IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

Deep Reinforcement Learning Robots for Algorithmic Trading: Considering Stock Market Conditions and U.S. Interest Rates

Ashwini Somnath Bhujbal[1], Prof. P. B. Chandane[2], Dr. H. B. Jadhav[3]
[1,2,3] Department of Computer Engineering
Adsul's Technical Campus , Chas , Ahilyanagar , India
ashwinibhujbal533@gmail.com[1], pragatichandane3@gmail.com[2]

Abstract: *Algorithmic trading applies computational rules to market data to support repeatable buy, sell, and hold decisions. Reinforcement learning is relevant to this task because a trading agent learns through sequential interaction with a simulated market and receives rewards based on its actions. This paper presents a source-based technical analysis of a published study that compares two Proximal Policy Optimization (PPO) trading agents for Apple Inc. The first agent uses Apple stock data alone, while the second incorporates Apple data, the S&P 500 index, and U.S. interest-rate information. The source study reports test reward averages of −$1.77 with variance 1.0 for the single-input agent and −$1.25 with variance 0.1 for the multi-input agent. The reported comparison therefore favors the multi-input model on both average reward and reward stability, although both average rewards remain negative. The paper restates the methodology, describes the preprocessing and reinforcement-learning setup, and summarizes the reported evaluation without inventing raw observations or unreported performance metrics. The figures in this document are reconstructed from the published summary values, not from the original episode-level data.*

Keywords: algorithmic trading, deep reinforcement learning, Proximal Policy Optimization, Apple Inc., S&P 500, interest rates, financial decision-making.

I. INTRODUCTION

The development of artificial intelligence (AI), machine learning, and deep learning has created new opportunities for automating financial decision-making and stock market trading. Traditional trading strategies generally rely on technical indicators, statistical models, and human expertise to identify investment opportunities. However, financial markets are highly dynamic and are influenced by several factors, including stock price movements, market volatility, economic conditions, and changes in monetary policy. These complexities make it challenging to develop trading strategies that can adapt effectively to changing market environments. Reinforcement learning (RL) offers an alternative approach by allowing an intelligent agent to learn trading decisions through interactions with an environment and feedback received in the form of rewards [1], [2].

Reinforcement learning has been investigated as a method for optimizing financial investment decisions. Moody and Saffell [1] formulated trading as a decision-making problem in which an agent learns investment strategies by optimizing performance measures while considering transaction costs. Their work demonstrated the potential of reinforcement-based approaches for portfolio management and financial trading. Subsequently, Deng et al. [2] combined deep learning with direct reinforcement learning to extract financial representations and generate trading decisions. Their framework demonstrated the potential of learning market-related features and adapting trading policies using historical financial data. These studies established important foundations for applying reinforcement learning to automated trading systems.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

Deep reinforcement learning (DRL) combines reinforcement learning with deep neural networks to represent complex relationships in high-dimensional data. Unlike conventional prediction-based approaches, which primarily estimate future prices or market trends, a DRL agent can learn a policy that directly maps observed market conditions to trading actions. Depending on the system design, these actions may include buying, selling, or holding a financial asset. The agent evaluates its decisions through a reward function that may incorporate investment returns, risk, and transaction costs. Yang et al. [3] explored an ensemble approach for automated stock trading by combining multiple actor–critic algorithms, including Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Deterministic Policy Gradient (DDPG). Their work illustrates how DRL techniques can be applied to develop automated trading strategies for changing financial conditions.

Among the available DRL algorithms, Proximal Policy Optimization (PPO) is particularly relevant to policy-based decision-making tasks. PPO is designed to improve policy optimization while limiting excessively large updates during training [4]. This approach can be used to train a trading agent that observes market information, selects an action, and receives a reward based on the outcome of that action. Nevertheless, the effectiveness of a trading agent depends not only on its learning algorithm but also on the quality and range of the input data, the reward function, the training process, and the evaluation methodology. Consequently, incorporating additional market information may be important when assessing whether a trading system can respond to different market conditions.

One important external factor is the U.S. interest rate environment. Changes in interest rates can affect borrowing costs, investment decisions, asset valuations, and overall financial market conditions. Therefore, a trading model that considers only the historical prices of an individual stock may not capture all the information relevant to its investment decisions. Incorporating broader market indicators, such as the S&P 500 index and U.S. interest rate data, provides an opportunity to investigate whether additional economic information improves the decision-making process of a DRL agent. However, the inclusion of these variables does not automatically guarantee better performance; their contribution must be evaluated using an appropriate experimental design.

