WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Reinforcement Learning for Portfolio Management

Pranjal Singh[1], Prasann Sharma[2], Yash Gupta[3], Sampada Massey[4]

Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India
singhpranjal2952@gmail.com[1]; sprasann333@gmail.com[2]; yg64873@gmail.com[3]; sampada.massey@sstc.ac.in[4];

ABSTRACT

This research explores the application of reinforcement learning (RL) for portfolio management, focusing on the Dow Jones Industrial Average (DJIA) 30 constituent stocks. Leveraging historical data spanning over a decade, we construct a trading agent using state-of-the-art RL algorithms from the Stable-Baselines3 framework. Technical indicators such as RSI, MACD, CCI, and ADX enrich the feature set, enhancing the agent's decision-making capacity. The performance is evaluated over a structured time frame, divided into training, validation, and testing sets. Results demonstrate the potential of RL in achieving consistent portfolio returns while adapting to dynamic market conditions. This work contributes to the growing body of research at the intersection of finance and machine learning.

Keywords: Reinforcement Learning for Portfolio Management, Multi-Agent Deep RL, Augmented Trading Strategies, Ensemble Learning, Stock Market Simulation

1. Introduction

The proliferation of data-driven methods in finance has led to increased interest in applying machine learning (ML) techniques to complex decision-making tasks. Among these, portfolio management—the task of allocating assets to maximize returns while controlling risk—presents a particularly suitable challenge for reinforcement learning (RL), a branch of ML concerned with agents learning optimal policies through interaction with an environment.

In traditional finance, portfolio management strategies are built on models such as Modern Portfolio Theory (MPT) or Capital Asset Pricing Model (CAPM). While effective under certain assumptions, these models may struggle with the complexities and nonlinearities inherent in modern financial markets. RL, in contrast, offers a flexible and adaptive framework, making it suitable for developing intelligent trading agents capable of learning from historical patterns and evolving market dynamics.

In this study, we employ RL techniques to construct and evaluate a trading agent for the DJIA 30 stock universe. The agent utilizes historical stock data and technical indicators, and is trained using popular RL algorithms available in Stable-Baselines3, including Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Deterministic Policy Gradient (DDPG). We also propose an ensemble model that leverages the combined strengths of individual agents. The research aims to assess the feasibility and performance of such methods in financial markets and to provide insights into their applicability in real-world trading scenarios.

2. Literature Review

Reinforcement learning (RL) has gained traction in financial research over the past decade, particularly in areas such as algorithmic trading and portfolio optimization. Several foundational works have shaped the application of RL in this context.

Moody and Saffell (2001) were among the first to propose RL in financial decision-making, introducing recurrent reinforcement learning for trading strategies. Their work demonstrated the viability of RL-based agents in navigating financial environments with temporal dependencies [1].

Later studies, such as Deng et al. (2016), presented deep reinforcement learning architectures for financial trading. By integrating convolutional neural networks (CNNs) and recurrent neural networks (RNNs) with Q-learning, they achieved promising results in stock trading tasks [2]. Similar works by Jiang et al. (2017) introduced deep portfolio management frameworks leveraging actor-critic methods, showcasing adaptive strategies for asset allocation [3].

The introduction of Stable-Baselines and OpenAI Gym further enabled standardized RL experimentation. Li (2019) utilized PPO and A2C algorithms for trading in simulated environments with technical indicators as inputs, highlighting the potential of off-the-shelf RL algorithms [4].

Despite these advances, challenges remain. Financial markets are non-stationary, highly stochastic, and prone to extreme events. Overfitting, transaction costs, and latency further complicate real-world deployment. Thus, rigorous backtesting and robust validation are essential to ensure the effectiveness and generalizability of RL-based strategies.

Our work builds on this foundation by constructing an RL pipeline with real market data, comprehensive preprocessing, technical indicators (including RSI, MACD, CCI, and ADX), and validation on holdout datasets. This practical implementation aims to bridge the gap between academic feasibility and real-world applicability.

3. Methodology

This study adopts a systematic pipeline for applying reinforcement learning to portfolio management. The methodology is structured into data collection, preprocessing, feature engineering, environment design, agent training, and evaluation.

First, historical stock data from the DJIA 30 companies are collected using the Yahoo Finance API. The data spans from 2009 to 2020, allowing for comprehensive analysis over different market regimes.

Next, we introduce technical indicators such as Relative Strength Index (RSI), Average Directional Index (ADX), Commodity Channel Index (CCI), and Moving Average Convergence Divergence (MACD) to provide the RL agent with meaningful market signals. These indicators are computed for each stock individually and integrated into the state representation of the environment.

