Integrating Machine Learning with Black-Litterman for Hybrid Crypto-Stock Portfolios

Ayoola Lawal, Rith Pansanga

WorldQuant University
lawalayoola132@gmail.com, rithpansanga@gmail.com

Abstract

This capstone project investigates and compares the effectiveness of Deep Learning (DL) and Machine Learning (ML) based approaches to portfolio optimization with two traditional portfolio optimization methods: Mean-Variance and Black-Litterman models for cryptocurrencies and stocks. Traditional portfolio optimization methods are known for their reliance on rigid assumptions about asset return distributions and correlations, static market relationships, therefore makes them less effective in today's volatile and data-rich markets. We develop 10 predictive DL and ML models: MLP, LSTM, CNN, XGBoost, Transformer, Decision Tree, SVR, Linear Regression, Ridge Regression, and Ensemble Learning, and evaluate their performance to determine whether they can address the inefficiencies in the traditional methods. Historical daily returns of two crypto assets and nine selected U.S. stocks were obtained from Yahoo Finance from 2018 to 2024 and used to forecast asset behaviors. Performance was evaluated through a back-testing framework under realistic trading constraints, using Annualized returns: Cumulative Return, Sharpe Ratio, Sortino Ratio, and maximum drawdown. We expect ML/DL-based strategies to outperform the traditional models, most especially under volatile market conditions.

Keywords: *Portfolio Management, Portfolio Optimization, Deep Learning for Finance, Machine Learning for Finance*

1. Introduction

The rise of cryptocurrencies alongside traditional stocks presents new challenges for portfolio optimization. The capability and robustness of traditional portfolio models such as Mean-Variance and Black-Litterman are limited in today's fast and dynamic financial markets, hence, we need portfolio optimization to be more adaptive and intelligent in the current world to optimize strategies for investment. This study explores portfolio optimization using ML and DL models to determine whether these advanced models can outperform traditional models. We aim to integrate ten predictive models, including MLP, LSTM, CNN, XGBoost, Transformer, Decision Tree, SVR, Linear Regression, Ridge Regression, and ensemble learning. We will evaluate these models by using daily returns data from two cryptocurrencies and ten stocks from 2018 to 2024.

1.1 Overview

We develop a portfolio optimization engine for handling diverse assets such as cryptocurrencies and stocks in a portfolio. Our engine combines predictive models for asset returns and assets allocation using traditional methods.

This progress has been documented in the following GitHub public repository:

Link to GitHub

1.2 Current Challenges in Financial Engineering

There are several emerging challenges facing portfolio optimization in the finance world today, many of which researchers have yet to find lasting solutions to, some of which are highlighted below:

- **Data Quality:** Financial data can be noisy, incomplete, and inconsistent, especially for cryptocurrencies where standardization is limited.
- **Limitations in dynamic markets using only traditional portfolio optimization methods:** Traditional methods assume stable market conditions. However, markets are highly dynamic, especially in the crypto market; this study addresses this by using machine learning to predict asset returns, which are then used as 'views' in the Black-Litterman model for asset allocation in real time.
- **Market Opening Hours and Data Alignment:** Cryptocurrencies and stocks can be challenging in terms of data alignment. Equity markets open within fixed hours, while cryptocurrency markets are active 24 hours/7 days. These differences in frequency of data and time zones need to be carefully managed, otherwise, it can lead to biased model outputs.
- **Execution Constraints:** In real-world implementation, we must consider rebalancing frequency.
- **Modeling-related Issues:** Machine learning models are often prone to overfitting, similarly, the lack of interpretability remains a common ML pitfall.

1.3 Expected Outcomes

After completing this project, the following outcomes are expected to be achieved:

- **Benchmarking of Traditional vs ML/DL Models**: This study aims to determine whether machine-learning-based asset prediction and allocation models perform better than traditional or classical portfolio methods – Black Litterman and Markowitz's Mean-Variance models.
- **Cross-asset generalizability** by validating performance across two different asset classes – cryptocurrencies and stocks. This study expects to understand whether a machine learning framework can adapt effectively to the statistical properties and trading behaviors of each market.
- **Generate strategic insights under market regimes:** identify which modelling approach performs best under:
    - Stable markets vs volatile markets: This study expects to identify which models remain robust during market stress and those that overfit to stability
    - High vs low correlation environments: This study expects to understand which architectures are effective in exploiting diversification
- Deliverable: A modular, reusable prediction and allocation engine for financial applications.

1.4 Other Potential Applications

The developed engine offers wide applicability for multifarious situations in real-world scenarios, including:

- Cryptocurrencies and Stocks trading strategies with portfolio optimization
    - Our engine can support traders in optimizing portfolios of cryptocurrencies and stock assets as we integrate machine learning and deep learning with

traditional portfolio optimization methods. Our engine can help traders dynamically adjust asset allocations for profitable investments.

- Back testing environment for educational purposes
    - Our engine can provide environment for traders or students to simulate their portfolio by using our engine. It will help users to understand the impact of different asset allocation approaches and risk management.

