Stock Trading on Reinforcement Learning

https://reurl.cc/04Ee0k 待補 網址會放在裡面

林秉陞
National Chengchi University
111307050
111307050@g.nccu.edu.tw

林靖迪
National Chengchi University
111301029
111301029@g.nccu.edu.tw

Abstract

This paper examines the use of reinforcement learning (RL) in stock trading. By employing actor-critic algorithms, the RL agent interacts with a simulated financial environment to optimize trading decisions based on market data and technical indicators. Focusing on stocks like Tesla (TSLA), Broadcom (AVGO), the study integrates sentiment analysis and evaluates the agent's performance using risk-adjusted metrics, such as the Sharpe ratio. Expected results will highlight RL's potential in improving trading strategies and performance.

1. Introduction

Stock trading has become increasingly complex, with traditional methods struggling to keep up with the dynamic nature of markets. The need for intelligent systems that can adapt and optimize trading strategies in real-time is critical. Reinforcement Learning (RL), which enables agents to learn optimal actions through interactions with the environment, provides a promising solution. By learning from market data and continuously improving, RL can offer more flexible and adaptive trading strategies compared to conventional methods.

1.1. Problem formulation

An agent interacts with an environment in RL. In stock trading, trading strategies will be performed by the agent; the data of the stock will be the environment, including stock market data, technical indicators, and more.

State space, the state of the environment at any given time, is set at intervals of one trading day, such as market data (open, high, low, close, volume), technical indicators (moving average, Relative Strength Index), and maybe market sentiment (news articles, social media signals), or macroeconomic factors (interest rate, exchange rate).

Action space, the actions that the agent needs to decide. For the study, the agent can buy, hold, sell, and short sell if possible.

Reward function, the key to guide the agent to maximize long-term success. Return on investment (ROI) and Sharpe ratio can both be a fine indicator for the task. The accurate reward function will be discussed in the near future.

1.2. Literature Review

Taylan and Ekrem [1] had utilized Deep Reinforcement Learning (DRL) approach in automation trading. They defined the problem as Partially Observed Markov Decision Process (POMDP) model and solved it by Twin Delayed Deep Deterministic Policy Gradient (TD3). They considered not only trading but also asset allocation. Moreover, they simulated real life by taking transaction costs like commission fees into account. To implement the trading, market data (i.e., close price), technical indicators (i.e., moving average), and sentiment scores classified by FinBERT [2], a fine-tuned Bert model especially for analyzing sentiments for financial text were set as input. Eventually, with all the data, it got a Sharpe ratio of 3.14 and ±0.40 standard error for a portfolio of Qualcomm (QCOM) and Microsoft (MSFT), whereas it also got 2.68 Sharpe ratio on 10 assets, which are both an outstanding performance. In our study, we are going to modify some of the experiment to observe different insights or even improve the performance.

1.3. Reinforcement Learning Algorithm

RL algorithms are selected based on the complexity of the task and the environment’s properties. Value-based algorithms like Q-learning and Deep Q-Networks (DQN) estimate the value of taking certain actions in particular states. Policy-based algorithms learn the policy directly, maps states to actions, and aim to optimize the policy itself. There are many more algorithms or methods like Monte Carlo methods or Temporal-difference methods, but we prefer actor-critic algorithms at the time. Since it combined the traits of value-based and policy-based approaches at once, which is designed to conquer the constraints of both approaches and enhance the learning efficiency in complex situations. The actor is responsible for choosing actions (the policy) and interact with the environment directly, while the critic evaluates the actions taken by the actor by estimating a value function, which provides feedback to the actor on how to adjust its policy.

2. Method

2.1. Data Collection and Preparation

The study utilizes historical market data from companies such as Tesla (TSLA), Broadcom (AVGO), Taiwan Semiconductor (TSM). Data spanning from January 2015 to November 2024 was sourced from Yahoo Finance. Essential features include Open, High, Low, Close, and Volume prices. Additionally, technical indicators (TIs) such as Relative Strength Index (RSI) and Simple Moving Averages (SMA), were computed to enhance the feature set.

Datasets were divided into training (80%) and testing (20%) subsets. To evaluate the impact of TIs, experiments were conducted using datasets with and without these indicators.

2.2. Reinforcement Learning Algorithms

Three RL algorithms were selected for this study:
The agent operates within a discrete action space:

- **Deep Q-Network (DQN)**: A value-based algorithm that estimates the Q-values of state-action pairs to guide the agent's decisions.
- **Advantage Actor-Critic (A2C)**: A hybrid algorithm combining policy-based and value-based approaches, where the actor selects actions, and the critic evaluates them.
- **Proximal Policy Optimization (PPO)**: A policy-based algorithm that stabilizes training by constraining policy updates, improving learning efficiency.

These algorithms were implemented using the Stable-Baselines3 library, with tailored hyperparameters to optimize trading performance.

2.3. Environment Design

A custom trading environment was built using OpenAI Gym. Key components include:

- **State Space**: A vector comprising market data (OHLC prices, Volume), TIs (e.g., RSI, SMA), and additional features like cash balance and shares held.
- **Action Space**: Discrete actions for buying, selling, or holding a stock.
- **Reward Function**: Defined as the change in total portfolio value, encouraging the agent to maximize returns while minimizing drawdowns.

