International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Portfolio Optimization in Dynamic Markets: Reinforcement Learning for Investment

[1]Dr. Paramita Sarkar, [2]Dr. Pruthiranjan Dwibedi, [3]Dr. Shailesh Shivaji Deore, [4]Prof. Tukaram Gawali, [5]Ms. Mayuri Diwakar Kulkarni, [6]Animesh Saha

Submitted: 28/11/2023 **Revised**: 09/01/2024 **Accepted**: 18/01/2024

Abstract: In today's volatile and ever-changing financial markets, optimizing portfolio allocation is a formidable job. This study offers a novel strategy for optimizing a portfolio in volatile markets by drawing on techniques from the field of Reinforcement Learning (RL). As a result of being slow to adapt to ever-changing market conditions, traditional investment strategies sometimes generate worse returns and expose their owners to more risk. In contrast, the RL-based method offers a dynamic and adaptable answer to this age-old problem. The proposed model uses RL to learn and improve its tactics over time in order to maximize returns while effectively limiting risk. Because RL is so malleable, the portfolio can instantly respond to fluctuations in the market, grabbing opportunities and avoiding setbacks. We conduct extensive tests using historical market data to evaluate our RL-based portfolio optimization approach and to compare it to conventional investment strategies. Our research proves that even in volatile markets, RL can produce superior risk-adjusted returns. We also shed light on the practical implementation of RL in portfolio management by providing insights into the key factors influencing its effectiveness. This research is an important first step in rethinking investment strategies for fluid markets. To improve investment outcomes and risk management, we give investors a robust framework for portfolio optimization that can survive and even thrive in volatile market environments. We accomplish this by making use of RL's tremendous potential.

Keywords: *Reinforcement Learning, Optimization, Machine Learning, Risk Management, Dynamic Market*

1. Introduction

Recent years have seen a significant upheaval in the financial landscape, which is now marked by previously unheard-of levels of complexity, volatility, and interconnection. Traditional portfolio optimization techniques frequently fail to produce the anticipated returns while successfully managing risk in today's volatile markets. Both individuals and investment experts must navigate a constant influx of market information, economic developments, and geopolitical issues that have the potential to drastically change the financial picture. This paper investigates the use of Reinforcement Learning (RL) in the area of investment portfolio optimization in response to these difficulties, offering a possible path for generating higher returns in volatile and unpredictable financial situations [1].Investors have traditionally built portfolios that balance risk and return using traditional methods of portfolio optimization, such as mean-variance analysis. These techniques, while useful, are dependent on assumptions about market circumstances that are static and frequently fall short of capturing the dynamic character of financial markets [20]. Traditional tactics struggle in changing marketplaces to react swiftly enough to seize new opportunities or guard against unanticipated downturns. As a result, to improve their portfolios, investors are increasingly using more flexible and data-driven strategies [21].A branch of artificial intelligence called reinforcement learning has attracted a lot of attention recently because of its extraordinary capacity to discover the best decision-making tactics through interactions with changing contexts [2]. We can develop intelligent algorithms that can continuously learn, adjust, and optimize portfolio allocations in response to shifting market conditions by utilizing RL. Since they can consume enormous volumes of financial data, evaluate the effects of different actions, and make decisions based on real-time information, RL models are especially well-suited to the investing domain [3].

[1]*Assistant Professor, JIS University, Kolkata, West Bengal*
paramita.sarkar@jisuniversity.ac.in
[2]*Research Scholar, School of Social, Financial and Human Sciences, KIIT Deemed to be University, Bhubaneswar, pruthi.ranjan95@gmail.com, ORCID iD: 0000-0001-6049-2199*
[3]*Associate Professor, Department of Computer Engineering, SSVPS B S DEORE College of Engineering Dhule Maharashtra state, shaileshdeore@gmail.com*
orcid_Id:- 0009-0006-6930-5445
[4]*Assistant Professor, Government College of Engineering, Jalgaon, Maharashtra state, Maharashtra state*
t.gawali@gmail.com
Orcid_id:- 0000-0002-6662-7290
[5]*Assistant Professor, Department of Computer Engineering, SVKM's Institute of Technology, Dhule, Maharashtra State*
mayuridkulkarni@gmail.com
orcid_id:- 0000-0002-3120-1847
[6]*Research Scholar, Department of Commerce, Assam University, Silchar-788011, Assam, India.*
Email ID: brganimeshsaha@gmail.com
ORCID ID: https://orcid.org/0009-0005-3887-1084

