Bachelor of Science - Ecole polytechnique

BACHELOR THESIS

Academic year 2024 - 2025

Dynamic Portfolio Optimization Using Machine Learning and Markowitz Theory

Gabin Lassus

Under the supervision of

Gauthier Vermandel

Geoffrey Barrows

1 Abstract

This study proposes a dynamic portfolio optimization framework that integrates Markowitz's Mean-Variance Optimization with machine learning techniques to enhance risk-adjusted returns. We utilize Long Short-Term Memory (LSTM) networks to predict asset returns and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models to estimate time-varying volatility. By integrating these forecasts into a Sharpe ratio-maximizing allocation strategy, we construct a portfolio that dynamically rebalances every quarter to reflect updated risk and return estimates. Our empirical analysis is conducted on the Dow Jones Industrial Average (DJI) stocks from 2005 to 2025, and compares the performance of our dynamic strategy against a static Markowitz portfolio and an equal-weighted benchmark. Our findings suggest that combining machine learning with classical financial models can significantly enhance portfolio optimization, offering a more adaptive and resilient investment strategy in dynamic markets.

Contents

2 Introduction

2.1 Background and Motivation

Portfolio optimization is a fundamental problem in financial economics, aiming to allocate capital in a way that achieves the best possible combination of risk and return. In 1952, Harry Markowitz [1] established the theoretical framework for portfolio optimization, later known as Modern Portfolio Theory (MPT). At the core of MPT lies the idea that investors should not consider investments individually, but rather evaluate their collective risk and return. This approach, commonly referred to as mean-variance optimization, conceptualizes investment decision-making as a trade-off between maximizing expected return and minimizing risk.

However, despite its significant influence, the traditional mean-variance optimization model exhibits certain limitations. For instance, it assumes returns follow a normal distribution, while empirical evidence shows that real-world returns frequently deviate from this assumption, demonstrating features such as fat tails and volatility clustering. Additionally, mean-variance optimization heavily relies on historical data, assuming that past mean returns and covariances remain stable over time. In reality, the dynamic nature of financial markets can lead to unexpected shocks that invalidate past trends.

Consequently, static allocation strategies based on traditional MPT often lack the flexibility needed to respond effectively to sudden market shifts, such as those experienced during the 2008 Global Financial Crisis or the market disruptions caused by the COVID-19 pandemic. To address these issues, extensions to the traditional MPT model have been proposed. By employing econometric techniques like Generalized Autoregressive Conditional Heteroskedasticity (GARCH) to capture time-varying volatility, and advanced machine learning models such as Long Short-Term Memory (LSTM) networks to predict future returns, it becomes possible to enhance both risk management and real-time adaptability to market changes.

2.2 Objective of the Study

The primary objective of this study is to construct and compare two portfolio allocation strategies: a static portfolio based on the classical mean-variance optimization framework proposed by Markowitz, and a dynamically optimized portfolio extending MPT through advanced forecasting techniques. Specifically, this research examines whether integrating volatility forecasts from GARCH models and return predictions from LSTM networks can improve portfolio performance compared to traditional static allocation strategies. The study empirically tests the effectiveness of these approaches by comparing their respective risk-adjusted performance metrics, including cumulative returns, volatility, Sharpe and Sortino ratios, and maximum drawdown.

3 Theoretical Framework

3.1 Markowitz Portfolio Theory

Markowitz's Modern Portfolio Theory [1] provides the following mathematical framework for optimizing asset allocation by balancing risk and return:
Let $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$ denote the vector of portfolio weights, where w_i represents the proportion of capital allocated to asset i. Similarly, let $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^T$ be the vector of expected returns for each asset. The **expected return** of the portfolio, denoted as μ_p, is given by:

$$\mu_p = \mathbf{w}^T \boldsymbol{\mu}$$

where μ_p represents the weighted sum of individual asset returns, determining the overall portfolio return.

While expected return is a key consideration, investors are also concerned with the risk associated with their portfolio. In MPT, **risk** is mathematically measured by variance (σ_p^2), but in practice, it is often expressed as standard deviation (σ_p) because it is easier to interpret. The variance of the portfolio is given by:

$$\sigma_p^2 = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$$

where $\boldsymbol{\Sigma}$ is the **covariance matrix** of asset returns:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}$$

where $\sigma_{ij} = \mathrm{Cov}(r_i, r_j)$ represents the covariance between asset i and asset j. The diagonal elements σ_i^2 correspond to the variance of individual assets, while the off-diagonal elements capture the relationships between asset returns.

Since investors seek the best possible trade-off between return and risk, the goal of mean-variance optimization is to identify the **efficient frontier**, the set of portfolios that achieve the highest expected return for a given level of risk. The **mean-variance optimization problem** is formulated as follows:

$$\min_{\mathbf{w}} \quad \sigma_p = \min_{\mathbf{w}} \quad \sqrt{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}}$$

subject to:

$$\begin{cases} \mathbf{w}^T \boldsymbol{\mu} = \mu_p^* & \text{(Target expected return constraint)} \\ \mathbf{w}^T \mathbf{1} = 1 & \text{(Budget constraint)} \end{cases}$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$ ensures that the sum of all portfolio weights equals 1.

The solution to this problem leads to a parabolic risk-return relationship in mean-variance space, forming the **efficient frontier**.

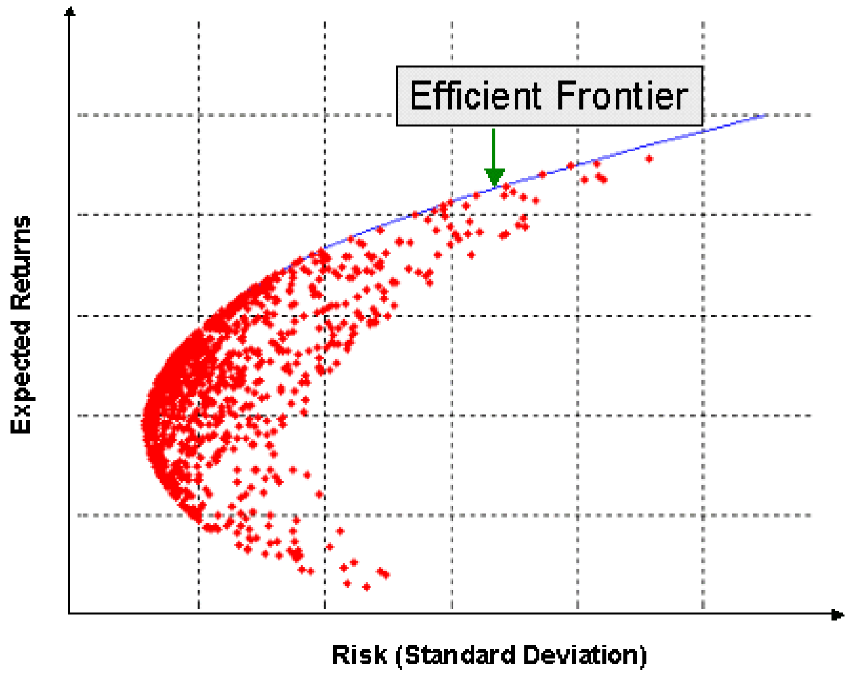

Figure 1: Efficient frontier illustration (image reproduced from FinanceTrain [2])

3.2 Sharpe Ratio

An essential extension to MPT is the **Sharpe ratio**, introduced by Sharpe (1966)[3], which measures the risk-adjusted return of a portfolio. The Sharpe ratio is defined as:

$$S = \frac{\mu_p - r_f}{\sigma_p} \tag{1}$$

where:

- μ_p represents the portfolio's expected return.
- r_f is the **risk-free rate**, typically associated with government bonds or other low-risk assets.
- σ_p denotes the portfolio's standard deviation, measuring overall risk.

where r_f represents the *risk-free rate*, typically associated with government bonds or other low-risk assets. A higher Sharpe ratio indicates a *better trade-off between return and risk*, meaning the portfolio generates more excess return per unit of risk taken.

