Portfolio Optimization using Reinforcement Learning and Sentiment Analysis to Navigate Uncertain Market Conditions

Anish Patel and Na'Sir Miller
patela16@rpi.edu

millen11@rpi.edu

Abstract

In this paper, we explore reinforcement learning (RL) along with sentiment analysis for stock portfolio optimization. We implement a proximal policy optimization (PPO) based agent within a custom OpenAI Gymnasium environment. Our agent is trained to maximize the log returns across a portfolio of a number of stocks. We experiment with and without the use of sentiment data and find a 2% gain in returns when using sentiment data. Our results underscore the success of using RL for portfolio optimization while highlighting the potential of sentiment analysis within the financial sector.

1 Introduction

The stock market is a volatile environment, managing over $124 trillion USD in global equity (Visual Capitalist, 2023). Given the substantial financial value at stake, institutions and individuals alike aim to maximize their returns through the management of their portfolios. However effective portfolio management requires extensive commitment through market research such as company performances, balance sheets, etc. Unfortunately, the data necessary to both make informed and understandable decisions is not readily accessible to most shareholders due to paywalls and domain expertise. This disconnect between resources and the market is a significant inconvenience and barrier to entry for share owners. To address these concerns, robotraders and other autonomous portfolio allocators have become prominent through statistical and machine learning techniques alike.

Although the development of approaches ranging from moving averages to deep learning are performative, more advancements can be made to improve these systems to bolster shareholder's profits and satisfaction. As a result, we have developed a robust portfolio optimization system that refines reinforcement learning methods with the power of natural language processing.

2 Related Work

Previous existing ways for portfolio optimization like mean-variance optimization proposed by Markowitz, mostly looks at previous data and a lot of the time does not see the non-quantitative factors that influence the market, like investor sentiment and real-world events such as the financial crisis or COVID-19 pandemic. To mitigate this, recent research has shown using machine learning and deep reinforcement learning in financial decision-making.

In *"Deep Reinforcement Learning for Optimal Portfolio Allocation: A Comparative Study with Mean-Variance Optimization"* by Sood et al. [4], the authors use Proximal Policy Optimization to rebalance portfolios in a stochastic environment. This research was one of our inspirations for our approach, specifically because of PPO's suitability for continuous action spaces like portfolio weights. The results of this study show that RL models can outperform in volatile markets.

Jiang et al. (2024) talks about *Deep Reinforcement Learning for Portfolio Selection*, showing how RL agents can adapt to market's volatility and to learn complex policies that outperform static allocation strategies. These findings support RL is a good fit in environments that have randomness and uncertainty. These are conditions that closely follow the behavior of financial markets.

In addition to numerical financial indicators, sentiment analysis is also an important factor for quantitative finance. Financial text data such as news headlines have been shown to have helped with predictive power with asset prices. FinBERT, introduced by Araci (2019), is a domain-specific version of the BERT model, but fine-tuned specifically to financial sentiment classification. This model has widely been used in NLP-based financial research because of the high performance results when finding sentiments in financial text. Our approach builds off of the fact that sentiment is an important changing factor in market behavior; we use FinBERT for sentiments and directly integrate this into the state space of our RL agent.

Our work is distinguished by the usage of a sentiment time series as a dynamic feature in the RL state representation. Previous works either look for financial indicators or sentiment separately, however our combined framework feeds the RL agent both the price data and sentiment information to make decisions using better context.

Hong (2024) also gives a useful guide for implementing RL with stock trading strategies. His details relating to hyperparameter tuning, reward engineering, and model stability were tremendous help for creating our own experimentation and evaluation methods.

Our approach builds off previous research by using sentiment context into the reinforcement learning framework, to give a more comprehensive view of the market dynamics that looks for not only quantitative data, but qualitative as well.

3 Methodology

3.1 System Overview

Our system has two independent pipelines that integrate before training our agent, the first being related to stock data and the other regarding sentiment.

The first pipeline starts by obtaining all tickers on the NASDAQ through the U.S Stock Symbols repository (Reichel, 2024) and verifying eligible tickers via yfinace (Ran, 2024), since yfinance produces our stock data. After extensive verification that tickers are both available on yfinance and material in a given time frame, we sample a user-specified number of tickers from the validated set of tickers and use them as our portfolio's assets. After selecting our assets, we derive a T x N price matrix, that contains the adjusted closing price across time (T) for each asset (N).

Our second pipeline runs independent of pipeline one, as it leverages the Massive Stocks News Analysis DB for NLP/Backtests from Kaggle (Aenlle, 2023) for stock and market related news headlines. To truncate the data ranging from 2009 to 2020, we filter the busiest months containing the most sentiments. We then extract the market's sentiment and integrate it with pipeline one.

Upon the availability of asset and sentiment data, we update our T x N price matrix, to ensure its time frame matches the sentiment data. From there we train our agent on the provided data and report its performance.

3.2 Environment Design

Our Gymnasium environment encapsulated both the portfolio to be optimized and the market conditions its assets are subjected to.

