Systematic Literature Review of Testing Tools and Techniques for Reinforcement Learning Agents

Groza Iulia-Diana[a], Havirneanu Albert-Andrei[a] and Ilie Oana-Andreea[a]

[a]Babes-Bolyai University, Faculty of Mathematics and Computer Science, Str. Mihail Kogalniceanu Nr. 1, Cluj-Napoca, 400084, Cluj, Romania

ARTICLE INFO

Keywords:
reinforcement learning
automated RL testing
hyperparameter optimization

ABSTRACT

Reinforcement learning (RL) has applications across various fields, necessitating a systematic review of its methodologies and tools. This paper synthesizes the latest advancements and identifies critical gaps in developing and testing RL systems. We conducted a comprehensive search on Google Scholar for articles published since 2013, filtering the initial pool to select ten key studies. Our review focuses on the approaches, methods, datasets, and results reported in these studies, underscoring the urgent need for improved testing frameworks to ensure the robustness and reliability of RL applications.

1. Study design

1.1. Review need identification

Reinforcement learning (RL) is applied in various fields, such as financial trading, autonomous vehicles, and multi-agent systems. Despite its widespread use, the development and validation of RL applications remain largely exploratory, leading to challenges in ensuring reliability and robustness. For example, RL algorithms in financial trading optimize strategies by interacting with market environments without predictive models of future prices. RL enhances decision-making under uncertainty in autonomous systems, essential for self-driving cars and robotic control. However, these environments' complex nature makes systematic testing and validation essential. Therefore, there is a critical need for robust testing methodologies and quality assurance techniques to improve the performance and trustworthiness of RL systems before deployment

1.2. Research questions definition

In this review, we aim to address the following research questions to understand better the methods and tools used in reinforcement learning:

- How do the authors approach the reinforcement learning process in various applications?
- What methods and algorithms are employed to solve specific problems in RL?
- What datasets or environments are utilized to test and validate the proposed RL solutions?
- What are the performance outcomes and results obtained from these RL methods?
- How do the proposed tools and frameworks enhance the efficiency and effectiveness of RL systems?
- What challenges and limitations are identified in the current RL approaches?
- What future research directions are suggested to overcome these challenges?

1.3. Protocol definition

The protocol for conducting this systematic literature review on reinforcement learning involved several steps. First, we searched for relevant articles using Google Scholar, employing keywords such as "reinforcement learning," "deep reinforcement learning," "RL frameworks," and "RL tools." We initially identified 30 articles that fell within the broad topic of reinforcement learning.

Next, we filtered these articles by reviewing each abstract to identify those that focused on enhancing the efficiency and effectiveness of RL methods. We looked for terms such as "automated," "framework," "methodology," and "tools." Each article was then ranked on a scale from 1 to 5 based on its relevance to our subject matter, with 1 being irrelevant and 5 being completely on topic. A study was considered relevant if it addressed the development, testing, or improvement of RL methods and tools.

After ranking the articles, we selected the top ten studies for further investigation. We read these articles in detail and extracted key information, focusing on the approach, methods used, datasets or environments utilized, and the results obtained. This information was systematically recorded in a table to facilitate comparison and synthesis.

Finally, we analyzed the extracted data to answer our predefined research questions, ensuring a comprehensive understanding of the current state of reinforcement learning methods and tools.

2. Conducting the SLR

2.1. Search and selection process

2.1.1. Database search

We used Google Scholar as our primary search engine to collect a broad range of articles on reinforcement learning. The search terms included "reinforcement learning," "deep reinforcement learning," "RL frameworks," "RL tools," and

iulia.groza@stud.ubbcluj.ro (G. Iulia-Diana);
oana.andreea.ilie@stud.ubbcluj.ro (H. Albert-Andrei);
albert.havirneanu@stud.ubbcluj.ro (I. Oana-Andreea)
ORCID(s):

"RL methodologies." We focused on articles published after 2012 to ensure the inclusion of up-to-date methods and practices.

2.1.2. Merging, and duplicates and impurity removal

After selecting the initial 30 articles, we reviewed each abstract to identify and remove any duplicates and irrelevant studies. We determined the relevance of the articles based on their focus on reinforcement learning, specifically looking for those that discussed the development, testing, or improvement of RL methods and tools. Keywords such as "automated," "framework," "methodology," and "tools" were essential in this evaluation. Once the abstracts were reviewed, we filtered out the less relevant studies and retained the top ten articles that most closely aligned with our research objectives. This careful selection process ensured that our review included only the most pertinent and high-quality studies.