Since the Sharpe ratio represents risk-adjusted return, the optimal portfolio in mean-variance optimization corresponds to the **tangent portfolio**: the portfolio that maximizes the Sharpe ratio.

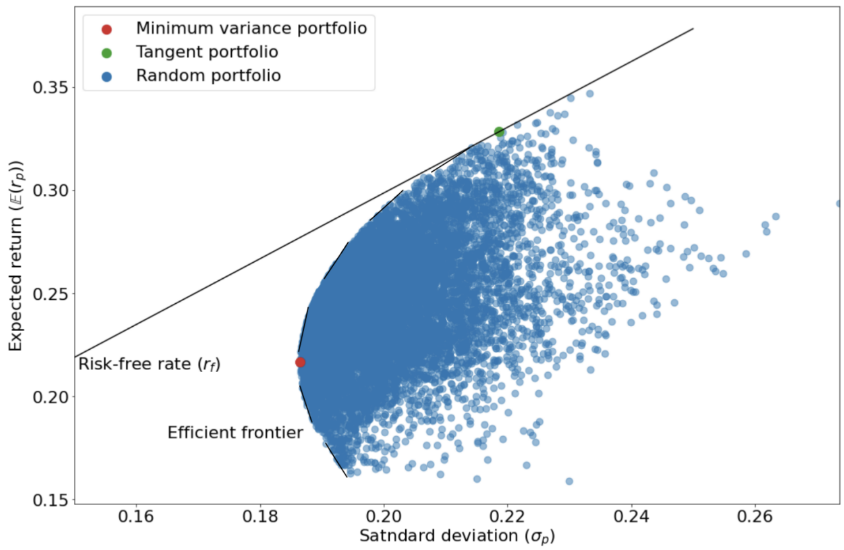

Figure 2: CML and Tangent Portfolio (reproduced from Durall, 2022 [4])

3.3 Sharpe Ratio Maximization

In dynamic portfolio strategies, assets allocations can be adjusted to reflect changing risks and returns. To do this, we determine the optimal portfolio weights that **maximize the Sharpe ratio** at every rebalancing period, ensuring that we always have the best possible trade-off between risk and return. Mathematically, the optimization problem can be formulated as:

$$\max_{\mathbf{w}} \quad \frac{\mathbf{w}^T \hat{\boldsymbol{\mu}} - r_f}{\sqrt{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}}} \tag{2}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{w}^T \mathbf{1} = 1 & \text{(Fully invested portfolio, ensuring all capital is allocated)} \\ \mathbf{w} \geq 0 & \text{(No short-selling, long-only constraint)} \end{cases}$$

where:

- $\mathbf{w}$ represents the portfolio weight vector.
- $\hat{\boldsymbol{\mu}}$ is the vector of expected returns.
- $\boldsymbol{\Sigma}$ is the covariance matrix of asset returns.

By solving this optimization problem, we identify the portfolio allocation that maximizes **risk-adjusted performance**, ensuring that the portfolio adapts to evolving market conditions.

3.4 Sortino Ratio

Finally, while the Sharpe ratio provides a general measure of risk-adjusted return, it treats all volatility—both upward and downward movements—as risk. However, this assumption does not always align with investor preferences, as losses tend to have a greater impact on portfolio performance than equivalent gains.

Therefore, we will also consider the **Sortino ratio**, an alternative risk-adjusted performance measure that only penalizes downside volatility. The Sortino ratio is defined as:

$$S_s = \frac{\mu_p - r_f}{\sigma_d} \tag{3}$$

where σ_d denotes the downside standard deviation, measuring only negative volatility.

A higher Sortino ratio suggests that the portfolio is delivering strong returns while effectively limiting downside risk, making it a more attractive choice for risk-averse investors.

Thus, by incorporating the Sortino ratio, we will assess whether our dynamic rebalancing strategy improves risk-adjusted returns at the cost of excessive downside risk or if it provides a more stable and resilient investment strategy.

3.5 Long Short-Term Memory (LSTM) for Return Prediction

Financial market returns exhibit non-linear, complex patterns that time series forecasting methods often struggle to fully capture. As discussed by Mandelbrot [6], traditional models such as the Autoregressive Integrated Moving Average (ARIMA) often fail because they assume the existence of linear relationships and stationarity, meaning that mean, variance, and covariance remain constant over time. However, real-world markets are highly dynamic and subject to sudden shifts caused by external factors.

To address these challenges, we use Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN) particularly well-suited for processing sequential data and capturing long-term dependencies.

LSTMs were introduced by Hochreiter and Schmidhuber (1997)[5] as a solution to the vanishing gradient problem. In standard RNN architectures, information is passed through multiple layers so that the model can learn sequential relationships. However, as the number of layers increases, gradients shrink exponentially, making it difficult for the network to retain long-term dependencies. This is a major issue for financial return prediction, as past market trends and cycles often influence future movements.

LSTM networks extend RNNs by addressing the challenge of learning long-term dependencies in sequential data. This is achieved through three key components, known as gates:

- Input Gate: Determines what new information to store in memory.
- Forget Gate: Decides what past information to discard or keep.
- Output Gate: Controls what information is used for the next prediction.

This structure allows LSTMs to retain important historical patterns while selectively incorporating new information, mitigating short-term fluctuations. This makes LSTMs particularly effective for financial time series forecasting, where recognizing patterns and adapting to dynamic trends is crucial.

However, LSTM models are not without limitations. They require careful tuning of hyperparameters to avoid overfitting or poor generalization. Additionally, while LSTMs can identify patterns, they are still sensitive to unforeseen shocks, such as macroeconomic shifts or financial crises.

3.6 GARCH Model for Covariance Forecasting

In portfolio optimization, the covariance matrix plays a fundamental role in measuring risk and diversification. By capturing how asset returns move relative to each other, it directly influences portfolio allocation decisions. The **covariance** between two assets i and j at time t is given by:

$$\sigma_{i,j,t} = \rho_{i,j,t} \cdot \sigma_{i,t} \cdot \sigma_{j,t} \tag{4}$$

where:

- $\rho_{i,j,t}$ represents the **correlation** between assets i and j at time t.
- $\sigma_{i,t}$ and $\sigma_{j,t}$ are the **volatilities** of the respective assets at time t.

To estimate time-varying covariance, we decompose the problem into two components:

1. Rolling Correlation Estimation: Continuously updating the correlation matrix to capture evolving relationships between assets.
2. Rolling Volatility Estimation: Modeling fluctuations in asset volatilities over time to better reflect changing market conditions.

