Hands-on Reinforcement Learning for Stock Trading using FinRL

Author 1 : Moez Kessemtini
Author 2 : Mazen Temani

SympactAI - TAIS NextGen

1 Introduction

Over the past decade, **AI** has revolutionized finance, enabling innovations from algorithmic trading to portfolio optimization. AI-powered systems process vast market data in real time, detect complex patterns, and execute trades within milliseconds. However, the dynamic and uncertain nature of markets demands adaptive models capable of autonomous decision-making. In this context, **Deep Reinforcement Learning (DRL)** offers a powerful framework for building intelligent agents to operate in volatile conditions.
This work explores RL algorithms for stock trading, evaluates the **FinRL** library and alternatives, and compares agent performance in realistic simulations to identify the most effective approach.

2 Methodology

The objective of this work was to design, train, and evaluate a DRL system capable of making **adaptive trading decisions** under realistic market conditions.
The approach integrated the **FinRL** framework with state-of-the-art RL algorithms from Stable-Baselines3, optimized for both long-term returns and adaptability to market shifts.

We began by thoroughly analyzing the **FinRL pipeline**, from raw data ingestion to agent training, in order to understand its architecture and functionalities. The framework is structured around three key pillars:

- **Data Pipeline** : Automated retrieval of historical market data using **YahooDownloader** [1]
- **Custom Trading Environments** : Designed according to the **OpenAI Gym** [2] standard for seamless RL integration.
- **Core RL Structure** : Definition of state space, action space, and reward function tailored to stock trading.

These components are organized within a layered architecture, as illustrated in Figure 1, where the application layer manages use-case-specific configurations, the environment layer encapsulates the trading logic and market simulation, and the agent layer implements RL algorithms for decision-making.

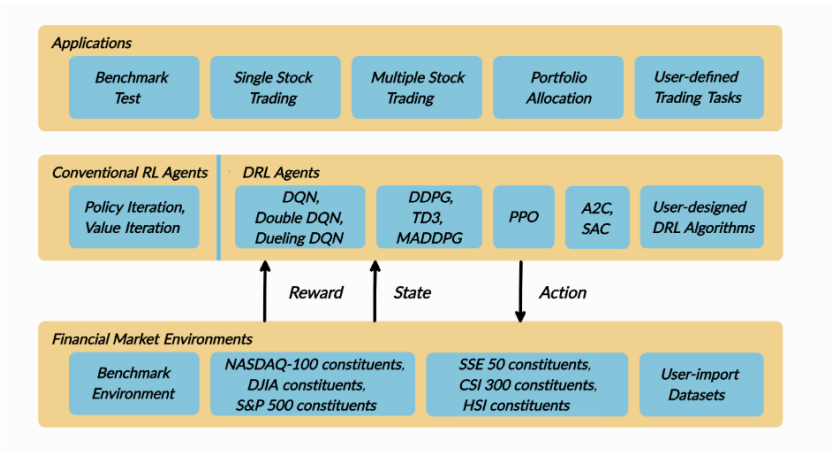

Fig. 1: Architecture and Data Flow of FinRL.

Data Configuration:

- Training period: 2010-01-01 to 2024-07-01
- Testing period: 2024-07-01 to 2025-07-01
- Source: Yahoo Finance [3] daily **OHLCV** data

Feature Engineering:

- The raw OHLCV market data was augmented with additional predictive features to enhance the agent's ability to detect and respond to market regimes.
- Eight widely used technical indicators were computed to capture different market dimensions, including **trend** (e.g., moving averages), **volatility** (e.g., Bollinger Bands), and **momentum** (e.g., Relative Strength Index).
- In addition to asset-specific indicators, **market-wide risk measures** were integrated:
  - The **VIX index** as a proxy for expected market volatility and investor sentiment.
  - A **turbulence index** quantifying abnormal market movements, used to detect periods of financial stress and potential regime shifts.
- These engineered features provided the agent with both micro-level asset information and macro-level market conditions, enabling more adaptive and robust trading decisions.

Environment Design:

- **State Space**: Encodes market features, recent price movements, and portfolio composition, giving the agent awareness of both market conditions and its current exposure.
- **Action Space**: Continuous target portfolio weights across assets, enabling fine-grained position rebalancing.
- **Reward Function**: Percentage change in total portfolio value per timestep, directly aligning optimization with capital growth.

Reinforcement Learning Algorithms:

- Off-Policy: Deep Deterministic Policy Gradient (**DDPG**), Twin Delayed DDPG (**TD3**), Soft Actor-Critic (**SAC**)
- On-Policy: Proximal Policy Optimization (**PPO**), Advantage Actor-Critic (**A2C**)

Training Strategy:

- **Full-period training** to establish baseline performance.
- **Rolling-window training** to adapt to changing market regimes.
- **Hyperparameter tuning** of learning rate, batch size, and exploration settings for optimal convergence.

The approach was designed to move beyond static backtesting toward robust, real-time decision-making in volatile and uncertain markets.

3 Experiments and Results

This section presents our framework, baseline agent performance, and improvements from retraining DDPG using an expanding rolling window.

