Reinforcement Learning for Finance: A Review

Aprendizaje por refuerzo para finanzas: una revisión

Diego Ismael León Nieto*

* Master in Finance. Research professor, Observatorio de Economía y Operaciones Numéricas (ODEON), Universidad Externado de Colombia. Bogotá (Colombia). [diego.leon@uexternado.edu.co], [ORCID ID: 0000-0003-1434-7569].
Artículo recibido: 26 de abril de 2023
Aceptado: 26 de junio de 2023

Para citar este artículo:
León-Nieto, D. I. (2023). Reinforcement learning for finance: A review. Odeon, 24, pp. 7-24.
DOI: https://doi.org/10.18601/17941113.n24.02

Abstract

This paper provides a comprehensive review of the application of Reinforcement Learning (RL) in the domain of finance, shedding light on the groundbreaking progress achieved andthe challenges that lie ahead. We explore how RL, a subfield of machine learning, has been instrumental in solving complex financial problems by enabling decision-making processes that optimize long-term rewards. Reinforcement learning (RL) is a powerful machinelearning technique that can be used to train agents to make decisions in complex environments. In finance, RL has been used to solve a variety of problems, including optimal execution, portfolio optimization, option pricing and hedging, market making, smart order routing, and robo-advising. In this paper, we review the recent developments in RL for finance. We begin by introducing RL and Markov decision processes (MDPs), which is the mathematical framework for RL. We then discuss the various RL algorithms that have beenused in finance, with a focus on value-based and policy-based methods. We also discuss the use of neural networks in RL for finance. Finally, we discuss the results of recent studies that have used RL to solve financial problems. We conclude by discussing the challenges and opportunities for future research in RL for finance.

Key words: Reinforcement learning; machine learning; Markov decision process; finance.

JEL classification: G10, G12, G13.

Resumen

Este artículo ofrece una revisión exhaustiva de la aplicación del aprendizaje por refuerzo (AR) en el dominio de las finanzas, y arroja una luz sobre el innovador progreso alcanzado y los desafíos que se avecinan. Exploramos cómo el AR, un subcampo del aprendizaje automático, ha sido instrumental para resolver problemas financieros complejos al permitir procesos de toma de decisiones que optimizan las recompensas a largo plazo. El AR es una poderosa técnica de aprendizaje automático que se puede utilizar para entrenar a agentes a fin de tomar decisiones en entornos complejos. En finanzas, el AR se ha utilizado para resolver una variedad de problemas, incluyendo la ejecución óptima, la optimización de carteras, la valoración y cobertura de opciones, la creación de mercados, el enrutamiento inteligente de órdenes y el robo-asesoramiento. En este artículo revisamos los desarrollos recientes en AR para finanzas. Comenzamos proporcionando una introducción al AR y a los procesos de decisión de Markov (MDP), que es el marco matemático para el AR. Luego discutimos los diversos

algoritmos de AR que se han utilizado en finanzas, con un enfoque en métodos basados en valor y políticas. También discutimos el uso de redes neuronales en AR para finanzas. Finalmente, abordamos los resultados de estudios recientes que han utilizado AR para resolver problemas financieros. Concluimos discutiendo los desafíos y las oportunidades para futuras investigaciones en AR para finanzas.

Palabras clave: aprendizaje por refuerzo; aprendizaje automático; procesos de decisión de Markov; finanzas.

Clasificación JEL: G10, G12, G13.

Introduction

Reinforcement Learning (RL) is a powerful machine learning technique that can be used to train agents for better decision making in complex environments. In finance, RL has been used to solve a variety of problems, including optimal order execution, portfolio optimization, option pricing and hedging derivatives, market making, smart order routing, and robo-advising.

RL is based on the idea that agents can learn to make good decisions by trial and error. In anRL problem, the agent is placed in an environment and must learn to take actions that will maximize its reward. The agent's environment can be anything from a simple game to a complex financial market. The agent's actions can be anything from buying or selling a stockto placing an order to trade. The agent's reward can be anything from making a profit to avoiding a loss.

The paper's analytical approach primarily involves a thorough examination of both seminal and recent academic papers, focusing on the application of RL algorithms in various financialsectors such as portfolio management, algorithmic trading, credit scoring, and risk management. Additionally, we highlight the different types of RL algorithms, their strengths,weaknesses, and the contexts in which they are most effective.