3.6.1 Rolling Correlation Estimation

Asset correlations are dynamic, influenced by market trends, economic cycles, and external shocks. A static correlation matrix, computed over a long historical period, assumes fixed relationships between assets. However, correlations evolve due to changing market conditions. A **rolling correlation approach**, where correlations are continuously recomputed using a fixed past window of observations, ensures that the most recent information is incorporated while reducing the impact of short-term noise. For instance, to compute the correlation between two assets at time T, we consider their correlation over the interval $[T - 60, T\ ]$. Using a rolling window of W trading days, the correlation matrix at time t is computed using the past W days of returns. The correlation between two assets is calculated as:

$$\rho_{i,j,t} = \frac{\sum_{k=0}^{W}(r_{i,t-k} - \bar{r}_i)(r_{j,t-k} - \bar{r}_j)}{\sqrt{\sum_{k=0}^{W}(r_{i,t-k} - \bar{r}_i)^2 \sum_{k=0}^{W}(r_{j,t-k} - \bar{r}_j)^2}}$$

where:

- W is the rolling window length
- $r_{i,t}$ and $r_{j,t}$ are the asset returns at time t.
- $\bar{r}_i$ and $\bar{r}_j$ are mean returns over the rolling window.

3.6.2 GARCH Model for Volatility Forecasting

Financial markets exhibit a phenomenon known as volatility clustering, where periods of high volatility tend to be followed by further high volatility, and periods of low volatility persist over extended periods. This challenges the assumption of the Markowitz model that historical estimates of covariance remain stable over time. In reality, markets exhibit heteroskedasticity, a condition where variance fluctuates over time due to factors such as economic cycles or liquidity shocks, and market sentiment shifts. To account for this, Engle (1982) [7] introduced the Autoregressive Conditional Heteroskedasticity (ARCH) model, allowing conditional variance to be dynamically updated based on the past squared returns. However, this model had limitations in capturing long-term volatility persistence, meaning it placed excessive importance on recent observation leading to recency bias. Therefore, to overcome this issue, Bollerslev (1986) [8] extended the ARCH framework, leading to the development of the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model.

3.6.3 GARCH Model Specification

The GARCH(p, q) process is defined as:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i r_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

where:

- σ_t^2 is the conditional variance at time t, representing the forecasted volatility.
- ω is a constant term capturing the long-run average variance.
- α_i represents the influence of past squared returns (ARCH term), reflecting the impact of recent shocks on volatility.
- β_j accounts for the persistence of past volatility (GARCH term), ensuring that high-volatility periods tend to persist.

By integrating GARCH-based volatility forecasts with rolling correlation estimates, we construct a dynamic risk model that adapts to changing market conditions. This allows for more responsive portfolio allocation and risk management decisions.

4 Data Collection & Preprocessing

The dataset we will be using consists of the 30 stocks currently constituting the Dow Jones Industrial Average (DJI) 14. It is composed of large, blue-chip companies across multiple sectors. These stocks provide a diversified universe for our portfolio optimization problem, ensuring both high liquidity and long-term data availability.

We obtain daily adjusted closing prices from Yahoo Finance (yfinance) [9], which accounts for dividends and stock splits. The dataset spans from 2005 to 2025, capturing a variety of market conditions, including major financial crises and recoveries.

However, due to incomplete or missing data, we removed Visa Inc (V) and Walgreens Boots Alliance (WBA) from our dataset, leaving us with 28 stocks for portfolio construction.

4.1 Data Splitting: Training and Testing

To ensure robust model evaluation, we divide the dataset into two periods:

1. Training set (2005–2019): Used for computing historical mean returns and covariance matrices in the Markowitz model, as well as training the LSTM and GARCH models.

2. Testing set (2020–2024): Used for out-of-sample model evaluation, assessing how the models perform on unseen data.

By doing so, we ensure that the training phase spans multiple market cycles, while the test period reflects a dynamic market environment. Training on pre-2020 data and testing on post-2020 data allows us to evaluate how well our models adapt to financial conditions that were not seen during training.

4.2 Data Transformation and Returns Calculation

From Yahoo Finance, we obtain raw adjusted closing prices at the end of each trading day. To transition from raw price data to returns, we compute daily simple returns using:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

where r_t represents the return at time t, and P_t is the adjusted closing price at time t.

4.2.1 Why Use Simple Returns?

Although log returns offer convenient properties such as additivity over time and improved normality assumptions, we will use simple returns due to their practical advantages. Simple returns align with the Markowitz framework where expected returns and covariances are computed using the arithmetic returns. Additionally, this enables us to easily compute the Sharpe Ratio (1), a key metric in risk-adjusted performance evaluation. Furthermore, we will incorporate scaling techniques for our LSTM and GARCH models, eliminating the need for log transformation. Lastly, using simple returns will provide an easy interpretability of the result.

4.3 Data Scaling

To improve model performance and numerical stability, we apply different scaling techniques for the LSTM and GARCH models. These transformations ensure that our models process data effectively while maintaining interpretability.

4.3.1 LSTM Scaling

LSTM models require normalized inputs since they use activation functions such as sigmoid and tanh, which perform optimally within bounded ranges. To ensure stable training, we apply **MinMax Scaling**, standardizing stock returns between $[-1, 1]$:

$$r'_t = \frac{r_t - r_{\min}}{r_{\max} - r_{\min}}$$

where $r_{\min}$ and $r_{\max}$ are the minimum and maximum observed returns in the training dataset. Since portfolio optimization requires actual return values, we revert LSTM-predicted returns back to their original scale before applying them.

4.3.2 GARCH Scaling

The GARCH model, used for volatility estimation, is sensitive to small numerical values. To enhance stability, we convert returns to percentages before inputting them into the model. Once volatility predictions are generated, they are rescaled back to their original units.

These transformations improve numerical precision, ensure consistency across models, and enhance the robustness of our portfolio optimization framework.

5 Model Implementation

5.1 Static Portfolio Optimization

As a baseline, we implement a **static portfolio** optimized using the *Markowitz mean-variance approach*. In this method, asset weights are determined based on *historical mean returns* and the *covariance matrix* computed from the training dataset (2005–2019).

At the beginning of the testing period, we select the optimal portfolio weights by solving the mean-variance optimization problem from earlier.

Once the optimal weights are determined, the **portfolio remains unchanged throughout the testing period (2020–2025)**, serving as a benchmark for comparison against the dynamic optimization approach.

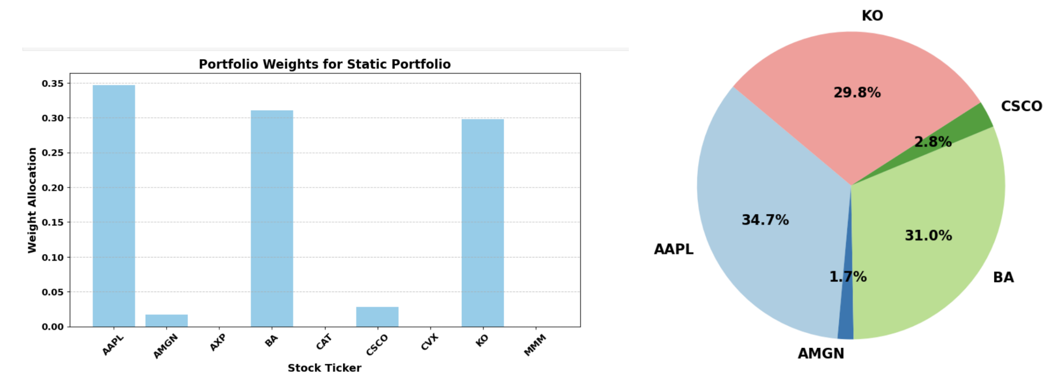

Figure 3: Repartition of weights for the static Markowitz portfolio

5.2 Implementing the LSTM Model

5.2.1 Hyperparameter Optimization for LSTM

The performance of an LSTM model is highly dependent on its hyperparameters. Therefore, to optimize the predictive power of our model, we use Bayesian Optimization with Keras Tuner, an approach that efficiently searches for the best configuration while minimizing computational cost. We optimize the following three parameters:

- **LSTM Units:** Controls the model's capacity to retain information across time steps. More units improve long-term pattern recognition, aiding the input and forget gates.
- **Dropout Rate:** Regularizes the model by randomly deactivating neurons, preventing overfitting and enhancing the forget gate's ability to discard less relevant information.
- **Learning Rate:** Determines how quickly the model updates its weights. A well-calibrated learning rate ensures smooth optimization of gate mechanisms without instability.