2. Literature Review

Numerous approaches have been developed by researchers and scholars with a focus on enhancing portfolio optimization for portfolio management. Through a review and combination of insights from previous studies in Machine Learning and Deep Learning for portfolio optimization, knowledge gaps can be identified, and we can conduct competitor analysis. We have limited our focus to recent studies and publications to ensure the foundation of this study is laid on up-to-date knowledge relevant to today's financial market.

Within the scope of the main research topic - *Integrating Machine Learning with Black-Litterman for Hybrid Crypto-Stock Portfolios* - the literature review is conducted and structured using themes. It begins by exploring the application of Machine Learning and Deep Learning in Portfolio Optimization, the use of Deep Learning and Machine Learning in Volatile Markets, the use of Deep Learning in Equity and Multi-Asset Markets, and then discusses the key challenges and gaps in prior studies that motivated this study.

2.1 Machine Learning and Deep Learning in Portfolio Optimization

The deployment of methods such as deep learning and machine learning in finance has acquired substantial popularity throughout the years, which could be credited to the rise in abundance of economic data and progress in computer generation capacity.

We choose to focus on the articles that explicitly connect to the specific models under consideration: MLP, LSTM, CNN, XGBoost, Transformer, Decision Tree, SVR, Linear Regression, Ridge Regression, and Ensemble Learning. Machine Learning and Deep Learning models serve as efficient predictors. Machine Learning techniques, such as Ridge Regression and tree-based ensembles like XGBoost, have been employed to predict and evaluate portfolio proportions.
Uhunmwhangho (2024) looks into the success rate of XGBoost and LSTM simulations for determining the oriented shifts in the the price of stocks and discovered that XGBoost is a more efficient instrument for short-term in nature decisions in stock market trading, highlighting the opportunity for enhancing the predictive accuracy in financial projecting as well as improving modeling predictive study results.
Deep learning models enhance their forecasting abilities through the detection of dynamic, multifaceted connections, demonstrating their efficacy in finding complex structures in vast datasets. Hochreiter and Schmidhumber (1997) presented LSTM networks and concluded that they function well for simulating time-varying financial trends. The work by Vuong et al. (2022) provided yet another illustration of the predictive power of models based on DL. The researchers demonstrated the way two models, XGBoost-LSTM, may be merged to employ various strategies to reap the advantages associated with structured analysis and historical information memory. After the COVID pandemic, academics have begun examining the idea of broadening the use of machine learning to algorithms that can advise on asset allocation techniques as opposed to relying just on forecasting yields. Zhan et al. (2020) introduced a deep learning approach that distributes investments to optimize the Sharpe ratio without first anticipating profits. They discovered that their methodology far surpassed the usual methods. In a comparable manner. Babiak and Barunik (2020) built an algorithm capable of infusing deep learning into the methodology for generating investment choices gradually. The downside to their analysis is that the portfolio's performance wasn't assessed across many independent parameters. They also explored an investor's adaptive portfolio selection using DL strategies to anticipate return on equity while designing the ideal portfolios. The research investigation found important advantages in employing DL techniques to build

optimum portfolios based on assurance akin to returns with Sharpe ratios. They proved that LSTM outperforms the different kinds of topologies researched. The authors of the paper, however, did not broaden their study to include other assets

2.2 The use of Deep Learning and Machine Learning in Volatile Markets, e.g., Cryptocurrency market (Bitcoin, Ethereum)

The cryptocurrency market lacks extensive periods of history compared with conventional securities, because their prices are always fluctuating, they have no straightforward trends and frequently show unpredictable behaviors. These characteristics made cryptocurrency an ideal playground for sophisticated DL methodologies. As an illustration, Li et al. (2024) used deep learning to detect pairings of cryptocurrencies that reveal mean reversion, or coins that frequently return to the same price range. Comparably, Sumith et al. (2024) created DL systems that can combine several sorts of information, including economic news, social media sentiment analysis, along with price charts, to generate better forecasts amid volatile market situations. We observed from our analysis of each of these literatures that concerns such as transaction charges, nonlinear trading, as well as government laws affecting crypto transactions, all of which are encountered in everyday trading operations, were not addressed by these simulations.

2.3 Deep Learning in Equity and Multi-Asset Markets

Recent research indicates that using advanced ML and DL approaches can help investors make decisions with greater insights and understanding. Babiak and Barunik (2020) reported significant advantages in distributing portfolios utilizing deep learning algorithms. They created adaptive prediction models to maximize return on investment, and their outcomes exceeded unchanging standard approaches. Similarly, in 2021, Zhang et al. created a methodology for selecting exchange-traded funds through the manner that boosts earnings compared to risk. Although the model performed better than traditional approaches, our assessment of their study revealed that the model did not give sufficient details into how it generated its conclusions. Likewise, a few researchers are expanding their research to investigate new technologies, among them are transformers, which are being used in AI chatbots especially. New studies conducted by Mozaffari and Zhang (2023) and Li et al. (2024) demonstrated that price trends can be forecasted more correctly than LSTMs, citing a prior model. They discovered that this is due to the Transformer's ability to narrow down on important data gradually.

