The Power of Rewards in Algorithmic Trading

Renato Laffranchi Falcão
Insper
São Paulo
renatolf1@al.insper.edu.br

***Abstract*—In recent years, algorithmic trading (AT) has experienced substantial advancements due to emerging technologies such as Artificial Intelligence (AI) and faster hardware solutions. Simultaneously, the AI field has seen significant progress, with algorithms evolving beyond classification and prediction problems to encompass behavior optimization. Reinforcement Learning (RL), an AI subfield reaching maturity, has demonstrated exceptional capacity to solve complex sequential decision-making problems, making it a promising approach for developing sophisticated trading algorithms. This paper investigates the potential of RL to enhance stock trading strategies by learning from market data and optimizing actions for maximum financial gain. Additionally, this study reviews the results achieved by the RL-based models and outlines future research directions to further explore the synergy between RL and AT.**

***Index Terms*—Algorithmic trading, quantitative finance, stock market, reinforcement learning, artificial intelligence**

I. Introduction

Algorithmic trading (AT) involves using computer programs to trade equities, currencies, or other financial instruments. These programs utilize algorithms to calculate current market conditions and execute trades automatically when the market meets predefined criteria for profitability. Recent advancements in Artificial Intelligence (AI) and high-performance computing have significantly enhanced AT, enabling the development of sophisticated and efficient trading algorithms.

Reinforcement Learning (RL), a subfield of AI, optimizes sequential decision-making processes by learning from interactions with the environment to maximize cumulative rewards. This makes RL particularly well-suited for AT, where the goal is to develop trading strategies that achieve maximum financial returns.

The primary objective of this paper is to investigate the effectiveness of RL in AT by conducting a backtesting study of an RL agent trained on three different stocks from the Brazilian Stock Market. Backtesting involves testing a strategy on historical data to evaluate its performance before real-world deployment. This study aims to assess the potential of RL-based models to enhance stock trading strategies and provide insights for future research directions in integrating RL with AT.

II. Methodology

In this study, a methodology was implemented to train and test RL agents on stock data from the Brazilian Stock Market. The methodology is divided into two main components: data collection and agent training/testing.

A. Data Collection

The data collection process involves retrieving historical stock data, collected from the past 10 years for each stock, from the Yahoo Finance API using a custom script. This process includes:

1) **Data Retrieval**: Historical data for specified stocks is retrieved from the Yahoo Finance API.
2) **Feature Engineering**: Several technical indicators and features are calculated to enhance the dataset. These features include:
   - **Percentage Change in Closing Prices**: Calculated as the percentage change between consecutive closing prices.
   - **Price Ratios**: Ratios of opening, high, and low prices to closing prices.
   - **Volume Relative to Maximum Volume**: Volume relative to the maximum volume over the past 7 days.
   - **Moving Average Convergence/Divergence (MACD) Indicator**: Includes MACD, MACD signal, and MACD histogram.
   - **Three Inside Up/Down Pattern**: A candlestick pattern indicator.
   - **Beta Indicator**: Measures the volatility of the stock relative to the market.

The features added to the dataset were selected to provide a comprehensive view of the stock's behavior and market conditions, essential for the RL agent's decision-making process. These features include the percentage change in closing prices to capture daily price movements, and ratios of opening, high, and low prices to closing prices to understand daily trading ranges. Volume relative to the maximum volume over the past 7 days helps gauge market activity and liquidity. The MACD indicator provides insights into market momentum, while the Three Inside Up/Down pattern helps identify potential trend reversals. Lastly, the Beta indicator measures the stock's volatility relative to the market, offering a perspective on the stock's risk and stability.

By incorporating these features, the dataset equips the RL agent with a rich set of information, enhancing its ability to make informed trading decisions based on price trends, trading volumes, momentum indicators, pattern recognition, and market risk assessment.

The processed data is stored in feather format for efficient storage and retrieval.

B. Agent Training and Testing

The training and testing of the RL agents are carried out through a structured process that involves data preparation, environment setup, agent training, and evaluation.