RL has been shown to be effective in solving a variety of financial problems. For example, RL has been used to develop trading algorithms that can outperform human traders. RL has also been used to develop portfolio optimization algorithms that can help investors to achievetheir financial goals. Our findings underscore the significant potential of RL in finance, evidencing its ability to outperform traditional methods in several applications. However, wealso identify several challenges and limitations, such as overfitting, instability, and the difficulty of interpreting RL models.

In this paper, we will review the recent developments in RL for finance. In the next section we will begin by introducing RL and Markov decision processes (MDPs), which is the mathematical framework for RL. The second section (ç) discusses the various RL algorithms thathave been used in finance and discusses the use of neural networks in RL for finance. Finally, we discuss the results of recent studies that have used RL to solve financial problems. We conclude by discussing the challenges and opportunities for future research in RL for finance.

1. Reinforcement Learning

The essence of Reinforcement Learning (RL) is to learn through interaction. An RL agent interacts with its environment, and by observing the consequences of its actions, can learn toalter its behavior in response to the rewards received. Therefore, without the presence of an agent and an environment, RL could not materialize. It should be clarified that within these elements, the environment is not understood as something deterministic, as even when the same action has been given in the same state, the results obtained are different.

However, beyond the agent and the environment, four additional elements can be identifiedthat provide inputs to the development of this type of learning. The first of these is the policy,which is understood as the behavior of the agent at a given moment. It is a mapping of theperceived states of the environment to the actions or decisions that must be taken in those specific states. In some cases, the policy may be a simple function, a lookup table, or involveextensive calculation, like a search process that, in the end, is carried out under behaviors performed from stimulus-response associations, attributed mainly to psychology. The actions studied are stationary, i.e., they do not depend on time.

At the beginning of each study, a goal must be established, or in RL, a reward, which is responsible for defining the objective through which the agent's behavior will be conditioned.From here arises the second element, the agent's pursuit of maximizing its benefit and obtaining an ever-greater reward, considering what are favorable or unfavorable events as stochastic functions and seeking as a priority its benefit.

In a biological system, we could think that rewards are analogous to experiences of pleasureor pain, they are the immediate and defining characteristics of the problem the agent faces, (Sutton & Barto, 2018). The correlation that exists between policy and reward is direct: if anaction governed by a policy

leads to a low reward, the agent opts for a change in the policy that generates a different action to finally obtain greater benefits.

Now, the long-term value function thought by the agent as the third element of RL should beconsidered. This means that reward signals are the immediate results of previous actions, while the value function gives the long-term results of the decision made. The value of a state is the total amount of reward that an agent can expect to accumulate in the future from that state, while rewards determine the immediate and intrinsic convenience of environmental states (Sutton & Barto, 2018). The value function becomes very important in later studies because, in the algorithmic application, it is these value functions that define decision-making that results in benefit maximization.

The last element is the environment model, established from the situations proposed to achieve low-risk experimentation. It is understood as a series of inferences about the possiblebehavior of the environment, given a state and an action, the model could predict the next state and its next reward.

In this way, the components that encompass the actors of RL can be synthesized. And additionally, there are a series of criteria related to the balance between exploration and exploitation, the acceleration of the learning process, and generalization, which influence learning.

The action of delegating to the agent the responsibility of determining the strategy to explore the environment, and controlling the training examples through the sequence of actions, provides RL with a defining characteristic. This is where the agent must find the balance between exploring new states to obtain new information and exploit already assimilated and learned actions with which they obtain a great reward, which guarantees an accumulated reward (Kaelbling et al., 1996).

