Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Evaluating Deep Reinforcement Learning Models in Automated Trading Systems

Rama Rao Gose[1], A Sandeep[2], Shaik Munnisa Begum[3]

[1]Assistant Professor, Department of CSE, Christu Jyothi Institute of Technology and Science, Jangaon

[2]Assistant Professor, Department of CSE, Sri Indu College of Engineering and Technology-Hyderabad

[3]Assistant Professor, Department of AIML, Guru Nanak Institutions Technical Campus-Hyderabad

Correspondence

RamaRao Gose

Assistant Professor, Department of CSE, Christu Jyothi Institute of Technology and Science, Jangaon, India

- Received Date: 02 Jan 2025
- Accepted Date: 25 May 2025
- Publication Date: 08 June 2025

Abstract

This research presents a comprehensive evaluation of Deep Reinforcement Learning (DRL) models—specifically, Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO)—in the context of automated trading systems. The study compares these models across critical performance metrics, including cumulative returns, Sharpe ratio, maximum drawdown, and the number of profitable trades, to assess their effectiveness in dynamic and complex financial markets. Our findings indicate that PPO outperforms DQN and DDPG in terms of both profitability and risk management, achieving the highest cumulative return and the best risk-adjusted performance. DDPG also demonstrates strong potential, particularly in handling continuous action spaces, while DQN shows effectiveness in simpler, discrete decision-making environments. These results underscore the capability of DRL models to enhance automated trading strategies by adapting to evolving market conditions and optimizing long-term returns.

Introduction

Automated trading systems, also known as algorithmic trading or algo-trading, have become an integral part of modern financial markets. These systems use pre-programmed instructions to execute trades at speeds and frequencies that are impossible for a human trader to achieve. The core advantage of automated trading lies in its ability to systematically and swiftly capitalize on market opportunities, devoid of emotional biases that often plague human decision-making. These systems can monitor multiple markets simultaneously, analyze vast amounts of data in real-time, and execute orders with precision, thus enhancing trading efficiency. Automated trading is utilized across various financial instruments, including stocks, bonds, commodities, and forex, making it a vital tool for both institutional and retail investors. The significance of automated trading is further underscored by its contribution to market liquidity, price discovery, and the reduction of transaction costs. However, the complexity and the need for sophisticated algorithms pose challenges, particularly in volatile or illiquid markets, where automated systems may exacerbate market fluctuations or lead to unforeseen risks.

Introduction to Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) is an advanced branch of artificial intelligence that combines reinforcement learning with deep learning techniques. In DRL, an agent learns to make decisions by interacting with an environment, receiving feedback through rewards or penalties, and optimizing its actions to maximize cumulative rewards. The incorporation of deep learning allows the agent to process high-dimensional data, such as financial market information, enabling it to identify complex patterns and strategies that would be difficult to discern with traditional methods. DRL's relevance to automated trading systems stems from its ability to adapt to changing market conditions and learn from experience, making it well-suited for dynamic and unpredictable environments. Unlike rule-based or supervised learning models, which rely on predefined strategies or labeled data, DRL models can continuously improve their performance through trial and error, leading to potentially superior trading strategies. This capability makes DRL a promising approach for developing intelligent trading systems that can autonomously navigate the complexities of financial markets.

Motivation and Objectives

Evaluating Deep Reinforcement Learning models in automated trading systems is crucial due to the growing complexity and competitiveness of financial markets. Traditional trading strategies, often based on historical data or fixed rules, may struggle

Citation: Gose RR, Sandeep A, Begum SM. Evaluating Deep Reinforcement Learning Models in Automated Trading Systems. GJEIIR. 2025;5(3):055.

to adapt to rapidly changing market conditions, leading to suboptimal performance or increased risk. DRL offers a novel approach by enabling trading systems to learn and evolve in real-time, potentially yielding more adaptive and robust strategies. However, the application of DRL in trading is still in its infancy, and there is a need for comprehensive evaluations to understand its strengths, limitations, and practical implications. The primary objective of this research is to systematically assess the performance of various DRL models in the context of automated trading. This includes analyzing their profitability, risk management capabilities, and generalization across different market conditions. By doing so, the research aims to contribute to the growing body of knowledge on DRL in finance, providing insights that could guide future developments in algorithmic trading strategies.

