Comp579 Final Project Report

Herbie He
McGill University
zizhan.he@mail.mcgill.ca

Mohamed Tliouant
McGill University
mohamed.tliouant@mail.mcgill.ca

Parag Jain
McGill University
parag.jain@mail.mcgill.ca

Abstract

Portfolio Management has always been a difficult problem to tackle due to the stochasticity of the stock market. Given the time-sensitive nature of stock trading, it is natural to formalize such a problem as an MDP, which allows us to seek solutions via DRL algorithms. In this project, we explored the Ensemble Strategy presented by Xiao-Yang et al. (2020), replicated the results of that paper, and improved the performance of the model by integrating additional indicators and sentiment analysis into the environment. We concluded that the agent learns better under limited resources when we explicitly provide pre-compute statistics on market dynamics to the agent. We have also successfully encoded sentiment analysis into the environment on a size-reduced dataset but failed to train agents under this environment due to the limited amount of financial news that we were able to acquire.

1 Background: Financial Markets and Reinforcement Learning

Reinforcement Learning (RL) is an area of machine learning where an agent learns to make decisions by performing actions and receiving feedback in the form of rewards or penalties. In finance, RL is employed to optimize investment strategies, manage portfolios, execute trades, and model market dynamics. Below, we have explored the current state and recent academic advances in the application of RL within financial markets.

Portfolio Management: RL algorithms can dynamically allocate assets in a portfolio to maximize the expected return based on the observed market conditions. A notable recent advancement is the development of deep RL models that can handle large state spaces (such as those with numerous assets and historical data points) and complex reward structures. For instance, researchers have been integrating deep neural networks with traditional RL frameworks to better predict and adapt to volatile market conditions.

Algorithmic Trading: RL is extensively used in developing autonomous trading systems that can learn profitable trading strategies. These systems are trained on historical data and are capable of executing high-frequency trades automatically. Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG) methods have been adapted to enhance the performance of trading algorithms under different market scenarios.

Risk Management: RL can help in adjusting the risk levels of a portfolio dynamically based on market conditions. Recent advancements involve the use of risk-sensitive reinforcement learning, where the RL models are specifically designed to consider the variance of returns as part of the reward function. This approach helps in constructing more robust financial models that can withstand high volatility and adverse market conditions.

Preprint. Under review.

Model-Based Approaches: These methods involve creating a model of the market dynamics and using it within the RL framework to predict future market states. This allows for more informed decision-making as the RL agent not only learns from historical data but also incorporates predictions about future market behavior. This is particularly useful in complex financial markets where accurate modeling of dynamics is crucial.

Interpretable and Trustworthy RL Systems: Recent research is addressing the challenge of the growing need for interpretable and trustworthy RL systems by developing methods to explain and visualize the decision-making process of RL agents. This is crucial for deployment in financial markets where stakeholders require transparency and accountability.

2 Literature Review

Xiao-Yang et al. (2020) have explored the ensemble learning strategies combined with deep reinforcement learning to enhance the stability and performance of automated stock trading systems. This is the paper that we are replicating and extending. Zhengyao et al. (2021) have presented a deep reinforcement learning framework for managing financial portfolios, demonstrating how DRL can outperform traditional portfolio management methods. Notably, authors (2021) have conducted a comprehensive survey of deep reinforcement learning in algorithmic trading, discussing the challenges and advancements in the field. Ritter (2021) has examined the application of reinforcement learning to optimize trade execution, focusing on minimizing market impact and transaction costs. Michael and Yuriy (2022) explored multi-agent reinforcement learning in trading, utilizing both simulations and real-world markets to develop dynamic trading strategies.

3 Introduction and Problem Definition

3.1 Purpose

The primary goals of our project are to ***a)*** replicate the results of the Ensemble Strategy presented by Xiao-Yang et al. (2020) and ***b)*** to investigate the correlation between the availability of information in the environment and the DRL agent's performance in tackling portfolio maximization. The motivation behind this is to find out whether the agent learns better when provided with the information that is usually available for human traders. Therefore, on top of the environment used in the paper, we added two more dimensions of data to the environment.

Sentiment Analysis

Sentiment analysis is a branch of natural language processing (NLP) that involves determining the emotional tone behind a body of text. This is particularly useful in algorithmic trading, where the sentiment derived from news articles, social media posts, financial reports, and other textual sources can influence trading decisions. Algorithmic trading systems use sentiment analysis to assess the general mood of the market or the sentiment toward specific assets or sectors. Tools like VADER (Valence Aware Dictionary and sentiment Reasoner) or custom NLP models can quantify sentiments as positive, negative, or neutral scores from textual data.

