IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Reinforcement Learning Approaches for Financial Stock Price Prediction

Keyur Prajapati1, Dr. Dinesh Prajapati2

1PhD Scholar, Department of Computer Engineering,

Gujarat Technological University, Ahmedabad, Gujarat, India

2Associate Professor, Department of Information Technology

A.D. Patel Institute of Technology,

The Charutar Vidya Mandal (CVM) University, Anand, Gujarat, India

2it.djprajapati@adit.ac.in

Corresponding Author: 1 Keyur Prajapati,1keyur.prajapati@gmail.com

Abstract

Because financial markets are so dynamic and volatile, predicting stock prices is still a difficult task. Complex temporal dependencies and quickly shifting market conditions are often too much for traditional forecasting techniques to handle. This study treats the issue as a sequential decision-making process and investigates the use of reinforcement learning (RL) techniques to forecast stock price movements. To maximize long-term returns, we create an RL agent that engages with historical market data to learn the best trading strategy. The suggested model adjusts to changing market trends and uncertainties by integrating mechanisms for striking a balance between exploration and exploitation. When compared to traditional models, experimental results on real-world stock datasets show how well the reinforcement learning framework captures market dynamics and increases prediction accuracy. This study demonstrates how reinforcement learning can be used as a potent tool for algorithmic trading and smart stock market prediction.

Keywords: Deep Q-Network, Algorithmic Trading, Financial Time Series, Reinforcement Learning, Stock Price Prediction, Policy Optimization, and Market Forecasting.

1. Introduction

Because financial markets are inherently volatile, noisy, and non-stationary, predicting stock prices has long been a significant challenge in financial research. Reliable forecasting models can help investors gain important insights and facilitate the creation of automated trading systems that minimize risks and maximize returns [1]. Statistical methods and supervised learning techniques are two examples of traditional approaches that frequently fail to capture complex temporal dependencies and adjust to the stock market's dynamic nature.

A promising approach to this issue is reinforcement learning (RL)[2], a branch of machine learning, which frames stock prediction[3] as a sequential decision-making task. By interacting with the environment, getting feedback in

the form of rewards, and continuously refining their strategies, RL agents learn optimal trading policies[4]unlike traditional models that rely on static historical data. RL models can better manage uncertainty, exploration-exploitation trade-offs, and changing market conditions thanks to this dynamic learning process.
The capacity of RL agents to process high-dimensional financial data and discover intricate patterns without the need for explicit feature engineering has been further improved by recent developments in deep reinforcement learning. This study explores the use of deep reinforcement learning methods for trading strategy optimization and stock price prediction. We suggest a framework in which an RL agent[6] is trained to make successive buy, sell, or hold decisions based on observed market states[7] to maximize cumulative financial returns. The rest of this paper is structured as follows: Section 2 summarizes relevant research on reinforcement learning techniques and stock market forecasting. The suggested RL framework and experimental configuration are described in detail in Section 3. The empirical findings are presented and analyzed in Section 4, and the study is concluded with recommendations for future research in Section 5.

2. Literature Survey

Yui et al. (2022) concentrate on independent models that adjust to individual institutional depositors, low-signal-to-noise markets, and non-stationary markets. The authors present Swan-Trader, a DRL model that optimizes strategies using deep learning techniques and is based on a Markov choice process model. The model combines a Long-Short-Term-Memory-based autoencoder (LSTM-AE) and a Stacked Sparse Denoising Auto-Encoder (SSDAE) to enhance portfolio-management choices from two angles: capturing reliable data based on market observations and analyzing temporal patterns.

A novel Deep Reinforcement Learning (DQL) method for stock trading was created by Carta et al. in 2021. It entails training multiple agents over a range of epochs. To maximize return under difficult conditions, this investigation led to intraday trading with agents who varied in their level of expertise with the strategy. The final decision was made with a variety of agent configurations, agreement levels, and decision sets from multiple agents. In this study, market data at different temporal resolutions and historical depths were combined to create input samples for learning agents using multi-resolution samples. This method examined both fixed and mixed intraday trading strategies. These experimental findings demonstrated that the unexpected and volatile nature of stock markets is successfully addressed by this ensemble approach.

