The University of Hong Kong

Faculty of Engineering

Department of Computer Science

Final Year Project
Detailed Project Plan

Topic

Portfolio Management with Technical Analysis using Reinforcement Learning

Supervisor

Prof. Zhiyi Huang

Students

Chan Hin Ting (3035928299)
Tam Pai Lok (3035926447)
Mak Wan Shing (3035930709)

Table of Contents

1. Project Background and Motivation

The financial market has been democratized by ever advancing technology [1] and easier access to financial data [2]. The rise in machine learning has driven a trend of stock prediction [3] and more technical analysis, especially facilitating short-term trading [4].

However, it is found that these analyses could not always “beat the market” [5] and people with limited knowledge in finance may overlook the hidden risk while pursuing high returns in their prediction and analysis. Small investors may only focus on a specific stock’s trend and assign all their capital into it. This could bring significant risk given the stock’s sensitivity to macroeconomic factors and idiosyncratic risk induced by the firm of the stock itself.

Therefore, this project aims to apply artificial intelligence in portfolio management instead in a bid to take into consideration the downside risk of investment and produce a more consistent expected performance. It also introduces the importance of portfolio diversification as stated by Markowitz [6] for a robust investment strategy.

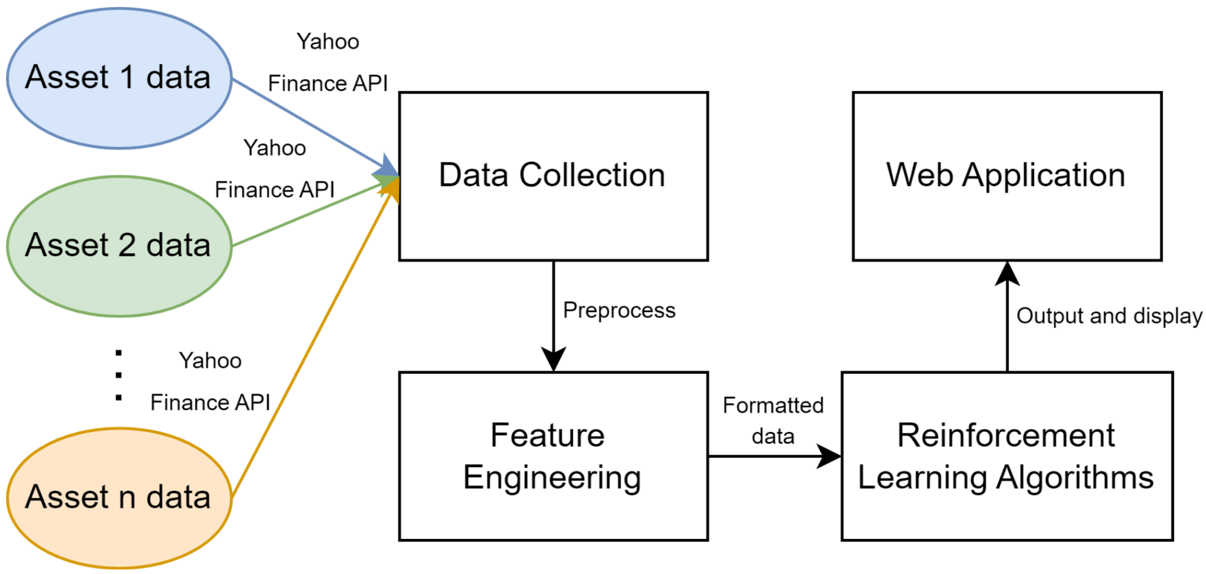
This project contributes in two aspects. First, it explores the application of Reinforcement Learning on Modern Portfolio Theory [7] and technical indicators including relative strength index (RSI), moving average (MA) and moving convergence divergence (MACD). Second, it develops a user interface that visualizes the portfolio weightings and the corresponding optimization applied to facilitate portfolio management.

