ISE 5405 Project Report I: Linearized Reinforcement Learning Enhanced Portfolio Optimization Model

Hung Tran & Asha Barua

8th Dec, 2024

Abstract

Linear portfolio optimization is a fundamental technique in quantitative finance that aims to construct investment portfolios by maximizing expected returns while minimizing risk. This paper examines the application of linear programming techniques for portfolio optimization, a major constituent within contemporary financial theory and the application of reinforcement learning in this field (Kolm and Ritter, 2019; Jiang et al., 2017). It outlines the possibility of efficiency that linear models may have in developing investment portfolios intended to obtain the highest expected return while attempting to limit exposure to risk at the same time. The technique for solving it is the formulation of an objective function or goal, usually to maximize returns, bound by a set of linear constraints that describe different investment variables (Becker et al., 2015; Poletaev and Spiridonova, 2021). It is based on historical financial data and uses a measure of statistical dispersion most of all covariance matrices when evaluating risk factors. Two base models in modern portfolio theory and finance are the mean-variance model of Markowitz and Capital Asset Pricing (CAPM) (Markowitz, 1991).

1 Introduction & Motivation of Problem

Recent developments in the quantitative finance field include increasing robustness to estimation errors and transaction cost factors through non-linear methods. Nonetheless, such findings confirm that linear portfolio optimization is still a useful tool in the asset allocation strategy implementation when considering changing markets (Saksham Jain, 2023). It further underlines the idea that the effectiveness of any model depends strongly on the estimation of input parameters provided and relative market dynamics. In so doing, we first formulate the problem following Markowitz Portfolio Theory (MPT) and later conduct the application of Reinforcement Learning in comparing the performance of profit maximization between traditional linear programming design and RL simulation (Hambly et al., 2021).

2 Literature Review

2.1 Relevant Literature

Portfolio optimization is the process of constructing a portfolio where we maximize returns for a given level of risk or minimize risk for a given level of return. The concept originated from Harry Markowitz's "Modern Portfolio Theory" (MPT) (Markowitz, 1952), which introduced the idea of an "efficient frontier," a set of optimal portfolios providing the best possible return for each level of risk. This concept underpins most portfolio optimization techniques in which MPT becomes the cornerstone of financial operations research. From this point, many scholars have explored and suggested multiple applications of the MPT to bridge the gap between the theory and real-life stock exchange in the financial markets. Becker et al. (2015) examines various methods to reduce the impact of estimation errors on optimal portfolio compositions in the context of the Modern Portfolio Theory (MPT). The paper compares two main portfolio optimization strategies and analyzes how to reduce the impact of estimation errors on optimal portfolio compositions. While previous studies comparing the traditional Mean-Variance (MV) optimization by Markowitz (1952) and the "resampled efficiency" by Michaud and Michaud (2008) had focused on specific settings, Becker et al. (2015) expands the analysis to encompass a wider range of realistic scenarios. Poletaev and Spiridonova (2021), on the other hand, proposes a method to reduce the dimensionality of data in the Markowitz portfolio optimization problem using hierarchical clustering of securities. The authors suggest that clustering securities based on their pairwise correlations using Pearson's correlation coefficient and constructing a covariance matrix of cluster returns will allow for solving the portfolio optimization problem with a reduced number of parameters, leading to faster computation (Poletaev and Spiridonova, 2021).

There are various applications of Markowitz's MPT using advanced mathematical OR modelings and Python initiation. Mallieswari et al. (2024) examine the NIFTY Pharma index of the Indian stock market, using data from 2020-2023 to analyze the performance of eight pharmaceutical companies and the index itself. They use Markowitz's theory to create an "efficient frontier" of portfolios that balance risk and return, calculating expected returns, volatility, and correlations between the chosen securities. The study also applies Monte Carlo simulations to predict the future end price of the NIFTY Pharma index and assess potential investment outcomes. Their findings suggest that the NIFTY Pharma portfolio offers a higher return (14.35%). Meanwhile, Xia (2023) scrutinizes specifically the case study of the U.S. Market using the MPT method and Python initiation. The paper analyzes the adjusted closing prices of five companies–Facebook, Amazon, Apple, Netflix, and Google–from 2011 to 2021 to demonstrate how MPT, which uses mean-variance analysis to measure portfolio performance (risk and return), can be used to create optimal portfolios (Xia, 2023). The authors constructed 25,000 portfolios with randomly generated weights and identified an efficient frontier, representing portfolios with the highest expected return for a given level of risk. The study found that a portfolio with 75.51% of the total investment in Amazon and Apple achieved the highest risk-adjusted return, while a portfolio with 75.04% invested in Google and Apple yielded the minimum risk. Similarly, Xidonas et al. (2021) describes the development of a Python-based decision support system for portfolio management that incorporates multiple criteria in both the security selection and portfolio optimization phases. The results are then aggregated to create a final ranking, allowing the selection of a specified number of top-ranked securities for inclusion in the portfolio. For the portfolio optimization phase, the system offers a range of models to determine the optimal allocation of capital to the selected securities. The authors highlight the system's ability to accommodate the decision-maker's preferences and investment strategies and the capacity to handle real-time problems and large datasets. The paper also indicates a large-scale application of the system, analyzing a substantial number of securities across various sectors and stock markets, including NYSE, NASDAQ, Paris, and Tokyo (Xidonas et al., 2021).

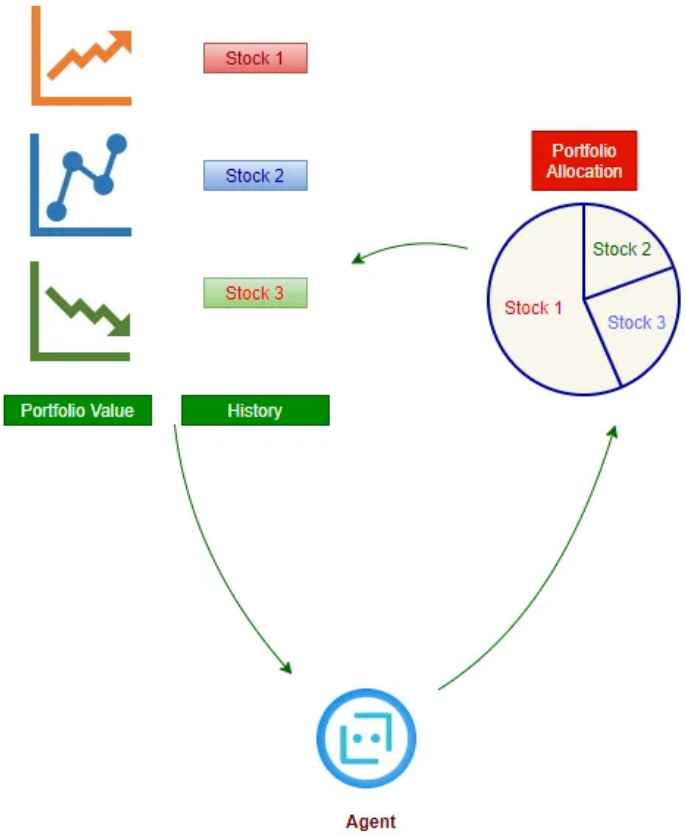

Figure 1: Flowchart of Portfolio Allocation Strategy (Samsudin, 2021)

The chart indicates the strategy of updating the policy over time to construct an optimized portfolio for future investment.

