Automated Multiple Stocks Trading

Muzhi Lyu mlyu25@uwo.ca
Master of Data Analytics
Western University
London, ON 437-580-8158, CA

Zirong Liu zliu659@uwo.ca
Master of Data Analytics
Western University
London, ON 519-661-2111, CA

Editor: Muzhi Lyu, Zirong Liu

Abstract

Our study explores the use of Reinforcement Learning (RL) to optimize automated trading systems for the U.S. stock market, using a dataset of hourly historical stock data from Kaggle. We evaluate the performance and adaptability of three RL algorithms: Advantage Actor Critic (A2C), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO), through a training and trading phase to test on unseen data. In particular, PPO shows a promising balance between stability and adaptability, which is essential in the volatile stock market. Our research suggests that an ensemble approach may yield more robust trading strategies, and future work should focus on hybrid models, improved exploration techniques, and the incorporation of complex market indicators to improve the effectiveness of RL models in real-world trading applications.

Keywords: Automated Trading Systems, Reinforcement Learning, Stock Market, Multi-stock Trading

1 Introduction

The proliferation of automated trading systems (ATS) that promise improved returns has been fueled by the application of machine learning techniques to financial markets. Within this landscape, reinforcement learning (RL) has emerged as a key tool for identifying optimal trading strategies amidst the dynamic nature of financial markets. Our study explores the application of RL to the complex domain of stock trading, with a particular focus on multi-stock trading within the US market. The central challenge is to develop profitable trading strategies in the volatile and diverse stock market environment. Traditional methods have been unable to consistently take advantage of market movements due to their static nature and the multitude of factors that influence stock prices. In addition, the need for dynamic strategies that can adapt in real time to optimize financial decisions is hampered by the delicate balance between risk and return. As the financial industry continually seeks strategies that provide a competitive advantage, this issue is of paramount importance.

2 Literature Review

Recent studies have extensively explored the application of Deep Reinforcement Learning (DRL) to automated trading systems (ATS), each providing unique insights into the optimization of trading strategies in the volatile stock market. Ansari et al. (2022) utilized a deep reinforcement learning-based decision support system for automated stock trading, employing a deep-Q network algorithm that integrates both past and predicted future stock price trends as inputs, enhanced with a forecasting network based on a Gated Recurrent Unit (GRU) to optimize buy, sell, or hold decisions. Tran et al. (2023) employed Deep Reinforcement Learning with a Double Deep Q-Network setting and a Sharpe ratio-based reward function to optimize parameters for strategies in automated stock trading, focusing on the cryptocurrency market.

Wu et al. (2020) developed adaptive stock trading strategies using deep reinforcement learning methods, specifically employing Gated Recurrent Units (GRUs) to handle the time-series nature of stock data and implementing two algorithms, Gated Deep Q-learning (GDQN) and Gated Deterministic Policy Gradient (GDPG), to drive trading decisions. Dempster and Leemans (2006) introduced an automated FX trading system using adaptive reinforcement learning, specifically employing a recurrent reinforcement learning (RRL) algorithm, to autonomously execute trades with a focus on risk management and dynamic utility optimization.

Moody and Saffell (2001) developed an automated stock trading system using RRL to optimize trading rules based on a utility function that adjusts for risk preferences, using an approach that dynamically updates the trade size and investment positions in real-time. Bao and Liu (2019) used multi-agent deep reinforcement learning to optimize stock liquidation strategies, focusing on minimizing the costs associated with market impact and risk aversion through simulations that adjust agents' reward functions to encourage cooperative or competitive behaviors among them. Bekiros (2010) introduced an adaptive fuzzy Actor-Critic reinforcement learning system for automated stock trading, which employs a fuzzy rule-based state space modeling and adaptive action selection to enhance prediction and management of financial market dynamics.

Lastly, We found the most helpful reference for us, which is the ensemble strategy by Yang et al. (2020). that combines three deep reinforcement learning algorithms: PPO, A2C, and DDPG. This approach effectively addresses the challenge of devising a profitable trading strategy in the dynamic stock market by leveraging the strengths of each algorithm to maximize investment returns. The study shows that the ensemble strategy not only adapts to different market conditions, but also outperforms individual algorithms and traditional benchmarks in terms of risk-adjusted returns. However, we would like to analyze the effect of these algorithms separately, as we want to analyze the performance for these algorithms whether, like the reading we read, are good for continuous state-action space support.