1) Data Preparation: The historical stock data is divided into training and testing sets, with 75% of the data allocated for training and the remaining 25% for testing. This ensures that the RL agents are trained on a substantial portion of the data and tested on unseen data to evaluate their performance.

2) Environment Setup: A custom trading environment is created using the `Gym Trading Env` package [3]. This environment is tailored to support a custom reward function, which is crucial for evaluating the trading strategies developed by the RL agents. The design of a good reward function is essential in RL applications, as it directly influences the agent's learning process and its ability to achieve the desired objectives. Two distinct reward functions were designed:

$$R(p_t) = \frac{\left(\frac{p_t}{p_0}\right)^3}{\text{risk-free rate} \times 100} \tag{1}$$

where p_t represents the portfolio's valuation at the current time step, p_0 the initial value of the portfolio and risk-free rate is the CDI rate, set to 10.40% annually. The risk-free rate is multiplied by 100 to make the reward function sparser. The purpose of this formula is to provide the agent with exponential rewards, offering substantial positive reinforcement for good returns while also imposing exponential penalties for negative outcomes. This approach ensures that the agent is strongly encouraged to achieve positive results and avoid significant losses.

$$R(p_t) = \left(\frac{p_t}{p_0}\right) - \left(\frac{\text{risk-free rate}}{365}\right) \tag{2}$$

The purpose of this reward function is to evaluate the agent's performance by comparing the final portfolio valuation to its initial valuation, adjusted for the daily risk-free rate. This encourages the agent to not only generate returns that exceed the initial investment but also outperform the baseline risk-free rate, ensuring that the trading strategy provides a meaningful advantage over low-risk alternatives.

The environment is configured with several parameters, including the initial portfolio value, trading fees, borrow interest rate, and the window size for historical data used in decision-making. Additionally, the actions available to the agent must be specified. The parameters used in the environment are presented in Table I.

TABLE I
PARAMETERS FOR THE ENVIRONMENT CREATION

Parameter	Value
Positions	[-1, 0, 1]
Trading fees	0.01%
Borrow interest rate	0.03%
Portfolio initial value	R$10,000
Windows	14

The "Positions" parameter defines the allowed positions: -1 for a short position, 1 for a long position, and 0 for no position in the asset. Further details on the parameters can be found in the official environment documentation [3].

3) Training: The RL agents were trained using the Proximal Policy Optimization (PPO) algorithm [1]. The training process involved configuring the algorithm with specific hyperparameters, including learning rate, number of steps, batch size, number of epochs, and gamma. The agents were trained on three stocks from the Brazilian Stock Market: COGN3.SA, ITUB4.SA, and PETR4.SA. This involved setting up the environment with historical data for each stock and configuring the PPO algorithm with the hyperparameters listed in Table II. Throughout the training, the rewards were monitored and plotted to visualize the agents' learning progress.

The implementation utilized Stable-Baselines3, a well-established and widely used library for reinforcement learning [2]. The training process iterated through the data 3,000 times, resulting in 5.211 million time steps of training for each stock under each reward scheme.

TABLE II
HYPERPARAMETERS FOR THE PPO ALGORITHM

Hyperparameter	Value
Learning Rate (α)	1×10^{-4}
Discount rate (γ)	0.9999
Number of Steps	Size of the training data
Batch Size	Training size $\times$ number of environments
Number of Epochs	20

4) Evaluation: The performance of the trained RL agents is evaluated on the test data set. The evaluation involves running the agents on the test data and plotting the portfolio valuation over time. This plot provides insights into how well the RL agents perform in terms of maximizing the portfolio value based on the learned trading strategies.

III. RESULTS

The results obtained detail the learning capabilities of the RL agents for both reward functions 1 and 2, alongside the outcomes of the backtesting within the trading environment. The evaluation encompasses an analysis of the agents' training performance using various reward functions and an assessment of their effectiveness in trading scenarios.

A. Training Capabilities

The training capabilities of the RL agents were evaluated using two distinct reward functions. The learning curves for each reward function on each stock provide insights into the agents' ability to optimize their trading strategies over time.