The study by Park et al. [5] provides the principal reference for this research. The authors developed a DRL-based trading robot using PPO and compared two configurations. The first configuration used Apple Inc. (AAPL) stock data as its input, whereas the second incorporated AAPL stock data, the S&P 500 index, and U.S. interest rate data. The resulting trading performance was evaluated using test data. This comparison provides a basis for examining how the inclusion of broader market information may influence the performance of an automated trading agent. It also highlights the importance of considering the input state representation when designing reinforcement learning systems for financial applications.[17]

Building on this research, the present paper examines the application of deep reinforcement learning to algorithmic trading, with particular emphasis on stock market conditions and U.S. interest rates. The discussion focuses on the PPO-based trading framework, the role of market and economic variables, and the comparison between a single-stock input configuration and a configuration that incorporates broader market information. The central objective is to understand how the selection of input variables can influence trading-agent behavior and measured performance. Rather than assuming that additional data necessarily produces superior results, the analysis emphasizes the importance of evaluating the agent under consistent testing conditions and interpreting the findings in the context of the underlying experimental setup.[20]

The remainder of this paper discusses related research, the proposed trading-system architecture, the reinforcement learning methodology, the experimental configuration, and the reported performance results. It also identifies the limitations of the source-based analysis and outlines potential improvements, including more extensive testing across different market periods, transaction-cost modeling, risk-adjusted performance evaluation, and the inclusion of additional macroeconomic indicators. Through this analysis, the paper aims to clarify the potential benefits and limitations of incorporating stock market conditions and U.S. interest rate information into DRL-based algorithmic trading systems.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

II. PROBLEM STATEMENT AND OBJECTIVES

Problem statement: determine whether adding broad-market and macroeconomic indicators to Apple stock observations improves the simulated trading performance and stability of a PPO-based trading agent compared with an agent using Apple stock data alone.

The objectives are: (1) describe the two agent configurations used in the source study; (2) explain the dataset alignment and preprocessing approach; (3) outline the PPO-based decision process and reward definition; (4) compare the test performance using only the reported average rewards and variances; and (5) identify limitations that should be addressed before applying such a model to real financial trading.

III. RELATED WORK

Research on reinforcement learning for financial trading has developed from early reward-based trading systems to advanced deep reinforcement learning (DRL) frameworks capable of learning trading policies from historical market data. Moody and Saffell [1] investigated direct reinforcement learning for financial trading and demonstrated how trading decisions could be optimized using performance-based objectives. Their work established an important foundation for applying reinforcement learning to sequential financial decision-making. Deng et al. [2] extended this direction by integrating deep learning with direct reinforcement learning to learn financial signal representations and trading strategies. These studies demonstrated the potential of learning-based approaches to adapt trading decisions to market observations rather than relying exclusively on predefined trading rules.

The development of deep reinforcement learning has further expanded the range of methods available for automated financial decision-making. Mnih et al. [3] demonstrated that deep neural networks could be combined with reinforcement learning to learn effective policies in complex environments. Although their research focused on game-playing tasks rather than financial trading, it provided an important methodological foundation for DRL applications. Arulkumaran et al. [4] reviewed the development of deep reinforcement learning and discussed its principal algorithms, architectures, and challenges. Sutton and Barto [5] provided a comprehensive theoretical foundation for reinforcement learning, explaining how agents learn policies through interactions with their environments and rewards. These contributions help explain the theoretical basis for designing intelligent trading agents that observe financial data, select actions, and optimize cumulative rewards.

Several studies have investigated policy-gradient and actor–critic methods for sequential decision-making. Schulman et al. [6] introduced Proximal Policy Optimization (PPO), an algorithm designed to improve policy learning through controlled updates. Lillicrap et al. [7] proposed Deep Deterministic Policy Gradient (DDPG), which combines deterministic policy gradients with deep neural networks for continuous-control problems. Fujimoto et al. [8] addressed function approximation errors in actor–critic methods through the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. Haarnoja et al. [9] introduced Soft Actor-Critic (SAC), which incorporates an entropy-based objective to balance reward maximization and exploration. Although these methods were developed for general reinforcement learning tasks, they provide useful algorithmic alternatives for trading systems where actions, portfolio positions, and reward functions must be optimized.