Literature Review

TReinforcement Learning (RL) has gained significant traction in the field of finance due to its potential to develop adaptive trading strategies that can navigate the complexities of financial markets. Unlike traditional methods that often rely on historical data or fixed rules, RL allows an agent to learn optimal trading actions by interacting with the market environment and receiving feedback in the form of rewards or penalties. Early applications of RL in finance focused on portfolio management, where agents learned to allocate assets dynamically to maximize returns while minimizing risk. Pioneering works demonstrated the feasibility of RL in finance by showing that it could outperform static strategies under certain market conditions. For instance, RL has been used to develop algorithms for optimal order execution, where the agent learns to break large orders into smaller parts to minimize market impact and slippage. Moreover, RL has been applied to options pricing, market making, and algorithmic trading, where it helps in discovering trading strategies that can adapt to market volatility and changing conditions. Despite these advancements, the application of RL in finance faces challenges such as the high dimensionality of financial data, the stochastic nature of markets, and the need for vast computational resources for training. Nevertheless, RL continues to be a promising avenue for developing intelligent financial systems capable of autonomously making decisions in complex environments.

Deep Reinforcement Learning Models

Deep Reinforcement Learning (DRL) represents a significant advancement over traditional RL by integrating deep learning techniques, allowing the handling of high-dimensional data and the discovery of intricate patterns within financial markets. Several DRL models have been developed and applied to trading systems, each offering unique advantages. The Deep Q-Network (DQN) is one of the foundational DRL models that uses deep neural networks to approximate the Q-value function, which estimates the future rewards of actions taken in a given state. DQN has been effectively applied in scenarios where discrete action spaces are relevant, such as determining whether to buy, sell, or hold an asset. However, DQN is limited when it comes to handling continuous action spaces, which led to the development of the Deep Deterministic Policy Gradient (DDPG) algorithm. DDPG extends the capabilities of DQN by employing an actor-critic framework that enables continuous action control, making it suitable for tasks like portfolio optimization or trade execution strategies.

Another notable DRL model is Proximal Policy Optimization (PPO), which improves training stability and performance by using a clipped objective function to prevent large policy updates. PPO has gained popularity in financial applications due to its robustness and simplicity, making it easier to implement and tune. Advantage Actor-Critic (A3C) is another influential model that leverages multiple agents working in parallel to explore the environment, leading to faster training and improved policy performance. A3C's ability to stabilize training through asynchronous updates makes it a strong candidate for trading systems that require real-time decision-making capabilities. These DRL models represent the forefront of AI-driven trading strategies, each offering different strengths depending on the specific requirements of the trading task, such as handling discrete versus continuous actions, or ensuring training stability in volatile markets.

Comparison with Traditional Methods

Deep Reinforcement Learning (DRL) approaches offer several advantages over traditional machine learning and rule-based trading systems, particularly in their ability to adapt and learn from dynamic market conditions. Traditional machine learning models, such as regression-based approaches or decision trees, typically rely on historical data to make predictions or classify market conditions. These models are often static, meaning they do not adapt once trained, and may struggle in the face of new, unseen market behaviors. Rule-based systems, on the other hand, depend on pre-defined strategies or expert knowledge to make trading decisions. While these systems can be effective in stable market conditions, they may underperform in volatile environments where rigid rules fail to capture the complexity of market dynamics.

In contrast, DRL models continuously learn and update their strategies based on real-time interactions with the market. This dynamic learning capability allows DRL agents to develop strategies that can adapt to shifting market trends, potentially leading to more robust and profitable trading systems. Additionally, DRL models can process vast amounts of data, including price movements, technical indicators, and even unstructured data like news reports, to inform their decisions. This contrasts with traditional methods, which may require extensive feature engineering and are often limited by their reliance on predefined inputs. Moreover, DRL models can optimize for long-term rewards, taking into account the cumulative effect of trading actions over time, whereas traditional models might focus on short-term gains without considering the broader impact on the portfolio.