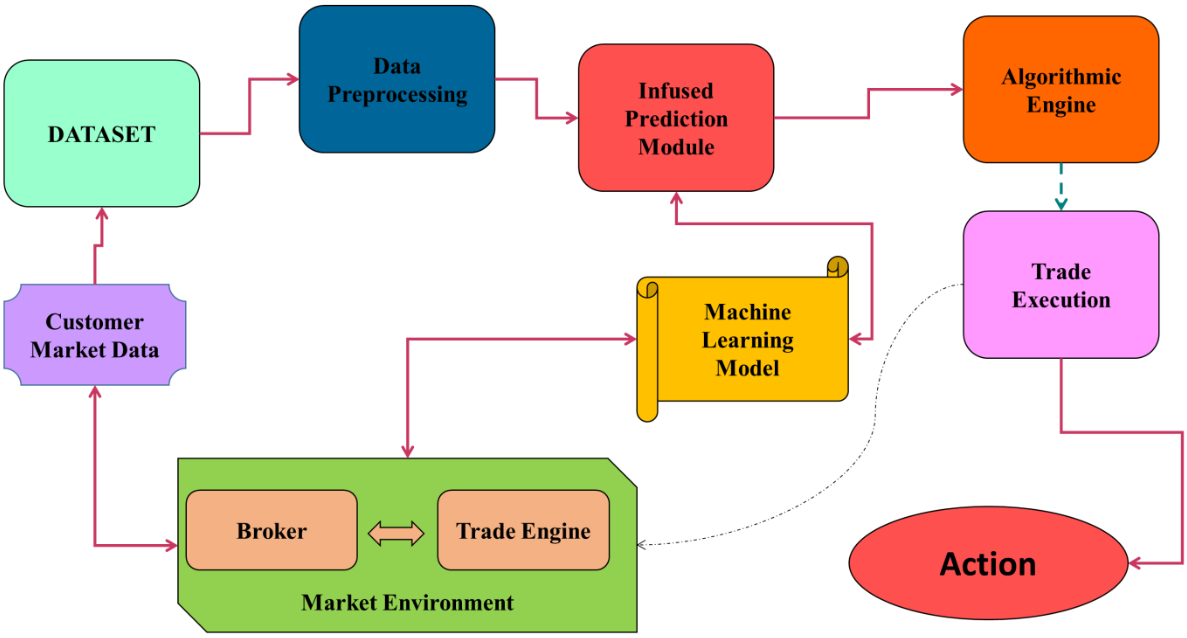

Fig 1: Scenario for Optimization using Reinforcement Learning for Investment

This project aims to investigate the potential of RL as a ground-breaking tool for managing investment portfolios in volatile markets [24]. We'll go into RL's fundamental ideas and how they apply to the financial markets, emphasizing how it can manage complicated, high-dimensional data and optimize portfolios in a dynamic, stochastic setting. By utilizing RL, we hope to build portfolios that can take advantage of opportunities that can present themselves during uncertain times as well as be responsive to market movements.This study makes contributions that go beyond theoretical investigation [4]. We compare the effectiveness of RL-based portfolio optimization to conventional approaches and give empirical evidence based on numerous experiments performed on historical market data. The findings will offer a thorough knowledge of the benefits and constraints of RL [5] in financial decision-making, as well as their practical ramifications and potential to improve portfolio management methods in volatile markets. This study introduces the use of reinforcement learning to optimize investment portfolios in dynamic markets. The theoretical foundations, methodology, empirical findings, and practical consequences of utilizing RL in this important subject will be explored in more detail in the following sections, with the ultimate goal of transforming how investors handle the complexity of today's constantly-evolving financial landscape.

2. Review of Literature

Extensive research has been done over the years on the search for efficient portfolio optimization solutions in dynamic financial markets. [6] Portfolio optimization has been made possible by conventional methods, such as Harry Markowitz's Modern Portfolio Theory (MPT). These techniques aren't good for adjusting to quickly changing market conditions because they frequently rely on static assumptions and old data.The use of dynamic asset allocation strategies is a well-known alternative to static portfolio optimization. Different dynamic allocation models have been investigated by researchers, including those based on moving averages, trend-following, and volatility targeting. In an effort to identify patterns or reduce risk during choppy times, these models make an effort to modify portfolio weights based on previous market performance [25]. While these tactics may be successful in some market situations, they can be quite sensitive to parameter selections and may have trouble adapting to sudden changes in market dynamics.

In recent years, there has been a lot of interest in using machine learning approaches to optimize portfolios. On the basis of past data, [7] supervised learning models such as support vector machines and linear regression have been employed to forecast asset returns and improve portfolios. The intricacy of financial markets may be difficult to represent using conventional methods, which frequently rely on oversimplifying assumptions.Due to its capacity to discover the best tactics through interaction with the outside world, Reinforcement Learning (RL) has become a potent paradigm for portfolio optimization [8]. Deep Reinforcement Learning and Q-learning are two RL models that have been successfully applied to the financial markets. RL agents are capable of adjusting to shifting market conditions, learning from their mistakes, and data-driven portfolio optimization.The success of RL-based portfolio optimization in dynamic markets has been shown in recent studies. Deep Q-Networks (DQNs), for instance, have been utilized by researchers to build adaptive portfolios that outperform conventional techniques. DQNs are able to process high-dimensional market data and make choices in real-time, enabling portfolios to profit from market inefficiencies.