The market conditions, see System Overview, and initial cash balance in USD allowed us to represent the environments RL parameters. More specifically, within the state space we track the prices of our assets, running cash balance, and asset allocation weights. To ensure stability throughout the allocation process, we normalized asset prices by the largest asset price on the day, with respect to each asset, normalized the cash balances by the initial cash amount, and allocation weights by the number of assets in the portfolio. Additionally the state space captures the sentiment across time, dependent on if the user tasks it with such.

At each time step, the agent can update the allocation weights from a percentage between 0 and 1. The criterion or reward function for altering allocation is to maximize log of the returns, where the return is the price relative movement. More precisely, the price relative movement is as follows. Given current price, p_t and the next price, p_{t+1},

$$reward = log(\frac{p_{t+1}}{p_t}) \tag{1}$$

We treat each timestep as the prices of the assets for the given timestep, such that we have

$$a_t = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix} \tag{2}$$

and terminate episodes if and only if we have processed the asset vector, a_t, at timestep, t.

3.3 Agent Design

To optimize the environment, we leverage Stable-Baselines3 (Raffin et al., 2021) and its PPO implementation for efficient and effective performance. Our agent follows a Multi-Layer Perceptron policy with the following hyperparameters

Hyperparameter	Value
Policy	MlpPolicy
Number of steps (n_steps)	128
Batch size	64
Learning rate	3×10^{-4}
Entropy coefficient (α)	0.01
Device	CUDA (if available)
Evaluation frequency	10,000 steps
Total training timesteps	100,000

Table 1: PPO agent hyperparameters used for portfolio optimization.

3.4 Sentiment Integration

To integrate sentiment into our environment, we process the news headlines using FinBERT. FinBERT is a transformer model based off of BERT and tuned specifically for financial sentiment classification (ProsusAI, 2020). The classifications are between the options, bearish, neutral, and bullish which we encoded to -1, 0, and 1, respectively .Moreover, normalized all produced sentiments between $[-1, 1]$ by aggregating the daily sentiment for each day in our time frame. Again, the sentiment was embedded into the state space of the environment.

3.5 Data Collection & Processing

All data we utilized throughout the project were retrieved through APIs. More specifically, Namely, yfinance for stock prices, US Stock Symbols repository for NASDAQ tickers, and Massive Stocks News Analysis DB for NLP/Backtests for news data.

To maintain consistency through three different sources of data, we ensured all data used in our analysis was aligned. For example we verified that any retrieved tickers were accessible through yfinance, and kept all stock data that matched the time frame of our news data.

3.6 Training Details

Our agent was trained for 100,000 timesteps, where we tracked the time the process took. We also used Stable-Baseline3 EvalCallback to monitor performance every 10,000 timesteps as well as perform early stopping and model retrieving elegantly through its check pointing.

4 Experiments

To evaluate our portfolio allocation agent, we perform four experiments, based around its performance with and without sentiment and across different time frames. We evaluated the agent on a different environment with .20% of the data, and the same sampled tickers.

Our experiments were formulated to quantify the impact of sentiment analysis on our optimization strategy and the presence of external factors. The data we were able to utilize from the massive database started from May 1st, 2019 to March 25th, 2020. On the latter half of the data was the emergence of COVID-19, which forced us to have to test the agent's performance pre and post COVID, as the period was a huge stressor on the stock market.

5 Results

Table 1 presents the cumulative return, average daily return, volatility and Sharpe ratios for agents trained with and without sentiments . As shown, sentiment-enhanced environments yielded higher returns and Sharpe ratios across both time spans.

As the test set ranged from 7 to 16 days, figure 1 indicates the returns of the portfolios across the given time span such as May 1st, 2019 to December 31, 2020 or March 25, 2025. As suggested from table 1, figure 1 illustrates that the sentiment assisted in portfolio growth.

6 Discussion

The results of our experimentation suggest that the incorporation of sentiment does impact portfolio optimization, and more precisely, improves the returns of portfolios. In fact, sentiment-induced portfolios, were less volatile and obtained better average returns displaying reduced downside risk. Moreover, we saw returns up to 3.02% across the evaluated portfolios, which beat the average performance of the NASDAQ which saw a return of 2.11% from May 2019 to December 2019.

However, as indicated by figure 1, the portfolio with sentiment still performs poorly in the presence of COVID-19. With respect to our 1,000,000 USD portfolio, we incurred a 14.26% loss, 16.09% without sentiment. The NASDAQ had also seen -1.69% return in this same period, indicating that our agent had struggled to navigate such an uncertain time.

7 Conclusion

Through our sentiment-empowered RL agent we have outperformed the NASDAQ in stable market conditions. However, in the presence of the unfortunate, COVID-19, we drastically underperformed the market. In our study we aimed to prove that sentiment analysis can better navigate uncertain market conditions, such as those seen in March 2025, but we were clearly disproven. In fact, we believe that sentiment can potentially harm your performance in bearish markets, given the market's inherent passiveness.