Sentiment scores are used as signals in predictive models to anticipate short-term movements in stock prices. For example, a surge in negative sentiment about a company might be used to predict a potential drop in its stock price. In an RL model, the state could include both traditional market data and sentiment scores as part of the input. Actions would be buy, sell, or hold decisions, and rewards would be defined based on the profitability of these actions. RL algorithms can continuously learn and adapt from new data, including sentiment analysis results. As the sentiment landscape changes, the RL agent updates its policy to optimize the expected outcomes based on both market data and sentiment trends. This is particularly useful in dynamic and volatile markets where trader sentiments can drastically change due to news events or social media trends.

Technical Indicators

Different indicators capture different dimensions of the trend of a stock. Theoretically, an agent should be able to learn the market dynamics implicitly when given enough data. However, with limited data & computing resources, we hypothesize the agent to learn better when we perform the computation beforehand and provide such information to the agent.

3.2 Problem Definition

The problem of portfolio optimization, which involves balancing the holdings on various stocks through buying and selling on each trading day, is modeled as a Markov Decision Process (MDP) problem. The training process involves observing the environment at a specific time stamp, taking an action, receiving a reward based on the change in portfolio value, and arriving in a new environment. The stock dataset we used is the Dow 30, which includes 30 prominent companies in the US (i.e. our profile contains some number of holdings on these 30 companies). The environment employed in the paper can be divided into the base environment and the supplementary environment.

Base Environment:

- $b_t \in R$: The available balance for us to trade at time step t
- $p_t \in R^{30}$: Adjusted closed price of the 30 stocks in Dow 30 at time step t
- $h_t \in Z_+^{30}$: The number of shares we hold on each stock

Supplementary Environment: Additional indicators, computed using the information in the base environment, capture various dimensions of the market dynamics. The purpose of them is to reduce the amount of knowledge an agent has to learn by explicitly feeding it with pre-computed statistics.

- $MACD_t \in R^{30}$: The Moving Average Convergence Divergence for each stock at time step t
- $RSI_t \in R^{30}$: The Relative Strength Index for each stock at time step t
- $CCI_t \in R^{30}$: The Commodity Channel Index of each stock at time step t
- $ADX_t \in R^{30}$: The Average Directional Index of each stock at time step t
- $Turbulence_t \in R^{30}$: The unstableness of the prices of each stock at time step t

Action Space: At each time step, an agent can sell, buy, or maintain its share on any of the 30 stocks. The agent cannot sell more shares than it currently owns nor spend more money than it currently has.

- $\alpha_t \in Z^{30}$:Where $\alpha[i] = k$ is the action taken on stock *i*. k>0, k<0, means the agent is buying and selling k shares on stock *i* respectively. k=0 means the agent neither buys nor sells stock *i*.

Reward Function: Suppose the agent took action α_t and transitioned from $S_t = [b_t, p_t, h_t]$ to $S_{t+1} = [b_{t+1}, p_{t+1}, h_{t+1}]$, then the reward is given by the change in portfolio value $r_t = (b_{t+1} + {p_{t+1}}^T h_{t+1}) - (b_t + {p_t}^T h_t)$. Given such a reward function, maximizing cumulative reward is equivalent to portfolio maximization.

3.3 Ensemble Strategy

The paper presented a novel solution to the problem of portfolio optimization by integrating various DRL algorithms. The mechanism is best illustrated with an example. Suppose our dataset is divided into the training dataset, from `2014-01-01` to `2019-09-30`, and the test dataset from `2019-10-01` to `2020-12-31`.

1. **Training Phase**: Multiple RL agents, including A2C, PPO, DDPG, SAC, TD3 are trained concurrently from `2014-01-01` to `2019-06-30`
2. **Validation Phase**: Before trading (testing), all agents undergo a 3-month validation period from `2019-06-30` to `2019-09-30` where we evaluate their performance through Sharpe Ratio and cumulative returns in profile value.
3. **Trading Phase**: The best agent is selected to trade for the subsequent period from `2019-09-30` to `2019-12-30`. Meanwhile, all other RL agents undergo retraining from `2014-01-01` to `2019-09-30` (the training window is expanded by the length of the validation window). Afterward, all RL agents are validated from `2019-09-30` to `2019-12-30`, where the best agent is picked to trade for the next 3 months from `2019-12-30` to `2020-03-01` . The procedure repeats til the end of the trading period which is `2020-12-31`.

A human intervention is incorporated into all DRL agents to handle market crashes. If the Turbulence Index exceeds a certain threshold, all DRL agents will sell all their holdings to minimize the loss.