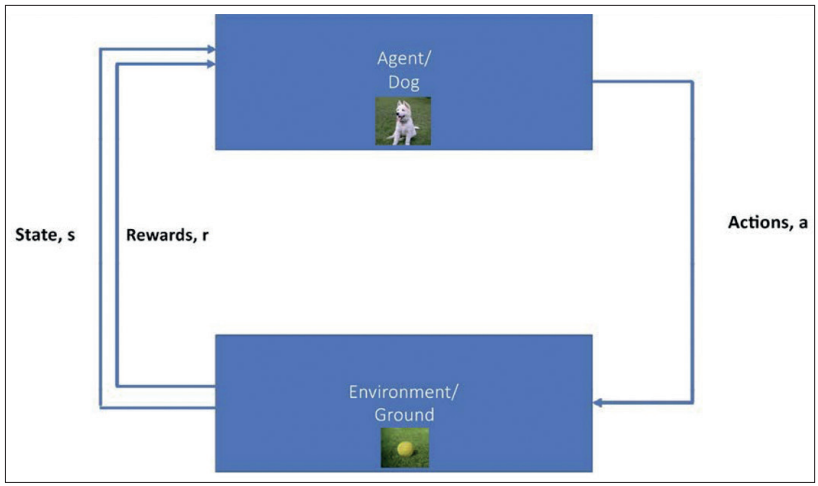
Since it is impossible to explore and exploit simultaneously with a single action selection, "conflicts" are created between exploration and exploitation and there are several proposals to achieve the balance between them (Kaelbling, 1993). There is the E-greedy strategy, optimistic initial values, action selection methods based on the Boltzmann distribution, interval estimation method, exploration bonus used in Dyna, and competition maps (Sutton, 1990, 1991).

Regarding the acceleration of learning, it seeks to attack the agent's need to reiterate actionsto learn the value function. A weakness that is mitigated with the incorporation of informationpredicted by an external observer or integrating learning with planning. And considering generalization, which is based on

the estimation of optimal values defined over the set of states, represented in a tabular manner if they are in small states (Thrun & Möller, 1991).

In that sense, the elements that interact within RL are exposed, as seen in Figure 1, within an environment of actions and rewards for the agent. Likewise, the criteria in which such learning is developed and where knowledge has been used for its employability, being State S_t the initial moment, with the interaction of an agent (Agent) who performs an action (Action a_t) in a certain environment (Environment) and that finally, obtains a reward (Reward r) that must be greater than the initial one (Reward S_{t+1}) (Kapoor et al., 2022).

Figure 1: Reinforcement learning: Interaction between agent and environment



Source: Kapoor et al. (2022).

One of the conclusions the evaluator must reach is that the studied agent must accept success or failure through the previously mentioned processes. However, such learning must be focused on understanding the environment and its behavior through rewards or punishments. Similarly, two cross-sectional stages are recognized for evaluating any model: prediction and control.

On the one hand, according to the associations of stimuli and their derivations, it is possible to propose an evaluation of the future given a policy, without the need to depend on time. On the other hand, control allows for the future to be optimized with the application of accurate conjectures to find the best policy.

Referring to the term of optimal control describes the problem of designing a dynamic controller over time, known in dynamic programming, which,

despite being considered one of the essential systems in solving general problems, suffers from "the curse of dimensionality" due to the exponential growth of computational requirements, and, when introduced in a stochastic version, Markovian decision processes (MDP) are produced (Gosavi, 2009). In this way, all these methods are RL because they require complete knowledge of the system to control or decipher.

Then, the modern field of trial and error learning begins, measuring the level of satisfaction or dissatisfaction that produces a strengthening or weakening in the agent's behavior and decisions (Thorndike, 1911). The above gives rise to the Law of Effect because it describes the effect of reinforcing events on the tendency to select actions and is widely considered a basic principle underlying many behaviors at the base of influential learning.

The trial and error methodology was also implemented in computers that converge at the beginnings of artificial intelligence. Minsky (1954) discussed computational models of reinforcement learning and described his construction of an analog machine composed of components he called SNARC (Stochastic Neural-Analog Reinforcement Calculators) designed to resemble modifiable synaptic connections in the brain. This found a lot of applicability in predictions and expectations compared to what is currently being had, all this affecting real-time decisions.

Likewise, in the functionalities of trial and error, there is the Stella system that learned by interaction with its environment (Andreae, 1963). Menace (Matchbox Educable Naughts and Crosses Engine) was also developed. It was a system to learn to play tic-tac-toe that consisted of a matchbox for each game position. Each box contained several colored beads, a different color for each possible move from that position (Michie & Chambers, 1968).

In the same way, the adoption of reinforcement learning with trial and error was carried out in classic economic models, specifically in game theory (Camerer, 2003), as one of the many uses of this branch.