3 Dataset

Our research on automated trading systems using reinforcement learning began with the selection of a robust dataset from Kaggle created by Oleg Shpagin Shpagin (2024) that includes hourly historical U.S. stock data from January 1, 2022 to March 15, 2024. This

dataset includes ten well-established stocks, randomly selected from the top 30 by market capitalization, such as Apple (AAPL), Microsoft (MSFT), and Nvidia (NVDA). These stocks provide a representative cross-section of the economy, offering a comprehensive view of market dynamics and serving as an ideal basis for developing and testing trading strategies.

In order to ensure that the data was suitable for our analysis, extensive data preparation and cleaning was performed. Initially presented in Moscow time, timestamps were converted to New York time to match NYSE trading hours. Each record was meticulously checked to include critical stock characteristics such as opening, high, low, and closing prices within the hour, as well as trading volume. This preprocessing step was critical to maintaining data integrity and relevance for our study's focus on the NYSE.

In addition, the S&P 500 Index was included as a benchmark indicator, enhancing our dataset by providing a reference point for the overall performance of the U.S. stock market. This integration helps contextualize individual stock movements within broader market trends. For data processing, Python scripts transformed this raw data into a structured format suitable for our reinforcement learning models, removing inconsistencies in date/time formatting and filling in gaps during non-trading periods to ensure a continuous dataset. This prepared data was then divided into two phases: a training phase from January 1, 2022 to June 30, 2023 and a subsequent trading phase from July 1, 2023 to March 1, 2024 to evaluate the performance of the models on unseen data.

In summary, the meticulous preparation of our dataset - enhanced by the strategic inclusion of a key market benchmark - ensures that our research is at the forefront of exploring efficient trading strategies via reinforcement learning. This structured approach allows us to rigorously test and refine the potential of RL algorithms to navigate the complexities of financial markets. However, the scope of the study, which is limited to ten major stocks and a specific post-pandemic period, as well as the reliance on some external data for dividends, introduce specific limitations that require careful interpretation of the results, especially when generalizing to broader market conditions.

4 Algorithms

After careful consideration, we plan to use three algorithms, which are listed below:

A2C is used for its stability and efficiency in processing large amounts of data. A key feature of A2C is its advantage function, which evaluates the relative merit of an action compared to the average. This function significantly reduces the variance of policy gradient updates, thereby increasing the robustness and stability of the algorithm. Such features are particularly beneficial in automated trading, where A2C's ability to manage synchronized gradient updates plays a critical role in adapting to rapid changes in market dynamics.

DDPG provides precise control over continuous action spaces, making it ideal for the nuanced demands of equity trading. DDPG combines the strengths of Q-learning and policy gradients to learn directly from the market's complex and unstructured data. Its deterministic approach helps formulate precise trading actions that optimize strategies for maximum returns. This capability is critical for developing strategies that consistently maximize investment returns.

Known for its efficiency and ease of implementation, PPO is well suited to the fast-paced environment of equity trading. PPO modifies the policy update mechanism to ensure minimal deviation from previous policies, promoting stability and consistent learning. Its rapid adaptability and execution speed are invaluable, allowing it to effectively take advantage of opportunities in volatile market conditions.

The PPO algorithm is a type of policy gradient method for reinforcement learning which balances the twin demands of easy implementation and data efficiency while achieving strong empirical performance. The core principle of PPO is to limit the size of policy updates, which improves training stability.

The application of A2C, DDPG and PPO are suitable for our case of trading multiple stocks, enhancing our ability to deliver superior investment strategies.

5 Environment Setup

Our research uses a stock trading simulation environment developed on top of the OpenAI Gym framework. This simulation, StockTradingEnv, intricately models the nuances of the market, accounting for hourly price fluctuations and transaction costs. Our environment is parameterized with a stock dimension that reflects ten selected stocks, resulting in a state space that includes the agent's account balance, stock prices, and holdings for each stock, with the S&P 500 index serving as the sole technical indicator. The resulting state space has dimension 31 and includes a single cash balance, a pair of open and close prices for each stock, and the value of the S&P 500 index. The action space is congruent with the stock dimension, allowing the agent to take different actions for each stock. Initial parameters, such as initial capital, were set to realistic values to mimic real-world trading conditions, while transaction costs were included to represent the frictional costs experienced in real-world trading scenarios. The environment evolves in one-hour increments, reflecting the hourly trading window in the stock market, thus providing agents with a temporal aspect to consider in their decision making.

When setting up the parameters for our selected algorithms using Stable Baselines 3, we carefully crafted the configuration to match the unique characteristics of stock trading. For the A2C algorithm, we chose a smaller n_steps value of 5 to allow for more frequent policy updates, which are essential for adapting to the volatility of the stock market. The entropy coefficient, set at 0.01, promotes sufficient exploration of the action space without deviating too drastically from the current policy, a balance necessary to navigate the unpredictable shifts in market trends.