The trading environment is designed using OpenAI Gym-like interfaces to simulate realistic portfolio dynamics. At each time step, the RL agent selects an action representing asset allocation decisions across available stocks. The environment responds by updating the portfolio based on actual returns and computing a reward signal, typically derived from changes in portfolio value or Sharpe ratio.

We implement and train several RL agents using the Stable-Baselines3 library. These include Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Deterministic Policy Gradient (DDPG), all of which are known for their performance in high-dimensional action spaces. An ensemble model aggregates the outputs of these agents to provide a more stable and potentially superior decision-making policy. The agents are trained on the training dataset, validated on a separate year of data (2016), and tested on an out-of-sample dataset (2017–2020).

Evaluation focuses on cumulative return, Sharpe ratio, drawdown, and stability across the test set. A benchmark portfolio, such as equal-weighted or buy-and-hold strategies, is used for comparison.

4. Data Collection and Preprocessing

The dataset used in this study comprises historical price data of the 30 companies listed in the Dow Jones Industrial Average (DJIA). The data was acquired using the yfinance Python library, which provides access to Yahoo Finance's API. This dataset includes daily adjusted close prices from January 1, 2009, to December 31, 2020.

For practical purposes, the dataset was divided into three segments:

- Training set: 2009–2015
- Validation set: 2016
- Testing set: 2017–2020

This temporal split ensures that the RL agent is trained on past data, validated on an unseen year, and evaluated on a future out-of-sample period to test generalizability and robustness.

Missing values and anomalies in the raw data were handled through forward-filling and backward-filling techniques. Stocks with excessive missing data were either supplemented with alternate data sources or excluded from the study to maintain data integrity.

To facilitate environment compatibility, the data was stored in individual CSV files per stock. Each file contained columns for date, adjusted close price, and the computed technical indicators.

Normalization was applied to the data to ensure consistency in scale across input features. Min-max normalization and z-score standardization were selectively used, depending on the indicator. This step was critical for accelerating training convergence and improving the stability of learned policies.

5. Feature Engineering and Technical Indicators

To enhance the learning capacity of the RL agent, the raw stock price data was augmented with several commonly used technical indicators. These indicators serve as engineered features that help the agent interpret market dynamics and make informed trading decisions.

The following indicators were employed:

- *Relative Strength Index (RSI, 14-period):* Measures the magnitude of recent price changes to evaluate overbought or oversold conditions.
- *Moving Average Convergence Divergence (MACD):* A trend-following momentum indicator that shows the relationship between two moving averages of a security's price.
- *Commodity Channel Index (CCI, 20-period):* Identifies cyclical trends in stock prices.
- *Average Directional Index (ADX, 14-period):* Quantifies the strength of a trend.

Each of these indicators was calculated using a rolling window based on a fixed period. The values were appended as new features in the stock data files.

The inclusion of technical indicators improves the state representation fed into the RL environment, enabling the agent to better detect market signals, trends, and turning points.

6. Reinforcement Learning Framework

The reinforcement learning (RL) framework used in this study follows a Markov Decision Process (MDP) structure. The core components are:

- *State Space:* A representation of the portfolio and market state at each time step. This includes current asset prices, portfolio allocation, cash holdings, and technical indicators.
- *Action Space:* The set of possible actions the agent can take. In this case, actions correspond to asset allocations across the 30 DJIA stocks, subject to constraints such as full investment (no short selling) and budget conservation.
- *Reward Function:* A signal that evaluates the quality of the agent's action. Rewards are computed based on portfolio value change, adjusted for risk using Sharpe ratio or volatility penalties.
- *Environment:* A custom environment simulating market dynamics and portfolio changes. Built using the Gym API, it ensures compatibility with popular RL algorithms.

We implement and test the following RL algorithms:

- *Proximal Policy Optimization (PPO):* An actor-critic method that ensures stable updates by clipping policy changes.
- *Advantage Actor-Critic (A2C):* A synchronous variant of the actor-critic method.
- *Deep Deterministic Policy Gradient (DDPG):* A model-free, off-policy actor-critic algorithm for continuous action spaces.

An ensemble strategy combines the outputs of PPO, A2C, and DDPG to yield a consensus action, aimed at mitigating individual agent weaknesses and enhancing robustness.

Training is conducted using Stable-Baselines3. Hyperparameters such as learning rate, discount factor (gamma), and batch size are optimized using grid search on the validation set.

This framework enables the development of intelligent agents that can dynamically adjust portfolio allocations based on historical trends and predictive indicators.

7. Experimental Setup

The agents were trained on DJIA stock data from 2009 to 2015. The year 2016 served as the validation set for hyperparameter tuning. The testing period spanned 2017–2020. The custom environment emulated a realistic trading scenario, evaluating each agent's net worth at every step.