The combination of LP models with RL in portfolio optimization leverages some crucial developments in finance and computational methods in an attempt to maximize returns while maintaining risk at an acceptable level. Markowitz's MPT pioneered the field of optimizing portfolios between risk and return, but solving its quadratic programming model for large portfolios turned out computationally intensive. To handle this problem, Konno and Yamazaki (1991) proposed the MAD model, which replaced variance by mean absolute deviation. The problem thus became a linear programming problem. The linearization made the portfolio optimization much quicker and applicable to big datasets with a little loss of precision. Thousands of assets could be included, and constraints could be handled more easily, as shown for the NIKKEI 225 stocks (Konno and Yamazaki, 1991).

While the MAD model improves computational efficiency due to the simplification of portfolio optimization, it still keeps it as a static framework that does not adapt to new market conditions. This deficiency has thus been instrumental in the integration of reinforcement learning, which can provide for an ongoing process of learning through feedback coming from markets. An RL agent iteratively updates portfolio weights in pursuit of maximum cumulative reward by making adjustments according to observed returns and penalizing deviations from target performance. This enables the model to respond to market volatility in real-time, which is a considerable improvement over the more traditional forms of LP, which are optimized only for a single point in time (Mansini et al., 2014) (Jiang et al., 2017) (Acero et al., 2024) (Yashaswi, 2021). Here, the linear constraints of the MAD model complement the flexibility of RL by allowing for adjustments in the weights of the underlying assets toward real-time optimization. The practical implementation means initialization of a portfolio, setting constraints, and allowing an RL agent to iteratively adjust weights by projecting updates onto feasible boundaries each time until such time that the model reaches a configuration that maximizes long-term returns within acceptable limits of risk, aligning with real-time demands in volatile markets. That gives a unique advantage to combining RL in a regime of linearized portfolio optimization, amalgamating the adaptiveness of RL with LP computational efficiency. For now, this overcomes the limitations of static LP models and provides real-time updates through dynamic adjustments in portfolio optimization. Moreover, the linear structure of the MAD model will support computational efficiency when RL iterates on updating the weights. The challenges could be sensitivities related to parameters, such as learning rate and nonlinear dependencies, which can only be captured in a limited manner. This is a very promising hybrid model for high-frequency trading, which, with active research, increases responsiveness and tunes reward functions against real-time performance (Konno and Yamazaki, 1991) (Mortaji et al., 2024).

2.2 Relevant Data Source

In the field of financial market and portfolio optimization, Yahoo Finance is one of the companies that provides daily updated data for the stock market. Many articles suggest the indispensable role of Yahoo Finance along the way of articulating their methods and concepts of portfolio allocation. Lawrence et al. (2018) utilize Yahoo Finance to run an experiment of investment from May 12 to July 28, 2016, with a 1% subset of users being randomly assigned articles about a firm's earnings announcement. The study focused on earnings announcements because the experiment's design required that a firm have a timely news article available for promotion. During the experiment, the most recently available news article related to the treatment stocks was displayed at the top of the article list on the Yahoo Finance home page for the experiment user sample. On the other hand, Jagwani et al. (2018) promotes Yahoo Finance as a key source of data to test the efficacy of their proposed stock price forecasting models. Specifically, they extracted monthly stock prices for Apple Inc. from January 2000 to January 2018 from the Yahoo Finance website. This period provided them with an 18-year time series to use for their analysis. The data included open, high, low, close, and adjusted close prices. The authors chose to use Apple Inc. because it is a popular company with a long history, which means that a lot of information about the company is available online. The ready availability of Apple Inc.'s historical stock prices on Yahoo Finance made the platform crucial to the design and execution of their research.

3 Background

The classical Markowitz model uses variance as a risk measure, which requires complex quadratic programming. Our objective is to create a model that efficiently balances return and risk while also being computationally efficient and

easier to understand by linearizing the risk measure. **In this study, we present Mean Absolute Deviation (MAD) as a linear risk metric of measure. It simplifies the complex computations of the Markowitz mean-variance framework by measuring the average deviation of portfolio returns from their expected (mean) return. Unlike variance, which squares deviations and emphasizes extreme values, MAD uses absolute deviations, making it less sensitive to outliers.** It is considered a more robust risk measure, especially for portfolios where returns may not follow a normal distribution. For real-time and large-scale applications, this framework—which incorporates reinforcement learning—offers a balanced method of managing risk and optimizing returns while striking a balance between computational efficiency and adaptability, where the portfolio weights are modified by the RL agent's learning framework in response to rewards and penalties obtained from risk deviations and portfolio performance. Moreover, the RL agent's goal is to maximize the reward function by using LP techniques efficiently.

3.1 Markowitz's Modern Portfolio Theory

Indices and Sets

- i: Index for portfolio's stocks, where $i = 1, 2, ..., d$

Decision Variables

- R_i: Return of stock i
- $\mathbb{E}(R_i)$: Expected return (reward) of stock i
- w_i: Assigned weight for each stock in the portfolio

Parameter

- c: Portfolio variance threshold for stocks i

Objective Function

Markowitz Portfolio Theory (MPT) introduces the framework for constructing an investment portfolio that aims to maximize the return for a given level of risk or minimize the risk for a given level of expected return. In this paper, we focus more on how portfolio selection will perform to maximize the return (rewards) for each investment combination. To achieve this goal while preserving the linearity of the model, we assign each stock with its specific weight.

$$\max_{i \in R^d} \mathbb{E}(R_i) = \max_{i \in R^d} \sum_{i=1}^{d} w_i R_i \quad \forall i = 1, 2, ..., d$$

The weights will be assigned on the basis of the historical performance of the stocks and determine which stocks are likely to have the highest return. Note that in this situation, we only take into account the return of portfolio regardless of the risk impact.

