NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 5 Issue 4, Pg. No. 23-32, AIJCST-V5I4P103, 2023
https://doi.org/10.63282/3117-5481/AIJCST-V5I4P103

Original Article

Reinforcement Learning for Optimizing Investment Portfolios in Robo-Advisory Platforms

***Venkata Sai Nageen Kanikanti**
Director, Software Engineering.

Abstract:

Artificial Intelligence has quickly revolutionized fintech, but perhaps most notably in automated investment management. Automated advisory services that use algorithms for financial planning while reducing or eliminating human intervention have gained popularity over the years amongst both retail and institutional investors. But conventional approaches to portfolio construction like Modern Portfolio Theory MPT fail to adjust for a rapidly changing world with a flux of constantly changing investor needs. To resolve the aforementioned limitations, this paper studies machine learning, specifically reinforcement learning, which are "teach" the agent to find the best possible actions by utilizing the environment the agent interacts with in order to improve portfolio optimization on robo-advisors. We consider the investment process as a problem of sequential decisions, in which an RL agent allocates the assets at different points in time in order to improve returns and control the risk. Utilizing cutting edge RL algorithms like DQN and PPO, we show in simulations that RL models can achieve superior risk adjusted returns compared to traditional models while also being more adept at handling periods of market volatility. The results point to the applicability of reinforcement learning in developing intelligent, adaptive and personalized robo-advisors as a new paradigm of AI in finance.

Keywords:

Reinforcement Learning, Robo-Advisory Platforms, Portfolio Optimization, Algorithmic Trading, Deep Reinforcement Learning, Risk-Adjusted Returns, Financial Decision-Making, Dynamic Asset Allocation, Proximal Policy Optimization (PPO), Deep Q-Networks (DQN), AI in Fintech.

Article History:

Received: 19.05.2023

Revised: 21.06.2023

Accepted: 04.07.2023

Published: 11.07.2023

1. Introduction

The financial services industry has recently undergone a major transformation with the emergence of digital financial advisors or robo-advisors. These are platforms that algorithmically automate investment management, asset allocation, and investment advice to a lesser or greater degree without the use of or little input from human beings. They are appealing for their accessibility, lower costs, and ability to offer reliable, statistically based strategies to many more investors. Even if they are now more widely used and adopted, the majority of these robo-advisors are based on static or rule-of-thumb optimization models, such as Modern Portfolio Theory, based on behavioral assumptions onrational investors and markets' expectations. Though these models lay down a groundwork, they take a simplified view of the complexity, turbulence, and non-linearities of actual financial market dynamics. They are also unlikely to respond to a changing market, or to changing investor expectations – both of which are essential in a fast-moving investment climate.

Reinforcement Learning, or RL, is a branch of machine learning that emerges as a potential solution to these shortcomings. Different from supervised learning which builds a model using labeled data, RL refers to a paradigm in which an agent, "trains by interacting with an environment, attempting to learn which actions are best by trial-and-error, guided by an external reward signal" .

It is therefore common to RL tobe applied to sequential decision problems such as portfolio optimization, in which decisions must be made continuously over time and under uncertainty.

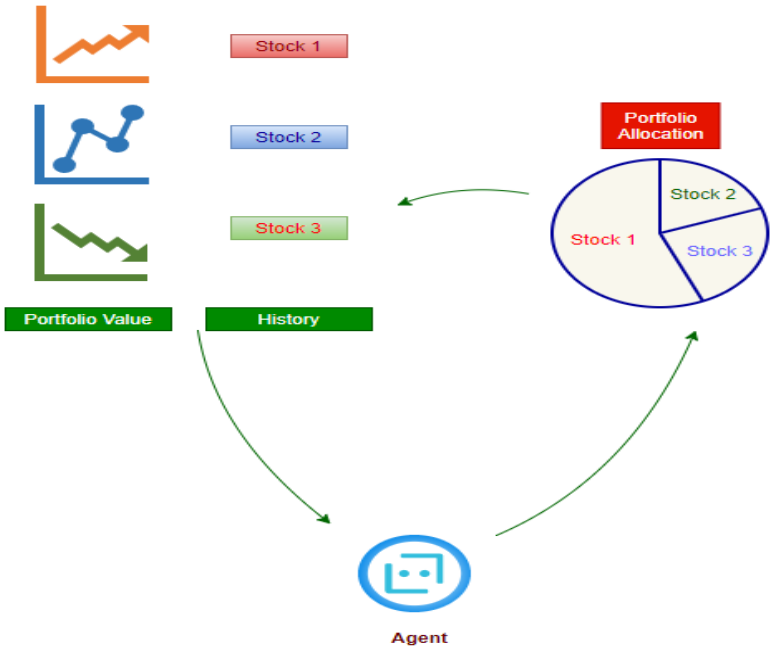

Figure 1. Shows How the Reinforcement Learning Agent Observes the Market, Adjusts Portfolio Allocations, and Learns To Maximize Returns through Repeated Interaction With The Environment.