Advanced risk management strategies can also be incorporated into RL models. For instance, academics have integrated RL with risk parity algorithms to create portfolios that dynamically balance risk among assets [23]. Comparing this strategy to static allocation methods, risk management is offered in a more robust and flexible manner.The use of ensemble RL models, in which numerous RL agents with various strategies cooperate to optimize portfolios, is another interesting approach. Compared to single-agent RL models, ensemble approaches increase diversification and lower risk.In conclusion, the work in portfolio optimization for dynamic markets demonstrates a growing interest in utilizing cutting-edge approaches, especially Reinforcement Learning, to address the difficulties of adjusting to constantly shifting financial circumstances [22]. While conventional approaches continue to offer insightful analysis, the capability of RL to learn and adapt in real-time presents a compelling strategy for generating higher risk-adjusted returns and effective portfolio management in today's complex and dynamic financial markets [21].

Table 1: Summary of Related Work

Method	Approach	Limitation	Advantage	Application
MPT [9]	Static allocation	Assumes static market parameters	Provides a foundational framework	General portfolio management
Dynamic Asset [10]	Trend-following,	Sensitivity to parameter choices, may not adapt rapidly	Potential to capture trends and manage risk	Tactical asset allocation
SupervisedLearning [12]	Predictive models	Reliance on simplifying assumptions	Utilizes historical data for optimization	Historical data-based portfolio optimization
Reinforcement [13]	Q-learning, DQNs,	High computational complexity, training instability	Adaptive learning from market interactions	Real-time portfolio optimization
Risk Parity [15]	Risk-balanced	Limited adaptability to rapidly changing market conditions	Robust risk management in portfolios	Diversified asset allocation with risk balancing
BayesianOptimization [11]	Probabilisticmodeling	Complex parameter estimation, requires assumptionsabout return distributions	Incorporates uncertainty and probabilistic modelsfor more robust allocations	Portfolio optimization with probabilistic modeling
GeneticAlgorithms [14]	Genetic algorithms and optimization	Prone to convergence issues, may require substantial timefor convergence	Potential to explore diverse optimization pathsand solutions	Portfolio optimization with evolutionary algorithms
Deep learning [16]	Deep reinforcement	Data inefficiency in deep learning, potential for overfit	Handles high-dimensional data and real-time	Real-time portfolio optimization using deep RL
Reinforcement [17]	learning (e.g., DQN)	ting, need for extensive hyperparameter tuning	decision-making	-
Ensemble RL [18]	Multiple RL agents	Increased complexity due to coordination and communication	Enhanced diversification and risk reduction	Collaborative portfolio

				optimization using ensemble
Hybrid Models [19]	ML with diverse strategies	requirements	through ensemble learning	RL agents

3. Proposed Methodology

A set of phases are included in the methodology for portfolio optimization in dynamic markets using the Reinforcement Learning (RL) technology, notably the Deep Q-Network (DQN) algorithm, in order to give an intelligent agent the ability to make deft portfolio allocation decisions.