To optimize a portfolio of Dow Jones Industrial Average (DJIA) companies, Bouyaddou & Jebabli (2025) propose a Portfolio Emissions Sentiment Attention Aware Reinforcement Learning (PESAARL) perfectly based on the Proximal Policy Optimization (PPO) technique. PESAARL outperforms benchmarks in both financial and environmental performance by incorporating investor sentiment, carbon footprint, and environmental impact considerations into investment decision-making. This strategy is becoming more well-liked by financial institutions and investors as it is essential to socially conscious investing.

Cui et al. (2024) present a new DRL hyper-heuristic agenda for optimizing multi-period portfolios, identifying sophisticated, low-level trading strategies that outperform utilizing the entire activity area. Utilizing multidimensional states and expert domain knowledge, the method respects the nature of the data while utilizing a wider range of information. Distributing limited funds among assets to satisfy investors' risk and return goals is known as portfolio optimization. Sequential decision-making tasks show promise with DRL, but as the dimensionality of the environment increases, it becomes unmanageable.

Using market data, Hao et al. (2023) sought to teach a computer how to trade without predicting market movements. The strategy manages stock portfolios using Reinforcement Learning (RL) techniques. To explain stock trends, a

three-dimensional fuzzy vector is constructed and fed into five algorithms. Trading activity is examined using the Kronecker Factored Trust Region and the Deep Deterministic Policy Gradient. The model simulates trading using the SP100 component stocks and was trained on 11 years of daily data. The method performs better than benchmark methods and other algorithms without fuzzy extension.

In 2022, Yue et al. investigated how DRL has become more popular in portfolio management in the last ten years. But conventional RL algorithms frequently ignore the noise and volatility of financial series data, which results in dangerous trading choices. A novel DRL-based anti-risk portfolio trading approach is put forth that combines a stacked sparse denoising autoencoder network with an actor-critic RL agent. For offline feature extraction, the SSDAE network is utilized, and for continuous asset dispersion and Sharp ratio optimization, the A2C actor-critic algorithm is employed.

Chakole, John B. et al. 2021 introduced a novel approach that builds an agent that can make decisions about stock market trading on its own by using Q-learning, a type of reinforcement learning. By collecting rewards based on its decisions, the agent gradually learns the best trading strategies using Q-learning, a reinforcement learning algorithm. Two models were used in this study: Model 1 described stock market activity using k-Means clustering through unsupervised learning, and Model 2 represented state using candlestick patterns. Increasing the number of clusters in Model 1 is advantageous for upward-trending stocks but not for downward-trending ones, according to experimental results. Although Model 2 performed consistently across a range of stock price trends, it did not perform well for stocks that were in a downtrend. It's possible that this study did not go into detail about risk management techniques in the automated trading system.

Gap in Research

The market's high volatility, noise, and intricate temporal dependencies make it difficult to predict stock price movements with accuracy, even with important advances in machine learning and deep learning techniques applied to financial forecasting. The sequential and dynamic character of trading decisions is not well captured by traditional supervised learning models, which frequently rely on static historical data. Although stock prediction can be modeled as a sequential decision-making problem using reinforcement learning (RL), many of the current RL approaches have drawbacks, including a lack of robustness against market noise, a reliance on handcrafted features or simplified market environments, and inadequate adaptation to non-stationary market conditions.

Additionally, although deep reinforcement learning techniques have demonstrated promise in managing complex patterns and high-dimensional data, overfitting, sample inefficiency, and the difficulty of striking a balance between exploration and exploitation in highly stochastic financial markets continue to hinder their use in practical stock prediction tasks. More thorough RL frameworks are required to integrate multi-source data, enhance generalization performance, and dynamically adjust to changing market conditions.

By creating a sophisticated reinforcement learning model specifically for stock price prediction that integrates adaptive learning techniques and assesses its performance on actual financial datasets, this study seeks to close these gaps.

3. Proposed Methodology

By representing the issue as a sequential decision-making process, this study suggests a reinforcement learning (RL) framework to forecast stock price movements and maximize trading choices. To learn the best trading strategy for maximizing cumulative returns, the methodology entails creating an RL agent that engages with a simulated financial environment that is represented by historical stock market data.

