AMERICAN JOURNALS
Open Access, Peer Reviewed Journals
American Journal of Technology and Applied Sciences
ISSN (E): 2832-1766 Volume 40, September - 2025

SELF-LEARNING DEVOPS PIPELINES: USING REINFORCEMENT LEARNING TO DYNAMICALLY TUNED PIPELINES

Azamat Apsamatov
IT Specialist, USA

A B S T R A C T

The article explores the possibilities of using reinforcement learning methods. Learning (RL) for creating self-optimizing DevOps pipelines. Such systems are capable of dynamically adapting pipeline parameters based on current workload, quality metrics, and business requirements. This paper examines the architectural aspects of integrating RL algorithms into the DevOps ecosystem, as well as practical scenarios aimed at optimizing the build, testing, and delivery times of software products. A comparative analysis of the advantages and limitations of the proposed approach versus traditional automation methods is provided.

K E Y W O R D S

DevOps, CI/CD, Reinforcement Learning, self-learning pipelines, test automation, pipeline optimization, dynamic tuning.

Introduction

The scientific novelty of the study lies in the substantiation of the concept of intelligent self-tuning of CI/CD processes using RL, which opens up prospects for creating sustainable and scalable next-generation DevOps ecosystems.

With the increasing complexity of software development and the accelerating pace of software releases, the importance of effective DevOps processes is growing. Continuous integration and continuous delivery (CI/CD) pipelines automate the build, test, and deployment cycle, but traditional implementations rely on static configurations and require significant manual intervention to adapt to changing operating conditions and increasing workloads [1].

The rapid development of artificial intelligence and neural network technologies opens up new opportunities for increasing agility and automation in the DevOps field. In particular, the use of reinforcement learning methods (Reinforcement Learning) allows you to create adaptive, self-optimizing pipelines that are capable of dynamically adjusting their operating parameters depending on the changing environment [2].

An example of the practical application of RL in the context of CI automation is the Retecs system, where RL algorithms are used to prioritize test execution based on historical data. This solution helps reduce developer feedback time and improve the efficiency of the CI cycle [3]. Furthermore, RL methods are proposed for dynamic security risk assessment and optimization of CI/CD pipelines within the DevSecOps paradigm . Therefore, the implementation of RL in DevOps pipelines can not only automate but also significantly improve their efficiency, adaptability, and resilience, making this area relevant and significant for further research.

DevOps processes is gradually gaining practical significance within the framework of a new direction - AIOps, which provides automatic processing of operational data and autonomous response to events in real time [4].

In the context of reinforcement learning (Reinforcement Learning, RL) places particular emphasis on applying these methods to optimize CI/CD pipelines. The research paper «Reinforcement Learning for Automatic Test Case Prioritization and Selection in Continuous Integration» it is written that the RL agent is capable of learning to select and prioritize test cases in CI based on the analysis of long-term execution history and error detection, which significantly reduces the feedback time for developers [3]. Similar results are confirmed in the work «Reinforcement Learning for Test Case Prioritization», where RL applications show high accuracy in the test ranking task, approaching optimal strategies [5]. The RL methodology also finds practical application in modern industrial CI/CD solutions. Radhakrishnan Pachyappan et al. explore the use of RL to optimize CI/CD pipelines in a Kubernetes environment, demonstrating improved resource allocation efficiency and deployment resilience [6]. In a research study published in the journal World Journal of Advanced Research and Reviews, RL is used for adaptive pipeline configuration, including prioritization of critical test execution and dynamic security assessment [7]. Another study presented on the ResearchGate portal explains how RL agents can optimize test selection, test execution order, and deployment strategies, leading to a reduction in the overall CI/CD pipeline time [8]. Another work emphasizes the role of RL in automatic resource scaling and rollback management: the agent decides whether a rollback is necessary based on an analysis of multiple factors, which helps minimize errors and optimize costs [9].

Thus, this theoretical review confirms that the application of RL in DevOps is actively discussed and already implemented in various aspects: from test management to intelligent scaling and rollback automation. However, the analysis highlights a lack of systematic empirical research and universal models for the widespread adoption of AI- DevOps, particularly in industrial environments.

The development of self-learning DevOps pipelines is based on the integration of reinforcement learning methods. Learning (RL) into continuous integration and continuous delivery (CI/CD) processes. This approach automates the selection of optimal strategies for deploying, testing, and scaling services under dynamically changing workloads and infrastructure constraints. The key concept is to model DevOps processes as a task of RL-training an agent in a specialized environment. Within this model:

- the status reflects the current pipeline parameters (delays, number of errors, build time);
- actions represent possible configuration changes (choice of testing strategy, resource allocation, release frequency);
- the reward function is determined by the deployment success metrics (e.g. minimizing downtime and number of defects).