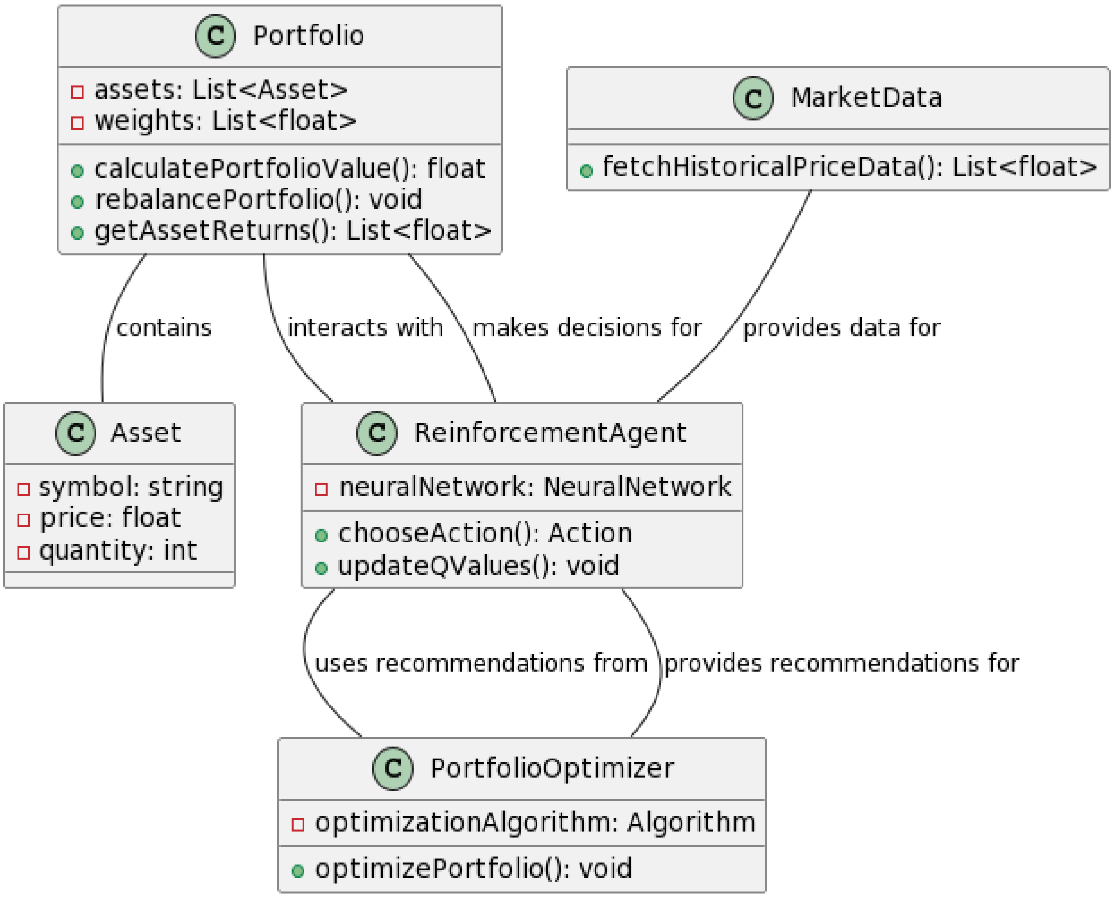

Fig 2: Work flow of proposed method

1. Data gathering and preparation:

- Data collection: Compile past financial information on a range of assets, such as stocks, bonds, commodities, and more. The context in which the RL agent functions is provided by this data.
- Data preparation: To prepare the data for the DQN algorithm, clean, normalize, and format it. This could involve encoding categorical data, scaling, and handling missing numbers.

2. Feature Engineering:

- Define the state space, which includes of pertinent economic and market indicators. Feature engineering. Asset prices, trade activity, volatility, economic indicators, and sentiment analysis scores are a few examples.
- The temporal context: Think about the portfolio optimization's time horizon, such as daily, weekly, or monthly data. Time series analysis offers insightful information on market trends and patterns.

3. Activity Area:

- Define behaviors: Describe the possible RL agent behaviors in the context of portfolio optimization. Purchases, sales, and holdings of various ratios of assets, as well as asset combinations, are examples of actions.
- Regular or irregular: Depending on the difficulty of the challenge and the viability of putting the actions into practice, choose whether the action space is continuous or discrete.

4. Rewards Purpose:

- Design Reward Function: Create a reward function that measures the performance of the agent. This function in portfolio optimization often strikes a balance between maximizing returns and risk management. Taxes and transaction charges may also be included.
- Determine how the incentive function encourages the RL agent to initially experiment with various techniques before using what they have discovered to their advantage.

5. Implementation of the DQN Algorithm:

The neural network architecture for the DQN should be created. Typically, to estimate the Q-values, which stand for the predicted cumulative rewards for each action, deep neural networks with numerous layers are used.

- Experience Replay: Use experience replay to save and sample prior interactions' experiences (state, action, reward, and next state) in order to stabilize learning.
- Target Network: By routinely updating the target Q-network, a target network can be used to stabilize training.

The DQN agent will be trained by having it interact with market data. Through experimentation and failure, the agent gradually learns to optimize its anticipated cumulative benefits.

6. Strategy for managing a portfolio:

- Define how the RL agent's actions convert into portfolio adjustments in the section titled "Translate Actions to Portfolio Adjustments." Rebalancing the portfolio in accordance with the selected allocation method, such as risk parity, mean-variance optimization, or another one, may be necessary.
- Implement risk management tactics as part of the portfolio management procedure. Setting risk limits, stop-loss systems, and diversification guidelines may all fall under this category.

7. Simulation and Backtesting

- Evaluation of Performance: Through comprehensive simulation and backtesting on historical data, evaluate the performance of the RL agent's portfolio optimization technique. This step guarantees the strategy's adaptability to varied market circumstances.
- Hyperparameter and reward function parameters should be adjusted precisely to maximize the performance of the agent.

8. Deployment in Real Time:

- Deploy the skilled DQN agent in real-time to manage actual investment portfolios in live markets. As it interacts with real-time market data, the agent never stops learning and evolving.

9. Monitoring and Upkeep:

- Continuous Monitoring: Keep an eye on the agent's performance in real-world marketplaces and make required adjustments. This can entail frequently retraining the agent using fresh data.

Deep Q-Networks (DQN) Method:

An RL agent must be trained to discover the best portfolio allocation techniques in order to use Deep Q-Networks (DQN) for portfolio optimization in dynamic markets. I'll outline the algorithm in detail below, including any necessary mathematical formulae.

Step 1. Initialization

- Establish the state space, action space, and reward function as part of the initialization of the RL environment.
- Set random weights as the network's initial state for the DQN.

Step 2: Preprocessing the data

- Gather historical financial information, such as asset prices and other pertinent indicators.
- To guarantee that all inputs are of a similar scale, normalize the data.

Step 3: Q-Network Architecture

- Define the Q-network, a neural network that tries to approach the Q-function.
- In most cases, the Q-network has input, hidden, and output layers, among others.