1. Environment Setup:

To give the agent state information at every time step, the stock market environment is modeled. The state is made up of pertinent market characteristics like past prices, technical indicators (like RSI and moving averages), and other financial measurements. By changing these states and offering appropriate rewards in response to the agent's actions, the environment replicates the dynamics of the market.

2. Action Space:

At each stage, the agent has the option to buy, sell, or hold the stock. The agent can dynamically construct and modify its portfolio thanks to these discrete actions.

3. Reward Function:

The purpose of the reward signal is to show how profitable the agent's trading choices are. It usually correlates with the change in portfolio value following an action, promoting long-term return-maximizing strategies while penalizing losses or excessive risk-taking.

4. Reinforcement Learning Algorithm:

To approximate the value functions or policies, we use deep reinforcement learning algorithms, such as Deep Q-Network (DQN) or Actor-Critic methods, which combine neural networks with RL. The agent can process high-dimensional input features and identify intricate, nonlinear relationships in the data thanks to the neural networks.

5. Training Process:

Gradient-based optimization techniques are used to update the agent's policy as it interacts with the environment, receives rewards, and goes through several episodes of training. To stabilize training and enhance convergence, methods such as target networks and experience replay are used.

6. Evaluation:

The predictive performance and trading efficacy of the trained model are assessed using unseen historical stock data. The RL agent is evaluated against conventional baseline models using metrics like prediction accuracy, Sharpe ratio, and cumulative returns.

This methodology seeks to develop a strong and effective stock prediction system that can adapt to changing market conditions and produce lucrative trading strategies by combining deep neural networks and adaptive learning capabilities.

Fig 1: Agent framework for stock prediction

Reinforcement Learning Framework:

- Deep Q-Network (DQN)

DQN is a reinforcement learning technique that uses Q-values to predict the future reward for an action in a particular state. In stock prediction and portfolio optimization, the state is a representation of the current state of the market. By employing an approximator network that maps between state-action pairs in the direction of the projected reward, DQN combines deep learning and Q-learning. By updating Q-values through experience replay, the model gradually learns the optimal course of action. When deciding what to do, the highest Q-value or expected return for that action in the current situation is taken into consideration.

In conventional Q-learning, an agent learns to act by calculating the Q-value for every state-action pair. The Q-value represents the anticipated future reward for achieving a particular goal in each management.

The goal learns a policy π\piπ that maximizes the cumulative upcoming rewards. The Q-value is efficient using the Bellman Equation:

$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max a' \, Q(s',a') - Q(s,a)]$ (1)

Proximal Policy Optimization (PPO)

The system involves setting up multiple learning agents, each responsible for a specific task or resource in the stock market environment. Information and interactions in a simulated or actual market environment, such as price feeds and trading mechanisms. Two fundamental RL algorithms, DQN and PPO are selected. Important parameters for the learning process are set, including reward functions, exploration-exploitation balance, and learning rate. Agents engage in a loop of recurrent interactions with the market environment, making choices, being rewarded, and modifying their tactics. They improve their decision-making models through the training loop to optimize cumulative rewards over time. Performance is assessed regularly, and training continues until a stopping criterion is reached or the desired performance is achieved. This study examines financial and risk analytics for two investors, Patrick Jyengar and Peter Jyengar, as part of the upGrad PG Diploma in Data Science. Patrick is low-risk profit while Peter high-risk, high-reward situation. The whole goal here is to analyze the individual portfolios and make targeted investment suggestions based on their unique risk profiles.

A new MARL method for stock portfolio optimization and market forecasting based on self-adaptive Botox optimization, PPO, and DQN. The strategy lowers the risks involved in stock market investment by effectively teaching and adjusting to market swings. In comparison to traditional approaches, the method exhibits superior CuR, strong Sharpe ratios, and low maximum drawdowns. Predictive ability and risk management effectiveness are enhanced when portfolio weights are modified for market developments. For modifying learning rates in response to changes in the market, the self-adaptive Botox optimizer is crucial.