As a third methodology, RL revolutionized the temporal dimension, driven by the differences between temporally successive estimates of the same quantity that influence the final decision of the agent and leveraged by the notion of secondary reinforcers. That is, temporal dimensions learn from primary events or primary reinforcers such as food or pain and acquire similar properties (Minsky, 1954), to the initial ones that would be the secondary reinforcers and support previous behavior.

In summary, Markov processes are created from decision processes. MDPs are sequential decision-making problems in which a control (action) must be selected at each decision- making state visited by the system in question. These methods integrate three important methods: again, Dynamic Programming, Monte Carlo, and Finite Differences Learning.

Sequential decision-making is part of processes with incentives for agents. In such a way that, the decision for the best policy will be influenced by two important factors: immediate rewards and subsequent rewards. And these decision processes can only present two unique states, either truth or lie. Given decision-making, and the use of methods and incentives that Markov's models presented, lies Dynamic Programming. The main objective is the calculation of optimal policies (Tesauro, 1995).

To achieve optimization within the choice and fulfillment of policies and their consequences, it is important that a division of the evaluated problem is carried out. This situation will create recurrent sub-problems, and if a solution is found, they should start with the next situation evaluated, which later leads to a state of combination between all the disputes (Kohl, 2004).

Recurrent sub-problems establish a compliance pattern. Firstly, the interaction of generalized policies, characterized by the interaction between evaluation and work in favor of policies. Second, the interaction of values, the one that intercalates policy evaluation among minors of policies, and where a scan is necessary. Lastly, the efficiency of dynamic programming interrelates between actions and states (Errecalde et al., 2000).

From the subdivision and a better evaluation of recurrent problems, model-free prediction must be made. Among these is Monte Carlo Learning, with sample episodes. Policy evaluation, first sight methods, and all sights. Finally, temporal differences learning, which contains both immediate experiences and dynamic programming. To understand the way these methods, behave and are fed back, you can see in the image that the RL algorithms in dynamic programming, temporal differences, exhaustive search, and Monte Carlo correlate in the bootstrapping techniques to simple supports, which are presented in the behavior of the Actor (Policy).

Even though Reinforcement Learning (RL) presented successful research and applications, the lack of scalability in its approaches and limitations to low-dimension problems became more and more apparent. For this reason, various studies began to emerge, branching out from RL, and seeking to solve the complexity constraints shared by RL algorithms like any other algorithm.

Among these problems, the complexity of memory, computational complexity, and sample complexity were found, which will be attacked from different approaches (Schlegel et al., 2019).

Considering that modern RL science has emerged from a synthesis of notions from four different fields: classic Dynamic Programming (DP), AI with temporal differences, stochastic approximation (simulation), and function approximation which contemplate regression, the Bellman error, and neural networks; it is necessary to analyze the connection of the mentioned problems and the techniques used in learning, which will be indispensable for future research (Torres et al., 2017).

It is stated again that these evolutions begin with a Markov Decision Process (MDP). The system is driven by underlying Markov chains that randomly jump from one state to another in discrete time steps, and where the probability of transition from the current state to the next depends only on the current state and not on where the system has been before (Taylor & Stone, 2009). Because of this, the system tries to find the policy that optimizes the performance metric and its infinite temporal horizon. MDPs are related to the evolution of Classic Dynamic Programming (CDP). Given that, it is responsible for the breakdown and for fulfilling the requirement to compose, store, and manipulate the transition probability matrices (TPMs) in the policy and value iterations. However, like many MDP processes, it has gaps like the phenomena of the curse of modeling; where it is not possible to calculate the values of the transition probabilities, the curse of dimensionality with storage processes, or manipulation of the value function (Foerster, 2016).

Analogously, RL with Q-Values is based on the dynamic programming of a discrete event. CDP is based on two forms of the Bellman equation: the Bellman Optimization Equation (BOE) and the Bellman Policy Equation (BPE). The Bellman optimality principle can be found applied to corporate financial structure, which seeks to find the optimal point of indebtedness, which in this case is represented by the maximum difference between the fiscal benefit obtained by contracting debt and the respective financial costs incurred (Ziebart et al., 2008).