2.4 Summary

The literature review presented above highlights the potential of the application of deep learning in portfolio optimization, and current models are often short in areas such as cross-asset integration, uncertainty modelling, and end-to-end learning. This project proposes a unified deep learning framework that incorporates structural relationships, supports multi-asset classes (including cryptocurrencies), and integrates uncertainty-aware predictions. This project intends to address these limitations by providing a more robust portfolio optimization strategy suitable for the ever-dynamic and changing financial markets.

3. Methodology

We define the methods used in our project for comparing traditional methods for portfolio optimization, such as Mean-Variance and Black-Litterman, with predictive models combined with traditional methods. Our goal is to determine that data-driven models are better for adapting to the market across assets such as cryptocurrencies and stocks.

3.1 Data Collection

We source and collect historical data of 2 cryptocurrencies and 8 selected U.S. equity stocks via the Yahoo Finance API.

- Cryptocurrencies - Bitcoin, Ethereum
- U.S. equity stocks – Mastercard, Coca-Cola, Bank of America, General Electric, Adobe, McDonald's, The Walt Disney Company, Pepsico, Chevron

Table 1. Table Label

Company	Group	Ticker
Bitcoin	Cryptocurrency	‘BTC-USD'
Ethereum	Cryptocurrency	'ETH-USD'
Mastercard	U.S. Equity Stock	'MA'
Coca-Cola	U.S. Equity Stock	‘KO'
Bank of America	U.S. Equity Stock	'BAC'
General Electric	U.S. Equity Stock	'GE'
Adobe	U.S. Equity Stock	'ADBE'
McDonald	U.S. Equity Stock	'MCD'
The Walt Disney Company	U.S. Equity Stock	'DIS.MX'
Pepsico	U.S. Equity Stock	'PEP'
Chevron	U.S. Equity Stock	‘CVX'

We clean all price series to adjust for missing data and use adjusted close prices. We compute returns as log-returns, and assets with excessive missingness or low liquidity are excluded.

3.2 Feature Engineering

To maximize our model performance, we represent each asset using a set of engineered features:

- Lagged Return: Daily/Weekly/Monthly returns
- Technical indicators: SMA, EMA, RSI, MACD

a. Simple Moving Average (SMA)

SMA is a useful tool used in time-series analysis in financial market. It calculates average of a selected range of prices by the number of periods in that range. SMA can be used to identify trend direction, and trading signals.

b. Exponential Moving Average (EMA)

EMA is a enhance version of SMA that gives more weight to recent data points makes it to be more responsive to new information. EMA is used to detect momentum shifts, trend confirmation.

c. Relative Strength Index (RSI)

RSI is a technical indicator used for determining the momentum of any asset’s price. It can help to identify potential overbought or oversold areas. RSI is a range between 0 and 100, where a range above 70 indicates overbought, and a range below 30 indicates oversold.

d. Moving Average Convergence Divergence (MACD)

MACD is a technical indicator used for the trend strength of the price of an asset. MACD captures both trend-following and momentum characteristics.

3.3 Predictive Model

We construct 10 predictive ML and DL models, and train them to predict relative asset performance. Each model was initialized with a specific asset ticker, a date range, and a dataset containing features and target values for various assets. The function for each model can be found in the python code for this project.

I. Machine Learning:

a. Ridge Regression

Ridge regression is a technique used to address overfitting in linear regression models. In general, it helps improve a model's performance by reducing error. Ridge Regression is especially useful when dealing with multicollinearity. It is given by the following equation:

$$\widehat{\beta_{\text{ridge}}} = (X^T X + \lambda I)^{-1} X^T y \qquad (1)$$

where $\widehat{\beta_{\text{ridge}}}$ is the vector, X is a matrix of features, X^T is the transpose of the matrix, y is the vector of the target, λ is regularization, I is the identity matrix.

b. XGBoost

XGBoost is a powerful machine learning tool for scalability and efficiency. Its strength is its ability to handle large datasets. XGBoost optimizes a regularized loss function, which helps prevent overfitting, and XGBoost can also handle missing values. The objective function is given by the following equation:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \qquad (2)$$

where $L(\phi)$ is an objective function, $l(y_i, \widehat{y_i})$ is a loss function, f_k is the tree function, Ω is a regularization term.

c. Support Vector Regression (SVR)

SVR is an extension of SVM (Support Vector Machines) built for regression tasks. One of the key strengths is to identify a function that fits data while keeping prediction errors as small as possible.

$$\mathrm{f}(x) = \sum_{i=1}^{n} (\alpha_i - \alpha^*{}_i)\, K(x_{i,x}) + b \qquad (3)$$

where α_i and $\alpha^* i$ are the dual variables, $x_{i,}$ represents the support vectors, $K(x_{i,x})$ is the kernel function applied to the support vector xi and the new input x,b is the bias term

d. Decision Tree Regressor

Decision Tree Regressor is a non-parametric supervised learning. It splits the data into subsets based on feature values to predict a continuous outcome. In finance, decision trees have been widely used.

e. Linear Regression

Linear Regression can provide us linear relationship between our independent variable and our target so that we can predict the probability of the outcome. Key benefits of linear regression are easy interpretation and implementation.