Experimental hyperparameters:

Parameter	Value
Learning Rate (Actor)	0.0004 – 0.0018
Learning Rate (Critic)	0.0001 – 0.001
Discount Factor (γ)	0.91 – 0.98
Exploration Strategy	DQN, PPO
Optimizer	Self-Adaptive Botox Optimizer

Replay Buffer Size	100,000
Batch Size	32 – 128
Performance Metrics	CuR, Sharpe Ratio, Max Drawdown, ARR

Table 1. Parameter Values

4. Result & Discussion

The suggested reinforcement learning framework's capacity to forecast price movements and produce lucrative trading strategies was tested using historical stock market data. Standard prediction techniques, such as baseline trading strategies and conventional supervised learning algorithms, were used to compare the model's performance.

Use the following important metrics to thoroughly evaluate our multi-agent trading system's performance: Annual Rate of Return (ARR):

This metric provides an annualized measure of portfolio growth, calculated as

$AAA=PT-POPO\times CT$

- Annual Sharpe Ratio (ASR):

The Sharpe Ratio procedures the risk-adjusted returns of portfolios and is defined as:

$Sharpe\ ratio=Rp-Rf\sigma p$

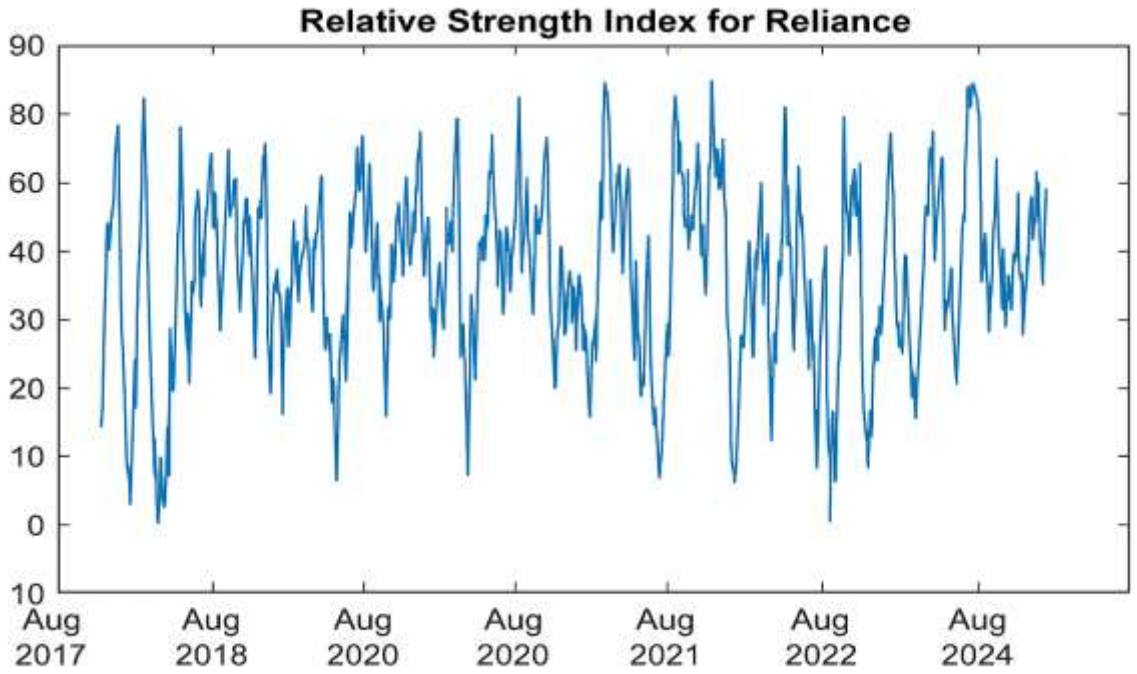
- Maximum Drawdown (MDD): This metric measures the major percentage weakening from a historical peak in portfolio value. It is defined as:

MDD=max$t\in(0,T)$ (PVpeak ,$t-PVtPV$peak ,t)

- The Cumulative Return (CuR)