Constraints

- **Non-negativity weight constraint**: In all possible portfolios $i \in R^d$, st.

$$w_i \geq 0 \quad \forall i = 1, 2, ..., d$$

- **Risk constraint**: For all possible portfolio combinations, we choose those with a variance (investment risk) below the threshold c. (Korn et al., 2001)

$$Var(R_i) \leq c$$

3.2 Mean Absolute Deviation (MAD) Formulation

MAD measures the average absolute deviation of portfolio returns from the mean return. **To linearize the problem, we will implement MAD as a risk measure by replacing the variance** (which is a non-linear constraint in our modified MPT [3.1]), as follows:

$$\text{MAD} = \frac{1}{T}\sum_{t=1}^{T}\left|R^t - \mathbb{E}(R^t)\right|$$

where,

- R^t: The return of portfolio at time t
- $\mathbb{E}(R^t)$: The expected return of portfolio over considered scenarios
- T : Number of scenarios

To incorporate MAD in a linear programming framework, we need to linearize it first. Here is the step-by-step procedure given:

Re-write the formulation as follows:

$$\text{MAD} = \frac{1}{T}\sum_{t=1}^{T}|y_t - \bar{y}|$$

where,

The return of the portfolio at time t is $y_t = R^t$. It can be expressed as a weighted sum of individual asset returns, which satisfies $\sum_{i\in I} w_i = 1,\ w_i \geq 0,\ \forall i \in I$. Therefore,

$$y_t = w_1 r_{1,t} + w_2 r_{2,t} + ... + w_n r_{n,t}$$

$$\Rightarrow y_t = \sum_{i=1}^{n} w_i r_{i,t}$$

Then, the expected return of the portfolio over considered scenarios is $\bar{y} = \mathbb{E}(R^t)$. It can be expressed as the weighted average of the returns of all possible scenarios, where the weights are the probabilities of the scenarios occurring and $\sum_t p_t r_{i,t}$ represents the expected return of asset i across all scenarios. It can be expressed as follows,

$$\bar{y} = \sum_t p_t y_t = \sum_t p_t \left(\sum_{i=1}^{n} w_i r_{i,t}\right) = \sum_i w_i \left(\sum_t p_t r_{i,t}\right)$$

Now, let us introduce, $|y_t - \bar{y}| = d_t^+ + d_t^-$. Then,
d_t^+: Positive deviation from the mean and $d_t^+ \geq 0$,

$$d_t^+ \geq y_t - \bar{y}$$

d_t^-: Negative deviation from the mean and $d_t^- \geq 0$,

$$d_t^- \geq -(y_t - \bar{y}) = \bar{y} - y_t$$

Hence, for each scenario t,

$$d_t^+ - d_t^- = y_t - \bar{y}$$

Now, the linearized form of MAD is,

$$\text{MAD} = \sum_{t=1}^{T} p_t \left(d_t^+ + d_t^-\right)$$

replacing $\frac{1}{T}$ for scenario probabilities p_t, where $\sum_{t=1}^{T} p_t = 1$.
Risk Penalty: The parameter λ incorporates into the MAD model to maintain the same trade-off mechanism and reflect the investor's risk tolerance.

$$\text{Risk Penalty} = \lambda \cdot \text{MAD} = \lambda \sum_{t=1}^{T} p_t \left(d_t^+ + d_t^- \right)$$

Depending on the market conditions and investor goals, λ values vary so that the portfolio ensures a satisfactory balance between risk and return for the investor's preferences.

3.3 Reinforcement Learning Formulation

- **State**(s_t): The state at time t is represented by the returns of each stock in the portfolio,
  i.e., $s_t = [r_{1,t}, r_{2,t}, ..., r_{N,t}]$
- **Action**(a_t): An action at time t is the distribution of weights across assets, $a_t = [w_{1,t}, w_{2,t}, .., w_{N,t}]$, where $w_{i,t}$ represents the weight of asset i in the portfolio at time t.
- **Reward**(r_t): The reward function evaluates the performance of a given action (weight distribution). A positive reward is given if the return from the portfolio weights results in a profit; otherwise, a negative reward (or punishment) is applied.
- **Policy**: A constrained gradient descent policy is used to iteratively adjust the portfolio weights to maximize rewards while satisfying constraints.

RL Settings

Objective Function:

Maximize cumulative returns, balancing expected return with risk.

$$\textbf{Maximize} \sum_{t=1}^{T} Reward(w_t, s_t)$$

Constraints:

- **Budget Constraint**: $\sum_{i=1}^{N} w_{i,t} = 1$
- **Non-negativity Constraint**: $w_{i,t} \geq 0, \ \forall i$
- **Policy Update**: The policy uses constrained gradient descent to adjust weights based on received rewards. If the reward is negative, the agent rejects the weight update and reverts to the prior action; if positive, it incorporates the weights for the next state. (Mansini et al., 2014) (Sutton and Barto, 2018)

4 Model Formulation

4.1 Nomenclature

- **Modern Portfolio Theory (MPT)** is used as a base model, wherein mean-variance optimization has been modified here to embody risk in the form of Mean Absolute Deviation (MAD).
- **Reinforcement Learning (RL)** adjustments are made to update the weights in a portfolio dynamically using cumulative feedback through some kind of iterative update process.

Indices and Sets

- i: Index for stocks, where $i = 1, 2, ..., N$
- t: Index for scenarios, where $t = 1, 2, ..., T$
- Set of all assets, $I = \{1, 2, ..., N\}$
- Set of all scenarios, $T = \{1, 2, ..., T\}$

Parameters

- R_i: Expected return of asset i
- $r_{i,t}$: Return of asset i in scenario t
- p_t: Probability of scenario t occurring, where $\sum_{t \in T} p_t = 1$
- λ: Risk aversion parameter, controlling the trade-off between return maximization and risk minimization.
- α: Learning rate for updating portfolio weights in the RL process.
- c: Optional risk threshold for MAD.

Decision Variables

- w_i: Weight of asset i in the portfolio, representing the proportion of the total investment allocated to asset i.
- y_t: Portfolio return in scenario t, calculated as $y_t = \sum_{i \in I} w_i r_{i,t}$
- d_t^+ and d_t^-: Positive and negative deviation variables, respectively, are used to linearize the Mean Absolute Deviation (MAD) risk measure.

4.2 Mathematical Formulation

The objective is to maximize the portfolio's expected return adjusted for risk (Moore, 1972), using the Mean Absolute Deviation (MAD) model as the risk measure, which implies that Reward = Expected Return - Risk Penalty.