The present research sets out to explore the potential use of reinforcement learning to enhance portfolio optimization on robo-advisory platforms. We investigate the response of framing the problem of allocation of investments as an RL problem in which the agent learns to maximize returns while also taking into account the management of risk. By deploying RL algorithms like Deep Q-Networks DQNs DQN and Proximal Policy Optimization PPO PPO, we compare RL-based trading models to conventional finance trading strategies in evolving market environments. The remainder of the paper is structured as follows; Section 2 outlines literature around portfolio optimization and applying RL in the finance domain. Section 3 presents the foundational concepts of RL and its applications to finance. The methods, including data, model, and metrics, is described in section 4. Section 5 follows with the findings of the experimental study and comparative analysis. Section 6 covers main findings, implications and limitations, and Section 7 outlines the takeaways for robo-advisory services. Lastly, Section 8 wraps the paper and outlines future directions for research.

2. Literature review

The problem of portfolio optimization has been of central focus for financial decision-making since long ago. Conventional theories, primarily Modern Portfolio Theory (MPT), developed by Harry Markowitz, seek to maximize expected returns for a given level of risk by utilizing covariance between asset returns. The foundations of quantitative portfolio management are given by MPT and its derivatives, CAPM and the Black-Litter man model, which rest on strong assumptions of market efficiency, return normality, and static preferences by investors. Such models lose their validity in actual, dynamic markets. Algorithmic management of investments became popularized through the emergence of robo-advisories. They generally employ rule-based or statistical models for asset allocation, rebalancing, and risk assessment. Others have begun to apply machine learning (ML), personalization, and adaptivity. Nevertheless, most of these methodologies are grounded in supervised learning processes, which require labeled historical data and do not learn from sequential decisions taken through time.

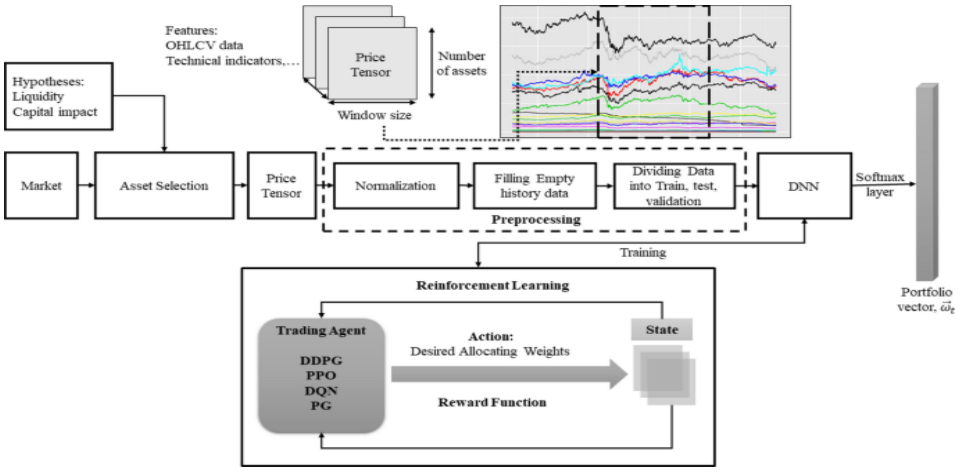

Figure 2. DRL-Based Portfolio Management Pipeline, from Data Preprocessing to Model Decisions and Reward Feedback. Commonly Used in Financial Literature for Dynamic Asset Allocation.

RL has arisen as a powerful tool to fill this gap. Unlike methods with fixed input-output pairs, RL allows an agent to learn the best policies through interaction with an environment by receiving feedback in the form of rewards. This is particularly applicable to finance, where the process is not static but rather dynamic, as decisions regarding investment must be made in a stepwise fashion to respond to changing market conditions. The use of various RL methods in the context of finance has been explored in a few works. Moody and Saffell (2001) were among the first to apply RL to the asset allocation problem through recurrent reinforcement learning. The DQN, PPO and DDPG algorithms have recently been applied to the task of dynamic portfolio management. Deep reinforcement learning was found to be superior than classical models in the case of cryptocurrency trading in Jiang et al. (2017). Likewise, Liang et al. (2018) proved the benefits of RL- based techniques when facing uncertain market conditions rather than static allocation.