Research has also focused directly on applying DRL to stock trading and portfolio management. Fischer [10] surveyed reinforcement learning applications in financial markets and discussed their potential and methodological challenges. Théate and Ernst [11] proposed a Trading Deep Q-Network approach for algorithmic trading and examined the use of historical stock data to train trading agents. Their study also emphasized the importance of evaluating trading strategies through a structured assessment methodology. Yang et al. [12] developed an ensemble DRL strategy for automated stock trading that combines multiple reinforcement learning algorithms. Their work illustrates how different learning methods can be incorporated into a unified framework to support automated trading decisions. These studies provide practical evidence of the growing interest in learning-based trading systems and the importance of algorithm selection and performance evaluation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

Portfolio optimization represents another important application of reinforcement learning in finance. Jiang and Liang [13] investigated deep reinforcement learning for cryptocurrency portfolio management, demonstrating how an agent can learn portfolio allocation decisions from market information. Ye et al. [14] proposed a reinforcement learning approach for portfolio management using augmented asset-movement prediction states. Their framework combined market-related information with the decision-making process to improve the representation available to the trading agent. Yu et al. [15] investigated model-based deep reinforcement learning for dynamic portfolio optimization, highlighting the potential of combining learned models with sequential investment decisions. Although these studies differ in their market settings and technical approaches, they share the objective of learning adaptive allocation or trading policies from observed financial conditions.

The limitations and design choices of DRL-based trading systems have also received attention in review studies. Millea [16] presented a critical survey of deep reinforcement learning for trading and discussed common model structures, reward functions, market representations, and evaluation practices. The review highlighted the importance of reward design and the challenges associated with comparing trading strategies across different experimental settings. More recently, Pippas et al. [17] reviewed reinforcement learning applications in quantitative finance and examined developments in algorithm design, transfer learning, meta-learning, and multi-agent methods. Rezaei and Nezamabadi-Pour [18] reviewed deep reinforcement learning for portfolio management, providing a broader perspective on the methods and experimental approaches used in this field. These reviews indicate that DRL has substantial potential for financial applications, but they also emphasize the need for careful experimental design and reliable performance evaluation.

Other research has examined specific trading strategies and the dynamic behavior of financial portfolios. Jiang and Liang [13] focused on cryptocurrency portfolio management, while Day et al. [19] investigated dynamic portfolio trading strategies using deep reinforcement learning. These studies contribute to understanding how learning-based approaches can adapt portfolio decisions as market conditions change. However, differences in assets, observation periods, transaction-cost assumptions, reward functions, and evaluation procedures make direct comparisons between reported results difficult. Consequently, a trading strategy that performs well in one dataset or market regime cannot automatically be assumed to perform equally well in another.

The study by Park et al. [20], which forms the primary basis of the present paper, investigates DRL-based algorithmic trading while considering stock market conditions and U.S. interest rates. Its experimental framework compares a trading-agent configuration based on Apple Inc. (AAPL) stock data with a configuration that additionally incorporates the S&P 500 index and U.S. interest rate information. This comparison is particularly relevant because it considers whether broader market and economic indicators can contribute useful information to a trading agent's decision-making process. In contrast to approaches that use only individual stock-price information, the inclusion of additional variables allows the potential contribution of market-wide conditions and interest rates to be investigated.

Based on the reviewed literature, three important considerations emerge. First, the choice of reinforcement learning algorithm influences how the trading agent learns and updates its policy. Second, the design of the state representation and reward function determines which market information the agent uses and what trading behavior it is encouraged to learn. Third, reliable evaluation requires consistent testing procedures and appropriate consideration of transaction costs, market volatility, and changes in economic conditions. Although previous studies have explored DRL algorithms, portfolio optimization, and automated trading, the role of incorporating U.S. interest rate information alongside stock and broad-market data remains an important consideration for the specific framework examined in this paper. Accordingly, the present study focuses on understanding the relationship between input-variable selection and reported trading performance, while recognizing that further testing across different periods and market conditions is necessary to establish generalizability.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