2.2. Data extraction

In this section, we highlight the important aspects of each of the ten articles from the previously described process. Table 1 contains publication information about each of the selected pieces of literature.

2.3. Data synthesis

2.3.1. "A Search-Based Testing Approach for Deep Reinforcement Learning Agents" by Zolfagharian, Amirhossein and Abdellatif, Manel and Briand, Lionel C. and Bagherzadeh, Mojtaba and S, Ramesh

In their seminal work, Zolfagharian et al. present STARLA, a Search-based Testing Approach for Reinforcement Learning Agents, which utilizes a combination of genetic algorithms and machine learning models to detect faults in Deep Reinforcement Learning (DRL) agents. The method involves generating and analyzing episodes of DRL behavior to identify and predict faults, thereby optimizing the search process within a constrained testing budget. By systematically exploring the state and action spaces, STARLA aims to uncover rare and critical faults that random testing might miss. STARLA was applied to two benchmark datasets from OpenAI Gym, specifically the Cart-Pole and Mountain Car problems, to validate its effectiveness. These benchmarks are well-known for their ability to represent fundamental challenges in reinforcement learning, making them ideal test beds for evaluating DRL agents. The results demonstrate that STARLA significantly outperforms random testing in identifying faulty episodes, showcasing its potential for enhancing the safety and reliability of DRL agents in simulated environments. This research highlights the importance of structured and intelligent testing methodologies in the development and deployment of DRL systems, suggesting that such approaches can lead to more resilient and trustworthy AI applications.

2.3.2. "Testing of Deep Reinforcement Learning Agents with Surrogate Models" by Biagiola, Matteo and Tonella, Paolo

The paper "Testing of Deep Reinforcement Learning Agents with Surrogate Models" by Matteo Biagiola and Paolo Tonella introduces Indago, a tool that employs a search-based testing approach utilizing surrogate models for evaluating Deep Reinforcement Learning (DRL) agents. Indago leverages genetic algorithms and hill climbing techniques, combined with classifiers and regressors trained on interaction data, to predict and maximize failure probabilities across various test environments. This method allows for the efficient exploration of potential failure scenarios by approximating the performance landscape of DRL agents, thereby identifying vulnerabilities that might be missed by conventional testing methods. Indago was rigorously tested on industry-relevant case studies, including autonomous vehicles and humanoid robots, showcasing its ability to detect a significantly broader and more diverse range of failure configurations compared to traditional methods. The extensive testing demonstrated that Indago could not only identify faults more effectively but also do so with greater efficiency, thus enhancing the power and reliability of DRL applications in complex, real-world scenarios. These findings underscore the critical role of advanced testing frameworks like Indago in the deployment of safe and dependable DRL systems, emphasizing the need for thorough validation techniques in the development of autonomous technologies.

2.3.3. "Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications" by Francesco Bertoluzzo and Marco Corazza

The article explores the use of Reinforcement Learning (RL) algorithms in developing automated Financial Trading Systems (FTSs). The study employs Temporal Difference methods, Q-Learning, and Kernel-based Reinforcement Learning to create trading strategies that learn and adapt by interacting directly with market environments. These RL-based FTSs were tested on both artificial and real financial data series to assess their effectiveness. The artificial data consisted of a simulated price series, while the real data comprised daily closing prices of a financial asset over an extended period. The study focused on evaluating the performance using the Sharpe ratio, calculated over both weekly and monthly intervals. Results demonstrated that the RL-based approaches, particularly Q-Learning and Kernel-based Reinforcement Learning, could generate profitable trading strategies and adapt effectively to market changes, outperforming traditional methods. The research underscores the potential of RL algorithms to enhance the efficiency and profitability of automated trading systems in the financial industry.