The first reward function offers exponential rewards for positive returns and exponential penalties for negative outcomes. The learning curves for each stock (COGN3.SA, ITUB4.SA, PETR4.SA) under this scheme are presented in Figure 1. The second reward function compares the final portfolio valuation to its initial valuation, adjusted for the risk-free rate. The

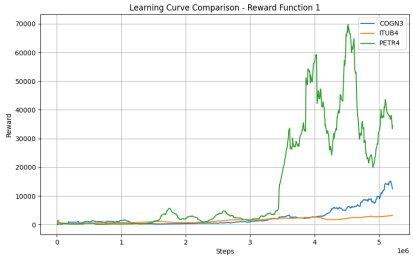

Fig. 1. PPO Learning Curves with reward function 1.

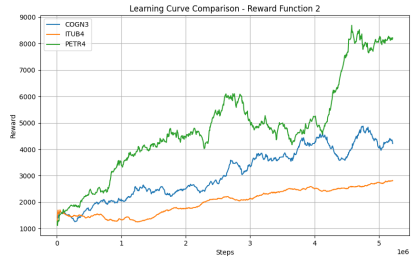

Fig. 2. PPO Learning Curves with reward function 2.

learning curves for each stock under this scheme are presented in Figure 2.

The first reward function results in varied learning trajectories. For PETR4, the agent shows high volatility and significant fluctuations, suggesting potential overfitting or sensitivity to market changes. In contrast, COGN3's learning curve is more stable with a steady increase in rewards, indicating consistent learning. ITUB4 shows the least volatility, with a gradual reward increase, suggesting stable learning.

The second reward function produces more consistent and smoother learning curves for all three stocks. PETR4 shows stable reward growth, indicating improved learning stability. COGN3 demonstrates a smooth upward trend, supporting the effectiveness of the second reward function in promoting stable learning. ITUB4 continues to show steady reward increases, albeit at a slower rate, indicating enhanced learning capability.

Overall, the second reward function yields more stable and consistent learning across all stocks, reducing overfitting and improving performance. The next section evaluates the RL agents on test data to compare the effectiveness of the reward functions in guiding the agents toward optimal trading performance.

B. Backtesting

The backtesting was conducted using the testing portion of the data after the training process. The portfolio valuations over time for each stock (COGN3.SA, ITUB4.SA, PETR4.SA) are shown in Figures 3, 4 and 5, where the trading was performed by the agent trained under the first reward scheme. Similarly, the portfolio valuations over time for the same stocks, traded by the agent trained under the second reward scheme, are depicted in Figures 6, 7 and 8.

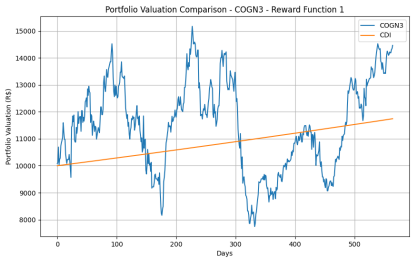

Fig. 3. Portfolio Valuation for stock trading of COGN3.SA. Agent trained on reward scheme 1.

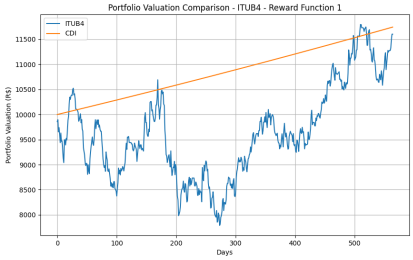

Fig. 4. Portfolio Valuation for stock trading of ITUB4.SA. Agent trained on reward scheme 1.

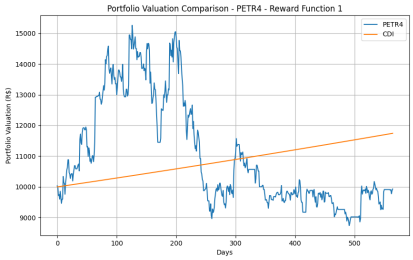

Fig. 5. Portfolio Valuation for stock trading of PETR4.SA. Agent trained on reward scheme 1.

