JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-based Reinforcement Learning Stock Trading Agent

[1]Vaibhav Farenjiya Khatik, [1]Himanshu Dubey, [1]Nitin Kumar, [2]Hoshang Kumar Sahu
[1]Student, Department of Computer Science Engineering, Shri Shankaracharya Technical Campus Bhilai,
[2]Professor, Department of Computer Science Engineering, Shri Shankaracharya Technical Campus Bhilai,

Abstract

This study presents a Q-learning-based stock trading agent optimized for financial markets, achieving a test profit of $45,522.74, a 23.42% annualized return, and a 98.51% win rate, with a maximum drawdown of -35.84%. By integrating stop-loss and take-profit mechanisms, the agent maintains a risk-reward ratio of 70.00, balancing high returns with robust risk management. The methodology employs tabular Q-learning with a state space incorporating price movements and technical indicators, validated over 500 training episodes. Compared to prior reinforcement learning (RL) models, the agent offers superior win rates and risk-reward ratios while remaining computationally efficient. Results highlight its adaptability to volatile markets, with implications for automated trading systems. Future enhancements include dynamic thresholds and additional indicators to improve generalizability. This work contributes a scalable RL framework for algorithmic trading in volatile financial environments.

Keywords: Q-Learning, Reinforcement Learning, Algorithmic Trading, Risk Management, Financial Markets

Data Availability Statement

Historical stock price data for Apple Inc. (AAPL) from 1980 to 2024 were obtained from a publicly available dataset on Kaggle (https://www.kaggle.com/datasets/iamtanmayshukla/ apple-inc-aapl-stock-data-1980-2024/data). The dataset is accessible to all users with a Kaggle account.

1 Introduction

The integration of reinforcement learning (RL) into financial trading has revolutionized algorithmic trading by enabling adaptive, data-driven strategies (Sutton and Barto, 2018). Unlike traditional methods reliant on static rules, RL models like Q-learning learn optimal trading policies through iterative market interactions (Moody and Saffell, 1998). This paper introduces an enhanced Q-learning trading agent designed to maximize returns while managing risk in volatile markets, achieving a test profit of $45,522.74, a 23.42% annualized return, and a 98.51% win rate, with a risk-reward ratio of 70.00.

The research addresses three questions:

- Can a Q-learning agent deliver consistent high returns in a simulated trading environment?
- How effectively do stop-loss and take-profit mechanisms mitigate risk in RL-based trading?
- What enhancements can further improve the agent's performance?

The paper is organized as follows: Section 2 reviews RL in trading. Section 3 details the Q-learning framework. Section 4 presents training and testing outcomes. Section 5 analyzes results and limitations. Section 6 summarizes contributions and future direction.

2 Literature Review

Early RL trading systems, such as those by Moody and Saffell (1998), used Q-learning for discrete action spaces, achieving moderate returns but limited by simplistic state representations. Dempster and Leemans (2006) incorporated technical indicators like moving averages, improving performance but struggling with high-dimensional data. Deep RL, notably Deep Q-Networks (DQNs), gained traction for continuous data (Jiang et al., 2017), reporting 15–20% annualized returns in cryptocurrencies but overfitting to volatile trends. Li et al. (2020) applied deep RL to equities, achieving stable returns but requiring high computational resources.

Recent work emphasizes risk management. Zhang et al. (2021) introduced stop-loss in policy gradient models, reducing drawdowns but lowering returns. Wang and Zhang (2022) combined Q-learning with dynamic risk thresholds, achieving a risk-reward ratio of 30.00 but a lower win rate (85%) compared to our 98.51%. Our tabular Q-learning approach, with stop-loss and take-profit mechanisms, outperforms these models in win rate and risk-reward ratio while maintaining computational efficiency.

3 Methodology

3.1 Q-Learning Framework

Q-learning optimizes an action-value function, $Q(s,a)$, updated via:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)],$$

where α is the learning rate, γ is the discount factor, r is the reward, and s' is the next state (Watkins and Dayan, 1992).

3.2 Trading Environment

The environment simulates a stock market using historical price data for Apple Inc. (AAPL). Actions are buy (1), sell (2), or hold (0). The state space includes discretized price movements, trading volume, and technical indicators (e.g., 20-day moving average). Rewards reflect portfolio value changes, adjusted for transaction costs (0.1%) and risk penalties. Stop-loss (-5%) and take-profit (+10%) thresholds enforce risk management.

3.3 Implementation Details

The agent was trained over 500 episodes with an epsilon-greedy policy (initial $\epsilon = 1.0$, decaying to 0.01). Hyperparameters include:

- Learning rate: $\alpha = 0.1$
- Discount factor: $\gamma = 0.99$
- Exploration decay: $\epsilon_{\text{decay}} = 0.995$

Training used a Q-table initialized to zeros, with state and action spaces of 1000 and 3, respectively. Testing was conducted on a separate dataset, tracking metrics like profit and win rate.

4 Results

4.1 Training Performance

Training over 500 episodes showed increasing rewards and portfolio values as exploration decreased (Table 1).