Y = mx + b

where y is predicted, m is the slope, x is the independent variable, and b is the y-intercept

II. Deep Learning:

a. Multi-Layer Perceptron (MLP)

MLP or Multi-Layer Perceptron is a neural network. In MLP, there are multiple layers, and its connected. MLP has an input layer, one or more hidden layers, and an output layer. MLP is very useful for pattern recognition or time series forecasting.

b. Long Short-Term Memory (LSTM)

LSTM or Long Short-Term Memory is a better version of RNN. LSTM can capture "long-term dependencies in sequential data," so it is suitable for tasks such as time series forecasting.

c. Convolutional Neural Network (CNN)

A Convolutional Neural Network is a deep learning algorithm originally developed for processing visual data, such as images and videos. However, in the financial world, CNN have been successfully adapted to analyze structured data by treating time series or financial indicators as visual patterns. CNN can be used to detect patterns in stock price movements.

d. Transformer

Transformer can be applied to model complex sequences in time-series data, such as stock prices. Transformers use self-attention to capture long-range dependencies and relationships across entire sequences simultaneously.

e. Ensemble Learning:

Weight Model Averaging

Weight Model Averaging, often referred to simply as model averaging, is a machine learning technique that combines the predictions of multiple models to generate a single, more reliable output rather than selecting only one model. This approach assigns weights to each model according to its performance.

For each of the models above, we define a custom function in Python to compute the above metrics

3.4 Portfolio Optimization – Evaluation Metrics

We forecast returns from predictive models and integrate them with optimization methods:

a. Markowitz Mean-Variance Optimization: based on predicted return vector and historical covariance
b. Black-Litterman Model: based on predicted return as 'view'

3.5 Back Testing and Evaluation of Portfolio Performance

We evaluate our portfolio performance using a rolling window back test over the test period. The portfolio is rebalanced weekly.

For metrics, we use:

a. Annualized Returns

This is a metric used to evaluate the performance of an investment over a specific period, on the assumption that the investment has grown at a constant rate each year. It is given by the following equation:

$$R_A = (PVt/PV_0)^{(1/t)} - 1 \quad (4)$$

where PV_t – portfolio value at time t, PV_o, - initial portfolio value, t – fraction of the year

b. Sharpe Ratio

Sharpe ratio is a metric used to evaluate the performance of an investment by comparing its return to the level of risk taken. It expresses the idea that higher returns may simply reflect greater volatility, not necessarily better investment decisions. It is given by the following equation:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p} \tag{5}$$

where Rp – portfolio return, Rf – risk-free return, σ_p - Volatility measured as the standard deviation of returns

c. Sortino Ratio

Sortino ratio focuses on downside risk. Sortino ratio helps us to evaluate downside risk.

$$Sortino\ Ratio = \frac{R_p - R_f}{\sigma_d} \tag{6}$$

where Rp – portfolio return, Rf – risk-free return, σ_d - Volatility measured as the standard deviation of downside returns

d. Maximum Drawdown

Maximum Drawdown measures the largest decline in a portfolio's value from its highest point (peak) to its lowest point (trough) before a new peak is reached. It is given by the following equation:

$$MaxDD = \frac{max(Peak - Trough)}{Peak} \tag{7}$$

where *Peak* – maximum realized portfolio value over the investment period, *Trough – minimum realized portfolio value over the investment period*

We define a custom function in Python to compute the above metrics, and we believe that the choice of the above metrics will provide a complete overview of the portfolio for both its return and riskiness.

Finally, we compare the results between:

- ML/DL-based portfolios and;
- Traditional portfolios.

4. Modeling

We detail the overall workflow of our project along with key insights into the data in the following sections.

4.1 Data

As discussed, we retrieved historical daily returns of two cryptocurrency assets and eight selected U.S. equity stocks from Yahoo Finance (yfinance), covering the period from 2018 to 2024. We selected a diversified set of assets between cryptocurrencies and stocks across sectors and industries, for their liquidity and relevance across institutional trading and retail sectors. We also selected stocks with high market capitalization which are less volatile and combined them with crypto assets which are more volatile in the market. To account for dividends and splits, we used adjusted close prices.

As we can see, the portfolio comprises two cryptocurrency assets and U.S. equity stocks. This composition was based on the following considerations:

- We selected Bitcoin and Ethereum because they are both decentralized blockchains, the top two by market capitalization, and have long historical data available, making them suitable for portfolio analysis.
- We selected assets from different sectors, including Financials, Consumer Goods and Services, Industrials, Technology, and Energy. This combination suggests a potential macroeconomic link: the companies within the four sectors are global giants that are highly regulated in the U.S., hence may be driven by similar trends and risk sentiments.
- Furthermore, we selected the period between 01/01/2018 and 19/04/2025 to incorporate pre- and post-COVID market eras.