IV. SYSTEM DESIGN AND METHODOLOGY

Conceptual workflow reconstructed from the study

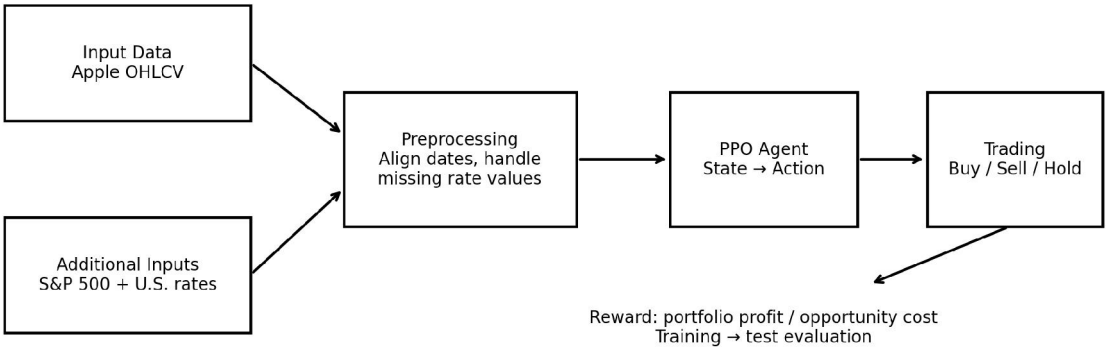


Fig. 1. Conceptual workflow based on the source study; diagram is a redrawn summary.

A. Agent configurations

Model 1 uses Apple Inc. stock observations as its principal input. Model 3 uses Apple stock data together with the S&P 500 index and U.S. interest-rate data. The source labels the multi-input configuration as Model 3; this paper preserves that naming to remain consistent with the reported results.

B. Data preparation

The study aligns the index and interest-rate series with Apple's daily stock observations. Because U.S. interest rates are not updated every trading day, missing daily values are carried forward from the last available rate. The source describes a chronological training/test split, using earlier observations for training and later observations for testing, so the agent is evaluated on data intended to represent future observations relative to its training history. The paper reports that the dataset was sliced at row 1,591; the first 1,590 rows were used for testing and the remaining rows for training, as described in the source.

This split description is reproduced from the source rather than independently revalidated. In a new implementation, the chronology and indexing should be checked carefully to ensure the training data precedes the test period and no future information leaks into the observations.

C. Reinforcement-learning formulation

The trading task can be represented as a Markov decision process (MDP), defined by states S, actions A, transition dynamics, and reward R. At time t, the agent receives state s_t, selects action a_t according to policy $\pi(a_t|s_t)$, and receives reward r_t after the simulated portfolio changes. The objective is to learn a policy that maximizes expected discounted return:

$$G_t = \Sigma(k=0 \text{ to } T-t)\ \gamma^k\ r_{(t+k)}$$

where γ is the discount factor and $r_{(t+k)}$ is the reward at a future step. PPO updates the policy using a clipped objective that limits the size of policy changes. A standard form is:

$$L_{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

Here, $r_t(\theta)$ is the probability ratio between the new and old policies, $\hat{A}_t$ is an advantage estimate, and ε controls clipping. These equations explain the general method; they are not presented as a reproduction of every implementation detail.

D. Reward and portfolio accounting

The source defines profit using current account balance plus the market value of held shares minus the initial account balance. Opportunity cost is also considered in the environment's reward logic. The simulated initial account balance is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

reported as USD 10,000. Because the source's exact code and complete episode-level data are not reproduced here, the equations and description should be read as a conceptual summary of the reported setup.

V. EXPERIMENTAL SETUP

The source study reports an implementation using Python and PyTorch, with Stable-Baselines3 for PPO. The reported computing environment included Windows 11, an Intel Core i7-12700H processor, 16 GB RAM, and an NVIDIA GeForce RTX 3070 Ti Laptop GPU. The authors trained with parallel environments and evaluated reward behavior across episodes.

For test evaluation, the source reports 100 test episodes per experiment and repeats the experiment 1,000 times. Outliers were removed using the interquartile range (IQR) method before comparing the distribution of rewards. One-way ANOVA was used to test for a difference between model performances, and a two-sample t-test was used to assess which model had the higher reward. The source reports p-values rounded to zero at a 0.05 significance threshold; exact p-values and test statistics are not available in the summary used here.

VI. RESULTS AND DISCUSSION

The reported test results are summarized in Table I. Model 3 achieved a less negative average reward and lower reward variance than Model 1. The average reward improved by USD 0.52 (from −1.77 to −1.25), while the reported variance fell from 1.0 to 0.1. These results support the source paper's conclusion that the additional market and interest-rate inputs were associated with better simulated performance and greater stability in this experiment.