Table 1
Articles selected

Id	Citation	Title, Authors	Year Published
P1	Zolfagharian, Abdellatif, Briand, Bagherzadeh and S (2023)	A Search-Based Testing Approach for Deep Reinforcement Learning Agents Zolfagharian, Amirhossein and Abdellatif, Manel and Briand, Lionel C. and Bagherzadeh, Mojtaba and S, Ramesh	2023
P2	Biagiola and Tonella (2024)	Testing of Deep Reinforcement Learning Agents with Surrogate Models Biagiola, Matteo and Tonella, Paolo	2024
P3	Bertoluzzo and Corazza (2012)	Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications Francesco Bertoluzzo and Marco Corazza	2012
P4	Eimer, Lindauer and Raileanu (2023)	Hyperparameters in Reinforcement Learning and How To Tune Them Eimer, Theresa and Lindauer, Marius and Raileanu, Roberta	2023
P5	Foley, Tosch, Clary and Jensen (2019)	TOYBOX: Better Atari Environments for Testing Reinforcement Learning Agents Foley, John and Tosch, Emma and Clary, Kaleigh and Jensen, David	2019
P6	Tappler, Córdoba, Aichernig and Könighofer (2022)	Search-Based Testing of Reinforcement Learning Martin Tappler and Filip Cano Córdoba and Bernhard K. Aichernig and Bettina Könighofer	2022
P7	Biagiola and Tonella (2022)	Testing the Plasticity of Reinforcement Learning Based Systems Matteo Biagiola and Paolo Tonella	2022
P8	Varshosaz, Ghaffari, Johnsen and Wąsowski (2023)	Formal Specification and Testing for Reinforcement Learning Mahsa Varshosaz and Mohsen Ghaffari and Einar Broch Johnsen and Andrzej Wąsowski	2023
P9	Guo, Chen, Hao, Yin, Yu and Li (2022)	Towards Comprehensive Testing on the Robustness of Cooperative Multi-agent Reinforcement Learning Jun Guo and Yonghong Chen and Yihang Hao and Zixin Yin and Yin Yu and Simin Li	2022
P10	Noel (2022)	Reinforcement Learning Agents in Colonel Blotto Joseph Christian G. Noel	2022

2.3.4. *"Hyperparameters in Reinforcement Learning and How To Tune Them" by Eimer, Theresa and Lindauer, Marius and Raileanu, Roberta*

The paper "Hyperparameters in Reinforcement Learning and How To Tune Them" studies the critical influence of hyperparameter settings on the performance and sample efficiency of Deep Reinforcement Learning (RL) algorithms. Recognizing the variability and sensitivity of RL algorithms to hyperparameter choices, the study highlights the necessity of systematic Hyperparameter Optimization (HPO) methods, drawing best practices from the field of Automated Machine Learning (AutoML). The authors advocate for principles such as separating tuning and testing seeds to avoid overfitting and employing comprehensive HPO across broad search spaces. They compare several state-of-the-art HPO tools, including DEHB, PB2, and Optuna, with traditional hand-tuning and grid search methods, demonstrating that HPO approaches often achieve superior performance and stability with significantly reduced computational overhead. The paper underscores the need for the RL community to adopt these advanced HPO practices to improve reproducibility and facilitate fairer comparisons across different studies. To support this adoption, the authors provide open-source implementations of the HPO tools they evaluated, aiming to make these techniques more accessible to RL researchers and practitioners.

2.3.5. *"TOYBOX: Better Atari Environments for Testing Reinforcement Learning Agents" by Foley, John and Tosch, Emma and Clary, Kaleigh and Jensen, David*

The paper "TOYBOX: Better Atari Environments for Testing Reinforcement Learning Agents" discusses the TOYBOX platform, an innovative approach aimed at enhancing the testing capabilities for reinforcement learning (RL) agents through a reimplementation of Atari games that are