Objective Function

$$\textbf{Maximize} \sum_{i \in I} R_i w_i - \lambda \sum_{t \in T} p_t (d_t^+ + d_t^-)$$

where:

- $\sum_{i \in I} R_i w_i$: The expected portfolio return.
- $\lambda \sum_{t \in T} p_t (d_t^+ + d_t^-)$: The MAD-based risk measure, weighted by λ to balance risk and return.

Constraints

- **The First Constraint**: To measure the deviation of each scenario return from the expected portfolio return, we introduce the deviation constraints for Mean Absolute Deviation (MAD):

$$d_t^+ - d_t^- = \sum_{i \in I} w_i (R_i - r_{i,t}),\ \forall t \in T$$

- **The Second Constraint**: To enforce a fully invested, long-only portfolio, we introduce the weight constraints:

$$\sum_{i \in I} w_i = 1,\ w_i\ \geq 0,\ \forall i \in I$$

- **The Third Constraint**: To ensure minimum level of risk:

$$\text{MAD} \leq c$$

In MAD formulation, it measures the average absolute deviation which implies that the deviations must be non-negative. As all scenario provides identical returns, for $c = 0$, there will be no deviation which is unrealistic. To ensure the feasibility of our optimization problem, we can bound c as $0 < c \leq max_t \left(d_t^+ - d_t^-\right)$. The upper bound of c represents the worst-case scenario of portfolio risk across all scenarios t. To explain it in general, c provides a concrete upper limit for average risk, aligning with the portfolio's actual observed risks across scenarios to ensure the validity and practicality of the optimization problem.
Why we didn't choose the constraint MAD $\geq c$? Because the constraint MAD $\geq c$ will ensure that the portfolio remains actively exposed to risk. In short, this constraint represents aggressive allocations with higher-risk assets by avoiding overly conservative portfolios to gain higher returns with greater volatility. On the other hand, our chosen risk constraint MAD $\leq c$ represents more stability with predictable returns by the conservative allocations with low-volatility assets.

Therefore, where:

- d_t^+ and d_t^- capture deviations of scenario returns above and below the expected return, respectively.
- Both deviation variables must be non-negative to ensure valid MAD calculation:

$$d_t^+ \geq 0,\ d_t^- \geq 0,\ \forall t \in T$$

- $\sum_{i \in I} w_i = 1$, The total portfolio weight equals 1 (full investment).
- $w_i \geq 0$ prevents short-selling, assuming a long-only portfolio.

4.3 Reinforcement Learning-Based Weight Adjustment Process

An RL agent iteratively updates weights w_i to maximize cumulative rewards under deterministic policy settings. The process uses a policy function to directly calculate actions (weights) without introducing randomness. The key steps are as follows:

- **Initialize Portfolio Weights:** Start with an initial allocation w_i and compute initial returns y_t using scenario-based returns $r_{i,t}$. The framework strictly follows a deterministic policy as actions (portfolio weights) are derived directly from the policy function $\pi(s_t; \theta_t)$. (Sutton and Barto, 2018)
- **Define the Reward Function:** This reward indicates how well the current weight configuration aligns with desired returns and risk tolerance. The reward function is based on the portfolio's return minus risk:

$$\text{Reward}(t) = \sum_{i \in I} R_i w_i - \lambda \sum_{t \in T} p_t (d_t^+ + d_t^-)$$

- **Policy Function - Gradient Descent Update:**

  1. Use constrained gradient descent to adjust weights iteratively, taking the gradient of the reward function:

$$\nabla_{w_i} = \frac{\partial}{\partial w_i} \left(\sum_{i \in I} R_i w_i - \lambda \sum_{t \in T} p_t (d_t^+ + d_t^-) \right)$$

  2. Update weights in the direction of the gradient:

$$w_i \leftarrow w_i + \alpha \cdot \nabla_{w_i}\ ,\ \text{where } \alpha \text{ is a learning rate.}$$

- **Projection Step to Satisfy Constraints:** Project weights back into the feasible region by ensuring non-negativity and normalizing:

$$w_i = max(0, w_i),\quad w_i = \frac{w_i}{\sum_i w_i}$$

- **Iteration:** Repeat steps until the reward function stabilizes or reaches a maximum, meaning optimal weights are found based on cumulative returns and penalties.

5 Experimental Setup

The companies we choose to invest in are: Software-application companies, Airline/Aviation firms, Biotechnology agents, Semiconductor agents such that -
AMZN: Amazon.com, Inc. (Internet Retail)
GOOG: Alphabet Inc. (Internet Content & Information)
AAPL: Apple Inc. (Consumer Electronics)
TSLA: Tesla, Inc. (Auto Manufacturers)
META: Meta Platforms, Inc. (Internet Content & Information)
AAL: American Airlines Group Inc. (Airlines)
LUV: Southwest Airlines Co. (Airlines)
DAL: Delta Airlines, Inc. (Airlines)
UAL: United Airlines Holdings, Inc. (Airlines)
BNTX: BioNTech SE (Biotechnology)
BIIB: Biogen Inc. (Drug Manufacturers - General)
BIO: Bio-Rad Laboratories, Inc. (Medical Devices)
TECH: Bio-Techne Corporation (Biotechnology)
NBIX: Neurocrine Biosciences, Inc. (Drug Manufacturers)
CHT: Chunghwa Telecom Co., Ltd. (Telecom Services)
TLK: Perusahaan Perseroan (Persero) PT Telekomunikasi Indonesia Tbk (Telecom Services)
TEF: Telefónica, S.A. (Telecom Services)
TDY: Teledyne Technologies Incorporated (Scientific & Technical Instruments)
NVDA: NVIDIA Corp (Chips and Semiconductors Manufacturing)

In Figure 2 and Figure 3, we have worked with both trained and test datasets, respectively, for each of the given stocks by following the traditional MPT non-linear model. For the trained dataset, our output is as follows- from 2020-01-02 to 2024-09-30, **Total Earning is 54516.0 Dollars** and **Total Portfolio's Risk is 45.91**.
For the test dataset, our output is as follows: from 2024-11-04 to 2024-11-22, **Total earning is -66.41 Dollars** and **Total Portfolio's Risk is 45.91**. For both the trained and test datasets, our solution's runtime is 0.00 seconds.