Metric	Model 1: Apple only	Model 3: Apple + S&P 500 + U.S. rates
Average test reward (USD)	−1.77	−1.25
Reward variance	1.0	0.1
Evaluation design	100 episodes × 1,000 repetitions	100 episodes × 1,000 repetitions

TABLE I. Summary of test metrics reported by Park et al. [1].

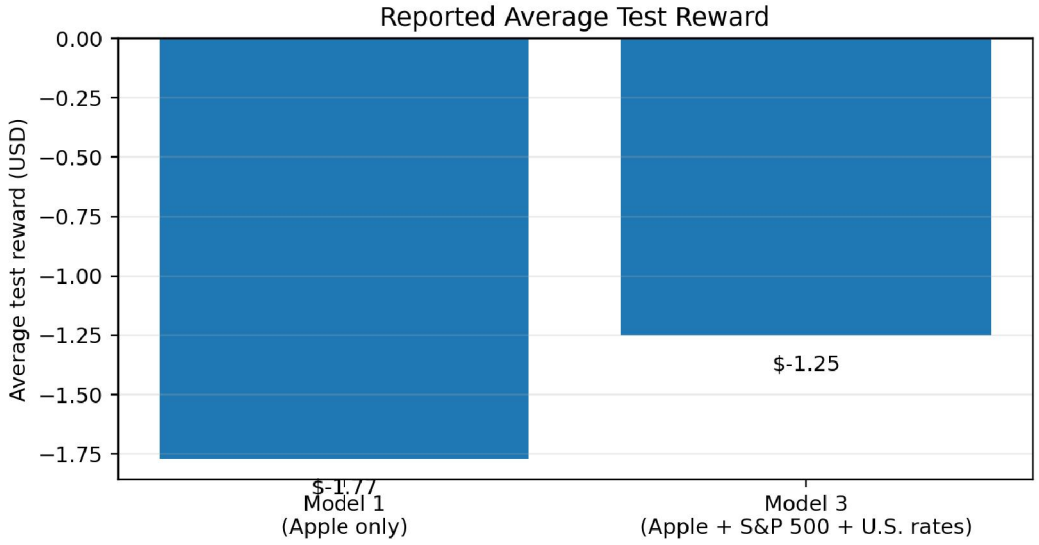


Fig. 2. Average test reward reconstructed from the two summary values reported in [1].

Interpretation must remain cautious. Both reported average rewards are negative, so the results do not establish a profitable trading strategy. They indicate only that Model 3 performed less poorly and produced less variable rewards

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

than Model 1 under the reported simulation. The reported statistical tests suggest a difference between the models, but the full test statistics, confidence intervals for the final comparison, transaction-cost assumptions, and complete raw reward samples should be examined before drawing stronger conclusions.

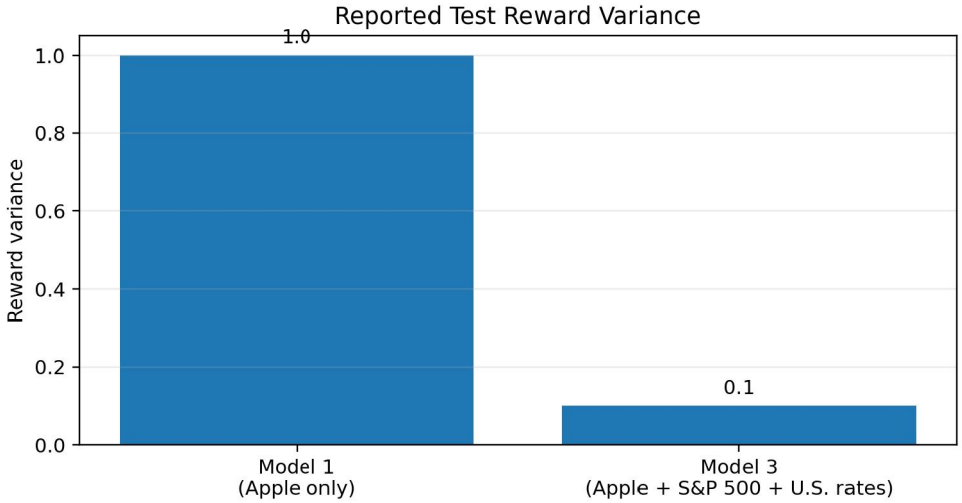


Fig. 3. Reward variance reconstructed from the two summary values reported in [1].