Table 2
Approach Summary Table

Study	Approach	Tools	Case study
P1	Search-Based Testing (STARLA)	Genetic algorithm, Machine Learning models	Academic (OpenAI Gym environments as benchmarks)
P2	Search-Based Testing with Surrogate Models (Indago)	Genetic Algorithm, Hill Climbing, Surrogate Models (Classifiers and Regressors), Saliency-Based Input Attribution	Industry (Autonomous vehicles, humanoid robots, self-driving car simulation)
P3	Reinforcement Learning for Financial Trading Systems	Temporal Difference, Q-Learning, Kernel-based Reinforcement Learning	Industry
P4	Hyperparameter Optimization (HPO) for RL	DEHB, PB2, Optuna, Random Search, StableBaselines3, Hydra	Academic
P5	Parameterizable Atari Game Reimplementation	TOYBOX	Academic
P6	Search-Based Safety and Performance Testing	Backtracking-based Depth-First Search Fuzz testing	Industry
P7	Quantification of Adaptation and Anti-Regression Capabilities	AlphaTest	Academic
P8	Formal Specification and Testing for Reinforcement Learning	TQuickcheck, Stryker4s for Scala, SARSA and Q-learning	Academic
P9	MARLSafe	SMAC	Academic
P10	Q-Learning	OpenSpiel	Industry

fully parameterizable and observable. This setup allows for precise manipulations and observations of game states, overcoming the limitations of the Arcade Learning Environment (ALE), which is generally non-transparent and fixed. TOYBOX enables a range of new analyses, such as dynamic behaviour analysis during training, acceptance testing, and test set generation by manipulating game parameters. The specific game of Breakout is used for testing, where the authors apply the Proximal Policy Optimization (PPO2) algorithm to train RL agents, evaluating them based on three behavioural requirements: brick elimination, start angle invariance and tunnel exploitation. The results show that while the agents performed well in responding to different ball start angles, they struggled with consistently removing bricks and exploiting tunnels to achieve higher scores, thus highlighting areas where the RL agents' performance could be improved.

2.3.6. *"Search-Based Testing of Reinforcement Learning" by Martin Tappler and Filip Cano Córdoba and Bernhard K. Aichernig and Bettina Könighofer*

The paper presents a search-based testing framework for evaluating the safety and performance of deep reinforcement learning (RL) agents, specifically applied to Nintendo's Super Mario Bros. game. The approach employs a four-step process:

1. using depth-first search (DFS) to identify reference traces and boundary states, which highlight safety-critical situations;
2. generating safety test-suites to evaluate the agent's ability to navigate near boundary states;
3. creating fuzz traces via fuzz testing to generate diverse state conditions for robust performance evaluation;
4. comparing the agent's performance across these generated states with its performance on fuzz traces.

(4) comparing the agent's performance across these generated states with its performance on fuzz traces. The results demonstrate that the framework effectively identifies safety violations and evaluates performance across varied states, providing a comprehensive testing method for RL agents in complex, stochastic environments.

2.3.7. *"Testing the Plasticity of Reinforcement Learning Based Systems" by Biagiola, Matteo and Tonella, Paolo*

The paper "Testing the Plasticity of Reinforcement Learning Based Systems" proposes a novel approach to assess and quantify the adaptability and regression characteristics of deep reinforcement learning (DRL) agents in varied environments. This method specifically evaluates how these agents handle changes in environment parameters, focusing on their ability to maintain performance or adapt without significant degradation. The approach involves creating a parameterized version of the environment, adjusting these parameters, and then monitoring how the agent's performance varies. This variability is mapped and analyzed through what the authors call an "adaptation frontier," which visually depicts successful and failed adaptations as changes are made.

The results demonstrate the tool's ability to effectively measure and visualize the extent of an agent's adaptability (or plasticity) and regression in a controlled, systematic manner. This is done by generating a heatmap that shows the adaptation and anti-regression capacity of the agent across different environmental configurations.

2.3.8. "Formal Specification and Testing for Reinforcement Learning" by Mahsa Varshosaz and Mohsen Ghaffari and Einar Broch Johnsen and Andrzej Wąsowski

The paper focuses on enhancing the systematic development of reinforcement learning (RL) applications through formal specification and property-based testing, specifically targeting temporal difference (TD) methods. The authors present a formal specification framework that uses a domain-specific language for defining backup diagrams and update rules in RL problems, which can then be tested using property-based approaches. A significant portion of the paper is dedicated to demonstrating this approach by applying it to a variety of RL algorithms, including SARSA and Expected SARSA, using a test harness that employs a mutation testing method. The dataset used consists of a set of case studies that highlight typical use cases in RL, which serve as test subjects for the mutation testing framework. The results are impressive, showing a high mutation score above 90% in 75% of the cases, indicating the efficacy of the testing approach in detecting and handling faults in RL algorithms.