Table 1 shows a sample of the data frame to demonstrate the data:

Table 1. Sample Data

Ticker	ADBE	BAC	BTC-USD	CVX	DIS.MX	ETH-USD	GE	KO	MA	MCD	PEP
Date											
2018-01-01	NaN	NaN	13657.2002	NaN	NaN	772.6410	NaN	NaN	NaN	NaN	NaN
2018-01-02	177.70	25.0441	14982.0996	92.3530	2152.6511	884.4440	81.4831	36.2722	145.6647	145.2608	94.2381
2018-01-03	181.04	24.9603	15201.0000	93.0262	2163.9773	962.7200	82.2535	36.1926	147.4961	144.6486	93.9906
2018-01-04	183.22	25.2870	15599.2002	92.7367	2162.4805	980.9220	83.9756	36.7023	149.4044	145.6633	94.4536
2018-01-05	185.34	25.4042	17429.5000	92.5846	2137.5325	997.7200	84.0209	36.6944	152.5016	145.9568	94.7250
...	...	...	...	...	...	...	...	...	...	...	...
2025-04-15	350.38	37.7674	83668.9922	132.9210	1708.0000	1588.6324	185.6700	71.8600	517.3800	310.7177	141.2897
2025-04-16	344.19	37.1113	84033.8672	133.7311	1650.2700	1578.1053	182.4500	71.6800	513.4300	307.3568	138.5695
2025-04-17	348.80	37.1908	84895.7500	136.2109	NaN	1582.5483	181.7900	73.0000	517.3300	309.5444	141.2897
2025-04-18	NaN	NaN	84450.8047	NaN	NaN	1588.9247	NaN	NaN	NaN	NaN	NaN
2025-04-19	NaN	NaN	85063.4141	NaN	NaN	1612.9229	NaN	NaN	NaN	NaN	NaN

Source: own elaboration in Python

4.2 Benchmark Portfolio

As a preliminary step, we constructed benchmark portfolios using Markowitz and Black-Litterman models to provide benchmark comparisons as a baseline for evaluating the performance of our predictive models. The benchmark portfolio composition is as listed below:

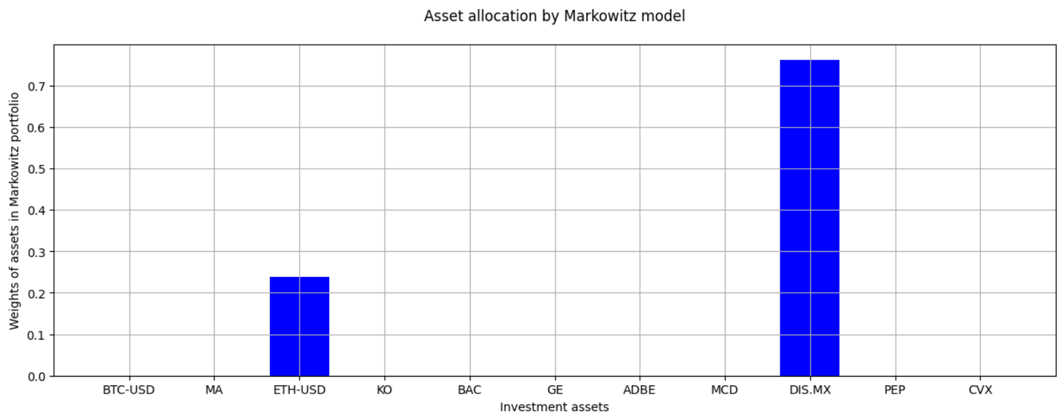
- Markowitz Mean-Variance, we relied on historical mean and covariance.
- The Black-Litterman model incorporates views based on historical performance.

We used the Markowitz mean-variance optimization formula to calculate the weights among the 11 assets. We implemented a standard convex optimization approach to determine asset weights, using inputs such as annualized mean historical returns and the covariance matrix of daily returns. We constrained the weights to be non-negative and to sum to 1.
The portfolio weight is given as:

$$x^* = argmax\left(\mu^T x - \frac{1}{2} x^T \sum x\right) \quad (15)$$

where x^* - represents the optimal allocation of capital across assets

Table 2: Benchmark portfolio allocation by Markowitz model



Source: Yfinance and own elaboration in Python

In the resulting allocation from Markowitz's mean-variance optimization, ETH received an allocation of approximately 25% while Disney received a significantly higher weight of around 75%. The remaining assets received 0% of the allocation, indicating exclusion from the portfolio. This outcome shows us that the Markowitz model is susceptible to historical volatility and correlations. Disney has comparatively low volatility during the selected time window, resulting in the algorithm heavily favouring it in pursuit of the highest return-to-risk trade-off. On the other hand, assets with comparatively higher or weaker risk-adjusted returns received 0% allocation, i.e., were completely excluded. This shows that the Markowitz model tends to allocate more low-risk assets when no diversification constraints are imposed.

4.3 Model Training

As a next step, we generated predictive views for the Black-Litterman by training and consolidating prediction returns using ten machine learning and deep learning models for each asset. For each asset, we developed ten machine learning and deep learning models: Classical ML Models: Linear Regression, SVR, Decision Tree Regressor, Ridge Regression, XGBoost; Deep Learning Models: MLP, LSTM, CNN, Transformer, Ensemble Learning.