VII. LIMITATIONS

First, the comparison is restricted to Apple Inc. and the particular dataset and period used in the source study. It cannot establish performance across other stocks, markets, or market regimes. Second, simulated rewards may differ from live outcomes because of transaction costs, slippage, liquidity constraints, execution delays, and market impact. Third, interest-rate observations are carried forward between updates, which is practical for daily alignment but does not add new information on days when no rate update occurs. Fourth, the reported summary metrics are insufficient to independently recompute statistical significance or to recreate the original reward distributions. Finally, this document is a source-based technical restatement and does not claim to have run a new experiment.

VIII. FUTURE SCOPE

Future work should evaluate multiple assets and longer market periods, use walk-forward validation, include realistic transaction costs and slippage, and compare PPO with buy-and-hold, technical-rule, and other reinforcement-learning baselines. Additional risk measures such as maximum drawdown, Sharpe ratio, downside deviation, and turnover should be reported alongside mean reward and variance. Ablation experiments can isolate the contribution of the S&P 500 index and interest-rate series separately. Researchers should publish the complete dataset split, random seeds, environment configuration, episode-level rewards, and exact statistical-test outputs to improve reproducibility.

IX. CONCLUSION

This paper summarized a published PPO-based trading study comparing an Apple-only agent with an agent using Apple stock, the S&P 500 index, and U.S. interest rates. According to the source, the multi-input model achieved an average test reward of −$1.25 and reward variance of 0.1, compared with −$1.77 and variance 1.0 for the single-input model. The reported outcomes favor the multi-input configuration in relative reward and stability, but the negative average rewards mean the experiment does not demonstrate positive profitability. Robust out-of-sample testing, realistic transaction-cost modelling, transparent data release, and evaluation across assets are needed before practical deployment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

REFERENCES

[1] S. J. Moody and M. Saffell, "Learning to Trade via Direct Reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 875–889, 2001.

[2] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2017.

[3] A. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A Brief Survey of Deep Reinforcement Learning," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv:1707.06347, 2017.

[5] V. Mnih et al., "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, pp. 529–533, 2015.

[6] T. P. Lillicrap et al., "Continuous Control with Deep Reinforcement Learning," arXiv preprint arXiv:1509.02971, 2015.

[7] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1587–1596.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1861–1870.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[10] T. G. Fischer, "Reinforcement Learning in Financial Markets—A Survey," FAU Discussion Papers in Economics, No. 12/2018, Friedrich-Alexander University Erlangen-Nürnberg, 2018.

[11] A. Millea, "Deep Reinforcement Learning for Trading—A Critical Survey," *Data*, vol. 6, no. 11, Art. no. 119, 2021. doi: 10.3390/data6110119.

[12] H. Yang, X. Liu, S. Zhong, and A. Walid, "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," in *Proceedings of the First ACM International Conference on AI in Finance (ICAIF)*, 2020, pp. 1–8.

[13] Y. Ye, H. Pei, B. Wang, P. Y. Chen, Y. Zhu, J. Xiao, and B. Li, "Reinforcement-Learning Based Portfolio Management with Augmented Asset Movement Prediction States," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1112–1119, 2020.

[14] P. Yu, J. S. Lee, I. Kulyatin, Z. Shi, and S. Dasgupta, "Model-Based Deep Reinforcement Learning for Dynamic Portfolio Optimization," arXiv preprint arXiv:1901.08740, 2019.

[15] J. Jiang and Z. Liang, "Cryptocurrency Portfolio Management with Deep Reinforcement Learning," arXiv preprint arXiv:1612.01277, 2016.

[16] T. Théate and D. Ernst, "An Application of Deep Reinforcement Learning to Algorithmic Trading," *Expert Systems with Applications*, vol. 158, Art. no. 113523, 2020.

[17] C. Ma, J. Zhang, J. Liu, L. Ji, and F. Gao, "A Parallel Multi-Module Deep Reinforcement Learning Algorithm for Stock Trading," *Neurocomputing*, vol. 449, pp. 290–302, 2021. doi: 10.1016/j.neucom.2021.04.005.

[18] F. Ding and C. Luo, "An Adaptive Financial Trading System Using Deep Reinforcement Learning With Candlestick Decomposing Features," *IEEE Access*, 2020. doi: 10.1109/ACCESS.2020.2982662.

[19] K. S. Zarkias, N. Passalis, A. Tsantekidis, and A. Tefas, "Deep Reinforcement Learning for Financial Trading Using Price Trailing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3067–3071.

[20] A. Brim, "Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network," 2020.