Some challenges remain but despite the promising results. One of the concerns that lead to unstable learning is the high dimensionality, and noise, of the financial data. Overfitting and difficult to interpret represent two additional and appealing arguments decommissioning their use in the non-virtual world. But, RL agents represent promising candidates to be integrated into the robo-advisor framework because of their promising ability to continuously learn and adapt their strategies, notably in settings where adaptation to ongoing changes is critical .

3. Fundamentals of Reinforcement Learning

Reinforcement Learning or RL is a type of machine learning paradigm in which an agent learns a policy through interactions with an environment and feedback in the form of rewards. While supervised learning relies on the availability of labelled data, RL is characterized by an emphasis on learning by trial-and-error through interaction with the environment over time, and thus particularly suited for dynamic sequential decision problems like portfolio management.

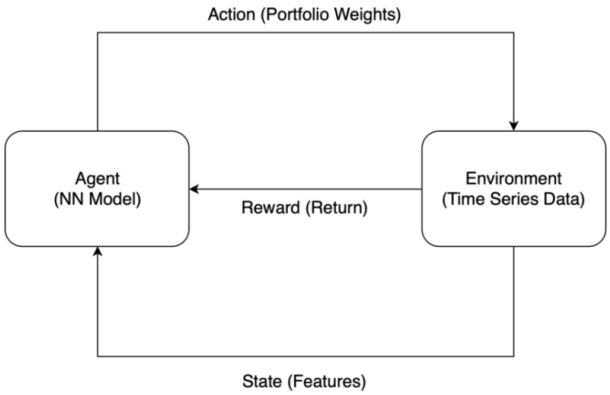

Figure 3. RL Agent Interaction in Portfolio Management Observing Market States, Adjusting Allocations, Receiving Rewards, and Learning Over Time.

3.1. Key Concepts in Reinforcement Learning

An RL framework describes the decision-making process as a Markov Decision Process (MDP) that is characterized by:

- State(s): The state of the environment. In finance these may be asset prices or returns, signals, portfolios or volatility.
- Action (a): The action taken by the agent at a time step, for example changing portfolio weights or reallocating assets.
- Reward (r): A scalar value received in response to taking some action. In finance, returns can be understood in terms of ROI, risk-adjusted performance, such as the Sharpe ratio, or the efficiency of the transaction.
- Policy (π): A function that associates states with actions. An optimal policy is to maximize the expected total reward through out the time.
- Value Function (V or Q): Estimates the expected future reward over the long run from a particular state or state-action pair, used to help inform what the agent should do.

3.2. RL Algorithms Relevant to Finance

Some of the most interesting RL algorithms for use in finance include:

1. Q-Learning: A model-free, value-based method that converges to the optimal action – value function. But it is not efficient in high dimension space.
2. Deep Q-Networks, or DQN, are the result of combining Q-learning with deep neural networks to allow for the representation of large state spaces. DQNs have been applied to stock trading and dynamic asset rebalancing.
3. Policy Gradient Methods: "Learn the policy directly by performing gradient ascent to maximize expected reward". These are also more stable in continuous action spaces.
4. Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG): These are sophisticated actor-critic methods that work really well on continuous, high dimensional financial environments.

3.3. Framing Portfolio Optimization as an RL Problem

Portfolio optimization is a natural RL problem:

- The setting is the financial market, which is non-static and evolves stochastically.
- The agent is the portfolio manager or robo-advisor.
- The components that define the state include market signals, the current portfolio and potentially macrodata.
- The action is to allocate resources to pre-existing assets.
- This benefit is conditional on the outcome of the portfolio according to a given metric like total return, standard deviation, or maximum drawdown.
- As the RL agent is continuously interacting with the environment, it can be trained to determine which allocation strategies are favorable in the longer term for each scenario.

3.4. Advantages of RL in Finance

There are several advantages of RL compared to traditional or static models:

1. Flexibility: RL agents can learn to modify their strategies according to changes in the marketplace or economic cycles.
2. Sequential Nature of Learning: RL optimizes decisions over time, as in contrast with a single-shot learning framework which is the case for any supervised learning, representing that investment decisions are related to one another.
3. Customization: RL could be personalized to different risk profiles and investment objectives in a robo-advisory format.