We run the tuning on the following range:
– **LSTM Units:** 32, 64, 96, 128

– **Dropout Rate:** 0.1 to 0.5 (step size = 0.1)
– **Learning Rate:** 0.0001, 0.001, 0.01

To obtain the best parameters for our model, we run the tuning process separately for each individual stock. Then, for the continuous parameters (learning rate and dropout rate), we take the average across all stocks, while for the discrete parameter (LSTM units), we take the mode value, which is the most frequently occurring value.

After running the tests, we obtain the following results:

Optimal Hyperparameters for LSTM Model

Hyperparameter	Optimal Value	Determination Method
LSTM Units	107.0	Mean
Dropout Rate	0.11	Mean
Learning Rate	0.01	Mode

Figure 4: Optimal Hyperparameter for our LSTM model obtained after tuning

Hence, our final model configuration consists of a 0.01 learning rate, 107 LSTM units, and a 0.11 dropout rate.

5.2.2 Model Performance: Predicted vs. Actual Returns

To assess the predictive power of our model, we compare its predicted returns against both the realized returns from the test period and the rolling mean of the past 60 trading days. The graph below showcases the differences between these three methods over the test period (2020 to 2025) for two representative stocks: Boeing Co. and 3M Co.

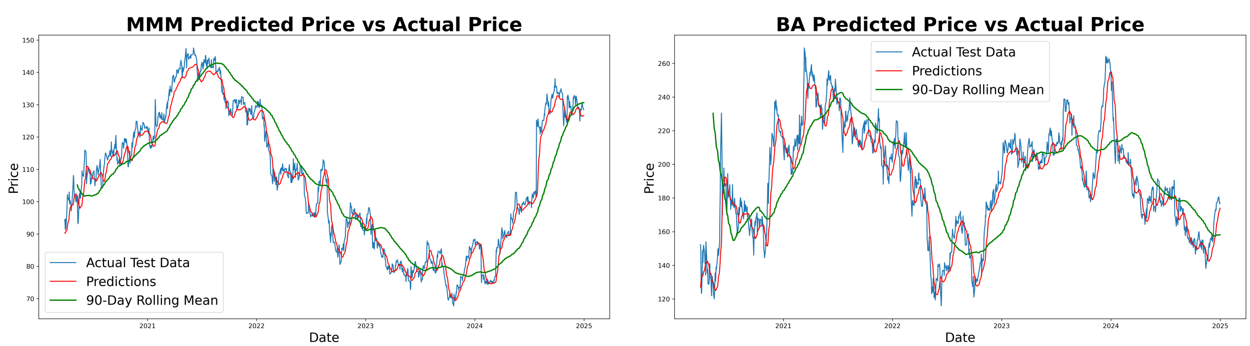

Figure 5: Actual vs Predict return by GARCH model

This visualization illustrates that our LSTM predictions provide a more dynamic alternative to traditional rolling averages but remain imperfect. Compared to the rolling mean, which smooths out short-term variations, the LSTM model is more reactive yet still exhibits lags and occasional deviations. Additionally, the model struggles during periods of rapid market movements, failing to capture sudden price changes effectively. While it outperforms the rolling mean in responsiveness, further improvements are needed to enhance accuracy during volatile conditions.

5.3 GARCH Model for Volatility Forecasting

To implement the GARCH model, we tested various combinations of parameters p and q ranging from 1 to 5 on each asset. To determine the best parameter combination, we assessed different models using the following error metrics:

- **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted volatility.
- **Root Mean Squared Error (RMSE):** The square root of MSE, which retains the original units of volatility.
- **Mean Absolute Error (MAE):** Represents the absolute difference between predicted and actual values, reducing the impact of large outliers.

Based on these evaluations, the GARCH(1,2) model ($p = 1, q = 2$) exhibited the lowest forecasting errors across multiple assets, making it the optimal choice.

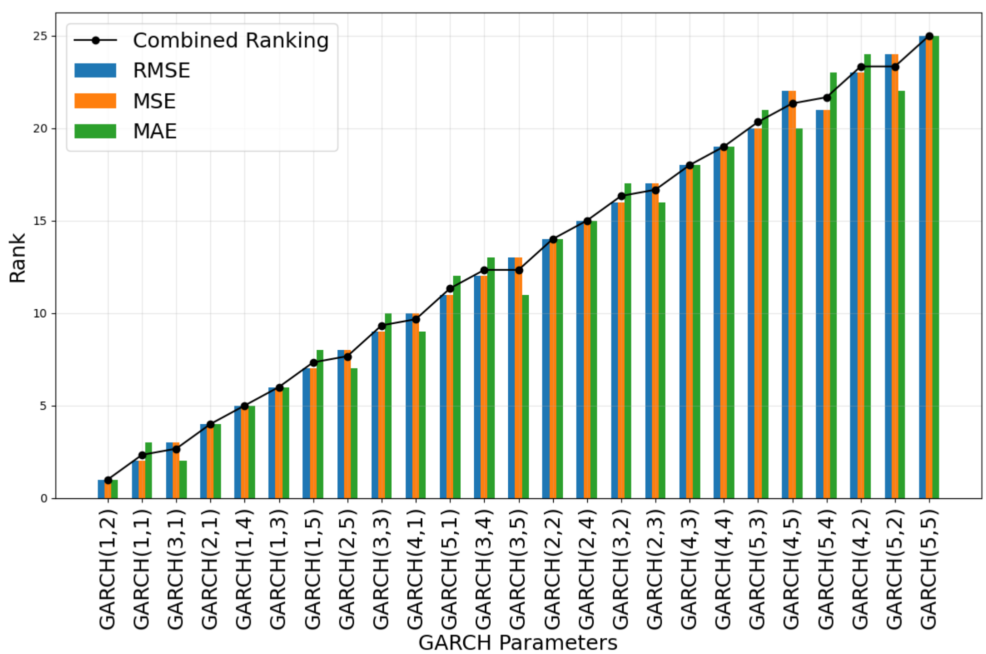

Figure 6: Ranking of GARCH Parameters Across Different Metrics

Therefore, we employ a GARCH(1,2) model to estimate time-varying volatility, which is given by:

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2$$

where:

- σ_t^2 is the conditional variance at time t.
- ω represents the long-run average volatility.
- α_1 captures the impact of past shocks (ARCH effect).
- β_1, β_2 represent the persistence of past volatility (GARCH effect).

As we can see in the figure below, our GARCH(1,2) model effectively captures volatility clustering, where periods of high volatility are followed by high volatility and vice versa. And while our model does not predict the full amplitude of the volatility peaks, it is generally able to capture the sudden shocks which are the source of uncertainty.