2. Project Objective

1. Increase portfolio management performance and accuracy by connecting Modern Portfolio Theory with reinforcement learning. We link up stocks' correlation and historical prices together as input tensor and by convolution neural network and Tucker Decomposition [8] to output portfolio weight as an action function for reinforcement learning.
2. Improve portfolio management efficiency by automating the process using Reinforcement Learning. The optimization involves heavy computation and regular monitoring based on the evolving market. Training online learning model reduces labor work on management.
3. Build a visualization dashboard to monitor the portfolio performance like Omega Ratio [9], maximum drawdown, and indicators optimizations by adjusting windows of historical data and factors strength to evaluate MA, RSI, MACD etc.
4. Reduce cost and enhance accessibility of portfolio management. Instead of outsourcing the work to portfolio managers, this project aims to automate the task with comparable performance in a self-sustainable manner.

3. Project Methodology

3.1 Architectural Overview



Our project aims to manage the user's investment portfolio by maximizing the portfolio value under an acceptable level of risk. This involves trading data of multiple assets being fed to our reinforcement learning model in real time. We then stream the trading data to the reinforcement learning model for training, according to our optimization policy. We have abstracted the details of the model at this stage as the research on model selection and optimization strategy is still ongoing. The model is expected to deliver the optimal portfolio weights to reach the target return or return/risk ratio and the output will be displayed via the web application.

3.2 Data Collection and Preprocessing

To get our stock trading data for training, we will go for the yfinance API, which is a Python tool to download stock data via Yahoo Finance [10]. The easiest and safest way to obtain a bulk amount of market data is to download the data (e.g. closing price, volume, etc…) for each stock chosen in our portfolio and export them into a CSV file. Although not common, we may have to deal with missing data. To fill in the holes caused by missing data, we can utilize interpolation to compute the mean value between the previous and the next data point. For features such as closing price, this could give a reasonably good estimate As the data fetched from the API is already well-structured, we can directly load the data batch-by-batch for processing before feeding them into the Reinforcement Learning algorithms.

3.3 Feature Engineering

By technical analysis principles, we can compute technical indicators for each stock at each timestamp and include them in a vector. Indicator values are often not easily retrieved by common publicly available APIs on the market. However, we can compute the indicators based on the closing prices we collected previously. Furthermore, we can also compute the correlation matrices between different stocks for different indicators in different timestamps. These structured and formatted data provide the basis for our reinforcement learning models.

3.4 Reinforcement Learning

We have identified numerous commonly used reinforcement learning algorithms for this specified task as we are able to find some related work based on these algorithms, such as Deep Q-Learning (DQL) [11], Deep Deterministic Policy Gradient (DDPG) [12] and Proximal Policy Optimization (PPO) [13]. However, since the action output of our learning output would be a set portfolio weightings for our assets, which is continuous in nature, we therefore will not use DQL, which relies on discrete action space, in this circumstance [12]. On the other hand, DDPG adds policy learning on top of Q-Learning and supports continuous action space [14]. PPO also supports continuous action space, has high performance and is easy to implement [13]. Hence, we decided to start with DDPG and PPO as our initial research direction to model our strategy with reinforcement learning algorithms.

3.5 Front-end Development

To support user interaction with our portfolio management system, we will develop a web platform for users to set their portfolio targets and monitor the portfolio status. We have chosen ReactJS for the front-end development as we can benefit from its wide range of robust libraries and plugins available in the community. The team also has previous experience in front-end development with ReactJS. Hence, we believe the development process could be sped up with the use of ReactJS.

3.6 Back-end Development

Depending on the complexity of our model and the scope of the web application, we would either use the Flask or Django framework built on Python to develop our back-end of the web platform. We choose to stick with Python as we would rely heavily on Python with the Reinforcement Learning model training. By using either Flask or Django, it will be easier for us to integrate the machine learning system with our web application. By comparison, Flask is a more lightweight framework, while Django is more robust and scalable [15]. Hence, upon finalizing our Reinforcement Learning model and designing the scope of our web application, we would pick one of the two frameworks which better suits our needs.