Table 3: Benchmark portfolio performance measures over the training period

Models	Sharpe Ratio	Sortino Ratio	Annualized Return	Maximum Drawdown
Markowitz Mean-Variance	0.11	0.18	0.81%	-59.37%
Black-Litterman	0.29	0.38	3.22%	-55.14%

Source: Own elaboration in Python

From a snapshot view, it is not surprising to see that the Black-Litterman model showed superior performance across all evaluation metrics over the Markowitz Mean-Variance model. Firstly, Black-Litterman model delivered better return per unit of risk 0.29 measured by Sharpe ratio, relative to Markowitz model that yielded only 0.11 return per unit of risk. Similarly, based on the Sortino ratio, Black-Litterman – 0.38 significantly outperformed Markowitz model - 0.18 by more than double, indicating superior performance in higher returns and less downside volatility. Black-Litterman's annualized returns of 3.22% exceeded Markowitz' return of 0.81% by more than four times, and this suggests that by incorporating investor views, we can better capture return opportunities. Finally, we observed a significantly high drawdown in both models (i.e., over 50%), with Black-Litterman indicating a drawdown of -55.14% while Makorwitz has -59.37%, although the former slightly reduced the maximum loss from peak to trough

In conclusion, we demonstrated through our findings that it is important to integrate theoretical structure with data-driven insights when constructing a portfolio. This proves that if we combine historical data with predictive models, we can achieve balanced risk and return expectations more effectively and efficiently, rather than using just a historical method like the Markowitz model

4.4 Portfolio Construction and Back Testing

Once we have constructed our benchmark portfolio, as a next step, we made use of machine learning models to predict the future performance of each asset in the future. We then combine our predictions by integrating these ML-based views into the Black-Litterman model to optimize the portfolio. We aimed to obtain the highest return possible for the amount of risk taken. We monitored the performance of this portfolio by rebalancing it every 3, 6, and 12-month intervals. This helped us to see how the overall results are affected by different rebalancing intervals.

5. Conclusions

We conclude the research by comparing the obtained performance metrics and identifying the most efficient portfolio optimization strategy. As a final step, we identify the potential further steps that can be taken to improve this project.

5.1 Results and Observations

We evaluated the performance of portfolio optimization strategies: two traditional models, Markowitz and Black-Litterman, and machine learning and deep learning (ML/DL) models integrated with Black-Litterman and rebalancing every 3-, 6-, and 12-month frequency. Comparison was based on these metrics: Sharpe Ratio, Sortino Ratio, Annualized Return, and Maximum Drawdown.

Table 4 below compares the performance of the portfolio optimization strategies. The Markowitz model showed the lowest performance across all the evaluation metrics, delivering a Sharpe Ratio of 0.11, Sortino Ratio of 0.18, Annualized Return of 0.81%, and a -59.37% Maximum Drawdown. The low value of the Sharpe and Sortino ratios indicates poor risk-adjusted returns. As reflected in the significantly high value of maximum drawdown, the model is susceptible to downturns in a single asset because of the tendency to concentrate allocations, as indicated by the allocation of 75% to Disney.

As we move further to the Black-Litterman model (without incorporating ML/DL views), we observe a significant improvement over the Markowitz baseline by almost three times across all metrics, as indicated by the Sharpe ratio increase from 0.11 to 0.29, Sortino moved up from 0.18 to 0.38, annualized return moved up from 0.81% to 3.22%, and we see a reduction in the Maximum Drawdown from -59.37% to -55.14%. We investigated the root cause of this significant increase and found that better returns and improved drawdown can be achieved by incorporating equilibrium market returns. This is because the incorporation of equilibrium market returns allows for a more diversified portfolio. We then concluded that the performance of the model will be limited without forward-looking information, which can be seen from the low performance of the Markowitz model.

We expanded our analysis by integrating ML/DL views into the Black-Litterman model. The objective was to arrive at a smarter model that has been trained with 10 ML/DL models. By so doing, we noted that the performance of the ML/DL-integrated Black-Litterman model with a 3-month rebalancing period was enhanced significantly above our baseline portfolio, the Markowitz model, by almost 4 times across all metrics. Specifically, Sharpe Ratio moved up from 0.11 to 0.35, Sortino moved from 0.18 to 0.47, Annualized Return moved up from 0.81% to 4.14%, and Maximum Drawdown moved down from -59.37% to -54.18%.
We also noticed a slight improvement across all the metrics when we compared the portfolio performance under ML/DL-Integrated Black-Litterman with portfolio performance und Black-Litterman without ML views. We concluded that our portfolio can adapt to changing and dynamic market conditions through quarterly rebalancing.