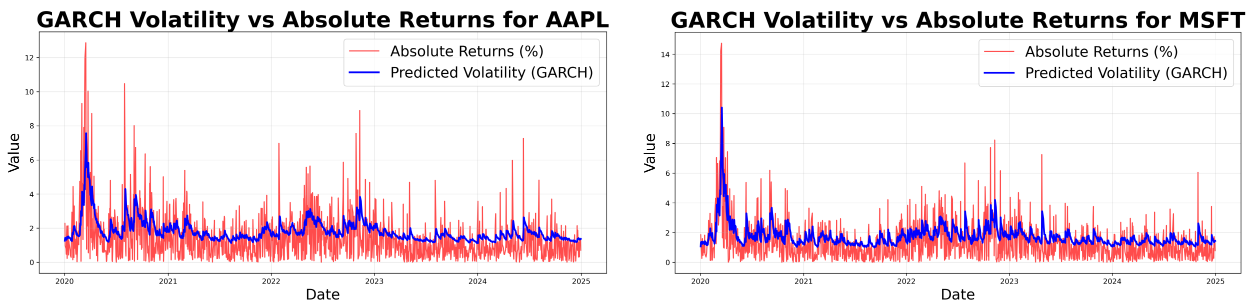

Figure 7: Actual vs Predict return by GARCH model for Apple and Microsoft stocks

5.3.1 Rolling Correlation and Dynamic Covariance Matrix

Next, we compute the rolling correlation matrices to ensure that our covariance matrix remains adaptive to market changes. Since we will later use the dynamic variance matrix to update the portfolio quarterly, we select a 60-day rolling window to compute our rolling correlation estimates.

This choice aligns with the quarterly rebalancing schedule (discussed in Section 5.4), as a typical quarter consists of approximately 63 trading days. Using a rolling window slightly shorter than the rebalancing period allows us to capture the most recent correlation patterns without excessive lag while maintaining consistency with portfolio adjustments.

Beyond the practicality, a 60 day rolling window is a common practice for correlation estimation in portfolio optimization (Fama & French (2018) [10]), but also risk management ((J.P. Morgan, 1996) [11]).

Using a shorter window could lead our model to overreact to temporary shocks, introducing unnecessary noise. On the other hand, selecting a longer rolling window could make the model less responsive to market changes due to its reliance on outdated information.

Finally, we can combine the rolling correlation with GARCH-based volatility forecasts to construct the time-varying covariance matrix. By doing this, we now have a covariance matrix that adapts in real-time, to market fluctuations, rather than relying on outdated historical estimates. Overall, by incorporating both GARCH volatility forecasting and rolling correlation adjustments, we obtain a more adaptive risk model that improves portfolio rebalancing decisions over time.

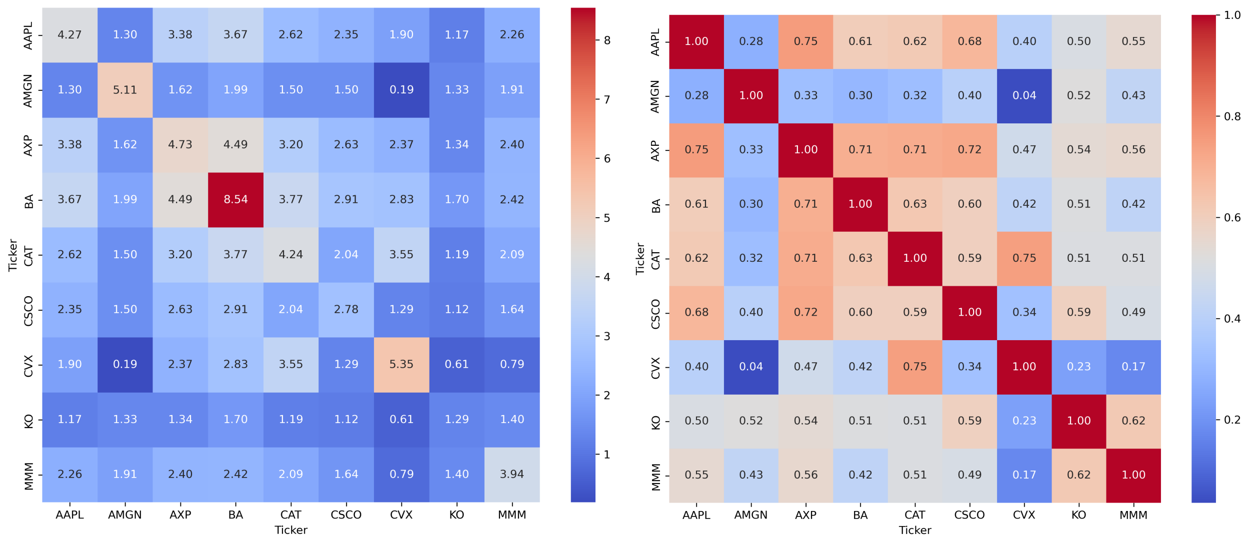

Figure 8: Dynamic covariance (left) and rolling correlation matrices (right) for a subset of stocks at the date 10/12/2022

5.4 Dynamic Portfolio Optimization

5.4.1 Portfolio Rebalancing Strategy

Now that we have constructed a dynamic covariance matrix and can predict future returns, we implement a dynamic portfolio strategy. By adjusting asset allocations at regular intervals, we ensure that the portfolio remains optimized based on current risk and expected return estimates, rather than relying on historical averages that may not reflect evolving market conditions.

To balance adaptability and trading efficiency, we adopt a **quarterly rebalancing strategy** (every 63 trading days). This frequency strikes a balance between capturing changes in asset relationships and avoiding excessive turnover costs, as explained in Donohue & Yip (2003) [12]. This paper showed that quarterly rebalancing is the optimal trade-off between minimizing transaction costs and maintaining an allocation close to the target portfolio. Choosing a more frequent rebalancing strategy would increase costs without significantly improving returns, while on the other hand, less frequent adjustments could lead to excessive allocation shifts.

5.4.2 Optimizing Portfolio Allocation: Maximum Sharpe Ratio

At each rebalancing period, we optimize portfolio weights to **maximize the Sharpe Ratio** as defined in (2), ensuring the highest possible return per unit of risk. The optimal problem is defined as:

$$\max_{\mathbf{w}} \quad \frac{\mathbf{w}^T \hat{\boldsymbol{\mu}} - r_f}{\sqrt{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}}}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{w}^T \mathbf{1} = 1 \\ \mathbf{w} \geq 0 \end{cases}$$

where:

- $\hat{\boldsymbol{\mu}}$ is the **quarterly-averaged LSTM return forecast**.
- $\boldsymbol{\Sigma}$ is the **quarterly-averaged GARCH covariance matrix**.

5.5 Quarterly Averaging of LSTM and GARCH Predictions

Since LSTM forecasts only **one-step-ahead** returns, using a single daily prediction for portfolio rebalancing would introduce excessive noise and instability. Instead, we take the **quarterly average** of LSTM return predictions to obtain a more reliable estimate. So, let [t;t+63] represent trading quarter time interval we want to estimate, then the average LSTM return prediction is:

$$\hat{\mu}_{\text{quarter}}^{\text{LSTM}} = \frac{1}{63} \sum_{i=1}^{63} \hat{\mu}_{t+i}^{\text{LSTM}}$$

Similarly, for the **quarterly volatility estimation**, we take the average of GARCH-predicted volatilities over the same period:

$$\hat{\sigma}_{i,\text{quarter}} = \frac{1}{63} \sum_{t=1}^{63} \sigma_{i,t}^{\text{GARCH}}$$

Finally, we compute the **quarterly covariance matrix** by combining the rolling correlation estimates with the averaged volatilities using (4):

$$\hat{\sigma}_{i,j,\text{quarter}} = \hat{\rho}_{i,j,\text{quarter}} \cdot \hat{\sigma}_{i,\text{quarter}} \cdot \hat{\sigma}_{j,\text{quarter}}$$

This formulation ensures that the covariance matrix reflects the most recent correlation patterns while smoothing short-term fluctuations in volatility, leading to a more stable and adaptive risk estimation for portfolio optimization.