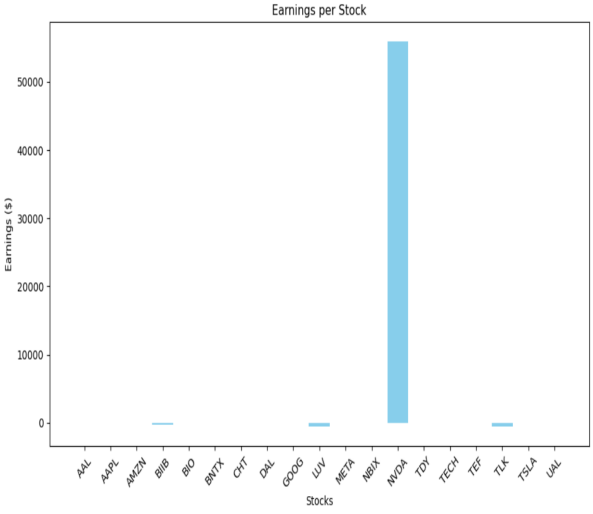

Figure 2: Traditional MPT: Trained dataset

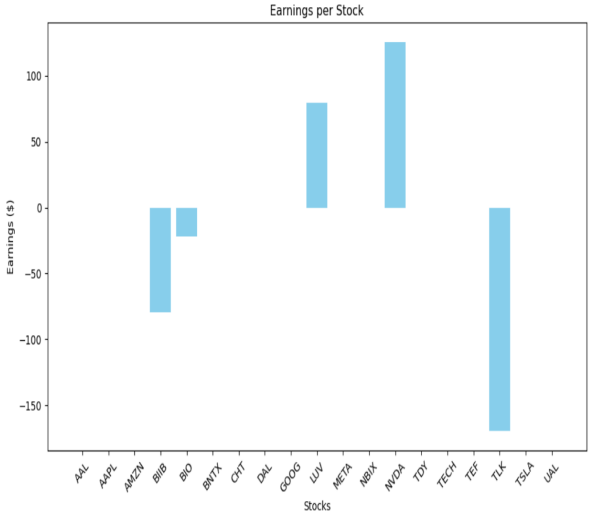

Figure 3: Traditional MPT :Test dataset

In Figure 4 and Figure 5, we have shown the portfolio weight allocations for both trained and test datasets, respectively, by following modified MPT to find the difference of the output results and how the non-linear risk constraints impact

the modified MPT formulation of maximized expected return. For both of the datasets, **Variance is 3.5** and the solution's runtime is 0.00 seconds.
For the trained dataset, the number of trades are 6 and we have received **Expected return = 13001.64**. For the test dataset, the number of trades are 4 and we have received **Expected return = 421.10**.

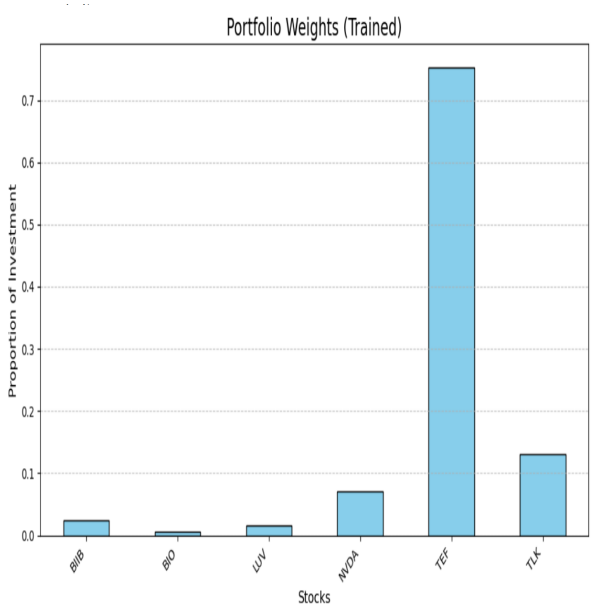

Figure 4: Modified MPT: Trained dataset

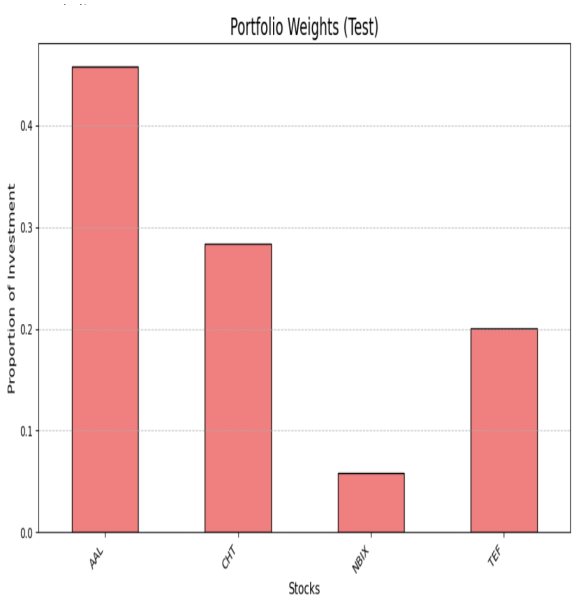

Figure 5: Modified MPT :Test dataset

Figure 6 represents the main model formulation of linearized portfolio optimization using MAD(Mean Absolute Deviation). In this case, we get **the Objective Value of 9.78e-03** represents the best trade-off between return and risk based on the defined reward function. This function incorporates both the expected return and the penalty for deviations (MAD) scaled by the risk aversion parameter λ. The allocation of weights reflects the optimal investment distribution across the five assets to achieve this trade-off in percentage with respect to the budget constraints. The optimal portfolio weights are- Stock TSLA: 0.0494 (4.94%), Stock LUV: 0.5834 (58.34%), Stock BNTX: 0.0857 (8.57%), Stock TLK: 0.0196 (1.96%), Stock TEF: 0.2619 (26.19%). The portfolio achieves an **Expected Return of 0.02 (approximately** 1.98%**)** with a **MAD risk of 0.02 (approximately** 2.00%**)**. The entire process has solved in 1285 iterations and 0.10 seconds (0.39 work units). Therefore, the results demonstrate the model's ability to navigate the risk-return trade-offs ($\lambda = 0.5$) in risk-sensitive portfolio optimization with risk limit ($c = 0.02$).