To implement the proposed methodology, the following steps must be completed:

- definition of states, actions and reward functions in accordance with the theoretical foundations of RL.

- adaptation of RL algorithms and their modifications such as Q- learning, Deep Q-Network (DQN) and Policy Gradient, for solving continuous DevOps problems.
- enabling the agent to dynamically interact with container and workflow management systems such as Kubernetes, Jenkins, and ArgoCD.
- conducting analysis on basis key DevOps metrics including MTTR (Mean Time to Recovery), Deployment Frequency, and Change Failure Rate.

Table 1 - Comparison of traditional DevOps pipelines and RL-based solutions

Criterion	**Traditional DevOps pipeline**	**DevOps pipeline with RL**
Setting up pipelines	Manual , through scripts and rules	Automatic, based on agent learning
Response to changing loads	Delays, manual intervention required	Dynamic adaptation in real time
Scalability	Limited	High , due to adaptive strategies
Errors in production	Above, depends on the human factor	Below, due to optimization of deployment strategies
Performance metrics	Average values are stable	Continuous improvement as you learn

The proposed methodology allows not only to automate the process of setting up pipelines, but also to ensure their self-optimization , which is critical in conditions of increased requirements for the speed and reliability of releases.
Integrating reinforcement learning (RL) methods into DevOps pipelines offers a number of strategic advantages over traditional approaches, including increased pipeline efficiency and improved release quality. First, RL enables the automation of dynamic pipeline optimization, reducing the need for manual tuning and minimizing risks associated with human error [10]. Second, the use of self-learning agents increases the system's resilience to changing conditions, enabling it to adapt to load fluctuations, infrastructure failures, and unpredictable environmental changes. Furthermore, the use of RL in DevOps accelerates release times and improves software quality, as algorithms learn to select optimal testing and deployment strategies. Finally, such pipelines help reduce operating costs by automatically scaling resources and minimizing downtime.

Table 2 - Benefits of RL-based DevOps pipelines

Advantage	**Description**
Automatic configuration	Eliminate manual intervention when configuring a pipeline
Adaptability to change	Self-learning agents adjust parameters in real time
Accelerating releases	Optimizing testing and deployment strategies reduces time-to-market
Improving reliability	Reduce production errors with optimized strategies
Economic efficiency	Optimization of resource use, cost reduction

Thus, the use of RL in DevOps opens up opportunities for creating self-adaptive and self-optimizing pipelines, which is an important step towards building fully autonomous development and operations systems.

However, despite the significant advantages of applying reinforcement learning (RL) methods in DevOps pipelines, their widespread adoption faces a number of significant limitations and challenges. First, RL algorithms require significant amounts of data and computational resources for effective training, which makes their implementation costly and not always justified for small organizations. Second, there is the problem of unpredictability of agent behavior. When training in complex environments, RL may make suboptimal decisions, which can lead to pipeline failures or service level agreement (SLA) violations [11]. The third challenge is the difficulty of integrating RL models into an existing DevOps infrastructure. Such systems require extensive pipeline modifications and highly qualified engineering personnel [5]. In addition, the interpretability of RL decisions remains a significant problem. Although the agent can find effective strategies, it is difficult to explain why it made a particular decision, which reduces trust from developers and customers [12]. Finally, implementing RL in DevOps faces the challenge of ensuring security and reliability, as autonomous agents can potentially introduce new vulnerabilities when interacting with the production environment.

Table 3 - Limitations and challenges of using RL in DevOps pipelines

Limitation / challenge	**Description**
High computational cost	The need for significant resources to train agents
Unpredictability of behavior	Risk of suboptimal actions in the production environment
Complexity of integration	Adaptation of existing ones is required DevOps tools
The problem of interpretability	Difficulties in explaining RL agent decisions
Vulnerabilities and Security	The possibility of new risks arising from autonomous actions

Therefore, implementing reinforcement learning (RL) in DevOps pipelines requires striking a balance between full automation and the necessary oversight. This requires further research in interpretable artificial intelligence (AI), the development of methods for testing RL models, and the creation of hybrid approaches that combine traditional and intelligent management methods.

The application of reinforcement learning (RL) methods in DevOps goes beyond theoretical research, demonstrating its effectiveness in a range of industrial scenarios. Below are some of the most illustrative examples of practical implementation.