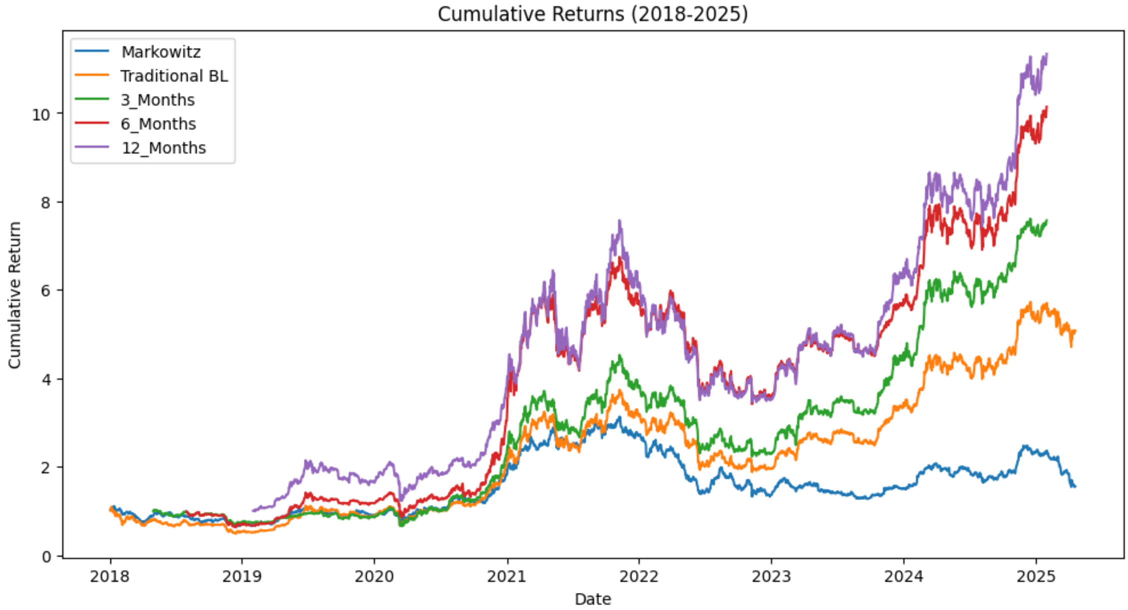
We further explored the benefits of rebalancing and then expanded the frequency to 6 months. Relative to our baseline Markowitz model, we noted a further enhancement in portfolio performance by more than 4 times across the return metrics. Specifically, Sharpe Ratio increased from 0.11 to 0.45, Sortino moved from 0.18 to 0.56, Annualized Return from 0.81% to 5.46%, while Maximum Drawdown went down from -59.37% to -52.67%. Overall, we concluded that 6-month rebalancing provided the best balance of performance and stability. Additionally, we also noticed that while this model reacts to market changes, overfitting was reduced, and this resulted in higher returns with a moderate drawdown.

Finally, we expanded the rebalancing to annual. This produced the highest returns - Sharpe and Sortino Ratios of 0.47 and 0.62, respectively, over the Markowitz baseline of 0.11 and 0.18 – representing about 5 times improvement. This suggests that long trends are captured by the model. Surprisingly, the maximum drawdown showed a slightly worse performance under the annual rebalancing when compared to the 6-month case, suggesting that it shows a slow response to market corrections.

In summary, the Markowitz model had the least performance, mainly driven by overconcentration and too much reliance on backward-looking estimates of returns. Our findings highlight the importance of integrating ML/DL forecasts into the Black-Litterman model. As demonstrated above, integrating ML/DL models with the Black-Litterman framework significantly enhances portfolio performance across all evaluation metrics. Additionally, rebalancing frequency also plays a very important part in portfolio performance. We see from our analysis above that rebalancing every 6 months and every year yields the highest performance, providing a fair balance of risk of adaptability, and overfitting.

Table 4: Model evaluation results

Models	Sharpe Ratio	Sortino Ratio	Annualized Return	Maximum Drawdown
Markowitz Mean-Variance	0.11	0.18	0.81%	-59.37%
Black-Litterman	0.29	0.38	3.22%	-55.14%
ML/DL Integrated with BL Rebalancing every 3 Months	0.35	0.47	4.14%	-54.18%
Integrated with BL Rebalancing every 6 Months	0.45	0.56	5.46%	-52.67%
Integrated with BL Rebalancing every 12 Months	0.47	0.62	5.86%	-55.14%



5.2. Discussion and Further Development

As we develop a portfolio optimization engine for handling diverse assets such as cryptocurrencies and stocks in a portfolio. Our engine combines predictive models for asset returns and assets allocation using traditional methods. However, there are promising potentials to develop this project further.

Firstly, we could heavily tune on hypermeter of each model to enhance model's performance. For example, increasing batch size and epochs of Deep Learning models. Furthermore, parameters tuning in XGBoost could gain more benefits for prediction results which are crucial for "views" in Black-Litterman.

Secondly, we could explore possibilities of more features, especially from Technical Analysis, such as BB (Bollinger Band), Golden Cross, Death Cross, and Price Volume Trend. Furthermore, Elliott wave principle combines with Fibonacci Levels could be developed to detect the stage of wave. These features could be benefits for portfolios and strategies by increasing their performance in four metrics (Annualized Return, Sharpe Ratio, Sortino Ratio, Maximum Drawdown)

Thirdly, we could explore potentials to apply each model to particularly sectors or industries. For example, as cryptocurrency market is high volatility, we could spend more time to use deep learning models in learning the pattern in 1 hours, 4 hours period to extract valuable data from this volatility. Stock market and cryptocurrency market are different in many ways, if we build models for each suitable market or sectors to maximize performance, it would be benefits for our portfolios.

Lastly, we could explore more trickers from cryptocurrency market such as Solana, Doge Coin, or even meme coins. Even though, cryptocurrency market has high volatility of market, there are coins as we indicated above which are more high volatility than Bitcoin, and Ethereum. These coins could be benefits to our portfolios after appropriate tuning in models.

