observations.

4 Analysis

4.1 Selected Benchmarks

We use the average holdings of all assets as the benchmark.

4.2 Description of Metrics

4.2.1 annualized return

$$ann\_return = (1 + cum\_return)^{\frac{250}{days\_Held}} - 1. \tag{17}$$

4.2.2 sharpe ratio

$$sharpe\_ratio = \frac{r_p - r_f}{\sigma_p}, \tag{18}$$

where:
r_p = return of portfolio,
r_f = risk-free rate,
σ_p = standard deviation of the portfolio's excess return.

4.2.3 max drawdown

$$MDD = \max_{\tau \in (0,T)} \left[\min_{t \in (0,\tau)} (\frac{p_t}{p_\tau} - 1) \right], \tag{19}$$

where p_t is the net value of the portfolio at time t.

4.2.4 omega ratio

$$\Omega(\theta) = \frac{\int_\theta^\infty [1 - F(r)] dr}{\int_{-\infty}^\theta F(r) dr}, \tag{20}$$

where F is the cumulative probability distribution function of the returns and θ is the target return threshold.

4.2.5 calmar ratio

$$calmar\_ratio = \frac{ann\_return}{MDD}. \tag{21}$$

4.2.6 sortino ratio

$$sortino\_ratio = \frac{ann\_return - \theta}{\sqrt{\frac{1}{T}\sum(r_{p_t} - r_f)^2}}, \tag{22}$$

where θ is the target return threshold.

4.2.7 turnover

$$turnover = mean(sum(abs(w_t - w_{t-1})))/2 * 250, \tag{23}$$

where w_t is the weight vector of assets at time t.

4.3 Numerical Results

	annual_return	sharpe_ratio	max_drawdown	calmar_ratio	omega_ratio	sortino_ratio	annual_turnover
benchmark	0.0614	0.4603	-0.5191	0.1184	1.0284	0.1920	0.0000
max_return	0.3999	1.9212	-0.2513	1.5913	1.3954	2.5970	159.6108
MVO	0.3409	1.9516	-0.2430	1.4028	1.3763	2.4994	154.9664
MVO_ds	0.3557	1.9393	-0.2190	1.6239	1.3762	2.5057	157.7217
MVO_le	0.3328	1.9008	-0.2334	1.4260	1.3646	2.4229	155.2693

Figure 5: portfolio performance

The above results do not consider transaction costs; hence, the real-world performance of the strategies would be worse.

Except for the benchmark, both turnover rates and annualized returns of the strategies are high, mainly due to the significant influence of short-term reversal factors. From the figure, it is evident that the max_return strategy achieved the highest annualized returns, but its maximum drawdown is slightly worse than that of MOV, MVO_ds, and MVO_le.

The figures of strategy net value are included in the appendix.

The Sharpe ratio in Figure 7 is high due to the gradient ascent algorithm by the trading strategy. The strategy owns risk-adjusted performance enhancement, as the metrics listed in the figure shown.

	annual_return	sharpe_ratio	max_drawdown	calmar_ratio	omega_ratio	sortino_ratio	annual_turnover
value	0.020646	0.540556	-0.14295	0.14443	0.841653	-0.962484	5.314928

Figure 6: Reinforcement learning for weights optimization performance

	annual_return	sharpe_ratio	max_drawdown	calmar_ratio	omega_ratio	sortino_ratio
value	1.26332	3.163444	-0.312708	4.03994	2.487503	5.807456

Figure 7: Reinforcement learning for trading performance

5 Conclusion

The pursuit of constructing a potent and effective portfolio has always been at the forefront of investment management. Since the inception of Modern Portfolio Theory (MPT) by Harry Markowitz [1], there has been a continuous endeavor to improve and refine portfolio construction methodologies. The research presented here exemplifies this pursuit, incorporating Merton's dynamic approach [2] to intertemporal consumption and portfolio optimization, which addresses the evolving and intricate nature of contemporary financial markets. Our research has taken a dual approach. We utilized the Bayes-Stein estimator, which is adept at curbing the effects of estimation errors, and secondly, we employed a multifactor regression model for predicting long-term returns. Both of these methods were subjected to rigorous validation processes, ensuring their efficacy. For the covariance matrix, three distinct models were evaluated: the Fama-French 5-factor model, the double shrinkage model with both L1 and L2 regularization, and the Ledoit-Wolf model. When these were applied within the mean-variance optimization framework, they demonstrated similar performances, underscoring the importance of selecting the right model based on the specific needs and characteristics of the portfolio in question. Among these, the double shrinkage model emerged as a slight frontrunner, delivering an impressive annual return of 0.3556. This indicates that while all three models can be instrumental in portfolio optimization, the double shrinkage model might offer a slight edge in certain scenarios.