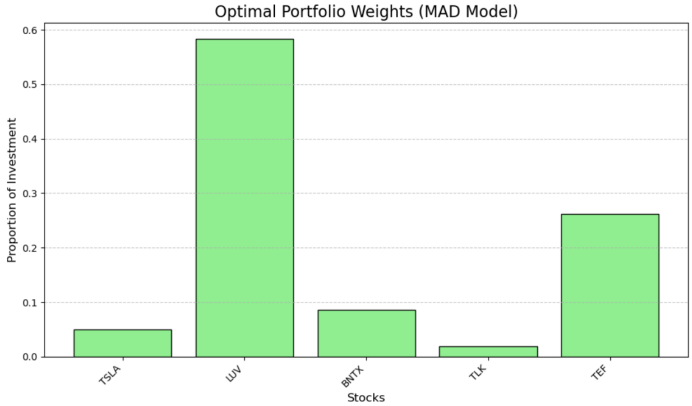

Figure 6: Linearized Portfolio using MAD

The linearized MAD model of portfolio optimization in Figure 7 illustrates how portfolio weights change across different values of the risk-return tradeoff parameter, λ. On the x-axis, λ reflects the balance between return and risk, with smaller values ensuring returns and larger values emphasizing risk minimization (as measured by MAD). The y-axis indicates the proportion of the portfolio invested in each stock, with each line representing a stock actively included in the portfolio for at least one λ value. Stocks with zero allocation across all λ values are excluded from both the plot and the legend. A horizontal line suggests consistent allocation to a stock regardless of risk-return preference. A decline in weights for increasing λ value suggests a preference for less risky stocks, aligning with a more conservative investment strategy. The chart effectively captures the balance between risk and return, revealing the dynamic shifts in portfolio composition as λ varies, and provides a clear visualization of how diversification and risk preferences influence investment decisions.

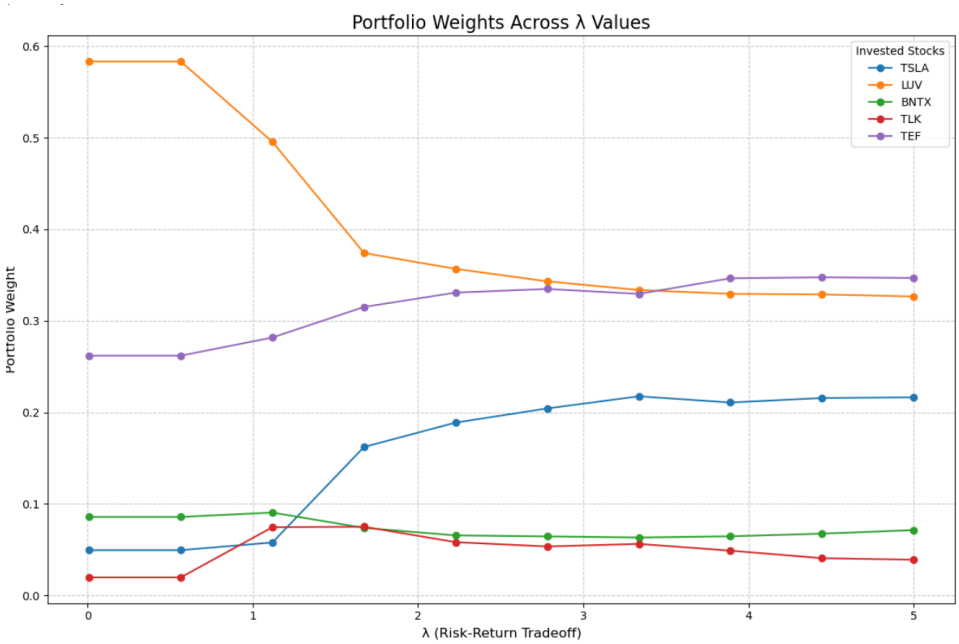

Figure 7: Linearized Portfolio using MAD for different values of the risk-return tradeoff (λ)

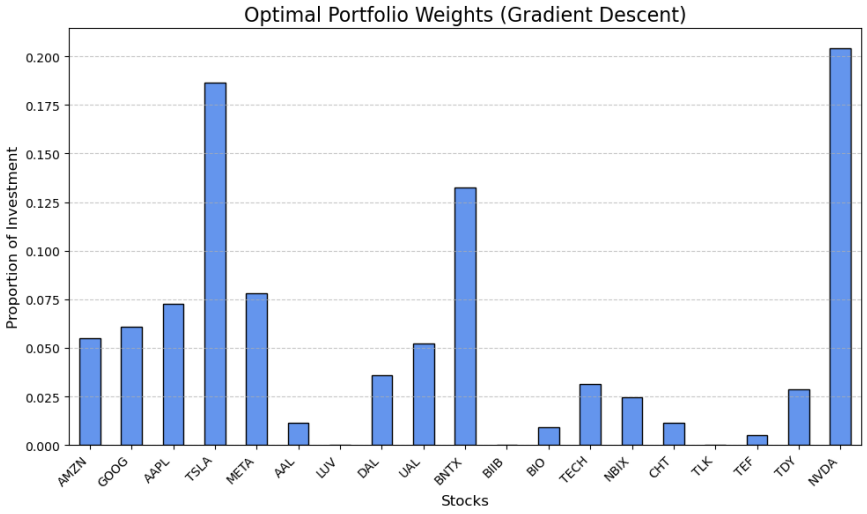

Figure 8: Optimal Portfolio Weights using fix $\lambda = 0.5$ in the Linearized Portfolio model with Gradient Descent

In Figure 8, we have shown the allocation of weights for each of the stocks by implementing the linearized MAD model of portfolio optimization with gradient descent. Our result has demonstrated the model's ability to navigate risk-return trade-offs ($\lambda = 0.5$) with learning rate $\alpha = 0.01$ and convergence threshold $= 1e - 6$. The process takes **213 number of iterations**. The optimal portfolio weights of each of the stocks are- AAL (0.011249), AAPL (0.072680), AMZN (0.055090), BIIB (0.000000), BIO (0.009348), BNTX (0.132469), CHT (0.011456), DAL (0.036042), GOOG (0.060806), LUV (0.000000), META (0.078057), NBIX (0.024724), NVDA (0.204263), TDY (0.028614), TECH (0.031481), TEF (0.005151), TLK (0.000000), TSLA (0.186457), UAL (0.052112). Our final **Outputs** are:

Final reward: 0.23
Expected return: 0.47
Mean Absolute Deviation (MAD): 0.47

We can see the significance of adding gradient descent in our model, which has lessened the number of iterations and at the same time, the reward reflects the balance between a reasonable return while keeping the risk (MAD) within acceptable bounds.

Figure 9 visualizes the relationship between the expected return and mean absolute deviation (MAD) across different values of the risk-return tradeoff parameter, λ. It uses a dual y-axis approach, where the expected return is plotted on the left y-axis (in blue), and MAD is plotted on the right y-axis (in red). As λ increases, the portfolio's expected return remains relatively stable, while the MAD tends to decrease. This suggests that as the investor places more emphasis on minimizing risk (increasing λ), the overall risk (MAD) of the portfolio decreases, but at the cost of potentially sacrificing return. The chart represents the tradeoff between return and risk tolerance as we adjust λ.