6 Performance Comparison and Evaluation

To analyze the results of the dynamic implementation, we compare the performance of the following three portfolios over different metrics:

- **Equal-Weighted Portfolio**: A simple approach where each stock in the portfolio receives an equal weight (approximately 3.5% per stock). This serves as a fundamental benchmark that closely follows market movements and has historically yielded returns comparable to major indices such as the S&P 500 16. a simple approach where each stock in the portfolio receives an equal weight (around 3.5
- **Static Portfolio (Markowitz MPT)**: This portfolio is constructed using the Markowitz Mean-Variance Optimization. At the beginning of the test period, we determine the optimal weights for each stock and keep it untouched for the rest.
- **Dynamic Portfolio (LSTM/GARCH)**: The portfolio we constructed using the LSTM and GARCH prediction to quarterly optimize the weight .

6.1 Cumulative Return Comparison

First, we can graph the cumulative return of the different portfolio over time:

Figure 9: Cumulative return comparison for diffrent portfolios

In the graph above, we see that the Dynamic Portfolio (represented by the red line) consistently outperforms both the Static Portfolio (blue) and the Equal-Weighted Portfolio (green) in terms of return over the entire period. However, we also notice that it introduces higher short-term fluctuations compared to the other models. In comparison, it seems the equal weight portfolio provides a more steady growth while the Static portfolio stands somewhere in the middle. We will therefore explore further if these observation are true.

6.2 Sharpe and Sortino Ratio

For each portfolio, we compute the Sharpe (1)and Sortino (3) ratios and get the following results:

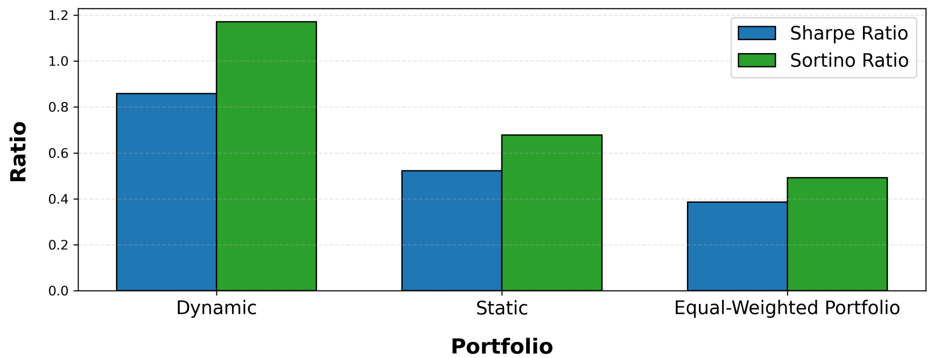

Figure 10: Sharpe and Sortino Ratios for Different Portfolios

Furthermore, we can also track the evolution of the Sharpe and Sortino Ratios over the test period. To do so, we compute rolling Sharpe and Sortino ratios using a quarterly window, allowing us to observe how the risk-adjusted performance of each of the three portfolios evolves over time.

Figure 11: Rolling Sharpe Ratio (Quarterly)

Looking at the results, the Sharpe and Sortino ratio trends illustrate the inherent trade-offs in the three strategies. The dynamic portfolio, while showing the highest performance for risk-adjusted return (higher Sharpe and Sortino ratios), does not always provide a consistently smooth return profile, as seen in periods of sharp declines. In comparison, the static portfolio is less volatile but struggles to consistently match the dynamic portfolio's peaks. Lastly, we see that the equal-weighted portfolio performs the worst of the three since it consistently underperforms in risk-adjusted return and also fails to provide a more stable alternative in that regard.

6.3 Risk-Return Tradeoff: Efficient Frontier

Now, we can plot the efficient frontier to visually and compare the risk-return trade-offs of our portfolios. Since the efficient frontier represents the set of optimal portfolios that offer the highest expected return for a given level of risk, we can assess how well each method balances return and volatility by positioning the static, equal-weighted, and dynamic portfolios alongside the frontier. To help us compare our results, generate 10 000 arbitrary portfolios, by assigning randomly weights and using the Mean Returns and Covariance from the test data to obtain their volatility and returns.

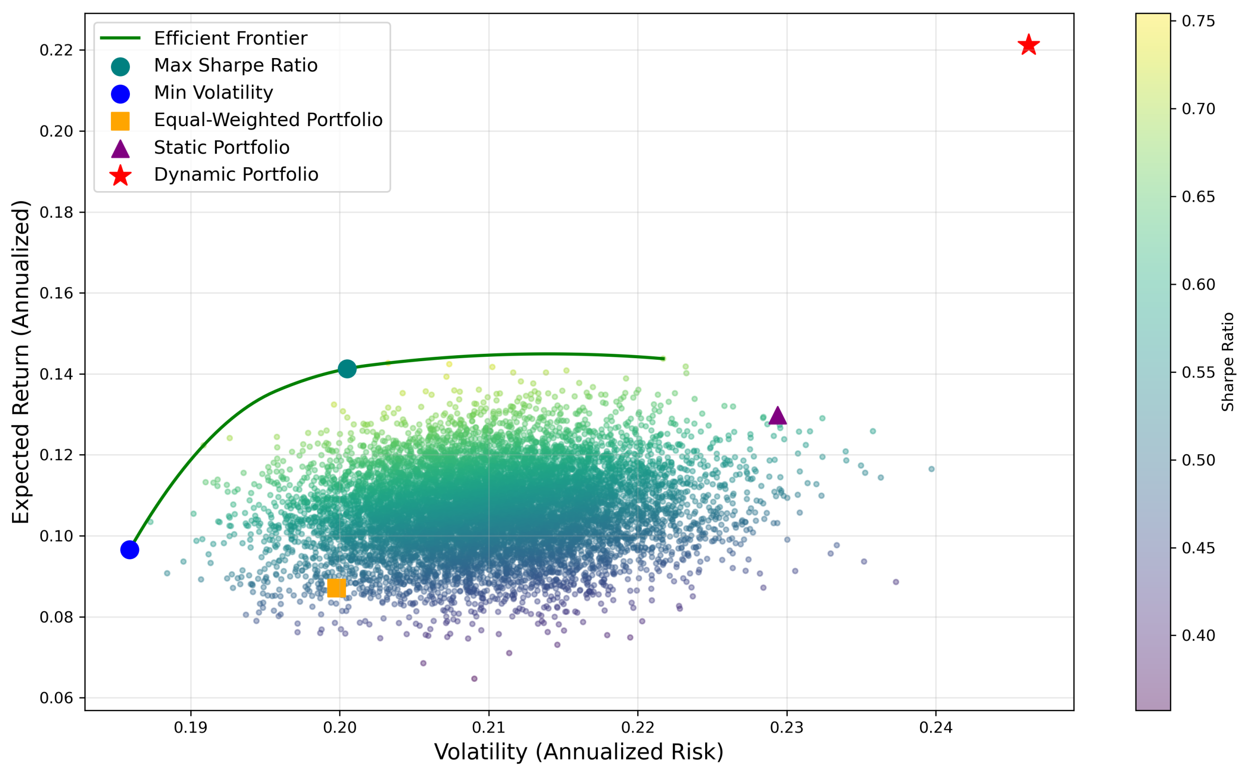

Figure 12: Efficient frontier and portfolio performance

This visualization of the efficient frontier gives us a founded benchmark for assessing the efficiency of our different portfolio strategies. As we can see, the static portfolio lies just below the frontier, suggesting a fairly balanced trade-off between risk and return, the dynamic portfolio deviates significantly from the frontier. While it does achieve the highest return of all portfolios, this deviation indicates that the dynamic approach enhances returns at the cost of higher risk, making it less aligned with mean-variance efficiency. Indeed, since adaptive rebalancing captures short-term opportunities, it also increases volatility and drawdown exposure, particularly in turbulent markets.