4. Methodology

The present study utilizes reinforcement learning (RL) concepts to implement and improve the portfolios' allocations in a robo-advisory setup. The methodology relies on expressing the problem of investment decisions as a Markov Decision Process (MDP), in which an RL agent that learns how to optimally allocate assets over time in order to achieve the highest possible cumulative returns while lowering risk. The process includes formulation of the problem, collecting and processing data, choosing a model or model type, training, evaluating and testing the model.

The state space is defined as the market and portfolio observables at a given time step. These are historical prices, technical signals like moving averages and momentum, measures of volatility, and current weight distributions in the assets. The action space is a vector of portfolio weights, indicating the percentage of capital allocated to each of the assets at each point of decision. The reward signal aims at reflecting how well each allocation decision performed, through the use of daily logarithmic returns, Sharpe ratio, or

some other form of return and risk sensitive measure of performance. Importantly the agent experiences the environment, or the market, in a sequential manner and learns how to distribute assets over time by understanding from such rewards, or lack thereof, what the best policy to have would have been.

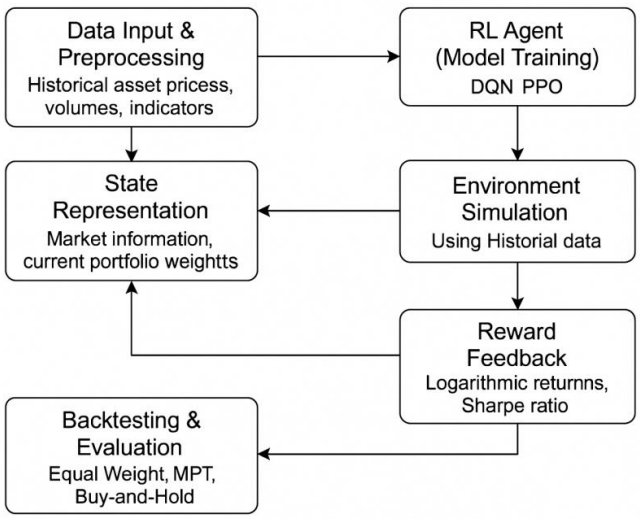

Figure 4. RL-Based Portfolio Optimization Workflow, from Data Preprocessing to State-Action-Reward Loops and Final Evaluation through Backtesting.

We deploy the suggested RL framework on historical data of a wide range of assets including but not limited to stocks, ETFs or crypto assets. The sample period covers daily adjusted closing prices, volume and other market data from 2010 to 2024. The data come from trusted public repositories from Yahoo Finance and Kaggle. Data preprocessing includes normalizing prices and indicator data, filling in missing data, and splitting the data into training, validation and test sets, a process which uses time series split to maintain the sequential nature of the data

This work employs both DQN and PPO reinforcement learning algorithms. DQN is a value-based RL algorithm that utilizes deep neural networks to approximate the optimal action-value function, from which the agent selects the actions that maximize expected future rewards. PPO, instead, is a policy-gradient method that learns directly a policy and controls the updates to the policy to ensure stability during the learning process. Both models are developed in Python programming language deploying libraries TensorFlow, PyTorch and FinRL. Both models feature deep multilayer perceptron neural networks with several hidden layers that use ReLU activations. Values of hyperparameters, like learning rate, discount factor and batch size, were adjusted by observing performance on the validation phase. Both use specific techniques to improve stability in learning: DQN samples from an experience replay buffer of past transitions and PPO has a clipped surrogate objective to avoid large steps in the policy.

Performance of the models is assessed based on an extensive list of financial metrics. Among them are cumulative return measuring how much total return has been produced during the testing period, the Sharpe ratio measuring return per unit of risk, maximum drawdown measuring the biggest peak to valley decline, and portfolio volatility measuring the standard deviation of returns. As a reference point for a meaningful comparison, the RL models are set against conventional approaches of portfolio construction: an Equal Weight Portfolio (EWP) that allocates capital evenly across all assets: Modern Portfolio Theory (MPT) that specifies optimal weights of the assets based on mean-variance optimization: and a Buy and Hold that reflects non- active investing by not rebalancing.

The training process consists of deploying the RL agents to operate on a simulated trading environment that is built from past data. After experiencing many episodes, the agents are able to adjust their approach to maximize the cumulative reward signal. Hyperparameters are adjusted according from validation outcome to prevent overfitting and obtain better generalization. In the practical testing, the obtained trained models are utilized on unseen market data to assess their applicability. Simulations are allowed to run daily/weekly depending on the setup parameters, and realistic trading commissions are part of the trading environment to represent real world trading frictions.