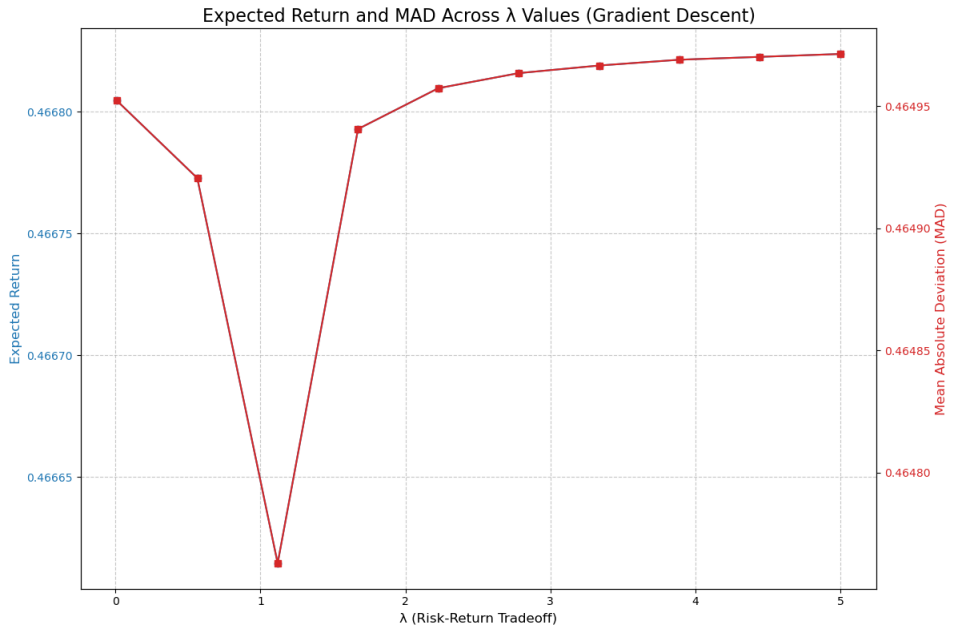

Figure 9: Expected Return and MAD across λ values in the Linearized Portfolio model with Gradient Descent

Figure 10 focuses on the portfolio composition at three different values of λ—one at the lower end, one in the middle, and one at the higher end. We keep all the previous data of the linearized MAD model of portfolio optimization with gradient descent as similar to before except ($\lambda = 0.01, 2.78, 5.00$). Using a stacked bar chart, the portfolio weights for each stock are displayed, showing how they vary as λ changes. At lower values of λ, the portfolio might place heavier weights on higher-risk, higher-return stocks, as the investor is more willing to accept volatility. As λ increases, the portfolio gradually shifts towards stocks with lower risk and more stable returns. This chart provides a clear visual of how adjusting the risk-return tradeoff influences the diversification and allocation within the portfolio. The legend helps track the specific value of λ for each set of portfolio weights.

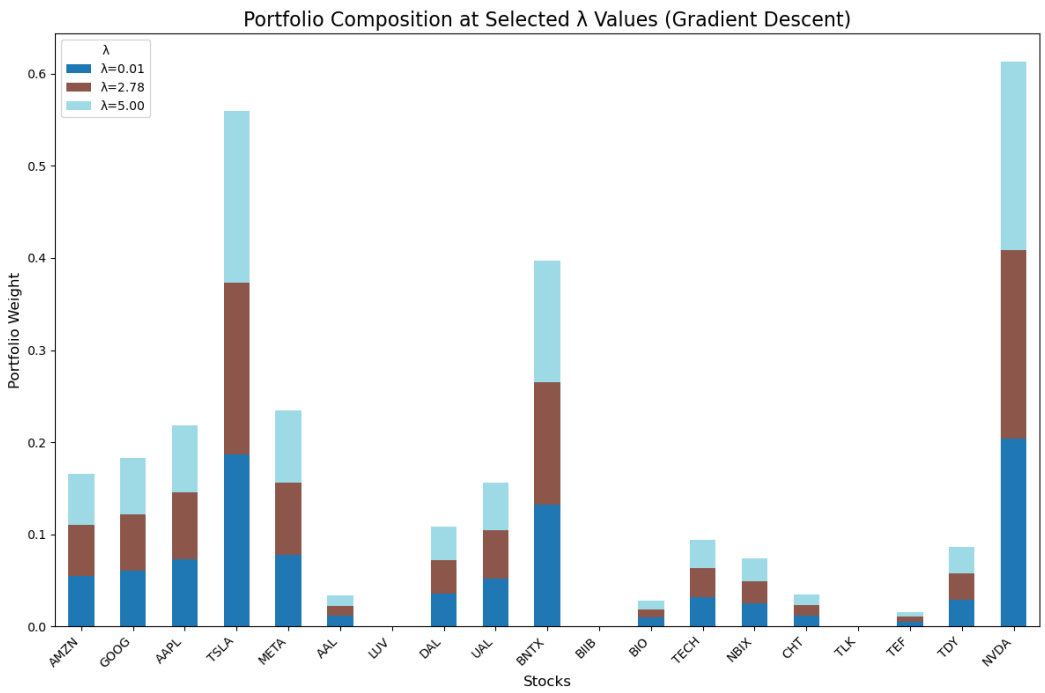

Figure 10: Portfolio Composition at selected λ values in the Linearized Portfolio model with Gradient Descent

6 Result

From the section [5], we can differentiate between the outputs for the non-linear traditional MPT model and the linearized MAD model of portfolio optimization with gradient descent. In our results, we have tried to focus on finding better outputs by comparing the linearized MAD model of portfolio optimization with gradient descent and without gradient descent. In figure 11, the data visualizes for different risk-return tradeoff parameters (λ), how the value of expected return and risk are changing. The near equality of return and risk suggests that the portfolio is finely balanced, with risk aversion dictating the compromise between higher returns and acceptable deviations.