Referring to the advances that have provided new tools for learning in deep neural networks, improving tasks such as object detection, voice recognition, and language translation (Bengio et al., 2013). The start is given to Deep Reinforcement Learning (DRL). The most important property of DRL is that deep neural networks can automatically find compact low-dimension representations

(features) of high-dimension data (images, text, and audio). By creating inductive biases in neural network architectures, particularly that of hierarchical representations, machine learning professionals have achieved effective progress in addressing the curse of dimensionality (LeCun et al., 2015). This advance allows RL to adapt to decision-making problems in high-dimension action and state environments.

The development of an algorithm for Atari videogames at a superhuman level trained by agents in raw and high-dimension observations, based on a single reward signal, is contributed by Deep Reinforcement Learning (DRL) as one of its first successes. Subsequently, the development of a hybrid system, AlphaGo (Silver 2016), which defeated the human world champion in Go, was presented. AlphaGo was composed of neural networks, trained using supervised and reinforcement learning, and combined with a traditional heuristic search algorithm. Parallel to IBM's Deep Blue's historic achievement in chess two decades earlier and IBM's Watson DeepQA system (Ferrucci, 2010), all these agents have the utility of "meta-learning" or learning to learn, allowing them to generalize complex visual environments they have never seen before (Duan et al., 2016).

Although algorithms can process high-dimension inputs, it is rarely feasible to train RL agents directly on visual inputs in the real world, due to the large number of samples required. To accelerate learning in DRL, it is possible to exploit previously acquired knowledge from related tasks, which come in various forms: transfer learning, multitask learning, and curriculum learning, to name a few (Nath et al., 2020).

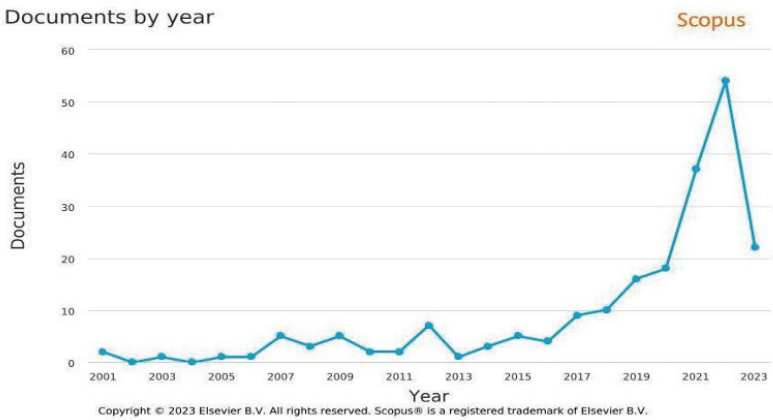
2. Reinforcement Learning in Finance

Reinforcement Learning (RL) has emerged as a significant tool in the field of finance, driving a surge in research and publications. The ability of RL to optimize decisions over time, learn from interaction with the environment, and adapt to changing circumstances makes it particularly suited to the dynamic and complex nature of financial markets. It has found applications in various areas of finance, including portfolio management, credit scoring, and algorithmic trading. The recent increase in papers published on this topic reflects the growing recognition of RL's potential in finance. Researchers are exploring innovative ways to apply RL to solve complex financial problems, improve financial decision-making, and create more efficient and robust financial systems. The

following figures show this relevance achieved in recent years by the academic community.

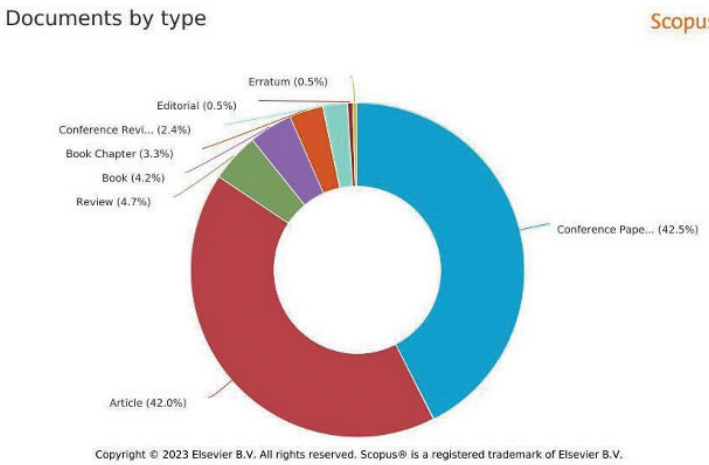
Figure 2 shows a relevant increase in the papers published about RL and finance, with special behavior from the last five years, with almost sixty documents published in 2020.