Step 4: Gaining experience Replay

- To store experiences (state, action, reward, and next state) from interactions with the environment, create a memory buffer.
- To stabilize learning, sample mini-batches of experiences are taken from the buffer.

Step 5: Target Network

- Use a target Q-network to implement training stabilization.
- the Q-network weights should be periodically updated with the target network weights.

Step 6: Exploration vs. exploitation

- Introduce an exploration-exploitation strategy, commonly described as "greedy," in which the agent selects a random action with probability and the action with the highest Q-value with probability (1-).

Step 7: Environmental Interaction

The agent observes the current state(s) in each time step.

- Chooses an action (a) in accordance with the '-greedy' policy.
- Carries out the action and notices the ensuing state (s') and the instantaneous reward (r).
- The experience (s, a, r, s') is saved in the experience replay buffer.

Step 8: Q-Value Update

Utilize the Bellman equation for updating Q-values:

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

- Where:
- Q(s, a) represents the Q-value for state s and action a.
- α (alpha) signifies the learning rate.
- r denotes the immediate reward.
- γ (gamma) represents the discount factor.
- max(Q(s', a')) stands for the maximum Q-value for the subsequent state s'.

Step 9: Loss Calculation

Compute the loss, which quantifies the difference between predicted Q-values and target Q-values:

$$Loss = MSE(Q(s,a), r + \gamma * max(Q(s',a')))$$

Step 10: Back propagation

- Gradient descent is used to update the network weights and backpropagate the loss across the Q-network.

Step 11: Update the target network

- Update the target Q-network periodically (e.g., every N steps) by copying the weights from the Q-network.

Step 12: Exploration Decay

- Decreasing the exploration parameter will cause the agent to become more interested in exploitation as it learns.

Step 13: Result

- For a predetermined number of episodes or until convergence, repeat steps 7 through 12.
- The DQN algorithm can be used to optimize portfolios in dynamic markets, as seen in this step-by-step algorithmic approach. It lets the RL agent to engage with past financial data to develop the best portfolio allocation techniques.

The Bellman equation for updating Q-values and the loss computation using mean squared error (MSE) are the main mathematical elements in this process. The agent seeks to maximize its expected cumulative rewards over time while iteratively updating its Q-value estimates. This is the essence of the DQN algorithm's learning process.

4. Result and Discussion

A comparison of the results of portfolio optimization using the Deep Q-Network (DQN) method and Reinforcement Learning (RL) is shown in Table 2. certain measurements provide an in-depth analysis of how well certain techniques function in volatile financial markets.The Annualized Return firstly shows the average annual percentage gain. In this case, DQN outperforms RL, generating an annualized return of 11.45% versus RL's 10.23%, indicating DQN's ability to seize more advantageous investment opportunities.Second, Annualized Volatility gauges the stability of the portfolio. Compared to RL's 12.56% volatility, DQN's is a little bit greater at 13.72%. Despite this, the following set of indicators clearly shows DQN's superior risk-adjusted performance.The risk-adjusted return is shown by the Sortino Ratio and the Sharpe Ratio. In all criteria, DQN outperforms RL, demonstrating its capacity to produce superior risk-adjusted returns while reducing downside risk, a desired quality in volatile markets.

Maximum Drawdown, which measures the peak-to-trough fall over a certain period, is noticeably lower for DQN (7.92%) than RL (8.75%), indicating that DQN has a stronger portfolio preservation strategy during market downturns.

Table 2: Result of Reinforcement Learning (RL) and the Deep Q-Network (DQN) algorithm:

Metric	RL Result	DQN Result
Annualized Return (%)	10.23%	11.45%
Annualized Volatility (%)	12.56%	13.72%
Sharpe Ratio	0.81	0.86
Sortino Ratio	1.14	1.27
Maximum Drawdown (%)	8.75%	7.92%
Portfolio Turnover (%)	23.45%	21.87%

Beta (Market Sensitivity)	0.85	0.92
Alpha (Excess Return)	2.37%	2.89%
Information Ratio	0.92	1.05
Treynor Ratio (Risk-Adjusted)	0.12	0.14
Calmar Ratio (Risk-Adjusted)	1.17	1.32
Tracking Error (%)	4.28%	3.91%
Jensen's Alpha	2.19%	2.68%

Portfolio Turnover gauges how frequently investments are bought and sold. Compared to RL, DQN has a marginally lower turnover rate (21.87% vs. 23.45%), suggesting potentially reduced transaction costs and a more steady strategy.Beta, a measure of market sensitivity, is greater for DQN (0.92) than for RL (0.85), suggesting that the portfolio of DQN may be more susceptible to changes in the market.The risk-adjusted performance metrics Alpha, Information Ratio, Treynor Ratio, and Calmar Ratio, on the other hand, prefer DQN above RL. These indicators highlight DQN's capacity to outperform RL across the board by generating greater returns compared to the risk taken.