The learning rate for PPO, set at 0.00025, was chosen to ensure gradual learning. This mitigates the risk of catastrophic policy updates that could result from overfitting to market noise, a common pitfall in financial applications where noise can easily be mistaken for a trend. A moderate batch size of 64 was chosen to provide a stable estimate of the gradients while allowing the model to update more frequently than larger batch sizes would allow, thus striking a balance between stability and responsiveness.

In the case of the DDPG, the chosen learning rate of 0.001 strikes a balance between fast adaptation and stable convergence. The batch size of 128 ensures that each training batch has enough diversity to prevent overfitting, while the buffer size of 50,000 allows the

algorithm to learn from a wide range of past experience, promoting robustness to market changes.

Overall, these parameter choices are the result of empirical tuning aimed at optimizing the learning process for a complex, non-stationary environment such as the stock market. They reflect a trade-off between rapid adaptation to new data and the need for stable, incremental improvement over time.

6 Empirical Evaluation

Our empirical evaluation begins with an assessment of the complexity of the RL algorithms employed in our trading system - A2C, DDPG, and PPO.

6.1 Complexity

The complexity of A2C stems from its requirement to manage multiple parallel environments, which, while beneficial for sampling diverse experiences, poses challenges for synchronization and averaging of gradients for efficient learning. Conversely, DDPG, an off-policy method, incorporates complex memory management to handle its replay buffer, which is necessary for learning in continuous action spaces and can lead to training stability issues. PPO offers a reduction in complexity compared to its predecessors, such as TRPO, by eliminating the need for complex second-order derivative computations through its clipped surrogate objective function.

6.2 Performance Evaluation

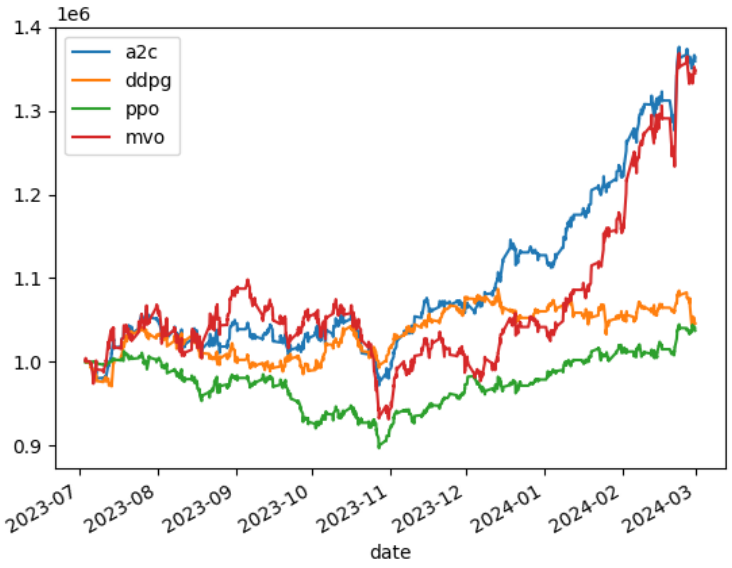

Figure 1: Performance comparison of different reinforcement learning algorithms.

Figure 1 presents a comparative analysis of the profitability of various reinforcement learning algorithms applied to automated multi-stock trading, charting their performance from July 2023 to March 2024. The y-axis denotes the value of the portfolio, starting from an initial fund of 1 billion. The MVO, which represents the classical Mean-Variance Optimization approach in portfolio management, experiences a stark downturn, establishing a baseline for evaluating the more advanced strategies. The PPO algorithm, represented by the green line, demonstrates resilience and a remarkable ability to capitalize on market opportunities, as evidenced by its steady and significant growth trajectory. In contrast, the MVO strategy, represented by the red line, suffers a sharp decline, highlighting the unpredictable and sometimes dangerous nature of algorithmic trading. The A2C algorithm, shown in blue, uses multiple workers to assimilate different market behaviors, demonstrating robustness. Meanwhile, the DDPG algorithm, shown in orange, supports stable and consistent trade executions that can lead to reliable investment growth.

6.3 Usability Considerations

In terms of usability, A2C, while simpler than its predecessor A3C, requires fine-tuning to navigate the bias-variance tradeoff. DDPG requires meticulous hyperparameter adjustments, especially in the exploration-exploitation dynamics, making it less beginner-friendly.PPO stands out for its user-friendly approach, with a simpler objective function and fewer hyperparameters to configure, which is particularly beneficial for novices in the field of RL. Robustness and Adaptability Robustness to market volatility is critical, as A2C's synchronous updates can lag in rapidly evolving market conditions. DDPG's replay buffer is a double-edged sword; it provides a rich learning experience from diverse historical data, but can lead to overfitting if not managed properly. PPO strikes a balance with its adaptive learning approach, allowing the model to respond to new market trends without deviating drastically from prior knowledge.