3.1 Experimental Setup

We evaluated multiple DRL algorithms in a standardized FinRL–Stable-Baselines3 **Gym** trading environment with continuous action space. **Features**: Technical indicators (MACD, Bollinger Bands, RSI, CCI, DX, SMA-30/60) and risk measures (VIX, turbulence index). **Metrics**: Annual/Cumulative Return, Volatility, Max Drawdown, Sharpe, Sortino, Calmar, Omega, Tail Ratio, Stability, and Daily VaR.

3.2 Baseline Results

Figure 2 shows cumulative returns for DRL agents vs. benchmarks (MVO, DJI). **DDPG** achieved the highest annual return (17.44%) and Sharpe ratio (0.926) but with high volatility (19.46%) and max drawdown (-20.32%). PPO ranked second in stability but with lower returns.

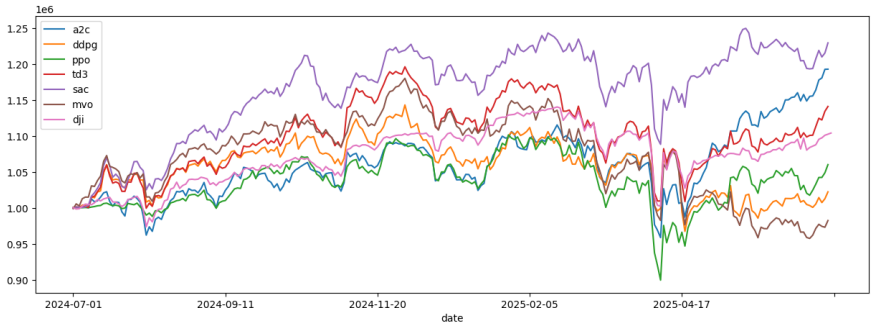

Fig. 2: Cumulative returns for DRL agents and benchmarks.

3.3 Rolling Window Retraining of DDPG

To improve adaptability and reduce risk, DDPG was retrained with an expanding rolling window: (1) train on an initial subset, (2) add new data incrementally, (3) retrain at each step. Figure 3 compares all agents, including the DDPG Rolling variant.

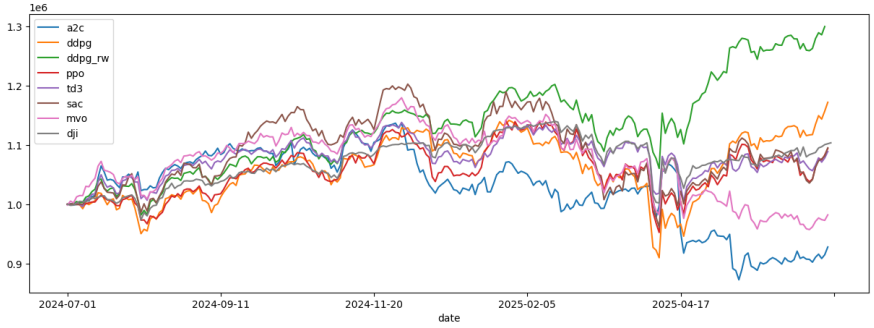

Fig. 3: Performance with DDPG retrained via expanding rolling window.

3.4 Performance Summary

The DDPG agent achieved the highest baseline annual return (17.44%) and Sharpe ratio (0.926), though with relatively high volatility (19.46%) and max drawdown (-20.32%). When retrained with an expanding rolling window (DDPG_RW), performance improved substantially: annual return rose to 30.56%, Sharpe ratio to 1.648, and max drawdown decreased to -11.79%. DDPG_RW also achieved the highest **stability** score (0.793), far surpassing other agents in this metric,

indicating consistent growth over time. Overall, the rolling window approach enhanced both profitability and risk-adjusted performance, making DDPG_RW the top-performing strategy across most metrics. A summary of the key metrics for all agents is provided in Table 1.

	PPO	DDPG	DDPG_RW	A2C	TD3	SAC
Annual return	0.095	0.174	**0.306**	-0.073	0.090	0.097
Sharpe ratio	0.603	0.926	**1.649**	-0.268	0.640	0.571
Calmar ratio	0.575	0.858	**2.591**	-0.313	0.768	0.481
Max drawdown	-0.166	-0.203	**-0.118**	-0.233	-0.117	-0.201
Stability	0.143	0.187	**0.793**	0.501	0.067	0.003

Table 1: Selected performance metrics for all agents (best in **bold**).

4 Conclusion

Agent performance varies by algorithm and settings. The rolling window strategy proved effective in boosting returns and reducing risk. Limitations include potential overfitting and market regime sensitivity. Future work will integrate **LLMs** for news analysis and explore real-time deployment.

5 Team Contributions

- **Author 1**: Collaborated on data preprocessing and feature engineering, assisted in implementing the backtesting setup, contributed to the comparative performance analysis, and took a lead role in drafting, restructuring, and finalizing the written report.
- **Author 2**: Led the development of the core methodology, including data preparation, training and evaluation of DRL agents, and designing the expanding rolling window retraining strategy. Contributed to structuring the experimental framework and writing key sections of the report.

Bibliography

[1] Yahoo Downloader. https://downloads.yahoo.com/, . Accessed: 2025-08-13.
[2] OpenAI-gym. https://github.com/openai/gym/, . Accessed: 2025-08-13.
[3] Yahoo Finance . https://finance.yahoo.com//, . Accessed: 2025-08-13.