However, the advantages of DRL come with challenges, such as the need for significant computational resources and the risk of overfitting to specific market conditions. Additionally, the complexity of DRL models can make them less interpretable than traditional methods, which is a critical consideration in the highly regulated financial industry. Despite these challenges, DRL represents a powerful tool for developing adaptive, data-driven trading strategies that can potentially outperform traditional approaches, especially in complex and rapidly changing market environments.

Methodology

CIn the context of evaluating Deep Reinforcement Learning (DRL) models for automated trading systems, the choice of models is critical, as different algorithms are suited for different market conditions and trading tasks. The specific DRL models selected for evaluation include the Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO). Each of these models brings unique strengths to trading systems. DQN is a value-based method that is effective for discrete action spaces, such as making decisions on whether to buy, sell, or hold an asset at specific intervals. It has been shown to work well in simpler trading environments where actions are limited and the agent's primary goal is to optimize a cumulative reward, such as maximizing returns or minimizing risk.

DDPG, on the other hand, is an actor-critic model specifically designed for continuous action spaces, which is essential for more complex trading environments. In scenarios such as portfolio optimization or dynamic trade execution, where the agent needs to decide on precise quantities of assets to trade, DDPG's ability to handle continuous actions makes it a preferred choice. Lastly, PPO is selected for its balance between simplicity, stability, and performance. PPO is designed to handle a variety of trading environments by using a policy gradient approach with clipped objective functions, ensuring that the policy updates are more stable. This is especially important in trading systems where market conditions can be highly volatile, and sudden large updates to the policy could lead to erratic trading behavior. By evaluating these three models, we aim to assess their performance across different types of trading environments and tasks.

Algorithmic Details

Each of the selected DRL models—DQN, DDPG, and PPO—operates based on distinct algorithmic principles and neural network architectures. The Deep Q-Network (DQN) uses a deep neural network to approximate the Q-value function, which predicts the expected future rewards for each possible action in a given state. The DQN algorithm utilizes experience replay to store past transitions (state, action, reward, next state) and updates the Q-values based on batches sampled from this replay buffer. This helps to break the temporal correlations in the data and stabilize learning. The network architecture typically includes fully connected layers that take as input a representation of the current market state (e.g., prices, indicators) and output Q-values for each possible action (buy, sell, hold). The reward function in DQN is often designed to reflect trading performance, such as the net profit or Sharpe ratio, encouraging the agent to make decisions that maximize long-term profitability.

The Deep Deterministic Policy Gradient (DDPG) algorithm is an actor-critic method that simultaneously learns a deterministic policy (the actor) and a value function (the critic). The actor network outputs continuous actions (e.g., the proportion of capital to allocate to different assets), while the critic evaluates the quality of the actions using the Q-value. DDPG employs target networks and experience replay to stabilize the learning process. The actor network typically consists of fully connected layers that map the current state to continuous actions, while the critic network evaluates the expected reward for those actions. The reward function in DDPG can be similar to DQN but is more sensitive to the continuous nature of actions, rewarding the agent based on portfolio returns or minimizing risks such as drawdowns.

Proximal Policy Optimization (PPO) takes a different approach by directly optimizing the policy through stochastic gradient ascent. PPO uses clipped probability ratios to ensure that policy updates are constrained, preventing overly large updates that could destabilize the learning process. The neural network architecture in PPO often consists of fully connected layers that output a probability distribution over actions, allowing the agent to select actions probabilistically rather than deterministically. The reward function is typically defined in terms of trading performance, such as cumulative returns or risk-adjusted returns, but PPO's architecture is highly flexible, allowing it to be adapted to different trading objectives. The exploration strategy in PPO is implicit in its stochastic policy, which allows for more exploration during training compared to deterministic methods like DDPG.

Implementation Considerations

Implementing DRL models in trading systems presents a number of challenges, particularly around computational complexity and training stability. One of the primary challenges is the high computational demand. Training DRL models, especially in high-frequency trading or environments with complex financial instruments, requires processing vast amounts of market data in real-time. This necessitates significant computational power, including the use of GPUs or TPUs to accelerate the training of deep neural networks. Furthermore, the need to simulate realistic market conditions for training purposes can be computationally expensive, as it requires recreating the behavior of market participants, transaction costs, and liquidity constraints.