4. Project Schedule

Note: Deadlines are bolded

Date	**Milestone**
Sep 1, 2024 - Sep 30, 2024	• Preliminary research
Oct 1, 2024	**Deliverables of Phase 1 (Inception)** • **Detailed project plan** • **Project web page (Set up)**
Oct 1, 2024 - Dec 8, 2024	• Trading data collection • Reinforcement Learning model research and prototyping • Portfolio management strategy formation
Dec 9, 2024 - Jan 19, 2025	• Reinforcement Learning model optimization and finalization • Backend design + prototype • Frontend design + prototype
Jan 13, 2025 - Jan 17, 2025	**First presentation**
Jan 20, 2025 - Mar 30, 2025	• Full stack development • System Integration
Jan 26, 2025	**Deliverables of Phase 2 (Elaboration)** • **Preliminary implementation** • **Detailed interim report**
Mar 31, 2025 - Apr 20, 2025	• Deployment • Final testing and review
Apr 21, 2025	**Deliverables of Phase 3 (Construction)** • **Finalized tested implementation** • **Final report**
Apr 22, 2025 - Apr 26, 2025	**Final presentation**
Apr 30, 2025	**Project exhibition + 3-min video**

5. References

[1] H. Hirsch-Kreinsen, “Financial Market and Technological Innovation,” *Industry & Innovation*, vol. 18, no. 4, pp. 351–368, May 2011, doi: https://doi.org/10.1080/13662716.2011.573954

[2] M. White, “CAREER PROFILES • Options and Insights: Meredith White,” *Oceanography*, vol. 34, no. 2, Jun. 2021, doi: https://doi.org/10.5670/oceanog.2021.208

[3] A. Subasi, F. Amir, K. Bagedo, A. Shams, and A. Sarirete, “Stock Market Prediction Using Machine Learning,” *Procedia Computer Science*, vol. 194, pp. 173–179, 2021, doi: https://doi.org/10.1016/j.procs.2021.10.071

[4] J. Ayala, M. García-Torres, J. L. V. Noguera, F. Gómez-Vela, and F. Divina, “Technical analysis strategy optimization using a machine learning approach in stock market indices,” *Knowledge-Based Systems*, vol. 225, p. 107119, Aug. 2021, doi: https://doi.org/10.1016/j.knosys.2021.107119

[5] D. M. Smith, N. Wang, Y. Wang, and E. J. Zychowicz, “Sentiment and the Effectiveness of Technical Analysis: Evidence from the Hedge Fund Industry,” *papers.ssrn.com*, Jan. 30, 2015. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2457289

[6] H. Markowitz, “Portfolio Selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952, Available: https://www.math.hkust.edu.hk/~maykwok/courses/ma362/07F/markowitz_JF.pdf

[7] J. Jang and N. Seong, “Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory,” *Expert Systems with Applications*, p. 119556, Jan. 2023, doi: https://doi.org/10.1016/j.eswa.2023.119556

[8] V. Bhatt, S. Kumar, and S. Saini, “Tucker decomposition and applications,” *Materials Today: Proceedings*, Mar. 2021, doi: https://doi.org/10.1016/j.matpr.2021.01.676

[9] J. Mitchell, “Optimizing the Omega ratio using linear programming,” *Journal of Computational Finance*, Jun. 03, 2014. https://www.risk.net/journal-of-computational-finance/2347959/optimizing-the-omega-ratio-using-linear-programming

[10] “GitHub - ranaroussi/yfinance: Yahoo! Finance market data downloader” *GitHub*. https://github.com/ranaroussi/yfinance

[11] Z. Gao, Y. Gao, Y. Hu, Z. Jiang, and J. Su, “Application of Deep Q-Network in Portfolio Management,” *arXiv.org*, 2024. https://arxiv.org/abs/2003.06365

[12] J. Jang and N. Seong, "Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory," *Expert Systems with Applications*, p. 119556, Jan. 2023, doi: https://doi.org/10.1016/j.eswa.2023.119556

[13] S. Lin and P. Beling, "An End-to-End Optimal Trade Execution Framework based on Proximal Policy Optimization." Available: https://www.ijcai.org/proceedings/2020/0627.pdf

[14] "Deep Deterministic Policy Gradient — Spinning Up documentation," *Openai.com*, 2014. https://spinningup.openai.com/en/latest/algorithms/ddpg.html#the-policy-learning-side-of-ddpg

[15] "Differences Between Django vs Flask," *GeeksforGeeks*, Feb. 25, 2020. https://www.geeksforgeeks.org/differences-between-django-vs-flask/