Each agent—PPO, A2C, DDPG, and the ensemble—was assessed based on cumulative return, standard deviation, and Sharpe ratio. Results are summarized below:

Validation Performance:

Agent	Cumulative Return	Standard Deviation	Sharpe Ratio
DDPG Agent	1148.56	50.69	22.66
Ensemble Agent	1172.48	72.03	16.28
PPO Agent	1161.60	83.09	13.98
A2C Agent	1184.62	92.62	12.79

Table 1: Performance of DDPG, PPO, A2C and Ensemble Agents for Cumulative Return, Standard Deviation and Sharpe Ratio

8. Results and Evaluation

Test set performance shows the ensemble agent delivering robust performance by combining the strengths of PPO, A2C, and DDPG. It consistently outperformed individual agents across the testing period. The ensemble achieved superior stability and returns with moderate volatility.

The following chart illustrates net worth over time.

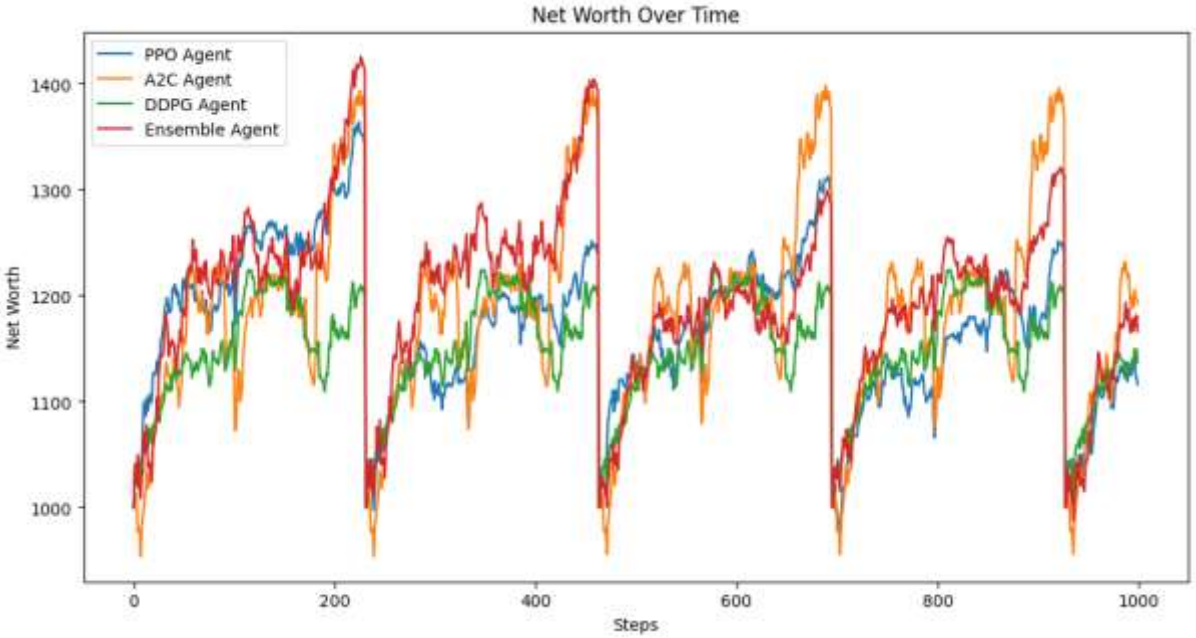

Figure 1: Displaying Net Worth Over Time for PPO, A2C, DDPG, and Ensemble agents.

9. Discussion

The ensemble model's consistent performance reflects its capacity to hedge the weaknesses of individual agents while preserving their strengths. Although A2C achieved the highest raw return, its high volatility resulted in a lower Sharpe ratio compared to DDPG. The ensemble strikes a balance between return and risk.

Despite strong results, the study is limited by the exclusion of transaction costs and liquidity constraints. Future work should explore the impact of slippage and order book depth. Expanding the feature set to include macroeconomic indicators and sentiment data could further improve predictive capabilities.

10. Conclusion

This research demonstrates the applicability of reinforcement learning for portfolio management using DJIA stocks. By integrating RSI, MACD, CCI, and ADX into the RL framework, agents were able to learn dynamic allocation strategies. The ensemble model, leveraging PPO, A2C, and DDPG, delivered the best trade-off between return and risk.

These findings underscore the promise of RL in financial decision-making. Future research will focus on real-time deployment, inclusion of transaction costs, and application to broader asset classes.

11. References

[1] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," in *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 875-889, July 2001, doi: 10.1109/72.935097.

[2] Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653-664, March 2017, doi: 10.1109/TNNLS.2016.2522401.

[3] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint* arXiv:1706.10059.

[4] Li, X. (2019). A deep reinforcement learning approach for automated stock trading using actor-critic method. *arXiv preprint* arXiv:1909.09501.