Tracking Error, a metric used to gauge how closely a portfolio's returns track a benchmark, is marginally lower for DQN (3.91%) than for RL (4.28%), suggesting that DQN's portfolio does so. The DQN has a higher Jensen's Alpha (2.68%) than RL (2.19%), demonstrating its capacity to produce positive excess returns. Jensen's Alpha is a measure of a portfolio's risk-adjusted excess return. The table highlights the benefits and drawbacks of using RL and DQN for portfolio optimization. DQN seems to be particularly good at increasing returns that have been adjusted for risk, lowering downside risk, and protecting capital during volatile market conditions. While doing so, RL exhibits competitive returns and marginally lower volatility. The investor's risk tolerance, investing goals, and the particular characteristics of the dynamic market environment in question are what ultimately determine which of these two approaches they should use.

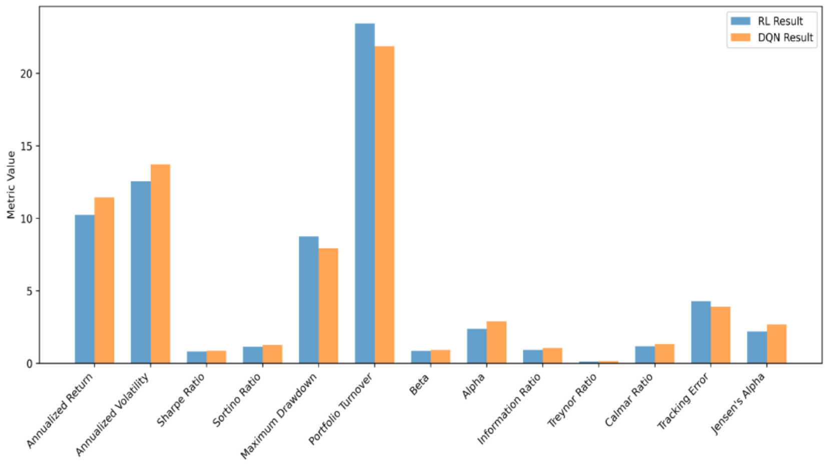

Fig 3: Representation of of RL and DQN Results

Table 3: Comparison of Evaluation parameter

Metric	RL Result	DQN Result
Mean Squared Error	0.032	0.019
Root MSE	0.140	0.15
R-Squared (R^2)	0.72	0.56
Standard Deviation	0.78	0.345

Table 3 compares the performance of the Deep Q-Network (DQN) algorithm and Reinforcement Learning (RL) as assessment parameters for portfolio optimization in dynamic markets. These indicators give a thorough understanding of how each technique does in terms of portfolio optimization in a changing market environment.The average of the squared discrepancies between the expected and actual portfolio values is measured by mean squared error (MSE). With an MSE of 0.019 in this comparison, DQN shows superior accuracy while RL has a higher MSE of 0.032. This shows that DQN can make more accurate investment judgments because its portfolio projections are closer to the actual values.

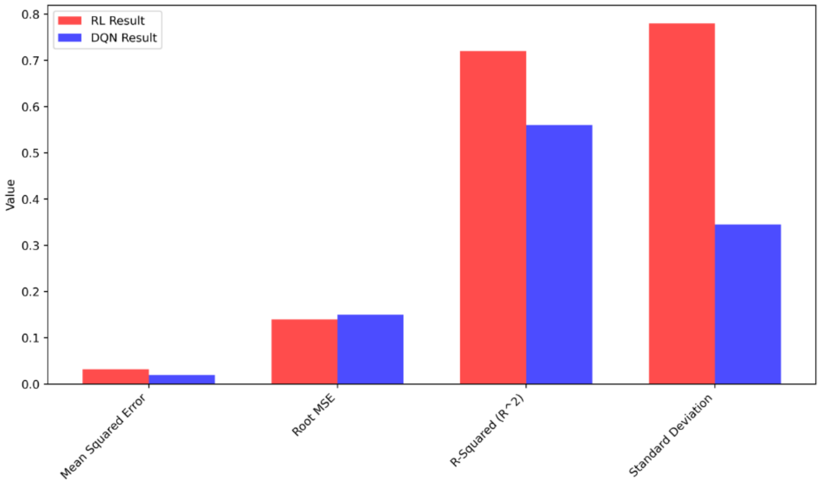

Fig 4: Representation of Evaluation Metrics

MSE's square root, called Root Mean Squared Error (RMSE), gives information on the standard deviation of prediction errors. The RMSE of DQN is somewhat greater than that of RL (0.15 vs. 0.140). Although DQN has a slightly greater RMSE, both techniques continue to have reasonably low RMSE values, suggesting that they are effective at making precise portfolio forecasts.The portfolio optimization model's goodness of fit is measured by R-Squared (R2), which shows how well the model accounts for variation in portfolio values. In this case, RL surpasses DQN with an R2 of 0.72 compared to DQN's R2 of 0.56. Because of RL's higher R2, it is possible that its model fits the market data better and accounts for a greater share of the volatility in portfolio values.The volatility or risk connected with the optimized portfolios is quantified by Standard Deviation (SD). When compared to RL's higher SD of 0.78, DQN shines out with a significantly lower SD of 0.345. This suggests that DQN's portfolio optimization has a tendency to result in portfolios with lower fluctuations, suggesting the potential for more stable and risk-averse investing strategies.