Figure 2: Documents published by year with the key "Reinforcement Learning and Finance"



Source: Scopus.

Figure 3: Documents published by type with the key "Reinforcement Learning and Finance"



Source: Scopus.

Figure 3 shows that scientific articles and conference are equivalent to 85% of the publications in RL and finance, revealing that is an important topic for cutting edge research in the world.

The rapid changes in the finance industry due to the increasing amount of data have revolutionized the techniques on data processing and data analysis and brought new theoretical and computational challenges. Given that traditional approaches to financial decision-making heavily rely on model assumptions, reinforcement learning (RL) can make full use of the large amount of financial data with fewer model assumptions and improve decisions in complex financial environments. This section aims to review the recent developments and use of RL approaches in finance, with a focus on value and policy-based methods that do not require any model assumptions. It also discusses the potential benefits of using RL approaches in finance, such as improving decision-making, reducing transaction costs, and capturing complex patterns in financial data.

RL approaches can make full use of the large amount of financial data with fewer model assumptions and improve decisions in complex financial environments. RL algorithms can be applied in a variety of decision-making problems in finance, including optimal order execution, portfolio optimization, option pricing and hedging, market making, and risk management. RL algorithms can help in developing trading strategies that can adapt to changing market conditions and improve the overall performance of the portfolio. RL algorithms can also help in reducing transaction costs and market impact costs by optimizing the execution of trades. The use of deep RL algorithms can help in capturing complex patterns in financial data and improve the accuracy of predictions (Hambly et al., 2021).

One of the most exciting implications of RL in finance, is portfolio management. Hu and Lin (2019) discuss the application of Deep Reinforcement Learning (DRL) for optimizing finance portfolio management. The authors address several research issues related to policy optimization for finance portfolio management. They propose the use of a deep recurrent neural network (RNN) model, specifically Gated Recurrent Units (GRUs), to weigh the influences of earlier states and actions on policy optimization in non-Markov decision processes. They also propose a risk-adjusted reward function for searching for an optimal policy.

The authors discuss the integration of Reinforcement Learning (RL) and Deep Learning (DL) to leverage their respective capabilities to discover an optimal policy. They explore different types of RL approaches for integrating

with the DL method while solving the policy optimization problem. They also discuss the challenges of applying DRL in optimizing finance portfolio management. These challenges include the impossibility of obtaining a real state space of the finance world, the need to deal with the non-Markovian property with dependence on earlier states and actions to learn and estimate future expected rewards, and the need to consider transaction overheads such as transaction fees and tax when computing the risk-adjusted reward function to obtain total effective rewards.

Finally, they propose using deep RNNs for DL and policy gradient for RL to search for the optimal policy function's parameters. They also discuss various DL and RL combinations and propose one of the DRL approaches, arguing why this one is better for optimizing finance portfolio management. The paper concludes with the intention to investigate all types of DL and RL combinations, find the best one, and discover its incentives for finance planning in future work.

Millea and Edalat (2022) discuss portfolio optimization, which is the process of selecting a combination of assets that will increase in value over time. The goal is to partition the available resources in a way that the overall portfolio value increases over time. The paper presents a hierarchical decision-making architecture for portfolio optimization on multiple markets, using a combination of Deep Reinforcement Learning (DRL) and Hierarchical Risk Parity (HRP) and Hierarchical Equal Risk Contribution (HERC) models. The experiments were performed on the cryptocurrency market, stock market, and foreign exchange market, showing excellent robustness and performance of the overall system.

Another framework in finance to using Reinforcement Learning (RL), is option pricing and hedging with derivatives. The QLBS Model: The Quantitative Learning from Buffer Stock (QLBS) model, proposed by Halperin (2019) and extended in Halperin (2020), learns both the option price and the hedging strategy in a similar spirit to the mean-variance portfolio optimization framework based in Q-Learning algorithms.

Buehler et al. (2019) used deep neural networks to approximate an optimal hedging strategy under market frictions, including transaction costs, and convex risk measures. They showed that their method can accurately recover the optimal hedging strategy in the Heston model without transaction costs and it can be used to numerically study the impact of proportional transaction costs on option prices.