	λ (Risk-Return Tradeoff)	MAD Model Expected Return	MAD Model MAD	GD Model Expected Return	GD Model MAD
0	0.010000	0.019776	0.020000	0.466805	0.464952
1	0.564444	0.019776	0.020000	0.466773	0.464920
2	1.118889	0.016322	0.016819	0.466615	0.464763
3	1.673333	0.007856	0.010262	0.466793	0.464941
4	2.227778	0.005895	0.009220	0.466810	0.464957
5	2.782222	0.004740	0.008758	0.466816	0.464963
6	3.336667	0.004138	0.008562	0.466819	0.464967
7	3.891111	0.003668	0.008430	0.466821	0.464969
8	4.445556	0.003296	0.008340	0.466823	0.464970
9	5.000000	0.003099	0.008298	0.466824	0.464971

Figure 11: Comparison between MAD Linearized model and Gradient Descent MAD model

In our framework, tuning weights are done through the fast response ability of RL in changing market situations using gradient descent, making the model perfectly fit for a real-time situation. Gradient descent is used to help the

RL agents trade-off between high profits and low risk, which is done through the rewards function of the portfolio. Gradient-based methods are efficient in terms of computation, which means RL can fast learn in high-dimensional spaces of assets.

7 Future Directions

In our work, we have introduced a new setup by **improvising MAD to linearize the portfolio model and later added gradient descent to improve the reward over time**. Our framework follows the deterministic policy with the capability of balancing risk and returns by varying the risk-aversion parameter (λ) and adhering to specific risk thresholds (c). To delve more in-depth into this formulation, future work can focus on improving a better bound for c and, at the same time, cover the time-dependent market behavior and real-world uncertainties. Moreover, the application of advanced RL algorithms like policy-gradient methods or actor-critic frameworks can help the portfolio be more adaptable by learning the very best allocation strategies based on market scenarios. The limitation of dependency on historical data can also be addressed with a more potential approach.

8 Conclusion

The proposed integration of the MAD-based linearized portfolio optimization model with RL in adaptive investment strategies is a strong framework. A portfolio weight can be adjusted with the level of risk, the approach allows risk-averse investors to use minimum risk as the threshold. By means of the introduction of restricted gradient descent, one can ensure not only efficient optimization but also the existence of feasibility constraints. The findings suggested that the model has the capability to dynamically change the allocations thereby, it involves both diversification and risk management considerations. This investigation offers one of the most efficient examples of the potential approach of modern portfolio theory by the use of MAD to create a pathway for further exploration in dynamic and real-world financial or economic environments.

9 Acknowledgement

Throughout the research project, we want to extend our sincere gratitude to Dr. Esra Buyuktahtakin Toy from Grado Department of Industrial and Systems Engineering, Virginia Tech for her essential advising contributions to the project construction stage and the idea brainstorming process.

10 Contributions

Asha Barua: Literature Review (From the diagram till the end), Background, Model Formulation, Experimental Setup(Explanation), Future directions, Conclusions, Report Writing in LaTeX.
Hung Tran: Abstract, Introduction, Literature Review (1st two paragraph), Relevant Data Source, Background(Markowitz's Modern Portfolio Theory), Experimental Setup(Coding and Explanation of Figure 7, 9, 10), Presentation Slides.

References

Acero, F., Zehtabi, P., Marchesotti, N., Cashmore, M., Magazzeni, D., and Veloso, M. (2024). Deep reinforcement learning and mean-variance strategies for responsible portfolio optimization. *arXiv preprint arXiv:2403.16667*.

Becker, F., Gürtler, M., and Hibbeln, M. (2015). Markowitz versus Michaud: portfolio optimization strategies reconsidered. *The European Journal of Finance*, 21(4):269–291.

Hambly, B. M., Xu, R., and Yang, H. (2021). Recent Advances in Reinforcement Learning in Finance. *SSRN Electronic Journal*.

Jagwani, J., Gupta, M., Sachdeva, H., and Singhal, A. (2018). Stock price forecasting using data from yahoo finance and analysing seasonal and nonseasonal trend. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 462–467.

Jiang, Z., Xu, D., and Liang, J. (2017). A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. arXiv:1706.10059 [cs, q-fin].

Kolm, P. N. and Ritter, G. (2019). Modern Perspectives on Reinforcement Learning in Finance. *SSRN Electronic Journal*.

Konno, H. and Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management science*, 37(5):519–531.

Korn, R., Korn, E., and Korn, R. (2001). *Option pricing and portfolio optimization: modern methods of financial mathematics*. Number volume 31 in Graduate Studies in Mathematics. American Mathematical Society, Providence, RI.

Lawrence, A., Ryans, J., Sun, E., and Laptev, N. (2018). Earnings announcement promotions: A yahoo finance field experiment. *Journal of Accounting and Economics*, 66(2):399–414.

Mallieswari, R., Palanisamy, V., Senthilnathan, A. T., Gurumurthy, S., Joshua Selvakumar, J., and Pachiyappan, S. (2024). A Stochastic Method for Optimizing Portfolios Using a Combined Monte Carlo and Markowitz Model: Approach on Python. *ECONOMICS*, 12(2):113–127.

Mansini, R., Ogryczak, W., and Speranza, M. G. (2014). Twenty years of linear programming based portfolio optimization. *European Journal of Operational Research*, 234(2):518–535.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1):77.

Markowitz, H. M. (1991). Foundations of Portfolio Theory. *The Journal of Finance*, 46(2):469–477.

Michaud, R. O. and Michaud, R. O., editors (2008). *Efficient asset management: a practical guide to stock portfolio optimization and asset allocation*. Financial Management Association survey and synthesis series. Oxford University Press, New York, 2nd ed edition.

Moore, P. (1972). Mathematical models in portfolio selection. *Journal of the Institute of Actuaries*, 98(2):103–148.

Mortaji, M., Khiat, A., and Benhouad, M. (2024). Reinforcement learning application in portfolio optimization: a comprehensive literature review. In *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–6. IEEE.

Poletaev, A. Y. and Spiridonova, E. M. (2021). Hierarchical Clustering as a Dimension Reduction Technique in the Markowitz Portfolio Optimization Problem. *Automatic Control and Computer Sciences*, 55(7):809–815.

Saksham Jain (2023). Optimizing Portfolio Management using Mean-Variance Optimization in Python. *Innovative Research Thoughts*, 9(5):33–41.

Samsudin, N. (2021). Portfolio Optimization using Reinforcement Learning.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Xia, T. (2023). The Application of Modern Portfolio Theory in US Stock Market with the Use of Python Programming. In Qiu, D., Jiao, Y., and Yeoh, W., editors, *Proceedings of the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022)*, volume 5, pages 152–160. Atlantis Press International BV, Dordrecht. Series Title: Atlantis Highlights in Intelligent Systems.

Xidonas, P., Doukas, H., and Sarmas, E. (2021). A python-based multicriteria portfolio selection DSS. *RAIRO - Operations Research*, 55:S3009–S3034.

Yashaswi, K. (2021). Deep reinforcement learning for portfolio optimization using latent feature state space (lfss) module. *arXiv preprint arXiv:2102.06233*.