The concepts of reinforcement learning and gradient ascent are used in two different ways in the study [10][11]. First, they are utilized for portfolio weights optimization. The Sharpe ratio converges to the maximum when the weights are updated iteratively. Therefore, the model outputs the weights with risk-adjusted performance enhancement. Second, the reinforcement learning model with the gradient ascent algorithm is used for trading. The weights from the last model are used in this strategy. By iteratively using gradient ascent to update the predicted position at each time step, traders can follow the estimated results to buy or sell assets when the Sharpe ratio converges. The reinforcement learning model greatly outperformed the benchmark (buy and hold) method. The strategy has a high Sharpe ratio and relatively lower turnover than traditional optimization methods.

Further improvements could be made through the integration of real-time market sentiment analysis, which can augment the predictive power of our methodologies. We can also do further study of advanced machine learning techniques, such as deep learning, which could further refine the accuracy and robustness of our portfolio construction approach.

References

[1] H. Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2975974 (visited on 08/13/2023).

[2] R. C. Merton. "Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case". In: *The Review of Economics and Statistics* 51.3 (1969), pp. 247–257. ISSN: 00346535, 15309142. URL: http://www.jstor.org/stable/1926560 (visited on 08/13/2023).

[3] P. Jorion. "Bayes-Stein Estimation for Portfolio Analysis". In: *The Journal of Financial and Quantitative Analysis* 21.3 (1986), pp. 279–292. DOI: 10.2307/2331042.

[4] E. F. Fama and K. R. French. "A five-factor asset pricing model". In: *Journal of Financial Economics* 116.1 (2015), pp. 1–22. ISSN: 0304-405X. DOI: 10.1016/j.jfineco.2014.10.010. URL: https://www.sciencedirect.com/science/article/pii/S0304405X14002323.

[5] G. De Nard and Z. Zhao. "Using, taming or avoiding the factor zoo? A double-shrinkage estimator for covariance matrices". In: *Journal of Empirical Finance* 72 (2023), pp. 23–35. DOI: 10.1016/j.jempfin.2023.02.003.

[6] O. Ledoit and M. Wolf. "Shrinkage estimation of large covariance matrices: Keep it simple, statistician?" In: *Journal of Multivariate Analysis* 186 (2021), p. 104796. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2021.104796. URL: https://www.sciencedirect.com/science/article/pii/S0047259X2100095X.

[7] M. A. M. Al Janabi. "Is optimum always optimal? A revisit of the mean-variance method under nonlinear measures of dependence and non-normal liquidity constraints". In: *Journal of Forecasting* 40 (2021), pp. 387–415. DOI: 10.1002/for.2714.

[8] Z. Zhang, S. Zohren, and S. Roberts. "Deep learning for portfolio optimization". In: *The Journal of Financial Data Science* 2 (2020), pp. 8–20. DOI: 10.3905/jfds.2020.1.042.

[9] G. Peralta and A. Zareei. "A network approach to portfolio selection". In: *Journal of Empirical Finance* 38.Part A (2016), pp. 157–180. DOI: 10.1016/j.jempfin.2016.06.003.

[10] J. Moody and M. Saffell. “Learning to trade via direct reinforcement”. In: *IEEE Transactions on Neural Networks* 12.4 (2001), pp. 875–889. DOI: 10.1109/72.935097.

[11] G. Molina. “Stock Trading with Recurrent Reinforcement Learning (RRL)”. In: *CS229, nd Web* 15 (2016).

Appendix

Appendix.A

Commodity Index

BCOMTR: Bloomberg Commodity Index Total Return

Equity Indices (US)

RU20VATR: iShares Russell 2000 Value ETF
RUMCINTR: iShares Russell Mid-Cap ETF
RUMRINTR: iShares Micro-Cap ETF
RUTPINTR: iShares Russell Top 200 ETF
RU10GRTR: iShares Russell 1000 Growth ETF
RU10VATR: iShares Russell 1000 Value ETF
RU20GRTR: iShares Russell 2000 Growth ETF
RU20INTR: Russell 2000 Total Return
SPXT: Proshares S&P 500 EX Technology ETF

Equity Indices (Sector Specific - US)