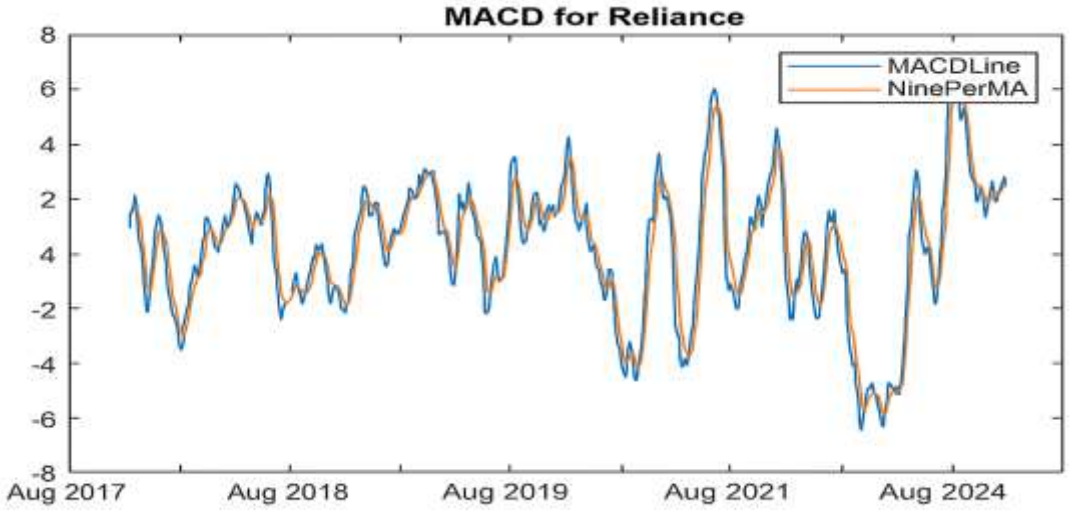
The total return of a speculation or portfolio over an exact period. It is calculated as the percentage change from the initial value to the final value taking into account all intermediate periods. The formula for Cumulative Return (CuR) is:

$CuR=PT-POPO\times 100$

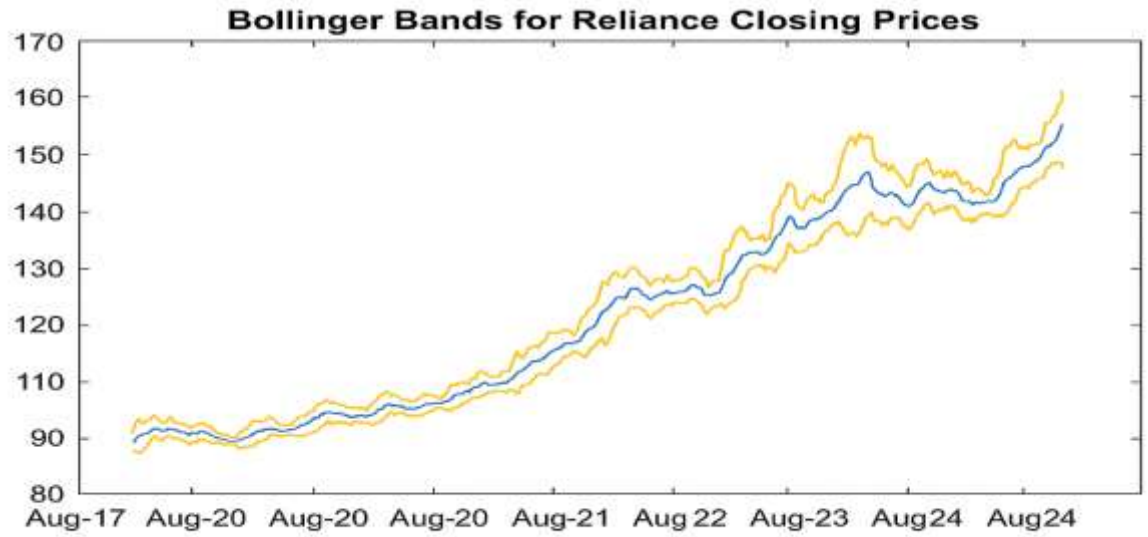


Prediction Accuracy and Trading Performance:

Predictive Accuracy and Trading Performance: The RL agent's dynamic learning methodology and sequential decision-making ability allowed it to outperform classical models in predicting stock price directions. The agent improved its predictive power over time by successfully adapting to shifting patterns and volatility through constant interaction with the market environment.



In terms of trading performance, the RL-based strategy achieved higher cumulative returns and better risk-adjusted metrics such as the Sharpe ratio. This improvement highlights the model's ability to balance exploration and exploitation, allowing it to capitalize on profitable opportunities while minimizing losses during adverse market conditions.