This strategy relies on the following two assumptions:

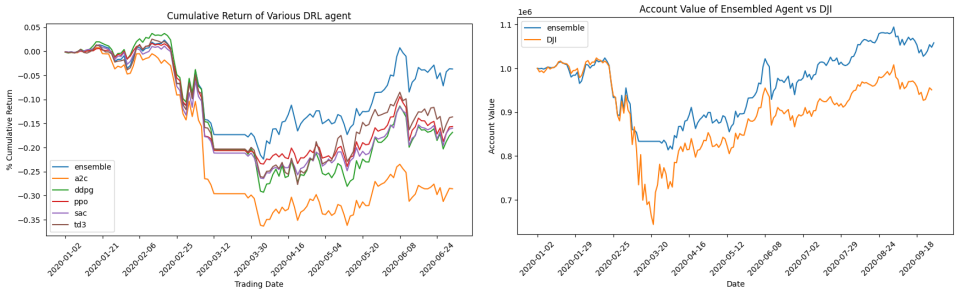
- **Different DRL agents work better under different market dynamics**. Therefore it is reasonable to apply different DRL algorithms in different quarters due to the fast-evolving nature of stock markets.
- **Trends in the stock market persist for more than 3 months**. In the Ensemble Strategy, the most successful agent during a given quarter is designated to trade in the subsequent quarter. Consequently, returns are maximized when these trends persist over the following quarter. On the contrary, if the trends change drastically from one quarter to the next, then it is ineffective to choose agents to trade based on their prior performance.

4 Experiment and Methodology

4.1 Replicating the Results of the Paper

Our goal in this section is to replicate the following results presented in the paper ***i)*** The Ensemble Strategy outperforms all other DRL agents, ***ii)*** the Ensemble Strategy outperforms the benchmark Dow Industrial Average, and ***iii)*** the Ensemble Strategy and all DRL agents outperform the Dow Industrial Average during the market crash in 2020-03.

To achieve this, we trained the Ensemble agent on stock data from '2014-01-01' to '2019-09-30' and tested the agent on stock data from '2019-10-01' to '2020-12-31'. The reason behind the selection of test data is to see how the Ensemble agent handles market crashes and how well it recovers from market crashes.



Analysis: Overall, the Ensemble Strategy outperforms all other DRL agents and DJI during the market crash and post-market-crash recovery phase.

Before the market crash, The performance of various DRL agents is indistinguishable in terms of cumulative return, with DDPG slightly outperforming other agents.

During the market crash from `2020-03-01` to `2020-04-01`, all agents detected the rapid drop in stock prices via the Turbulence Index and sold all their holdings at the same time. However, the Ensembled Strategy is smarter in risk management which minimizes its loss during unstable periods. DJI, which is not equipped with turbulence detection, has a consistently low account balance compared with RL agents.

The Ensemble strategy recovers from the market crash more quickly than all other DRL agents. Looking at `2020-04-01` to `2020-04-16`, we observe that the Ensemble Strategy made the biggest leap among all other agents. At the end of the trading period, the Ensemble Strategy outperformed the majority of DRL agents by 10

4.2 Comparing the environment in the paper against our customized environment

Our goal in this section is to investigate whether encoding more information into the environment will give us a better performance. To do this, we preprocessed the data to include 6 additional

indicators that are common in portfolio optimization and are, what we identified as, the most useful indicators on top of the environment used in the paper. They are Exponential Moving Average (EMA), Bollinger Band (BOLL), Average Directional Movement Index (ADX), Simple Moving Average (SMA), Volume Ratio (VR), and Supertrend.

State Space in Customized Environment: [b_t, p_t, h_t, $MACD_t$, RSI_t, CCI_t,DX_t, EMA_t, BOLL LB_t, BOLL UB_t, $BOLL_t$, ADX_t, SMA_t, VR_t, $SUPERTREND_t$,$Turb_t$]

State Space in Paper: [b_t, p_t, h_t,$MACD_t$, RSI_t, CCI_t,DX_t,$Turb_t$]

Then we trained and tested two Ensemble agents for 6 years and 1 year respectively under different environments, the one presented in the paper, and the customized environment (i.e. One agent is trained & tested with more information available in the environment).