S5COND: S&P 500 Consumer Discretionary Index
S5CONS: S&P 500 Consumer Staples Index
S5ENRS: S&P 500 Energy Index
S5FINL: S&P 500 Financials Sector GICS Level 1 Index
S5HLTH: S&P 500 Health Care Index
S5INDU: S&P 500 Industrials Index
S5INFT: S&P 500 Information Technology Index
S5MATR: S&P 500 Materials Index
S5RLST: S&P 500 Real Estate Index
S5TELS: S&P 500 Communication Services Index
S5UTIL: S&P 500 Utilities Index

Bond Indices

LBUSTRUU: Bloomberg Barclays US Aggregate Bond Index
LG30TRUU: Bloomberg Barclays Global High Yield Total Return Index Value Unhedged
LMBITR: Bloomberg Barclays Municipal Bond Index Total Return Index Value Unhedged USD

Equity Indices (International)

NDDUE15X: Amundi MSCI Europe Ex UK Ucits ETF
NDDUJN: MSCI Japan Index
NDDUNA: iShares MSCI North America UCITS ETF
NDDUPXJ: MSCI Pacific ex Japan UCITS ETF
NDDUUK: iShares MSCI UK ETF
NDDUWXUS: MSCI World ex USA total net return
NDUEEGF: SPDR MSCI Emerging Markets UCITS ETF