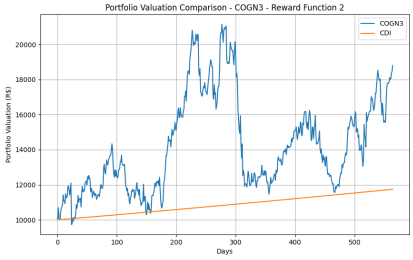

Fig. 6. Portfolio Valuation for stock trading of COGN3.SA. Agent trained on reward scheme 2.

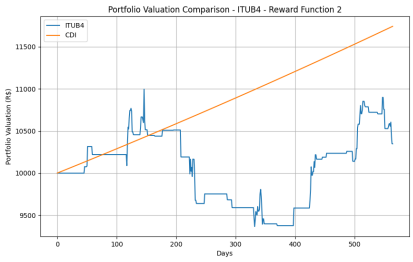

Fig. 7. Portfolio Valuation for stock trading of ITUB4.SA. Agent trained on reward scheme 2.

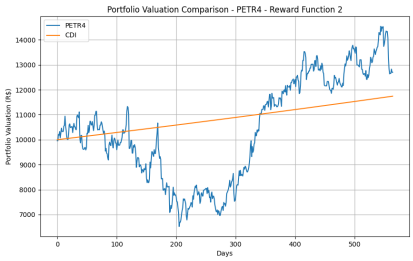

Fig. 8. Portfolio Valuation for stock trading of PETR4.SA. Agent trained on reward scheme 2.

These results, depicted in Table III, highlight the effectiveness of the second reward function in guiding the RL agents toward more stable and profitable trading strategies, as evidenced by the higher returns and smoother portfolio valuation curves. However, it is important to note that none of the agents managed to outperform the CDI benchmark of 10.40% per annum. The performance under the first reward function shows greater variability, particularly for PETR4, indicating potential overfitting or sensitivity to market fluctuations.

TABLE III
ANNUAL MARKET AND PORTFOLIO RETURNS FOR THE TESTING DATA UNDER TWO REWARD FUNCTIONS

Metric (Annual)	**COGN3**	**ITUB4**	**PETR4**
Market Return (%)	-9.20	8.55	4.78
Portfolio Return (%) Reward 1	17.67	6.77	-0.30
Portfolio Return (%) Reward 2	32.08	1.52	11.11

IV. CONCLUSION

In this study, Reinforcement Learning (RL) agents were trained and tested on stock data from the Brazilian Stock Market using two distinct reward functions. The training process revealed that the second reward function, which evaluates the agent's performance by comparing the final portfolio valuation to its initial valuation adjusted for the risk-free rate, provided more stable and consistent learning outcomes. This reward function facilitated better generalization of the trading strategies, as evidenced by the smoother learning curves and higher portfolio returns for most stocks.

The backtesting results demonstrated that while the RL agents achieved significant portfolio returns, particularly with the second reward function, they did not manage to outperform the CDI benchmark of 10.40% per annum. This suggests that while RL can enhance trading strategies, further optimization is needed to achieve superior performance consistently.

Future work could explore several avenues to improve the effectiveness of RL in algorithmic trading. First, refining the reward functions to better capture the complexities of financial markets and human behavior could lead to more robust trading strategies. Additionally, incorporating more sophisticated feature engineering techniques and utilizing alternative data sources, such as sentiment analysis from news articles and social media, could provide the RL agents with a richer context for decision-making.

Moreover, investigating the impact of different RL algorithms and their configurations on trading performance could yield insights into the most effective approaches for various market conditions. Finally, studying the interaction between RL agents and human traders could offer valuable perspectives on how automated strategies can complement human decision-making in trading environments.

By addressing these areas, future research can further enhance the capabilities of RL in algorithmic trading, leading to more effective and reliable trading strategies that can adapt to the dynamic nature of financial markets.

REFERENCES

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," CoRR, vol. abs/1707.06347, 2017. arXiv: 1707.06347. [Online]. Available: http://arxiv.org/abs/1707.06347.

[2] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html.

[3] Gym trading env documentation, https://gym-trading-env.readthedocs.io/en/latest/index.html, Accessed: 2024-05-28.