In future works, we aim to expand our study from 2009 to 2020, to get a better understanding of our agent's performance across varying markets and long-term performance. Additionally, we look to introduce technical indicators that can better guide our agent in its decision making.

Limitations

Given the machines utilized for development, we could not utilize all of the Massive Stocks News Analysis DB for NLP/Backtests dataset. As a a result, we were forced to use mid 2019 to early 2020 data, which was an incredibly hard time for the stock market – as reflected by portfolio performance. Additionally, we were limited to free data sources which also diminished our flexibility and data quality.

Moreover, we were limited to one portfolio for the study, with stocks that may not necessarily be popularly held such as Inotiv, Airgain, etc. Again, we sample tickers from the NASDAQ, so it is likely that the portfolio we obtain is not optimal across the index, considering potential omission of larger companies such as Apple.

References

Miguel Aenlle. 2023. Massive stock news analysis db for nlp/backtests. Accessed: 2025-04-21.

ProsusAI. 2020. Finbert - financial sentiment analysis model. https://huggingface.co/ProsusAI/finbert. Accessed: 2025-04-23.

Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. https://stable-baselines3.readthedocs.io/en/master/guide/quickstart.html. Accessed: 2025-04-23.

Tirthajyoti Ran. 2024. Yfinance: Yahoo finance market data downloader. Accessed: 2025-04-21.

Richard Reichel. 2024. Us stock symbols dataset. Accessed: 2025-04-21.

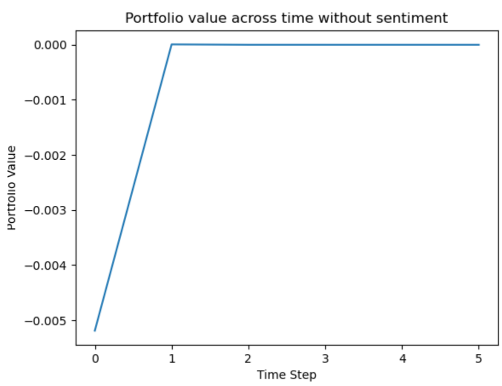
Visual Capitalist. 2023. The $124 trillion global stock market by region. Accessed: 2025-04-21.

8 Codebase

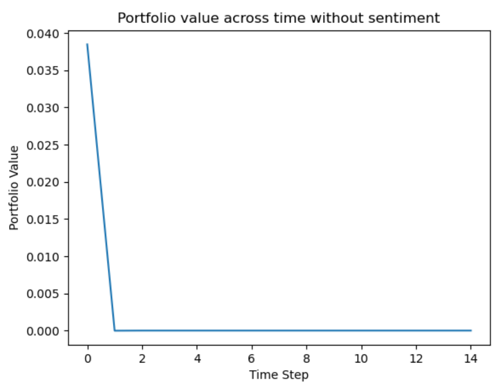
To access our codebase please naviagte to, SentimentPortfolioOptimizer

Sentiment	Period	Cum. Return	Avg Return	Volatility	Sharpe (Simple)	Sharpe (Log)
No	2019-05-01 to 2020-03-25	0.8391	-0.0097	0.0617	-0.6289	-0.7474
Yes	2019-05-01 to 2020-03-25	0.8574	-0.0084	0.0598	-0.5619	-0.6804
No	2019-05-01 to 2019-12-31	1.0140	0.0024	0.0090	0.6957	0.6839
Yes	2019-05-01 to 2019-12-31	1.0302	0.0050	0.0088	1.5024	1.4964

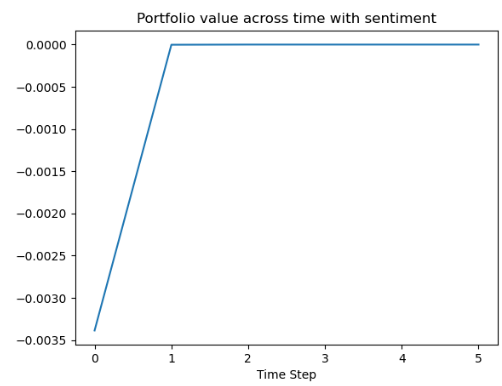
Table 2: Performance comparison with and without sentiment features across two time periods.



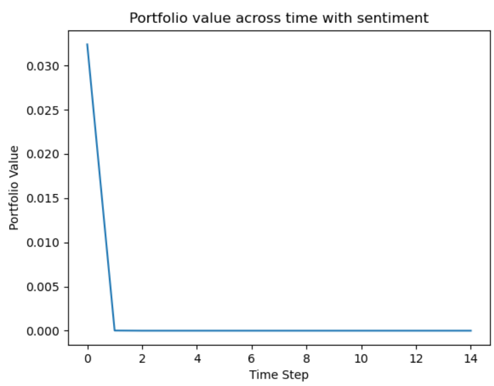
(a) No Sentiment, Pre-COVID



(b) No Sentiment, Post-COVID



(c) With Sentiment, Pre-COVID



(d) With Sentiment, Post-COVID

Figure 1: Portfolio returns across sentiment settings and time periods.