Lastly, the equal-weighted portfolio is positioned below the efficient frontier which reinforces the idea that diversifying naively, without any optimization, leads to suboptimal risk-adjusted returns.

6.4 Summary of Performance Metrics

	Total Return	Annualized Return	Volatility	Sharpe Ratio	Sortino Ratio	Max Drawdown
Static Portfolio	83.84%	12.98%	22.94%	0.52	0.68	-21.85%
Equal-Weighted Portfolio	51.68%	8.71%	**19.98%**	0.39	0.49	**-20.81%**
Dynamic Portfolio	**129.80%**	**22.13%**	24.62%	**0.86**	**1.17**	-30.31%

Figure 13: Performance Comparaison of the models over various performance metrics

The table above highlights the importance of portfolio adaptability and optimization. The dynamic model's ability to adjust the portfolio's weights regularly based on LSTM return predictions and the dynamic covariance built by the GARCH volatility estimates allows it to capture short-term opportunities, driving its performance upward. With a total return of **129.80%** and an annualized return of **22.13%**, it significantly outperforms both the static portfolio (**83.84%** total return) and the equal-weighted portfolio (**51.68%** total return). However, this comes at the cost of higher volatility (**24.62%**) and a maximum drawdown of **-30.31%**, indicating that frequent rebalancing can also introduce risk, especially in turbulent market conditions.

The static portfolio, while not as reactive as the dynamic model, maintains a reasonable balance between return and risk. As seen in figure 3 of section (5.3), the portfolio likely suffers from a lack of asset diversification with only five out of the 28 possible stocks bought. . This is reflected in its volatility of **22.94%**, which, while slightly lower than the dynamic portfolio, is still considerably high. In terms of risk-adjusted returns, it achieves a Sharpe ratio of **0.52** and a Sortino ratio of **0.68**, performing better than the equal-weighted portfolio but falling short of the dynamic portfolio.

The equal-weighted portfolio, although it showcases the lowest volatility and maximum drawdown (**19.98%** and **-20.81%**, respectively), performs the weakest in terms of risk-adjusted returns, underscoring the importance of asset weighting strategies to optimize returns.

Ultimately, this showcases the fact that no single strategy is universally superior, but that the choice of strategy should align with an investor's financial goals and market outlook.

7 Limitations and Extensions

7.1 Limitations

Overall, we see that the integration of the LSTM return prediction and Garch volatility estimates into a dynamic optimization framework allows us to outperform both the static and equal weighted portfolio in terms of cumulative return and risk adjusted performance. However, this ability to react to market shifts comes at the cost of greater risk exposure, and increases the potential source of error. Furthermore, while Mean-Variance Optimization have limitations as covered in the Introduction, our dynamic model also comes with some weaknesses. Firstly, while the LSTM model captures the non linear relationship in stock returns, their predictions still contain noise and are highly vulnerable to overfitting during periods of high volatility.In addition, the GARCH model assumes that volatility evolves in an autoregressive manner, meaning that past volatility influences future volatility in a predictable way. However, this is not always the case in financial markets, causing the model to struggle to predict unexpected, non-recurring risks. Finally, we assume frictionless trading in our dynamic framework. This means that we do not account for transaction fees and market impact, which must be factors into the optimization process to make the model realistic for real world applications.

7.2 Extensions and Future Work

To address these limitations, we propose some possible extensions to the project.

First and foremost, a solution to deal with our unrealistic assumption of frictionless trading would be to extend our dynamic model to incorporate transaction costs directly into the optimization framework. By doing this, we would ensure that our models do not make unnecessary trades that offer marginal improvements in performance but lead to relatively high fees. This would make the portfolio optimization process more realistic and practical for real-world applications.

Similarly, we could also introduce a cap on portfolio rebalancing to limit turnover to a certain level (10-20 percent) for instance). This would prevent massive shifts in our weight allocations, in turn reducing the risk associated with high-turnover strategies. Additionally, this would mitigate the issue that large portfolio rebalancing can have an impact on the market, making the model more suitable for large investors who need to carefully manage market impact. However, it must be said that these two extensions would also come with challenges, such as the model potentially underreacting to market changes, or creating a dependence on assumptions regarding trading costs.

Next, while backtesting does provide valuable insights, it is limited to historical data. This makes it difficult to assess the performance of our models during rare and extreme stress scenarios. Therefore, another possible extension to the project could be to apply Monte Carlo simulation to generate hypothetical market scenarios that evaluate our model under various market conditions (bull market, bear market, financial crises, etc.). This would provide a more robust assessment of the model's adaptability and risk management capabilities.

Finally, we could also try to account for macroeconomic variables in our models to improve the predictive power of our LSTM model. By incorporating factors such as interest rates, inflation rates, and GDP growth, we could enhance the model's ability to capture broader economic trends and refine return predictions, leading to more informed portfolio allocation decisions.

8 Conclusion

Overall, our work highlights the potential of machine learning-powered portfolio optimization as an alternative to static, rule-based allocation strategies. While traditional finance such as the Markowitz model relies on backward-looking models, AI-driven methods offer a way to predict market behaviors and adjust portfolio weights in real-time. By refining these models and addressing practical limitations, machine learning has the potential to reshape the way investors approach risk and return trade-offs.

The application of deep learning in portfolio management is still in its early stages. According to a report by Global Market Insights, the global AI in asset management market is projected to grow from \$5.75 billion in 2025 to approximately \$38.94 billion by 2034 [14]. As these technologies become more integrated into financial decision-making, ensuring transparency and accountability will be crucial. Ethical considerations such as maintaining human oversight will play a key role in fostering trust and adoption of these models.

Consequently, it is absolutely essential to understand these dynamics, as the future of investing will not solely rely on automation, but also on the seamless collaboration between data-driven intelligence and financial judgment.