2.3.9. "Towards Comprehensive Testing on the Robustness of Cooperative Multi-agent Reinforcement Learning" by Jun Guo and Yonghong Chen and Yihang Hao and Zixin Yin and Yin Yu and Simin Li

This paper introduces MARLSafe, a framework designed to test the robustness of cooperative multi-agent reinforcement learning (c-MARL) algorithms across multiple aspects: state, action, and reward robustness. The approach leverages adversarial attacks to identify vulnerabilities in c-MARL algorithms within the SMAC (StarCraftII Multi-Agent Challenge) environment. The paper demonstrates that most state-of-the-art c-MARL algorithms exhibit low robustness, making the need to enhance their security evident. Key results from the experimental evaluation show that the robustness of c-MARL algorithms against adversarial perturbations is significantly low, often leading to near 0% winning rates when subjected to robustness tests.

2.3.10. Reinforcement Learning Agents in Colonel Blotto by Joseph Christian G. Noel

The paper explores the application of reinforcement learning (RL) agents within the Colonel Blotto game, using a Q-learning approach to train agents. The study measures the performance of these agents against randomly acting opponents and observes how strategies adapt with the increase in opponents. The dataset includes numerous simulated games where the RL agent's performance significantly outstrips that of the random opponents, particularly as the number of opponents increases. The results show that the RL agent can effectively learn and adapt strategies that maximize its chances of winning across different scenarios.

3. Results

After parsing through all of these studies, we gained a better knowledge of the scientific community's different approaches to testing reinforcement learning (RL) agents. We can observe the dominant approach is reinforcement learning, generally paired with a possible improvement. In Table 2, we have stated for each study its approach, tools used, and type of case study. Overall, this systematic literature review was helpful in seeing how automating testing for RL agents could be implemented.

References

Bertoluzzo, F., Corazza, M., 2012. Testing different reinforcement learning configurations for financial trading: Introduction and applications. Procedia Economics and Finance 3, 68–77. URL: https://www.sciencedirect.com/science/article/pii/S2212567112001220, doi:https://doi.org/10.1016/S2212-5671(12)00122-0. international Conference Emerging Markets Queries in Finance and Business, Petru Maior University of Tîrgu-Mures, ROMANIA, October 24th - 27th, 2012.

Biagiola, M., Tonella, P., 2022. Testing the plasticity of reinforcement learning based systems. ACM Transactions on Software Engineering and Methodology 1, 1:1–1:53. URL: https://doi.org/10.1145/3511701, doi:10.1145/3511701.

Biagiola, M., Tonella, P., 2024. Testing of deep reinforcement learning agents with surrogate models. ACM Trans. Softw. Eng. Methodol. 33. URL: https://doi.org/10.1145/3631970, doi:10.1145/3631970.

Eimer, T., Lindauer, M., Raileanu, R., 2023. Hyperparameters in reinforcement learning and how to tune them, in: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.), Proceedings of the 40th International Conference on Machine Learning, PMLR. pp. 9104–9149. URL: https://proceedings.mlr.press/v202/eimer23a.html.

Foley, J., Tosch, E., Clary, K., Jensen, D., 2019. Toybox: Better atari environments for testing reinforcement learning agents. arXiv preprint arXiv:1812.02850 .

Guo, J., Chen, Y., Hao, Y., Yin, Z., Yu, Y., Li, S., 2022. Towards comprehensive testing on the robustness of cooperative multi-agent reinforcement learning. arXiv:2204.07932.

Noel, J.C.G., 2022. Reinforcement learning agents in colonel blotto. arXiv:2204.02785.

Tappler, M., Córdoba, F.C., Aichernig, B.K., Könighofer, B., 2022. Search-based testing of reinforcement learning. arXiv:2205.04887.

Varshosaz, M., Ghaffari, M., Johnsen, E.B., Wąsowski, A., 2023. Formal specification and testing for reinforcement learning. Proceedings of the ACM on Programming Languages 7, 1–34. doi:10.1145/3607835. https://doi.org/10.1145/3607835.

Zolfagharian, A., Abdellatif, M., Briand, L.C., Bagherzadeh, M., S, R., 2023. A search-based testing approach for deep reinforcement learning agents. IEEE Transactions on Software Engineering 49, 3715–3735. doi:10.1109/TSE.2023.3269804.