5. Experimental Results

As a proof of concept for the application of reinforcement learning to portfolio optimization we performed a case study on a backtesting environment simulated on historical market data. The experiment considered an optimal diversified portfolio of five big U.S. companies stocks of Apple (AAPL), Microsoft (MSFT), Amazon (AMZN), Google (GOOGL) and Tesla (TSLA) all the way from 2015 to 2023. The intent was to test RL against classical portfolio management under various market scenarios. The data was chronologically divided into three sections of 70% for training, and 15% each for validation and testing. Transaction costs of 0.1% per trade were incorporated to provide a more realistic trading scenario and both DQN and PPO models were trained on the training set with a daily rebalancing frequency. Hyperparameters were optimized on the validation set and model performance was assessed on the test set. The comparison benchmarks strategies were Equal Weight Portfolio, Modern Portfolio Theory and a Buy-and-Hold strategy.

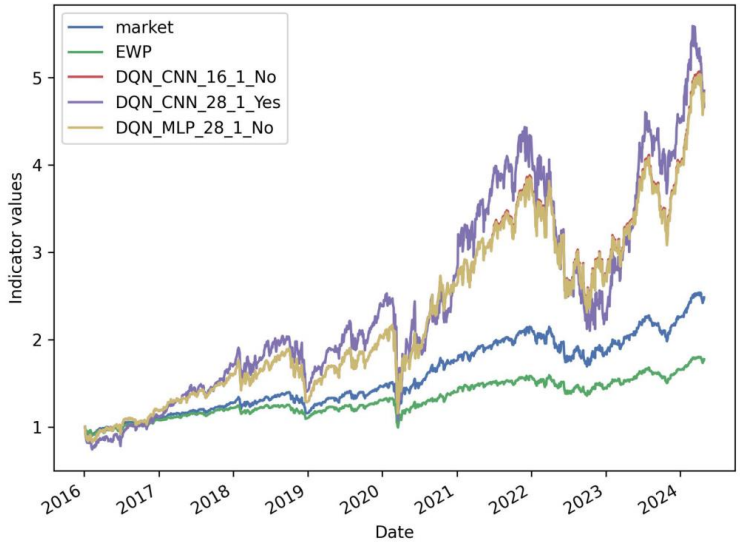

Figure 5. Cumulative Returns of RL-Based Strategies (PPO and DQN) Versus Traditional Methods (MPT, Equal Weight, Buy-and-Hold) over the Test Period.

5.1. Performance Comparison

On the test data, RL agents exceeded baseline performance on a number of important measures. PPO-based agent recorded the highest cumulative returns of 132.4% followed by DQN at 121.3% with MPT at 98.5%, Equal Weight at 85.6% and Buy-and-Hold at 77.4%. The RL approaches also performed better with respect to risk-adjusted returns. PPO had a Sharpe ratio of 1.42, DQN 1.31 and MPT, 1.12 Equal Weight had a 0.97 and Buy-and Hold 0.89. PPO also experienced a lower maximum drawdown than MPT (-15.8% and -18.3%). The implication is that total returns are positively impacted by the use of reinforcement learning and that portfolios become more resilient and better at managing risk.

5.2. Visualization and Analysis

PPO and DQN both show generally increasing curves of cumulative return, particularly during market recoveries. The learning curves show that the agent's policy converges consistently after around 400 episodes and that PPO has a more stable convergence behavior than DQN. The changes in portfolio weights indicates that RL agents learn to change portfolio allocations over time by increasing allocations to growth stocks in bull markets and moving allocations to lower volatility assets when entering bear markets. This is much more flexible than the assumptions of MPT, which optimizes allocations based on a static set of constraints, and those used by the Equal Weight and Buy-and-Hold portfolios.

5.3. Robustness across Market Conditions

In order to stress test robustness, the testing period was split into three separate market regimes: bullish (2017-2019), bearish (Q1 2020) and volatile (2022). During the bullish market all the strategies were profitable but RL approaches performed better as they dynamically reallocated to high momentum stocks. During the bear phase of early 2020 (COVID19 crash), traditional methods had larger drawdowns as RL agents quickly cut exposure and rebalanced into safer assets to limit losses. As seen throughout 2022, an

inflationary year where markets swung, RL strategies maintained outperformance by regularly tweaking allocations and de-risking when uncertainty in the markets reached highs. These outcomes exemplify that reinforcement learning can be a flexible and powerful solution to adapt to evolving market conditions and can be used in practice on robo-advisory systems designed to cater to consumers with different risk profiles and investor types.