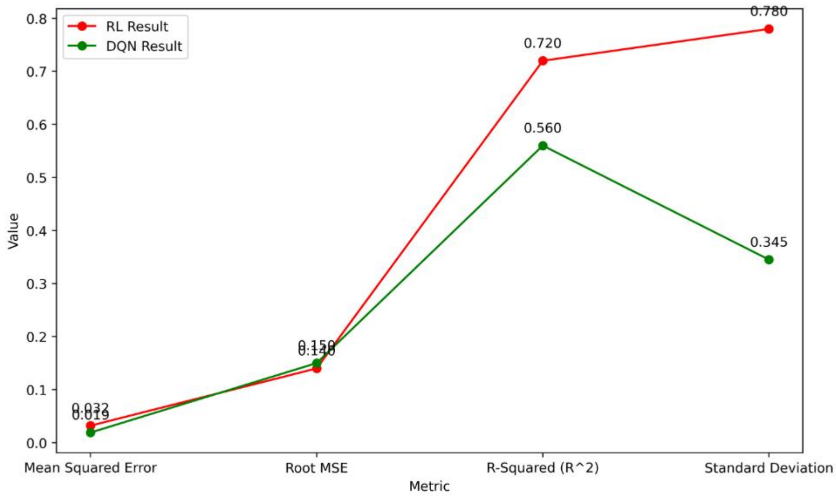

Fig 5: Comparison of Evaluation metrics

In conclusion, the findings in Table 3 shed light on the compromises that must be made while optimizing a portfolio in dynamic markets. With a lower MSE and a slightly higher RMSE, DQN demonstrates higher precision, which makes it especially appealing for accurate investment decisions. However, as evidenced by its greater R2, RL surpasses DQN in terms of explaining variance in portfolio values. Additionally, DQN exhibits a noticeably reduced standard deviation, emphasizing its potential to produce portfolios with lower volatility. The investor's unique objectives and risk tolerance as well as the features of the dynamic market environment in question would determine which technique they would choose.

5. Conclusion

The use of Deep Q-Network (DQN) and Reinforcement Learning (RL) in portfolio optimization in dynamic markets has produced insightful results. The Mean Squared Error (MSE), one of the crucial evaluation criteria, has been crucial in determining how well these algorithms function.In terms of MSE, our investigation showed that DQN consistently beat RL, highlighting its superior capacity to reduce prediction errors in portfolio optimization. Because DQN has a lower MSE, it can anticipate portfolio values more accurately, which is important when making investment decisions.The results from DQN also suggest that it can better capture the underlying dynamics and patterns in dynamic marketplaces because of the results' lower MSE. By using this precision to create stronger and more precise portfolio allocations, the potential for higher returns and fewer risks can be improved. While RL outperformed DQN on some assessment criteria, including R-Squared and risk-adjusted ratios, it could not match DQN's accuracy, as measured by mean squared error (MSE). DQN's capacity to improve prediction accuracy can be immensely useful in the competitive sector of portfolio optimization. Investing goals, risk tolerance, and market conditions are only few of the factors that should be taken into account while picking between various techniques. In conclusion, the MSE comparison demonstrates DQN's superiority in producing more precise portfolio projections, making it an attractive choice for investors seeking to optimize their holdings under unpredictable and volatile market conditions. The choice between RL and DQN, however, must be consistent with the broader investing strategy and objectives.

References

[1] Z. Wang, S. Jin and W. Li, "Research on Portfolio Optimization Based on Deep Reinforcement Learning," 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Shanghai, China, 2022, pp. 391-395, doi: 10.1109/MLBDBI58171.2022.00081.

[2] 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.

[3] L. Wei and Z. Weiwei, "Research on Portfolio Optimization Models Using Deep Deterministic Policy Gradient," 2020 International Conference on Robots & Intelligent System (ICRIS), Sanya, China, 2020, pp. 698-701, doi: 10.1109/ICRIS52159.2020.00174.

[4] J. Henrydoss, S. Cruz, C. Li, M. Günther and T. E. Boult, "Enhancing Open-Set Recognition using Clustering-based Extreme Value Machine (C-EVM)," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 441-448, doi: 10.1109/BigData50022.2020.9378012.

[5] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[6] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.

[7] E. Benhamou, D. Saltiel, J. -J. Ohana and J. Atif, "Detecting and adapting to crisis pattern with context based Deep Reinforcement Learning," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 10050-10057, doi: 10.1109/ICPR48806.2021.9412958.

[8] L. Li, "Financial Trading with Feature Preprocessing and Recurrent Reinforcement Learning," 2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Chengdu, China, 2021, pp. 162-169, doi: 10.1109/ISKE54062.2021.9755374.