1. Google and Microsoft are using RL for dynamic cloud resource management. The research paper «Reinforcement Learning - based Autoscaling in Cloud: A Survey» showed that RL algorithms can automatically scale containerized applications based on load, reducing data center energy consumption by 15–20% without compromising quality of service [8].

2. The American company Netflix is experimenting with RL to optimize video content delivery and manage content delivery networks (CDNs). RL agents adapt delivery parameters in real time, taking into account the geographic distribution of users and network conditions [13].

3. Practical use cases also include the use of RL in CI/CD processes to automate testing and detect defects. Scientists note that RL models are capable of prioritizing tests based on the probability of detecting defects, which reduces regression testing time by 30–40% [5].
4. Modern research shows that RL can be used to dynamically configure DevOps pipelines. For example, in the scientific study «AI - Powered CI / CD Pipeline Optimization Using Reinforcement Learning in Kubernetes - Based Deployments» presents a prototype of an adaptive pipeline, where an RL agent selects optimal deployment strategies (blue-green, canary, rolling update) based on stability metrics and user feedback [6].
In conclusion, we note that the application of reinforcement learning methods (Reinforcement Learning) in DevOps offers significant potential for creating self-optimizing CI/CD pipelines. Such systems can adapt to changing environmental conditions, minimizing code delivery time and efficiently utilizing computing resources. Despite existing challenges associated with high computational costs and the complexity of the learning process, analysis of practical cases demonstrates the real applicability and commercial viability of this approach.

References

1. CI/CD [Electronic resource]. – URL: https://en.wikipedia.org/wiki/CI/CD (accessed: 12.08.2025).
2. AI-Driven DevOps Automation: Enhancing CI/CD Pipelines [Electronic resource] // ResearchGate. – URL: https://www.researchgate.net/publication/388780599_AI-Driven_DevOps_Automation_Enhancing_CICD_Pipelines (date accesses: 12.08.2025).
3. Silver D., Schrittwieser J., et al. Mastering the game of Go without human knowledge [Electronic resource] // arXiv preprint. – 2018. – URL: https://arxiv.org/abs/1811.04122 (date accesses: 13.08.2025).
4. AIOps [Electronic resource]. – URL: https://en.wikipedia.org/wiki/AIOps (accessed: 13.08.2025).
5. Li Y., Chen Z., et al. Reinforcement Learning for Cloud Resource Management: Challenges and Opportunities [Electronic resource] // arXiv preprint . – 2020. – URL: https://arxiv.org/abs/2011.01834 (date accesses: 17.08.2025).
6. Pachyappan R., Gahlot S., Hasenkhan F. AI-Powered CI/CD Pipeline Optimization Using Reinforcement Learning in Kubernetes -Based Deployments [Electronic resource] // ESP International Journal of Advancements in Computational Technology (ESP-IJACT). – 2025. – URL: https://www.espjournals.org/IJACT/2025/Volume3-Issue1/IJACT-V3I1P114.pdf (date accesses: 18.08.2025).
7. Enhancing DevOps Efficiency through AI-Driven Predictive Models for Continuous Integration and Deployment Pipelines [Electronic resource] // ResearchGate. – URL: https://www.researchgate.net/publication/390943967_Enhancing_DevOps_Efficiency_through_AI-Driven_Predictive_Models_for_Continuous_Integration_and_Deployment_Pipelines (date accesses: 08/15/2025).
8. Chen L., Ali -Eldin A., Elmroth E. Reinforcement Learning-based Autoscaling in Cloud: A Survey // ACM Computing Surveys . – 2023. – Vol. 55, No. 6. – P. 1–36. – DOI: 10.1145/3520242.
9. Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps – Building and Scaling High Performing Technology Organizations. – Portland: IT Revolution Press, 2018. – 288 p.

10. Sutton RS, Barto AG Reinforcement Learning: An Introduction. – 2nd ed. – Cambridge: MIT Press, 2018. – 552 p.
11. Zhang C., Zhao Y., Xu J., et al. A Survey on Deep Reinforcement Learning in Cloud Resource Management // IEEE Access. – 2021. – Vol. 9. – P. 125–145. – DOI: 10.1109/ACCESS.2021.3049876.
12. Doshi -Velez F., Kim B. Towards A Rigorous Science of Interpretable Machine Learning // arXiv preprint. – 2017. – URL: https://arxiv.org/abs/1702.08608 (date accesses: 20.08.2025).
13. Mao H., Alizadeh M., Menache I., Kandula S. Resource Management with Deep Reinforcement Learning // Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets) . – ACM, 2019. – P. 50–56.