6.4 Key metrics for empirical comparison

For quantitative evaluation, we measure: Average return: The average return for each algorithm, reflecting the profitability of the trading strategy. Sharpe Ratio: This risk-adjusted return measure evaluates the performance stability and economic viability of the algorithms. Max Drawdown: A measure of potential loss that assesses the ability of algorithms to withstand adverse market movements. Computational Efficiency: Timeliness in training and inference is critical to real-time market responsiveness and is therefore a critical factor in our evaluation.

7 Conclusion

7.1 Optimal Technique

Our empirical analysis of three reinforcement learning techniques—A2C, DDPG, and PPO—unveils distinct strengths in automated stock trading. A2C excels in profitability, exhibiting a pronounced upward trajectory in portfolio value that signifies its potent return-generating capability.DDPG, on the other hand, offers commendable stability, providing a consistent and reliable trading experience.PPO proves to be versatile, adeptly balancing stability with

adaptability—qualities that are particularly valuable in the unpredictable financial market landscape. Its resilience in learning and capacity for incremental adaptation, without drastic policy shifts, highlight its practical applicability in trading. In summary, in the realm of profitability, A2C is the standout choice, as evidenced by its impressive growth in portfolio value, indicative of robust investment returns. For stability, a key consideration in the volatile financial market, DDPG is the strategy of choice due to its smooth and reliable performance.

7.2 Sufficiency for Problem Solving

While A2C's profitable trajectory and DDPG's stability are notable, and PPO's versatility offers a balanced approach, it is important to acknowledge the intricacies of the financial markets that require more than just algorithmic proficiency. Despite the promising results, it would be premature to assert any single algorithm as the ultimate solution for trading challenges. Financial markets are driven by an array of unpredictable factors that can often surpass algorithmic foresight.

7.3 Recommendations for future research

For future research, it would be beneficial to delve into the integration of diverse reinforcement learning approaches to develop hybrid models, enhancing feature engineering with market indicators and macroeconomic factors. Advancements in meta-learning can further tailor the adaptability of trading agents, and refined transaction cost models alongside sophisticated risk management strategies can optimize the financial aspects of trading algorithms. Additionally, employing multi-agent systems can simulate more complex market environments for training, providing insights into collective behavior dynamics. Real-world trials and validations of these systems are essential for practical applications. The promising results from PPO suggest a fertile ground for such explorations, yet it is the amalgamation of A2C's profit-driven performance, DDPG's dependable stability, and PPO's strategic flexibility in an ensemble framework that holds the potential to tackle the intricate nature of financial markets comprehensively. Such an ensemble could capitalize on each method's strengths to create a robust, efficient trading system that stands resilient in the face of market uncertainties.

References

Yasmeen Ansari, Sadaf Yasmin, Sheneela Naz, Hira Zaffar, Zeeshan Ali, Jihoon Moon, and Seungmin Rho. A deep reinforcement learning-based decision support system for automated stock market trading. *IEEE Access*, 10, 2022. doi: 10.1109/ACCESS.2022.3226629.

Wenhang Bao and Xiao-Yang Liu. Multi-agent deep reinforcement learning for liquidation strategy analysis. In *Proceedings of the 36th International Conference on Machine Learning*, page 97, Long Beach, California, 2019. PMLR.

Stelios D. Bekiros. Heterogeneous trading strategies with adaptive fuzzy actor–critic reinforcement learning: A behavioral approach. *Journal of Economic Dynamics & Control*, 34:1153–1170, 2010. doi: 10.1016/j.jedc.2010.01.015.

M.A.H. Dempster and V. Leemans. An automated fx trading system using adaptive reinforcement learning. *Expert Systems with Applications*, 30(3):543–552, 2006. doi: 10.1016/j.eswa.2005.10.012.

John Moody and Matthew Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, 2001. doi: 10.1109/72.935097.

Oleg Shpagin. Usa 514 stocks prices nasdaq nyse. `https://www.kaggle.com/datasets/olegshpagin/usa-stocks-prices-ohlc/data`, 2024. [Online; accessed 24-April-2024].

Minh Tran, Duc Pham-Hi, and Marc Bui. Optimizing automated trading systems with deep reinforcement learning. *Algorithms*, 16(23):1–17, 2023. doi: 10.3390/a16010023.

Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020. doi: 10.1016/j.ins.2020.05.066.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. Available at SSRN, 2020. `https://ssrn.com/abstract=3690996` or `http://dx.doi.org/10.2139/ssrn.3690996`.