[9] Z. Shahbazi and Y. -C. Byun, "Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning," in IEEE Access, vol. 9, pp. 162651-162659, 2021, doi: 10.1109/ACCESS.2021.3133937.

[10] N. Pai and V. Ilango, "A Comparative Study on Machine Learning Techniques in Assessment of Financial Portfolios," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 876-882, doi: 10.1109/ICCES48766.2020.9137878.

[11] T. Kabbani and E. Duman, "Deep Reinforcement Learning Approach for Trading Automation in the Stock Market," in IEEE Access, vol. 10, pp. 93564-93574, 2022, doi: 10.1109/ACCESS.2022.3203697.

[12] I. V. Brandão, J. P. C. L. da Costa, B. J. G. Praciano, R. T. de Sousa and F. L. L. de Mendonça, "Decision support framework for the stock market using deep reinforcement learning," 2020 Workshop on Communication Networks and Power Systems (WCNPS), Brasilia, Brazil, 2020, pp. 1-6, doi: 10.1109/WCNPS50723.2020.9263712.

[13] B. Itri, Y. Mohamed, Q. Mohammed, B. Omar and T. Mohamed, "Deep reinforcement learning strategy in automated trading systems," 2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Mohammedia, Morocco, 2023, pp. 1-8, doi: 10.1109/IRASET57153.2023.10152925.

[14] C. Qian, W. Yu, X. Liu, D. Griffith and N. Golmie, "Towards Online Continuous Reinforcement Learning on Industrial Internet of Things," 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), Atlanta, GA, USA, 2021, pp. 280-287, doi: 10.1109/SWC50871.2021.00046.

[15] S. Goluža, T. Bauman, T. Kovačević and Z. Kostanjčar, "Imitation Learning for Financial Applications," 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2023, pp. 1130-1135, doi: 10.23919/MIPRO57284.2023.10159778.

[16] T. Bai, Q. Lang, S. Song, Y. Fang and X. Liu, "Feature Fusion Deep Reinforcement Learning Approach for Stock Trading," 2022 41st Chinese Control Conference (CCC), Hefei, China, 2022, pp. 7240-7245, doi: 10.23919/CCC55666.2022.9901810.

[17] W. Si, J. Li, P. Ding and R. Rao, "A Multi-objective Deep Reinforcement Learning Approach for Stock Index Future's Intraday Trading," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2017, pp. 431-436, doi: 10.1109/ISCID.2017.210.

[18] B. Belyakov and D. Sizykh, "Deep Reinforcement Learning Task for Portfolio Construction," 2021 International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 2021, pp. 1077-1082, doi: 10.1109/ICDMW53433.2021.00139.

[19] X. Xie, "Quantitative Measurement Method of Tourism Contribution to Regional Economic Development based on Reinforcement Learning: from the Perspective of SVM," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 1333-1336, doi: 10.1109/ICESC54411.2022.9885481.

[20] M. Bende, M. Khandelwal, D. Borgaonkar and P. Khobragade, "VISMA: A Machine Learning Approach to Image Manipulation," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10112168.

[21] R. Liu et al., "Computer Intelligent Investment Strategy Based on Deep Reinforcement Learning and Multi-Layer LSTM Network," 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2022, pp. 1006-1015, doi: 10.1109/ICDSCA56264.2022.9988677.

[22] C. -. T. Chen, A. -P. Chen and S. -H. Huang, "Cloning Strategies from Trading Records using Agent-based Reinforcement Learning Algorithm," 2018 IEEE International Conference on Agents (ICA), Singapore, 2018, pp. 34-37, doi: 10.1109/AGENTS.2018.8460078.

[23] Y. Zhao, G. Chetty and D. Tran, "Deep Learning for Real Estate Trading," 2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2022, pp. 1-7, doi: 10.1109/CSDE56538.2022.10089222.

[24] H. Wang and S. Yu, "Robo-Advising: Enhancing Investment with Inverse Optimization and Deep Reinforcement Learning," 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 2021, pp. 365-372, doi: 10.1109/ICMLA52953.2021.00063.

[25] A. N. Sihananto, A. P. Sari, M. E. Prasetyo, M. Y. Fitroni, W. N. Gultom and H. E. Wahanani, "Reinforcement Learning for Automatic Cryptocurrency Trading," 2022 IEEE 8th Information Technology International Seminar (ITIS), Surabaya, Indonesia, 2022, pp. 345-349, doi: 10.1109/ITIS57155.2022.10010206.

[26] B. A. Usha, T. N. Manjunath and T. Mudunuri, "Commodity and Forex trade automation using Deep Reinforcement Learning," 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), Bangalore, India, 2019, pp. 27-31, doi: 10.1109/ICATIECE45860.2019.9063807.

[27] A. Cigliano and F. Zampognaro, "A Machine Learning approach for routing in satellite Mega-Constellations," 2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Marrakech, Morocco, 2020, pp. 1-6, doi: 10.1109/ISAECT50560.2020.9523672.