2.4. Evaluation Metrics

The following metrics were used to assess performance:

- **Total Return**: Percentage change in portfolio value over the test period.
- **Annualized Return**: Average yearly return based on total return and test duration.
- **Sharpe Ratio**: Risk-adjusted return, calculated as the ratio of excess return to standard deviation of returns.
- **Maximum Drawdown**: Largest peak-to-trough decline during the test period, indicating risk exposure.

3. Results and Analysis

3.1. Performance Comparison

	Return	Annually	Sharpe	MDD
TSM	127.88	51.96	0.08	22.49
DQN				
A2C	124.68	50.87	0.08	22.56
PPO				
AVGO	196.92	73.83	0.10	25.47
DQN				
A2C	192.51	72.52	0.10	25.47
PPO	192.51	72.52	0.10	25.47
TSLA	91.36	39.06	0.05	51.09
DQN	86.44	37.23	0.05	51.57
A2C				
PPO	86.44	37.23	0.05	51.57

The results for each RL model were compared to the Buy-and-Hold baseline across multiple companies. Notable findings include:

- **TSM and NVDA**: Despite significant volatility, Buy-and-Hold outperformed RL models in total and annualized returns. RL models struggled with the high drawdowns observed in these stocks.
- **AVGO**: PPO and A2C demonstrated competitive performance, with total returns and annualized returns marginally below Buy-and-Hold. DQN did not execute trades.

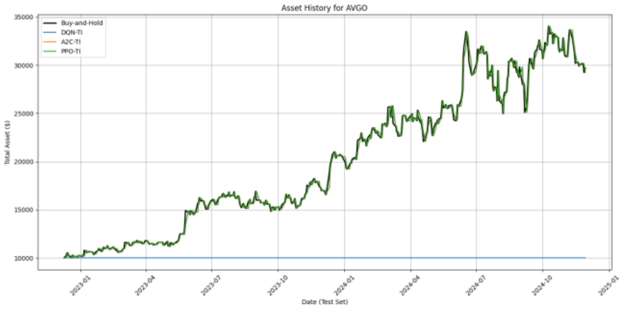
3.2. Analysis of DQN Performance

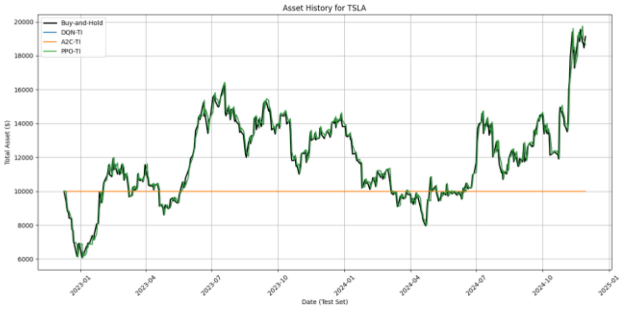
DQN struggled significantly, failing to execute trades for most companies. This can be attributed to the algorithm's reliance on discrete actions and sensitivity to reward sparsity in financial environments. Additionally, the lack of adaptive exploration likely hindered its ability to discover profitable trading patterns.

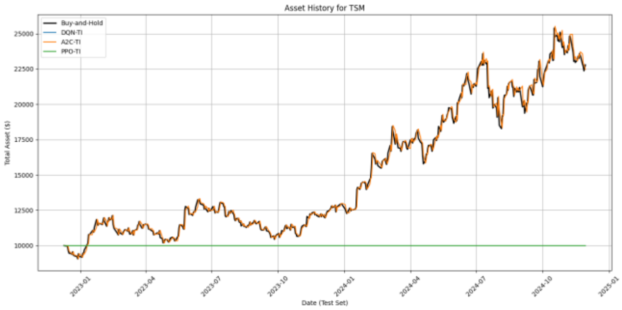
3.3. Visualization of Asset Histories

Figures 1 to 3 illustrate the total asset trajectories for selected companies (AVGO, TSLA, TSM). Key observations include:

- **AVGO**: RL models closely tracked the performance of Buy-and-Hold, with PPO and A2C showing robust asset growth.
- **TSLA**: Volatility impacted RL models' performance, with Buy-and-Hold maintaining higher returns overall.
- **TSM**: PPO demonstrated promising results but failed to surpass Buy-and-Hold, likely due to insufficient reward signal alignment with long-term gains.







3.4. Impact of Technical Indicators

Incorporating technical indicators improved the RL models' ability to make informed decisions, particularly for A2C and PPO. However, the reliance on TIs was not sufficient to overcome the limitations observed in volatile stocks like NVDA.

4. Future Work

Future research should focus on the following areas:

- **Reward Function Optimization**: Design reward functions that penalize high volatility and incentivize risk-adjusted returns to enhance model robustness.
- **Alternative Architectures**: Experiment with hybrid RL architectures (e.g., SAC or TD3) to improve decision-making in high-volatility environments.
- **Incorporation of Market Sentiment**: Integrate sentiment analysis from financial news and social media to enrich state representations.
- **Dynamic Exploration Techniques**: Develop exploration strategies to prevent premature convergence and improve trade discovery in DQN and other RL models.

References.

[1] T. Kabbani and E. Duman, "Deep Reinforcement Learning Approach for Trading Automation in The Stock Market," arXiv preprint arXiv:2208.07165, 2022.

[2] D. Araci, “FinBERT: Financial sentiment analysis with pre-trained language models,” *arXiv preprint arXiv:1908.10063*, 2019.