Appendix.B

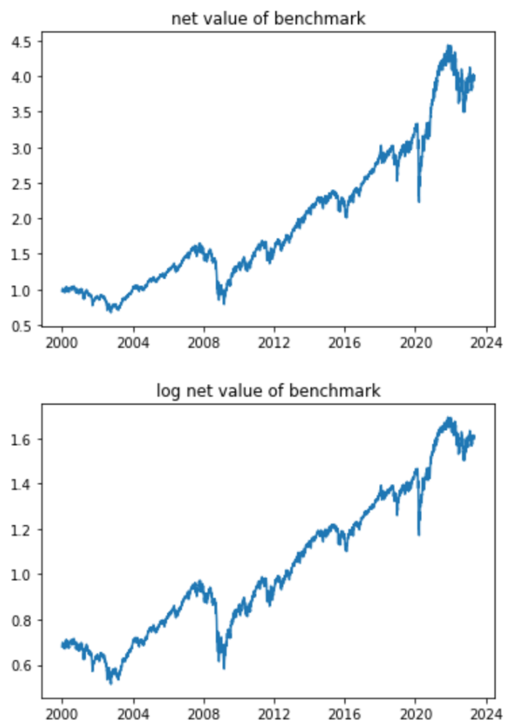

Figure 8: benchmark

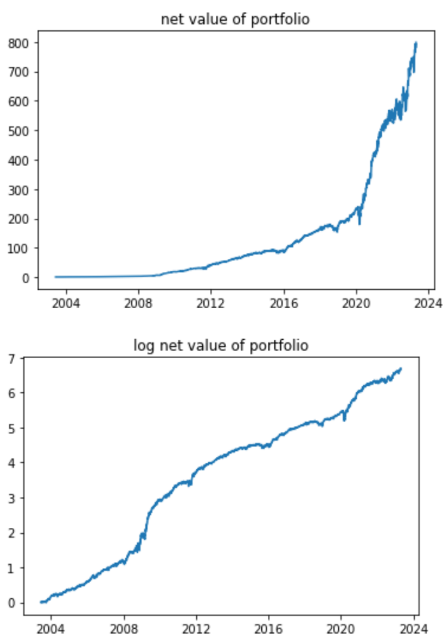

Figure 9: max_return

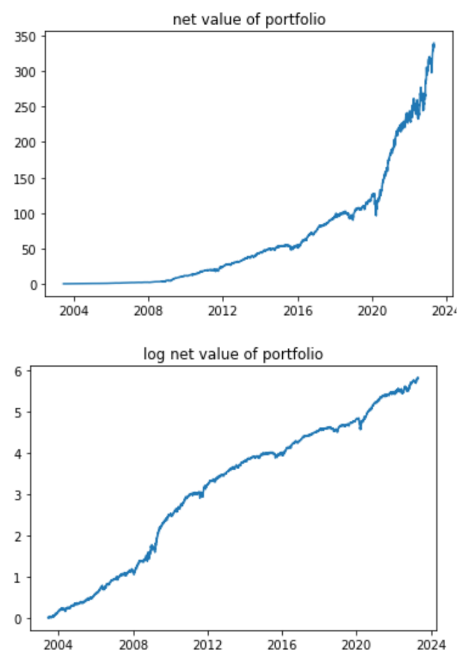

Figure 10: MVO

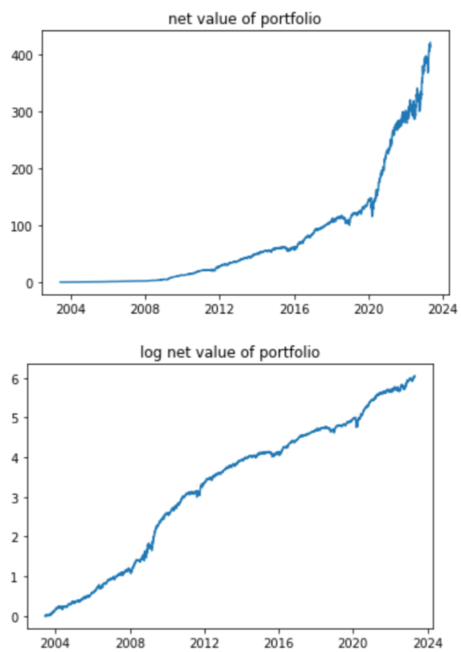

Figure 11: MVO_ds

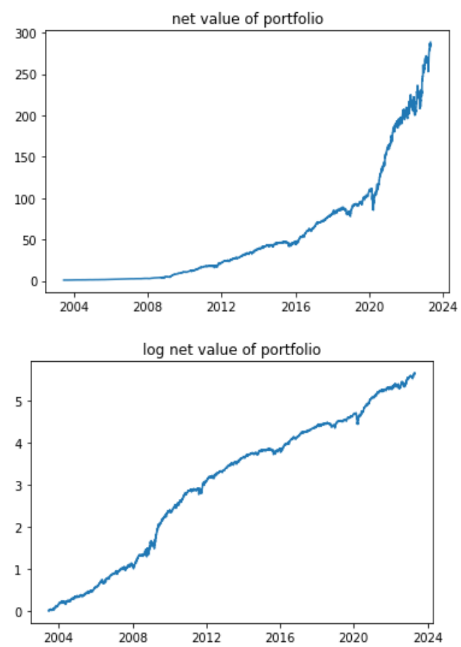

Figure 12: MVO_le

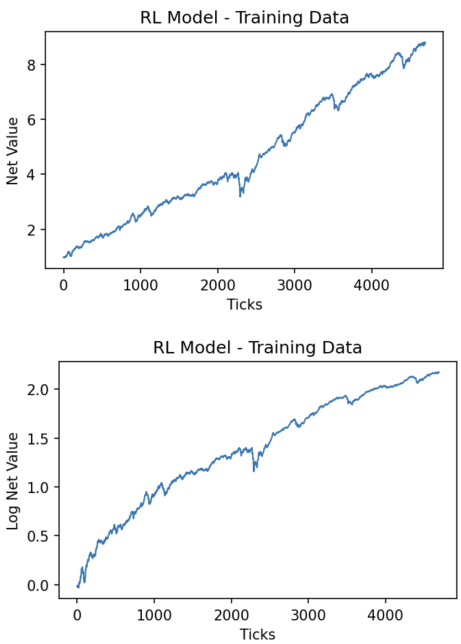

Figure 13: RL for weights optimization - training

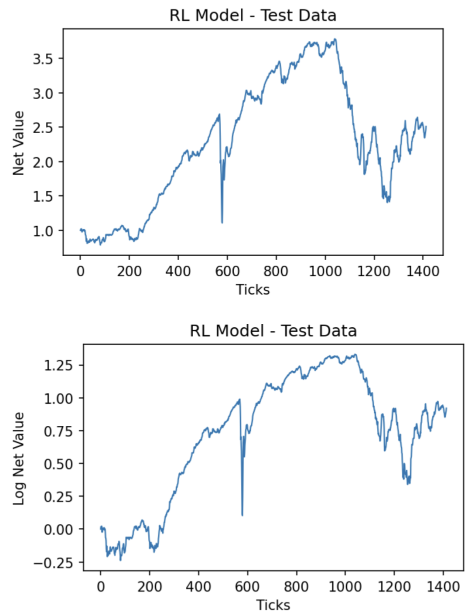