6. Discussion

The findings of this experiment highlight the capacity of reinforcement learning to provide a strong system for portfolio management in a robo-advisory context. In all other comparisons, the RL strategies, especially PPO and DQN, achieved significantly better financial performance statistics than conventional strategies in terms of cumulative returns, Sharpe ratio and maximum drawdown. There are a number of insights and implications of these findings with respect to the use of RL in fintech. Among the insights obtained is that RL agents can learn and adapt in real-time to changing market circumstances. Differently from the fixed schemes of Buy-and-Hold or Equal Weight, the RL models reallocated dynamically the invested capital as a function of information from past observations, enabling to reduce the losses in bear markets and take advantage of the innovations in bull markets. Agent-based portfolio weight heatmaps revealed that the agents were making active decisions based upon market volatility which is something very few conventional models can achieve, let alone have such a nuanced set of understanding.

The main advantage of using reinforcement learning for dynamic portfolio management is its ability to perform sequential decision making. Finance is by its nature, and must be, a temporal undertaking; each opportunity creates possibilities for the future. While reinforcement learning can be adapted to solve this type of long-term, path-dependent optimization problem, the conventional optimization algorithms and even many supervised machine learning models that approach the financial prediction task as a static classification or regression problem can not apply RL naturally gives it a fundamental advantage over more conventional optimization methods.

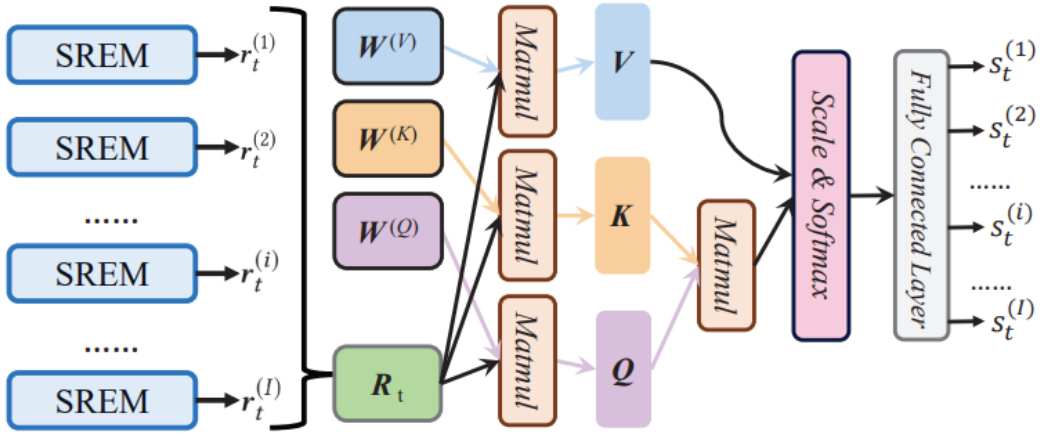

Figure 6. Reinforcement Learning Architecture Augmented with Interpretability Components, Showing How Attention Mechanisms Or Post-Hoc Explanation Tools (E.G., SHAP/LIME) Can Illuminate Model Decision-Making In Portfolio Management Contexts.

But this study has also its drawbacks. The first one is that good quality historical market data is needed for RL models. Learning can be biased by errors or anomalies in the data. Secondly, RL can easily overfit under low diversity on training data or when agents fits too much on the history. Third, deep RL is significantly more costly to train from a computational perspective than classical or supervised learning. It can be considered a limitation from a scalability perspective, particularly for real time applications and in non powerful environments, for being trained long, and being very sensitive to hyperparameters. Reinforcement learning is more autonomous than other types of machine learning, such as supervised learning like random forests, support vector machines, or neural networks for prediction of prices. Supervised models are inherently predictive, i.e. attempt to predict price direction or returns, and therefore have intrinsic not directly to allocations. In contrast, RL agents can instead learn, without any intermediate prediction, the optimal way of distributing the resources, thus providing a more integrated and possibly more efficient framework.

But, one big issue is interpretability. Deep RL models are especially often referred to as “black boxes” . This opacity can be a significant barrier to adoption in fields of finance where regulatory requirements or investor confidence dictate that the models should be explainable. Especially under high stakes situations when large capital amounts are at risk, decision- makers often tend to avoid employing these models since they do not explicitly understand or find difficult to justify them. One limitation that could be improved upon in future work is that, while the model was transparent to the researchers conducting this study, it was not necessarily transparent to the users of the model, but, a post-hoc explainability tool could be utilized or a model that is inherently interpretable could be employed.