Cannelli et al. (2020) formulated the optimal hedging problem as a Risk-averse Contextual Multi-Armed Bandit (R-CMAB) model and proposed a

deep CMAB algorithm involving Thompson Sampling. They showed that their algorithm outperforms DQN in terms of sample efficiency and hedging error when compared to delta hedging. Cao et al. (2021) considered Q-learning and Deep Deterministic Policy Gradient (DDPG) for the problem of hedging a short position in a call option when there are transaction costs. The objective function is set to be a weighted sum of the expected hedging cost and the standard deviation of the hedging cost. They showed that their approach achieves a markedly lower expected hedging cost but with a slightly higher standard deviation of the hedging cost when compared to delta hedging.

For American options, the key challenge is to find the optimal exercise strategy, which determines when to exercise the option as this determines the price. Li et al. (2009) used the Least-Squares Policy Iteration (LSPI) algorithm and the Fitted Q-learning algorithm to learn the exercise policy for American options.

Regarding algorithmic trading, Sun and Si (2022) discuss the use of Reinforcement Learning (RL) in automated trading for generating buy and sell signals in financial markets. RL is a method of training an agent to make optimal decisions based on the current state of the market and owned positions and cash. The paper proposes a novel framework called Supervised Actor-Critic Reinforcement Learning with Action Feedback (SACRL-AF) to address the issue of incomplete fulfillment of buy or sell orders in certain situations. The proposed framework uses Deep Deterministic Policy Gradient (DDPG) and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithms to achieve state-of-the-art performance in profitability.

Théate and Ernst (2021) present a new approach to solve the algorithmic trading problem using deep reinforcement learning (DRL). The proposed Trading Deep Q-Network algorithm (TDQN) is inspired by the popular DQN algorithm and is adapted to the specific algorithmic trading problem. The training of the reinforcement learning (RL) agent is based on the generation of artificial trajectories from a limited set of stock market historical data. The paper also proposes a novel performance assessment methodology to objectively assess the performance of trading strategies. Promising results are reported for the TDQN algorithm.

3. Conclusions

RL approaches can provide a powerful tool for decision-making in finance and can help in developing more efficient and effective trading strategies. These

approaches show how RL can be used to learn optimal strategies for option pricing and hedging, often outperforming traditional methods, also in the field of portfolio optimization and algorithmic trading; RL has shown remarkable results compared with traditional methods. However, it's important to note that these methods often require careful tuning and may not always be applicable in every market condition. DRL algorithms performs well on multiple markets, including the cryptocurrency market, the stock market, and the foreign exchange market. The system can learn when to switch between the low-level models, and the performance is better than the individual models. Additionally, possible future works needs to consider transaction costs, which can have a significant impact on the performance of the system in practice.

References

Andreae, J. H. (1963). STELLA: A scheme for a learning machine. IFAC *Proceedings Volumes*, *1*(2), 497-502. https://doi.org/10.1016/S1474-6670(17)69682-4

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine intelligence*, *35*(8), 1798-1828. https://doi.org/10.1109/TPAMI.2013.50

Buehler, H., Gonon, L., Teichmann, J., & Wood, B. (2019). Deep hedging. *Quantitative Finance*, *19*(8), 1271-1291. https://doi.org/10.1080/14697688.2019.1571683

Camerer, C. F. (2003). Behavioural studies of strategic thinking in games. *Trends in Cognitive Sciences*, *7(5),* 225-231. https://doi.org/10.1016/S1364-6613(03)00094-9

Cannelli, L., Nuti, G., Sala, M., & Szehr, O. (2020). Hedging using reinforcement learning: Contextual *k*-armed bandit versus Q-learning. Working paper, arXiv:2007.01623.

Cao, J., Chen, J., Hull, J., & Poulos, Z. (2021). Deep hedging of derivatives using reinforcement learning. *The Journal of Financial Data Science, 3(1)*, 10–27. https://doi.org/10.3905/jfds.2020.1.052

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., & Abbeel, P. (2016). RL^2: Fast reinforcement learning via slow reinforcement learning. Working paper, arXiv:1611.02779.

Errecalde, M. L., Muchut, A., Aguirre, G., & Montoya, C. I. (2000). Aprendizaje por Refuerzo aplicado a la resolución de problemas no triviales. In *II Workshop de Investigadores en Ciencias de la Computación*.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., … & Welty, C. (2010). Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31*(3),* 59-79. https://doi.org/10.1609/aimag.v31i3.2303

Foerster, J., Assael, I. A., De Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information processing systems*, *29*, 1-9.