Figure 14: RL for weights optimization - testing

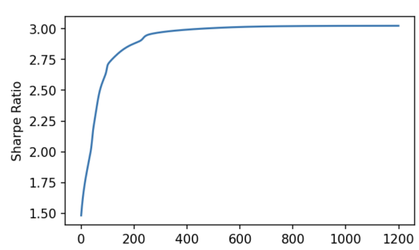

Figure 15: Sharpe Ratio Convergence

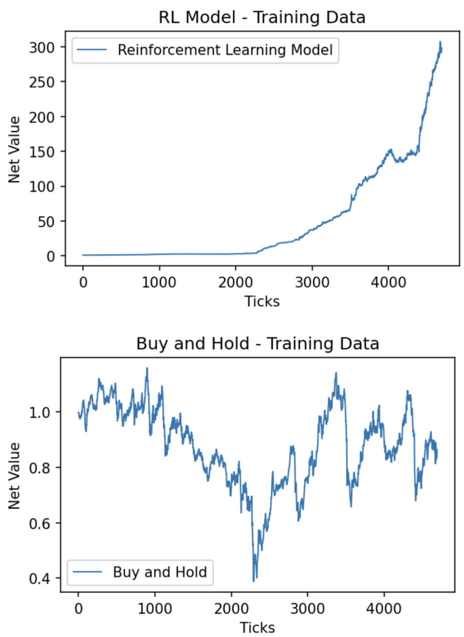

Figure 16: RL for trading (net value) - training

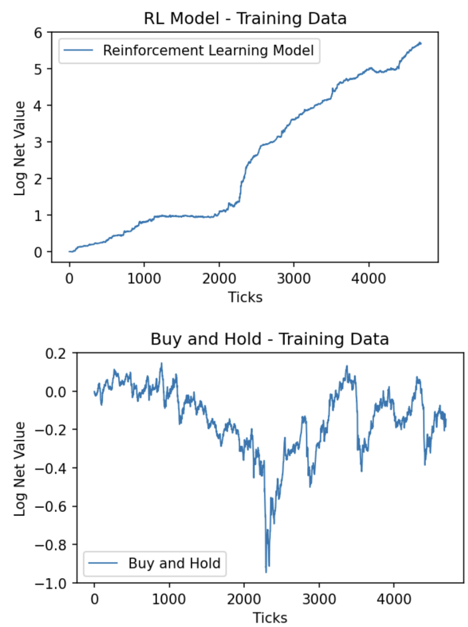

Figure 17: RL for trading (log net value) - training

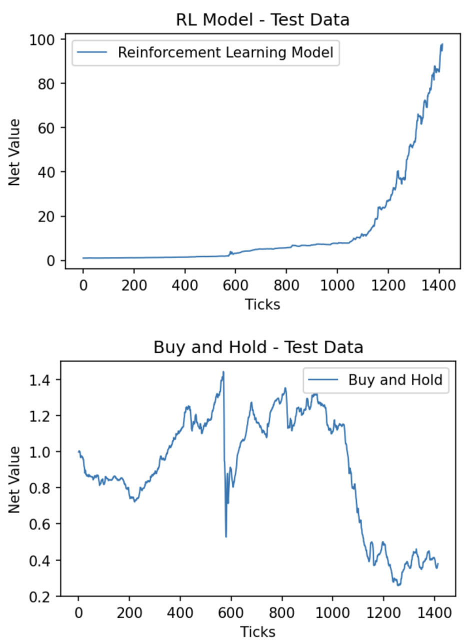

Figure 18: RL for trading (net value) - testing

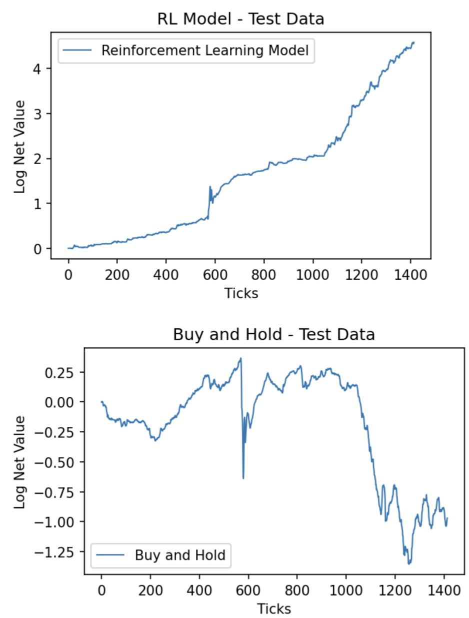

Figure 19: RL for trading (log net value) - testing