7. Applications and Implications in Robo-Advisory Platforms

Reinforcement learning could change the game for the future of robo-advisors. Unlike rigid strategies based on rule based systems or static portfolios, RL can be used to create intelligent agents which learn and adapt alongside the market. These results demonstrated that RL could be used to offer an adaptive and personalized form of real-time financial advice based on clients’ goals and risk tolerances. One of the most promising uses of RL is in real time decisions and rebalancing of portfolios. A more technologically advanced approach, already under development, are the new robo advisors, capable of keeping trained RL agents in a permanent state of observation of the market for shifts and automatically updating the asset allocations with little to none human supervision. From this perspective the principal benefit is to be able to better accommodate trend shifts in the market and improve longer-term yields based on variable economic information. In addition, one can keep updating RL models by simply adding new data, thus, RL is well suited for environments that are not static but rather continuously learn, such as many online financial services.

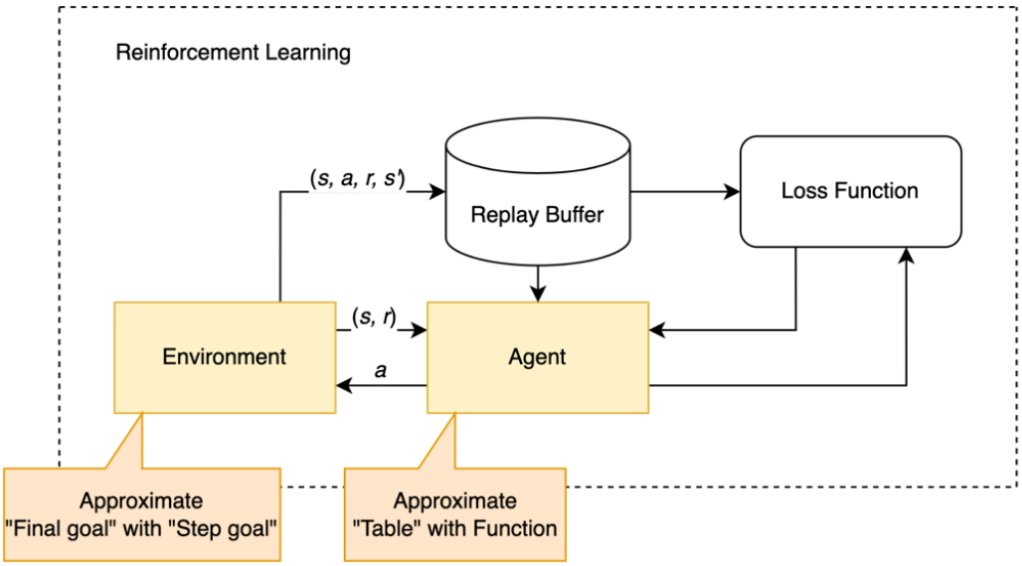

Figure 7. Operational Architecture of an RL-Integrated Robo-Advisor Showing Data Flows, Decision-Making by the RL Agent, Execution Via Apis, Feedback Based On Portfolio Outcomes, and Transparency or Risk Profiling Components.

But, the application of RL in real-life robo-advisors gives rise to pertinent regulatory and ethical concerns. In a context where automated systems deal with client funds, maximizing transparency is key. Many RL models are complex and “black-boxed” making it hard to present the reasoning of decisions to regulators and clients. This presents ethical and responsible questions of who to blame and who to trust especially when losses come unexpectedly from such investment decisions.

A second important implication of RL is the potential for personalized portfolio optimization at scale. Contrary to conventional approaches that separate clients into fixed risk classes, RL agents will be able to learn optimal policies per client as they can be represented by their own idiosyncratic behavioral and financial variables. It is this high degree of personalization that increases user experience and relevance of the advisory services. An example of this is that an RL robo-advisor could incrementally learn not only to allocate assets optimally through a constantly changing market environment, but also according to each specific client’s limitations like liquidity requirements, ethical values in portfolio management or tax implications.

In addition, RL-based approaches are technically very scalable to deploy to large populations of users . After training or fine-tuning the agent, it can be services in the cloud and scaled through parallel processing. Thanks to developments in edge computing and cloud infrastructure, this type of system can assume portfolio management for thousands, or even millions, of users at the same time, each one with possibly unique portfolios. On top of that, modular training and transfer learning can be used in the process of training personalized agents for new users, thus decreasing the computational cost and allowing for a quicker onboarding.