In summary, these are promising potentials to further develop project as indicated above. Nevertheless, there are more techniques to maximize portfolio outcomes by incorporating with text analysis, for example, from X (twitter), news, Facebook pages, blogs, and discord channels to do sentiment analysis which can be our good potential features.

Acknowledgments

We would like to express our sincere gratitude to Igor Tulchinsky, the founder of Worldquant University, for his invaluable commitment to making higher-quality education more accessible worldwide, to all the instructors and tutors for allowing us to share from their wealth of knowledge and most importantly, to our mentor and project supervisor, Professor Joe Byers, for his inspirational leadership, guidance, and support throughout this project.

References

Babiak, Mykola, and Jozef Barunik. “Deep Learning, Predictability, and Optimal Portfolio Returns.” arXiv preprint arXiv:2009.03394, 2020. https://arxiv.org/abs/2009.03394

Chen, Tianqi, and Carlos Guestrin. *XGBoost: “*A Scalable Tree Boosting System.” arXiv preprint arXiv:1603.02754, 2016. https://arxiv.org/abs/1603.02754

Conlon, Thomas, John Cotter, and Iason Kynigakis. “Machine Learning and Factor-Based Portfolio Optimization.” arXiv preprint arXiv:2107.13866, 2021. https://arxiv.org/abs/2107.13866

Gatev, Evan, William N. Goetzmann, and K. Geert Rouwenhorst. "Pairs Trading: Performance of a Relative-Value Arbitrage Rule." *Review of Financial Studies*, vol. 19, no. 3, 2006, pp. 797–827. https://dx.doi.org/10.2139/ssrn.141615

Hoang Vuong, Pham, Dat Tan Trinh, Tieu Khoi Mai, and Pham Hoang Uyen. “Stock-Price Forecasting Based on XGBoost and LSTM.” Computer Systems Science and Engineering, vol. 40, no. 1, 2022, pp. 237–246, https://doi.org/10.32604/csse.2022.017685

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." *Neural Computation*, vol. 9, no. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hopfield, John J. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, 1982, pp. 2554–2558.

Hormann, Siegfried, Lajos Horváth, and Ron Reeder. "The Functional ARCH Model for High-Frequency Data." *Journal of Financial Econometrics*, vol. 9, no. 3, 2011, pp. 456–493. https://www.researchgate.net/publication/51889824_A_functional_version_of_the_ARCH_model

Huck, Nicolas, and Koffi Afawubo. "Pairs Trading and Outliers: A Robust Cointegration Approach." *Economic Modelling*, vol. 50, 2015, pp. 487–500.

Johansen, Søren. "Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models." *Econometrica*, vol. 59, no. 6, 1991, pp. 1551–1580.

Kopeliovich, Yaacov, and Michael Pokojovy. “Portfolio Optimization with Feedback Strategies Based on Artificial Neural Networks.” arXiv preprint arXiv:2411.09899, 2024. https://arxiv.org/abs/2411.09899

Kumar, Yashaswi. “Deep Reinforcement Learning for Portfolio Optimization Using Latent Feature State Space (LFSS) Module.” arXiv preprint arXiv:2102.06233, 2021. https://arxiv.org/abs/2102.06233

Li, Tong & Liu, Zhaoyang & Shen, Yanyan & Wang, Xue & Chen, Haokun & Huang, Sen. (2024). “MASTER: Market-Guided Stock Transformer for Stock Price Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence.” 38. https://doi.org/10.1609/aaai.v38i1.27767.

Mozaffari, Arian, and Bozhi Zhang. “Transformers for Financial Time Series Forecasting.” arXiv preprint arXiv:2303.12916, 2023.

Osaigbokan Peter Uhunmwangho. “Comparative Analysis of XGBoost and LSTM for Stock Price Prediction.” arXiv preprint arXiv:2404.03792, 2024.

Sezer, Omer & Gudelek, Ugur & Ozbayoglu, Murat. (2020). “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing.” 90. 106181. 10.1016/j.asoc.2020.106181.

Said, Said E., and David A. Dickey. "Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order." *Biometrika*, vol. 71, no. 3, 1984, pp. 599–607.

Sujatha, R., and H. Sundaram. "An Application of Ridge Regression for Portfolio Optimization." *International Journal of Computer Applications*, vol. 25, no. 6, 2011.

Sumit, Nawathe, Ravi Panguluri, James Zhang, Sashwat Venkatesh. “Multimodal Deep Reinforcement Learning for Portfolio Optimization”. 2024. https://doi.org/10.48550/arXiv.2412.17293

Vaswani Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. "Attention is All You Need." *Advances in Neural Information Processing Systems*, vol. 30, 2017. https://doi.org/10.48550/arXiv.1706.03762

Yi, Jin. "A Pair Trading Strategy Using KMeans and Cointegration Test Based on High-Frequency Data." *Procedia Computer Science*, vol. 175, 2020, pp. 735–740.

Z. Zhang, S. Zohren, S. Roberts. “Deep Learning for Portfolio Optimization.” arXiv preprint arXiv:2005.13665, 2020. https://arxiv.org/abs/2005.13665