Gosavi, A. (2009). Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, *21*(2), 178-192. https://doi.org/10.1287/ijoc.1080.0305

Hambly, B., Xu, R., & Yang, H. (2021). Recent advances in reinforcement learning in finance. arXiv preprint arXiv:2112.04553. https://arxiv.org/abs/2112.04553

Halperin, I. (2019). The QLBS Q-learner goes NuQlear: Fitted Q iteration, inverse RL, and option portfolios. *Quantitative Finance, 19(9),* 1543–1553. https://doi.org/10.1080/14697688.2019.1622302

Halperin, I. (2020). QLBS: Q-learner in the Black-Scholes-Merton world. *The Journal of Derivatives, 28*(1), 99-122. https://doi.org/10.3905/jod.2020.1.108

Hu, Y. J., & Lin, S. J. (2019). Deep reinforcement learning for optimizing finance portfolio management. In *2019 Amity International Conference on Artificial Intelligence (AICAI)* (pp. 14-20). IEEE. https://doi.org/10.1109/AICAI.2019.8701368

Kaelbling, L. P. (1993). *Learning in embedded systems*. MIT Press.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237-285. https://doi.org/10.1613/jair.301

Kapoor, A., Gulli, A., Pal, S., & Chollet, F. (2022). *Deep Learning with Tensor Flow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models*. Packt Publishing Ltd.

Kohl, N., & Stone, P. (2004, April). Policy gradient reinforcement learning for fast quadrupedal locomotion. In IEEE International Conference on Robotics and Automation, 2004. https://doi.org/10.1109/ROBOT.2004.1307456

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning*. *Nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

Li, Y., Szepesvari, C., & Schuurmans, D. (2009). Learning exercise policies for American options. In *Artificial intelligence and statistics* (pp. 352–359). PMLR. https://proceedings.mlr.press/v5/li09d.html

Michie, D. & Chambers, R. A. (1968). BOXES: An experiment in adaptive control. In E. Dale & D. Michie (eds.), *Machine Intelligence*. Oliver and Boyd.

Millea, A., & Edalat, A. (2022). Using deep reinforcement learning with hierarchical risk parity for portfolio optimization. *International Journal of Financial Studies, 11*(1), 10. https://doi.org/10.3390/ijfs11010010

Minsky, M. L. (1954). *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Princeton University.

Nath, S., Liu, V., Chan, A., Li, X., White, A., & White, M. (2020). Training recurrent neural networks online by learning explicit state variables. In *International conference on learning representations*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489. https://doi.org/10.1038/nature16961

Schlegel, M., Chung, W., Graves, D., Qian, J., & White, M. (2019). Importance resampling for off-policy prediction. *Advances in Neural Information Processing Systems*, 32.

Sun, Q., & Si, Y. W. (2022). Supervised actor-critic reinforcement learning with action feedback for algorithmic trading. *Applied Intelligence*, 53, 16875-16892. https://doi.org/10.1007/s10489-022-04322-5

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990* (pp. 216-224). https://doi.org/10.1016/B978-1-55860-141-3.50030-4

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin, 2*(4), 160-163. https://doi.org/10.1145/122344.122377

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An introduction*. MIT Press.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, *38*(3), 58-68. https://doi.org/10.1145/203330.203343

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications,* 173, 114632. https://doi.org/10.1016/j.eswa.2021.114632

Thrun, S. B., & Möller, K. (1991). Active exploration in dynamic environments. *Advances in neural information processing systems*, 4. https://proceedings.neurips.cc/paper/1991/hash/e5f6ad6ce374177eef023bf5d0c018b 6-Abstract.html

Taylor, M. E., & Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, *10*(7), 1635-1685. https://doi.org/10.5555/1577069.1755839

Thorndike, E. L. (1911). *Animal intelligence: Experimental studies*. Transaction Publishers.

Torres Cortés, L. J., Velázquez Vadillo, F., & Turner Barragán, E. H. (2017). El principio de optimalidad de Bellman aplicado a la estructura financiera corporativa. Caso Mexicano. *Análisis Económico*, *32*(81), 151-181.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence 2008*.