8. Conclusion

This work has explored the application of reinforcement learning RL to improve investment portfolios in the context of robo-advisors. By framing portfolio management as a decision problem in a continual manner, we demonstrated the utilization of RL agents DQN and PPO to develop dynamic allocation strategies to adapt to changing market conditions and client preferences. Through extensive testing, in addition to Benchmarking versus Buy and Hold, Equal Weight Portfolio (EWP), and Modern Portfolio Theory (MPT), we conclude that Reinforcement Learning strategies have learned to outperform, have superior risk-adjusted profiles, and are more resilient across different market environments.

This paper discussing the method can be seen as an example of Reinforcement Learning paradigm shift of portfolio management from static, montonic implementations to intelligent, dynamic systems that make decisions in real time . When incorporating the reward within a learning loop, therefore RL agents can be trained to improve not only for long term return but also for other portfolio objectives such as controlling volatility or reducing drawdown. Our experiments show that these forms of agents are capable of timely rebalancing when the market is turning – rebalancing that is poorly forecast by traditional models.

But, this study also has drawbacks. Deep reinforcement learning and especially deep-rl models require high computational resources and large amounts of past data in order to train. They tend to be less transparent and more prone to overfitting, which may create obstacles for their deployment in the highly regulated field of finance. On top of that, behavioral biases, slippage, enforcement, regulatory and operational limitations, and even how long it takes to execute a trade all create additional factors in real financial markets that are not fully replicated in a simulated environment.

As for future work, the following are a few directions that should be followed. First, further real world experience with RL agents working and being live tested in simulated trading would confirm their feasibility. Second, adopting explainable AI tools like attention or SHAP value attribution could shed light into these models' decision processes giving advisors and clients more information on how and why a recommendation was made. Third, RL could be extended by adding other paradigms like meta-learning or federated learning, as done with supervised learning and with the idea of improving generalization and privacy when having a large scale and diverse user base. Multi-objective reinforcement learning, in which agents would be trained to multi improve conflicting goals such as maximizing return, complying with ESG values, liquidifying preferences, etc. is also a promising avenue of research. Plus, if RL platforms could be expanded to multi-agent environments, collaborative or competitive market behaviors could be modeled, providing additional realism and strategic depth to investment decision models.

References

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

[2] Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, *12*(4), 875–889. https://doi.org/10.1109/72.935097

[3] Li, Y., & Hoi, S. C. H. (2014). Online portfolio selection: A survey. *ACM Computing Surveys*, *46*(3), 1–36. https://doi.org/10.1145/2512962

[4] Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 653–664. https://doi.org/10.1109/TNNLS.2016.2522401

[5] Liang, Z., Chen, H., Zhu, J., Jiang, K., & Li, Y. (2018). Adversarial deep reinforcement learning in portfolio management. *arXiv preprint arXiv:1808.09940*.

[6] Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, *87*, 267–279. https://doi.org/10.1016/j.eswa.2017.06.023

[7] Lee, D., & O, J. (2012). A reinforcement learning approach to portfolio management. *Journal of Applied Mathematics*, *2012*, 1–16. https://doi.org/10.1155/2012/160247

[8] Corazza, M., & Sangalli, I. (2020). Portfolio optimization with reinforcement learning. *Journal of Economic Dynamics and Control*, *113*, 103855. https://doi.org/10.1016/j.jedc.2020.103855

[9] Xiong, Z., Liu, X., Zhong, S., Yang, H., & Walid, A. (2018). Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522*.
[10] Wang, Z., Liu, Y., & Li, Z. (2020). Deep reinforcement learning for dynamic portfolio optimization. *Neurocomputing*, *385*, 159–171. https://doi.org/10.1016/j.neucom.2019.12.002
[11] Bertoluzzo, F., & Corazza, M. (2012). Testing different reinforcement learning configurations for financial trading. *Journal of Economic Dynamics and Control*, *36*(10), 1494–1506. https://doi.org/10.1016/j.jedc.2012.04.007
[12] Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, *234*(2), 356–371. https://doi.org/10.1016/j.ejor.2013.10.060
[13] Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006). Reinforcement learning for optimized trade execution. *Proceedings of the 23rd International Conference on Machine Learning*, 673–680. https://doi.org/10.1145/1143844.1143929
[14] Yu, P., Yan, X., & Zhang, C. (2019). A deep reinforcement learning framework for the financial portfolio management problem. *IEEE Access*, *7*, 129789–129799. https://doi.org/10.1109/ACCESS.2019.2940221
[15] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.