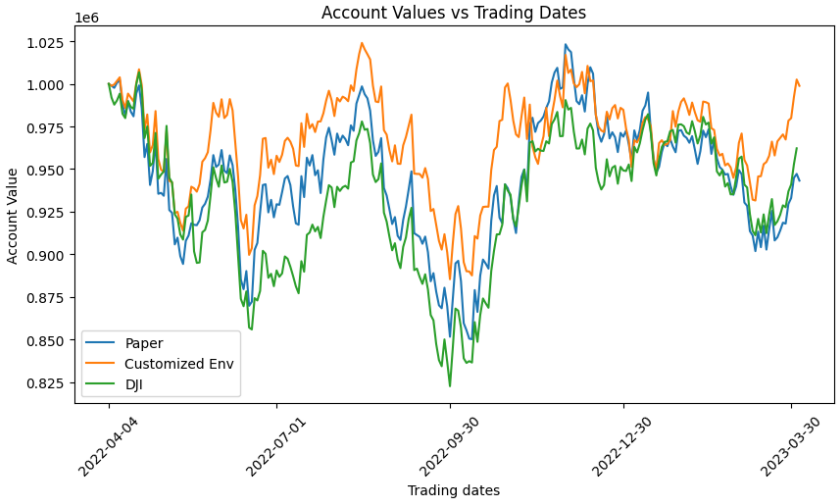


Figure 2: Comparing performance of agents trained in the customized environment, the environment presented in the paper, and the benchmark DJI.

Analysis: The agent trained in the customized environment, with more market dynamic information provided, consistently outperforms the agent trained in the environment with less information available throughout the 1-year trading period. This verifies our hypothesis that infusing pre-computed statistics into the agent facilitates the learning of the agent under limited training resources.

Compared with the benchmark DJI, the agent trained in the environment presented in the paper has a comparable performance, with a slightly better performance in a bullish market (stock prices rising). On the contrary, the agent trained & tested in a customized environment outperforms DJI significantly under all market dynamics.

4.3 Investigating the impact of each individual indicator

In this section, we are exploring the effects of adding individual indicators to the environment. The motivation behind this is to compare the human perception of the importance of certain indicators to their effectiveness in helping RL agents learn.

Firstly, we have manually defined a list of indicators ordered by human perception of their importance: Exponential Moving Average (EMA), Bollinger Band (BOLL), Average Directional Index (ADX), Simple Moving Average (SMA), Volume Ratio (VR), Supertrend, Stochastic RSI (SRSI), Triple Exponential Moving (TEMA), Stochastic Oscillator (SOSC).

Then we create an environment that includes one of the above indicators on top of the environment used in the paper and train & test the Ensemble agent under this new environment.

Analysis: Some indicators that we preceive as important do help RL agents to learn better. A prominent example would be EMA, which we have identified as the most useful indicator, boosted

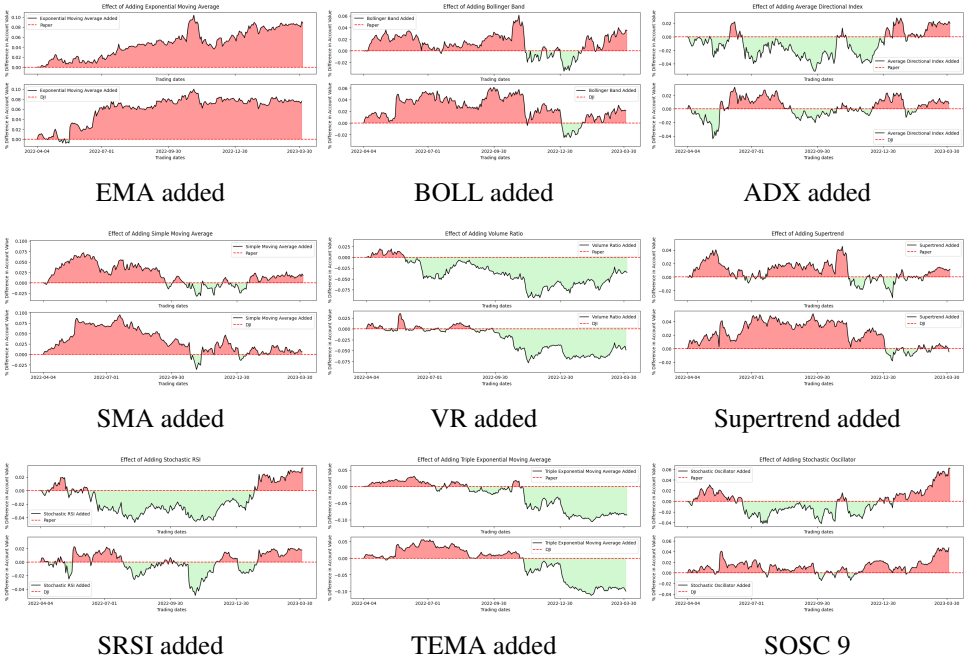


Figure 3: The effect of adding each indicator. Red/green areas mean the agent with such indicator added performs better/worse. The upper plot, for each indicator, outlines how much the agent with this indicator added outperforms the agent without this indicator added throughout the trading period. The bottom plot outlines how much the agent with this indicator added outperforms the benchmark DJI

the performance by 10% by the end of the trading period. Notable, agents trained with EMA outperformed agents trained without EMA on every trading day.