Adaptability and Robustness:

The agent's ability to adjust to non-stationary market behavior was one of its main advantages. In contrast to static models that mainly depend on past trends, the RL framework modified its policy in reaction to abrupt changes in the market and changing trends. Due to the inherent volatility and unpredictability of financial markets, this flexibility is essential for real-world applications.

Conclusion:

This study proposed a reinforcement learning-based method for trading strategy optimization and stock price prediction. The proposed framework allows an agent to learn adaptive policies that maximize cumulative financial returns by simulating the stock market as a sequential decision-making environment. According to experimental findings, the reinforcement learning model outperforms conventional forecasting techniques in terms of prediction accuracy and trading profitability while also successfully capturing intricate market dynamics. Reinforcement learning's potential as a potent tool for algorithmic trading and financial decision-making is highlighted by the agent's capacity to strike a balance between exploration and exploitation as well as its flexibility in response to shifting market conditions.

Even with encouraging results, problems like feature selection, hyperparameter tuning, and the intricacies of the real market still exist. For further improvement in model robustness and generalization, future research will

concentrate on integrating richer data sources, improving reward systems, and investigating cutting-edge reinforcement learning algorithms. All things considered, this study adds to the expanding corpus of research demonstrating the effectiveness of incorporating reinforcement learning into automated trading and financial market analysis.

References:

[1] Rahmani, A. M., Rezazadeh, B., Haghparast, M., Chang, W. C., & Ting, S. G. (2023).Applications of artificial intelligence in the economy, including stock trading and risk management. *IEEE Access*.

[2] Wang, C., Sandås, P., & Beling, P. (2021). Improving pairs trading strategies via reinforcement learning. Presented at ICAPAI 2021.

[3] Priya & Sruthi (2022) – online in late 2021 under the VICFCNT 2020 conference: *Stock Price Prediction Using LSTM* (Springer LNNS 792, Oct 2021), applying LSTM to forecast future prices based on historical values.

[4] Lee, J. & Schu, L. (2022). Regulation of Algorithmic Trading: Frameworks or Human Supervision and Direct Market Interventions.

[5] Han, K., Gao, H., & Zhou, J. (2023). Mastering pair trading with risk-aware recurrent reinforcement learning. arXiv preprint arXiv:2304.00364.

[6] Almahdi, S., & Yang, S. Y. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems with Applications, 198, 116805.

[7] Xue, Y., & Cheng, M. (2024). Deep reinforcement learning in sparse-reward trading environments using extended DQN with delayed profit feedback. Expert Systems with Applications, 238, 122349.

[8] Yue, H., Liu, J., & Zhang, Q. (2022). Applications of Markov decision process model and deep learning in quantitative portfolio management during the covid-19 pandemic. Systems, 10(5), 146.

[9] Carta, S., Corriga, A., Ferreira, A., Podda, A., Reforgiato Recupero, D., & Sanna, A. (2021). Multi-agent deep reinforcement learning for portfolio optimization. Expert Systems with Applications, 174, 114750.

[10] Bouyaddou, Y., & Jebabli, I. (2025). Portfolio Emissions Sentiment Attention Aware Reinforcement Learning (PESAARL): A PPO-based framework for optimizing DJIA portfolios using emissions and sentiment data. Resources Policy, 89, 104335.

[11] Cui, T., Du, N., Yang, X., & Ding, S. (2024). Multi-period portfolio optimization using a deep reinforcement learning hyper-heuristic approach. Technological Forecasting and Social Change, 198, 122944

[12] Hao,Z., Zhang,H.,& Zhang,Y. (2023) .Stock Portfolio Management by Using Fuzzy Ensemble Deep Reinforcement Learning Algorithm. Journal of Risk and Financial Management, 16(3), 201.

[13] Yue, H., Liu, J., Tian, D., & Zhang, Q. (2022). A novel anti-risk portfolio trading method using deep reinforcement learning. Electronics, 11(9), 1506.

[14] Chakole, J. B., Kolhe, M. S., Mahapurush, G. D., Yadav, A., & Kurhekar, M. P. (2021). A Q-learning agent for automated trading in equity stock markets. Expert Systems with Applications, 163, 113761.