9 Bibliography

References

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

[2] Finance Train. (n.d.). The Minimum Variance Frontier & Efficient Frontier. Retrieved March 2025, from `https://financetrain.com/the-minimum-variance-frontier-efficient-frontier`

[3] Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business*

[4] Durall, R. (2022). Asset Allocation: From Markowitz to Deep Reinforcement Learning. *arXiv preprint arXiv:2208.07158*

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*

[6] Mandelbrot, B. B. (1963). The Variation of Certain Speculative Prices. *The Journal of Business*

[7] Engle, R. F. (1982). Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*

[8] Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*

[9] Yahoo Finance. (2024). Market Data and Historical Prices. Retrieved from `https://finance.yahoo.com`

[10] Fama, E. F., & French, K. R. (2018). Choosing Factors. *Journal of Financial Economics*

[11] J.P. Morgan. (1996). *RiskMetrics Technical Document.*

[12] Donohue, C., & Yip, K. (2003). Optimal Portfolio Rebalancing with Transaction Costs. *The Journal of Portfolio Management*

[13] Fischer, T., & Krauss, C. (2018). Deep Learning With Long Short-Term Memory Networks for Financial Market Predictions. *European Journal of Operational Research*

[14] Precedence Research. (2025). AI in Asset Management Market Size to Hit USD 38.94 Billion by 2034.

[15] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neurala Information Processing Systems*

[16] Guidolin, M., Panzeri, G., & Pedio, M. (2024). Machine Learning in Portfolio Decisions. *BAFFI CAREFIN Centre Research Paper No. 233.*

[17] Garlappi, L., & Skoulakis, G. (2019). Mean-Variance Optimization Is a Good Choice, but for Other Reasons than You Might Think. *SSRN Electronic Journal*

[18] Sood, S., Papasotiriou, K., Vaiciulis, M., & Balch, T. (2023). Deep Reinforcement Learning for Optimal Portfolio Allocation: A Comparative Study with Mean-Variance Optimization. *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS 2023).*

[19] Smith, D. M., & Desormeau, W. H. Jr. (2006). Optimal Rebalancing Frequency for Stock-Bond Portfolios. *SSRN Electronic Journal*

10 Appendix

Ticker	Company Name	Sector
MMM	3M	Industrials
AXP	American Express	Financials
AMGN	Amgen	Health Care
AAPL	Apple	Technology
BA	Boeing	Industrials
CAT	Caterpillar	Industrials
CVX	Chevron	Energy
CSCO	Cisco	Technology
KO	Coca-Cola	Consumer Staples
DOW	Dow	Materials
GS	Goldman Sachs	Financials
HD	Home Depot	Consumer Discretionary
HON	Honeywell	Industrials
IBM	IBM	Information Technology
INTC	Intel	Information Technology
JNJ	Johnson & Johnson	Health Care
JPM	JPMorgan	Financials
MCD	McDonald's	Consumer Discretionary
MRK	Merck	Health Care
MSFT	Microsoft	Technology
NKE	Nike	Consumer Cyclical
PFE	Pfizer	Health Care
PG	Procter & Gamble	Consumer Staples
CRM	Salesforce	Technology
TRV	Travelers	Financials
UNH	UnitedHealth	Health Care
VZ	Verizon	Communication Services
WMT	Walmart	Consumer Defensive
DIS	Walt Disney	Communication Services

Figure 14: List of the stocks used in the experiment and their respective industry

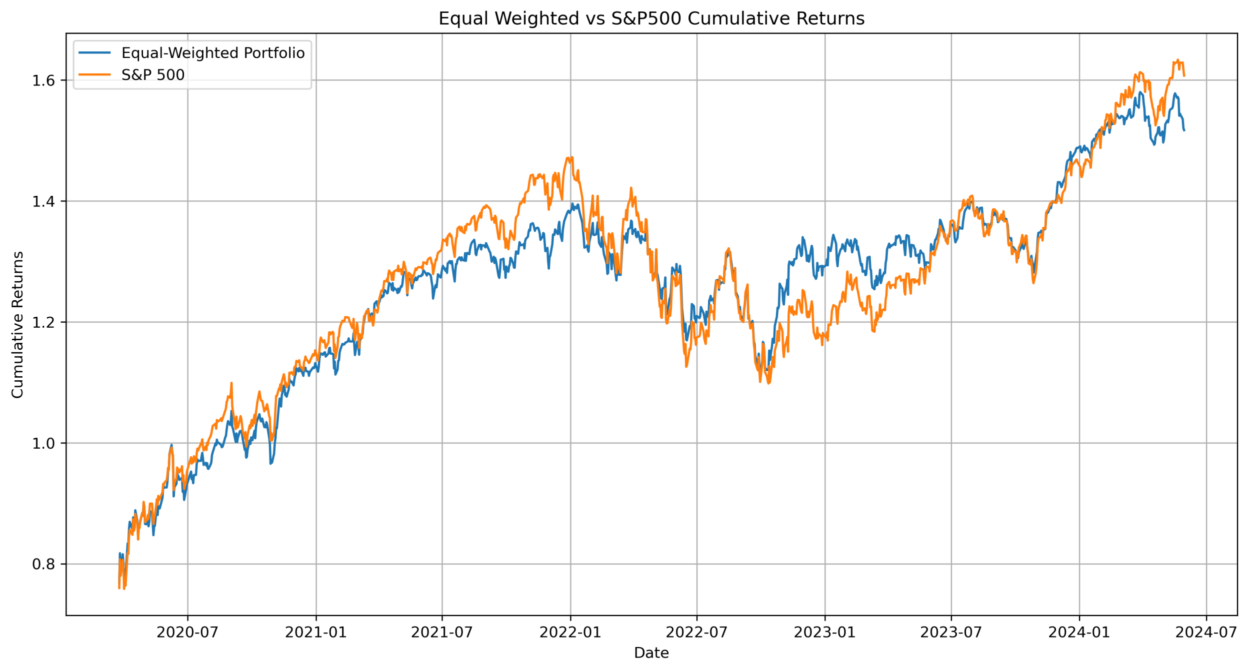

Figure 16: Cumulative return Comparaison of Equal Weighted Portfolio and S&P500

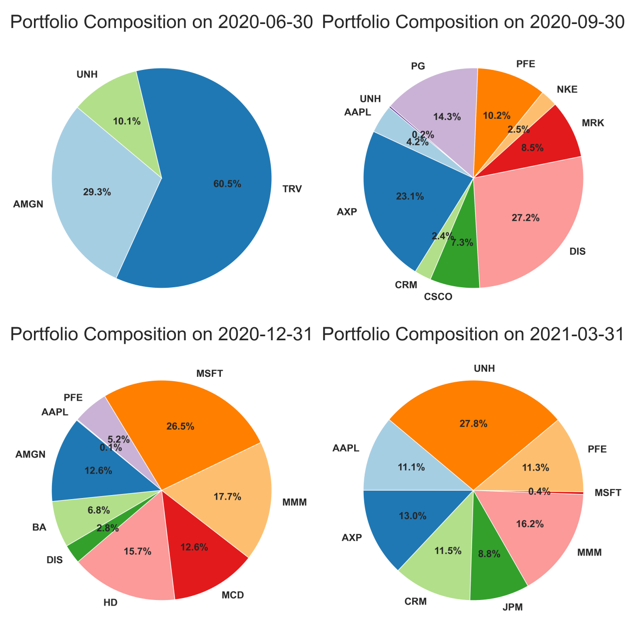

Figure 15: Portfolio Allocations Over Different Quarters

Statement of Academic Integrity Regarding Plagiarism

I, the undersigned ...Lassus Gabin.. [family name, given name(s)], hereby certify on my honor that:

1. The results presented in this report are the product of my own work.

2. I am the original creator of this report.

3. I have not used sources or results from third parties without clearly stating thus and referencing them according to the recommended rules for providing bibliographic information.

Declaration to be copied below:

I hereby declare that this work contains no plagiarized material.

Date 15 / 03 / 2025

Signature Gabin Lassus