On the other hand, some popular indicators common in human traders undermine the performance of RL agents when added to the environment. For example, the account balance of the agent trained & tested with TEMA added has dropped by 10% by the end of the trading period.

However, we cannot conclude that an indicator is detrimental based on the unsatisfactory result we observe for the following two reasons.

- Some indicators work better when paired with a specific combination of indicators. Nevertheless, determining the optimal combination of indicators requires an exhaustive search which is costly.
- Some indicators are not suitable under certain market dynamics. Experienced human traders know to value indicators differently under different market dynamics. Given the wide range of all possible states, it would be hard for the agent to learn to do so.

4.4 Sentiment Analysis

The core idea behind the Sentiment Analysis script is to leverage sentiment analysis for financial news to inform stock trading decisions. We integrated various Python libraries to automate the process of fetching relevant news articles about specific companies using the News API, and applies sentiment analysis on the text of these articles using the NLTK library's VADER tool. This approach aims to blend quantitative stock data with qualitative news sentiment, offering a holistic view of market influences that could impact stock prices, thus providing traders with deeper insights for making informed trading decisions.

More specifically, we fetched up to 100 most relevant articles for a given company on a given date (e.g 'Apple' on 2020-01-1), then we performed sentiment analysis on each article to obtain a real

number of sentiments $\in [-1, 1]$. Finally, we compute the mean of the sentiments across all articles and bind the result to our data set.

It is important to point out that we have failed to obtain the sentiments for the entire data set (which spans 6 years) due to the limitation on the number of HTTP requests we can send with a free‘newsapi‘ account. However, we have included a successful example of encoding sentiments into the environment of a size-reduced dataset in the attached notebook.

We have determined that the challenge of incorporating sentiment analysis is closely tied to the challenge of acquiring data. Given the time-sensitive nature of reinforcement learning (RL), it is difficult to obtain news data for large stock data sets of multiple companies spanning multiple years.

5 Conclusion and Future Work

In conclusion, we have found that explicitly providing pre-computed statistics facilitates the learning of the Ensemble agent. We also note that a costly exhaustive search is necessary to determine the optimal combination of indicators for the agent. We have shown that including sentiment analysis in the environment is challenging due to the time-sensitive nature of MDP. Nevertheless, we have demonstrated a successful size-reduced example of encoding sentiment analysis on various companies across time steps into the dataset.

5.1 The Future of RL and Fiannce

Looking forward, the integration of reinforcement learning with other advanced techniques such as natural language processing (to analyze financial news) and graph neural networks (to model the interconnectedness of financial entities) promises to further enhance the capabilities of financial market applications. Additionally, the increasing availability of high-frequency, high-dimensional data offers a fertile ground for developing more sophisticated and adaptive RL models.

Overall, reinforcement learning continues to be a vibrant area of research with significant potential to transform financial market operations through automation and optimization of decision-making processes.

5.2 Work Contribution

Herbie replicated the results of the paper and experimented on various indicators. Mohamed implemented sentiment analysis. Herbie and Mohamed co-wrote the report. Parag did initial explorations on FinRL.

References

Andrew, Y., James, X., Yi, W., Yifan, Y., Daniel, Y., Ryan, C., Bosheng, D., Vipin, C., and Xu, S. (2024). Learning the market: Sentiment-based ensemble trading agents.

authors, M. (2021). Algorithmic trading using deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Brigida, I. (2022). Sentiment analysis of financial news.

Chen, Y.-F. and Huang, S.-H. (2021). Sentiment-influenced trading system based on multimodal deep reinforcement learning author links open overlay panel.

Michael, K. and Yuriy, N. (2022). Multi-agent reinforcement learning for financial markets. *Journal of Financial Markets*.

Ritter, G. (2021). Reinforcement learning for optimized trade execution. *Quantitative Finance, 2021*.

Soon, P. C., Tan, T.-P., Chan, H. Y., and Gan, K. H. (2022). A review on sentiment analysis in reinforcement learning model for stock market analysis.

Xiao-Yang, L., Hongyang, Y., Qian, C., and Liuqing, Y. (2020). Learning for automated stock trading: An ensemble strategy. *IJCAI ECAI-2020 Workshop on AI in Finance*.

Zhengyao, J., Dixing, X., and Jinjun, L. (2021). A deep reinforcement learning framework for the financial portfolio management problem. *Journal of Financial Data Science*.