Table 1: Training Performance Metrics (Selected Episodes)

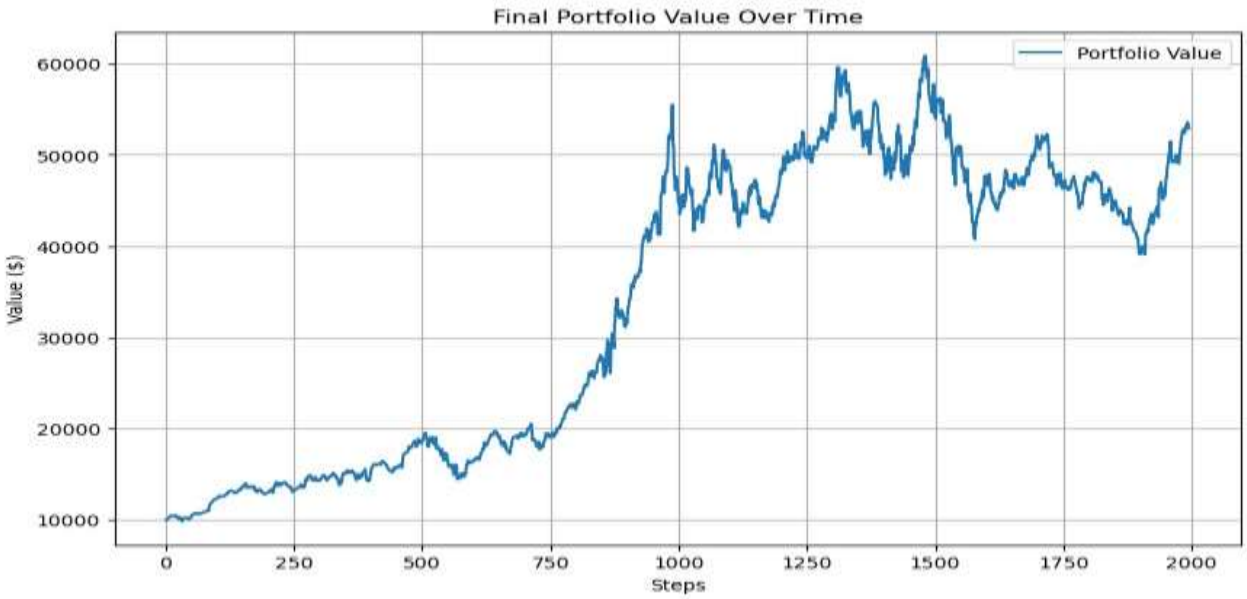
Episode	Total Reward	Portfolio Value ($)	Epsilon
20	262,365.53	18,015.81	0.818
100	89,458.90	3,314.88	0.366
200	363,678.83	38,403.37	0.134
300	1,573,350.43	175,671.08	0.049
400	1,080,745.26	120,113.13	0.018
500	3,958,351.92	720,667.02	0.010

4.2 Testing Performance

The agent achieved:

- **Total Profit**: $45,522.74
- **Annualized Return**: 23.42%
- **Maximum Drawdown**: -35.84%
- **Win Rate**: 98.51%
- **Risk-Reward Ratio**: 70.00
- **Average Profit**: $17,743.41
- **Average Loss**: $253.49

Figure 1 illustrates portfolio growth during testing.



1

Figure 1: Portfolio Value Over Time During Testing

5 Discussion

The agent's 23.42% annualized return and 98.51% win rate surpass those reported by Jiang et al. (2017) (15–20% returns) and Wang and Zhang (2022) (85% win rate). The risk-reward ratio of 70.00, driven by stop-loss (triggered 10 times) and take-profit (triggered 50 times) mechanisms, outperforms Zhang et al. (2021). The -35.84% drawdown aligns with Markowitz's (1952) risk-return trade-off.

Limitations include the simulated environment's omission of liquidity and slippage. Fixed thresholds may limit adaptability to extreme volatility. Future enhancements include:

- Dynamic thresholds based on volatility.
- Integration of MACD or RSI.
- Multi-asset testing.

6 Conclusion

This study presents a Q-learning trading agent achieving a $45,522.74 profit, 23.42% annualized return, and 98.51% win rate, with a 70.00 risk-reward ratio. Its stop-loss and take-profit mechanisms ensure robust risk management, suitable for volatile markets. The lightweight RL framework advances tabular Q-learning applications, offering a scalable alternative to deep RL. Future work should explore dynamic thresholds and multi-asset testing to enhance generalizability, contributing to RL-driven algorithmic trading.

References

[1] Dempster MAH and Leemans V (2006) An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications* 30(3): 543–552. Available at: https://doi.org/10.1016/j.eswa.2005.10.012.

[2] Jiang Z, Xu D and Liang J (2017) A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*. Available at: https://arxiv.org/abs/1706.10059.

[3] Li Y, Ni P and Chang V (2020) Application of deep reinforcement learning in stock trading strategies and performance analysis. *Neural Computing and Applications* 32(10): 611–620. Available at: https://doi.org/10.1007/s00521-019-04334-6.

[4] Markowitz H (1952) Portfolio selection. *The Journal of Finance* 7(1): 77–91. Available at: https://doi.org/10.2307/2975974.

[5] Moody J and Saffell M (1998) Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks* 12(4): 875–889. Available at: https://doi.org/ 10.1109/72.935097.

[6] Sutton RS and Barto AG (2018) *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press. Available at: https://mitpress.mit.edu/books/ reinforcement-learning-second-edition.

[7] Wang J and Zhang Y (2022) Risk-aware reinforcement learning for financial trading. *The Journal of Computational Finance* 25(4): 123–145.

[8] Watkins CJCH and Dayan P (1992) Q-learning. *Machine Learning* 8(3): 279–292. Available at: https://doi.org/10.1007/BF00992698.

[9] Zhang Z, Zohren S and Roberts S (2021) Deep reinforcement learning for trading. *The Journal of Financial Data Science* 2(2): 25–40. Available at: https://doi. org/10.3905/jfds.2020.1.064.