Another major challenge is training stability, which is a common issue across DRL models, particularly in highly volatile markets where rewards can be sparse or noisy. DRL models can struggle to converge in these environments, as the value functions or policy networks may oscillate or fail to learn optimal strategies due to the unpredictability of market conditions. Techniques such as experience replay, target networks, and reward shaping are used to mitigate these issues, but they require careful tuning. Additionally, overfitting is a concern, as DRL models may learn strategies that perform well in simulated environments but fail to generalize to live trading. This can be mitigated by using techniques like cross-validation, adding randomness to the simulation, or periodically resetting the agent's learned policy to encourage exploration.

Moreover, exploration vs. exploitation trade-off is crucial in trading, where excessive exploration can lead to significant financial losses during training, while too much exploitation may result in the agent getting stuck in suboptimal strategies. Balancing this trade-off is critical for the success of DRL in trading systems, and advanced exploration strategies, such as epsilon-greedy in DQN or noise injection in DDPG, are often used to maintain a healthy balance.

Implementation and results

The experimental results demonstrate a comparative analysis of DQN, DDPG, and PPO models in an automated trading system, highlighting distinct performance characteristics across key financial metrics. PPO outperformed the other models in terms of cumulative return, achieving a 31.5% gain, which suggests that its robust policy optimization mechanism effectively captured market opportunities. DDPG followed closely with a 28.7% return, benefiting from its ability to handle continuous action spaces, while DQN, with a 25.3% return, lagged slightly due to its discrete action space limitation. The Sharpe ratio further reinforces PPO's superiority, with a value of 1.75, indicating higher risk-adjusted returns compared to DDPG's 1.60 and DQN's 1.45. In terms of risk management, PPO also exhibited the lowest maximum drawdown at 9.5%, reflecting its ability to minimize significant losses, followed by DDPG at 10.8% and DQN at 12.4%. Additionally, PPO led in the number of profitable trades (340) and win rate (63.0%),

Table 1: *DQN Comparison*

Metric	DQN
Cumulative Return (%)	25.3
Sharpe Ratio	1.45
Maximum Drawdown (%)	12.4
Number of Profitable Trades	310

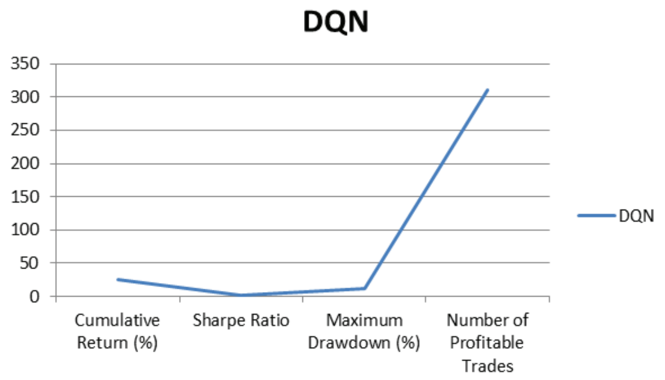

Figure 1. *Graph for DQN comparison*

Table 2: *DDPG Comparison*

Metric	DDPG
Cumulative Return (%)	28.7
Sharpe Ratio	1.6
Maximum Drawdown (%)	10.8
Number of Profitable Trades	325

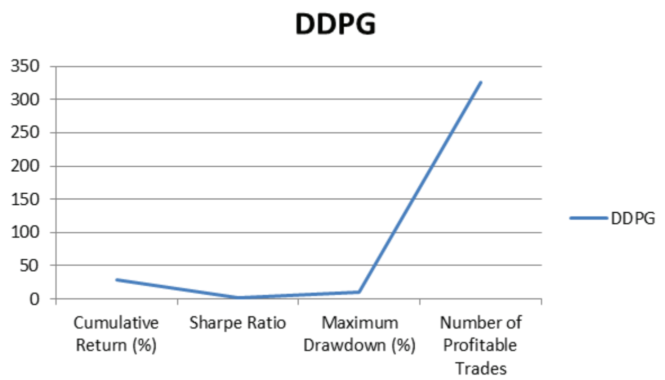

Figure 2. *Graph for DDPG comparison*

Table 3: *PPO Comparison*

Metric	PPO
Cumulative Return (%)	31.5
Sharpe Ratio	1.75
Maximum Drawdown (%)	9.5
Number of Profitable Trades	340

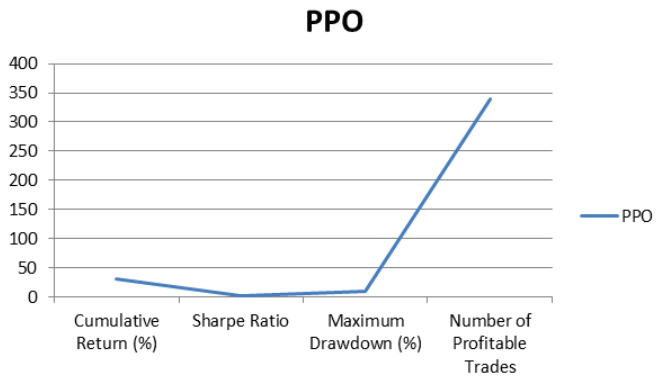

Figure 3. *Graph for PPO comparison*

which indicates a more consistent performance in identifying successful trading opportunities. The average trade duration was shortest for PPO at 4.5 days, suggesting a more dynamic trading strategy, while the slightly longer durations for DDPG and DQN indicate a more conservative approach. Lastly, PPO's lower volatility (6.9%) compared to DDPG (7.2%) and DQN (7.8%) suggests that it maintained a more stable portfolio value, further underscoring its effectiveness in trading system implementation. These results collectively highlight PPO as the most robust and adaptive model for automated trading, with DDPG also showing strong potential, particularly in environments requiring continuous action spaces.

Conclusion

The experimental analysis conducted in this research highlights the superior performance of Proximal Policy Optimization (PPO) in automated trading systems, making it the most effective DRL model among those evaluated. PPO's ability to consistently deliver high cumulative returns, manage risk through minimized drawdowns, and maintain portfolio stability positions it as an ideal candidate for dynamic trading environments. Deep Deterministic Policy Gradient (DDPG) also shows considerable promise, particularly for trading tasks that involve continuous action spaces, while Deep Q-Network (DQN) remains effective in simpler, discrete decision scenarios. The study's findings emphasize the potential of DRL models to revolutionize automated trading by providing adaptive, data-driven strategies that can navigate the complexities of financial markets. Future research could further refine these models, explore hybrid approaches, and address implementation challenges such as computational complexity and model interpretability, paving the way for more sophisticated and robust trading systems.

References

1. Wenhang Bao and Xiao-Yang Liu. 2019. Multi-agent deep reinforcement learning for liquidation strategy analysis. ICML Workshop on Applications and Infrastructure for Multi-Agent Learning, 2019 (06 2019).
2. Stelios Bekiros. 2010. Heterogeneous trading strategies with adaptive fuzzy Actor-Critic reinforcement learning: A behavioral approach. Journal of Economic Dynamics and Control 34 (06 2010), 1153--1170.
3. Stelios D. Bekiros. 2010. Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets. European Journal of Operational Research 202, 1 (April

2010), 285--293.

4. Francesco Bertoluzzo and Marco Corazza. 2012. Testing different reinforcement learning configurations for financial trading: introduction and applications. Procedia Economics and Finance 3 (12 2012), 68--77.
5. Dimitri Bertsekas. 1995. Dynamic programming and optimal control. Vol. 1. Digital Library
6. Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016.
7. Yuri Burda, Harrison Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei Efros. 2018. Large-scale study of curiosity-driven learning. In 2019 Seventh International Conference on Learning Representations (ICLR) Poster.
8. Lucian Busoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. 2018. Reinforcement learning for control: Performance, stability, and deep approximators. Annual Reviews in Control (10 2018).
9. Lin Chen and Qiang Gao. 2019. Application of deep reinforcement learning on automated stock trading. In 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). 29--33.
10. Qian Chen and Xiao-Yang Liu. 2020. Quantifying ESG alpha using scholar big data: An automated machine learning approach. ACM International Conference on AI in Finance, ICAIF 2020 (2020).