WJARR

World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Review Article)

Check for updates

AI-powered real-time data pipeline optimization using deep reinforcement learning

Deepika Annam *

Independent Researcher, USA.

World Journal of Advanced Research and Reviews, 2025, 26(02), 2647-2653

Publication history: Received on 08 April 2025; revised on 16 May 2025; accepted on 19 May 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.1957

Abstract

Deep Reinforcement Learning (DRL) represents a transformative paradigm for real-time data pipeline optimization across diverse industrial applications. Traditional optimization techniques often yield suboptimal results in dynamic environments with fluctuating workloads, while DRL enables autonomous systems to adapt through experience. This article examines how DRL integrates with distributed stream processing systems to address critical challenges, including workload unpredictability, resource dependencies, and infrastructure heterogeneity. The integration of neural networks with reinforcement learning principles allows for sophisticated decision-making that significantly improves resource utilization and operational efficiency. Various algorithms, including Deep Q-Networks, Proximal Policy Optimization, and Soft Actor-Critic, demonstrate particular efficacy in different application contexts. From healthcare to data centers, robotics to IoT systems, DRL implementation delivers measurable improvements in throughput, latency reduction, and resource optimization. Though implementation challenges exist, including hyperparameter sensitivity and sample efficiency considerations, the potential benefits of DRL-powered optimization for data-intensive industries are substantial, offering a path toward more intelligent, adaptive, and efficient data processing architectures.

Keywords: Deep Reinforcement Learning; Data Pipeline Optimization; Stream Processing; Resource Management; Adaptive Control

1. Introduction

In today's distributed stream processing systems, thousands of real-time streams may enter the system through processing nodes, where hundreds of nodes may be co-located or geographically distributed. Resource management for these systems is complicated by several factors: processing elements are constrained by producer-consumer relationships, data and processing rates can be highly bursty, and traditional measures of effectiveness, such as utilization, can be misleading [1]. The stream processing paradigm has always played a key role in time-critical systems, with applications ranging from real-time exploratory data mining to high-performance transaction processing [1].

Traditional optimization techniques for data pipelines, such as manual tuning and heuristics, usually yield suboptimal results and resource utilization, especially in changing environments with different workloads [2]. Resource management challenges include workload dynamicity, unpredictability, complex resource dependencies, heterogeneity of infrastructure, and multiple optimization objectives [2]. The classical solution to burstiness problems is to add buffers, but designing for very high data rates and scalability makes buffering increasingly expensive as system memory becomes a severe constraint [1].

Reinforcement Learning (RL) has gained pronounced recognition in recent decades as a powerful paradigm aimed at self-organizing and controlling complex systems [2]. In RL, an agent learns how to make the best decisions in interaction

[*] Corresponding author: Deepika Annam

with an environment by maximizing a cumulative reward signal [2]. The emergence of deep reinforcement learning techniques has further improved the applicability and effectiveness of RL in different fields [2].

Experimental results from case studies show promising improvements through RL applications. For Apache Spark, an RL-based resource allocation method completed tasks up to 20% faster than heuristic policies and used resources 25% more efficiently [2]. In Apache Flink, an RL-based approach for data flow control obtained a 30% reduction in end-to-end latency and a 20% increase in throughput compared to rule-based policies [2]. For Kubernetes task placement, the RL algorithm policy accomplished up to 15% fewer task completion times and 20% fewer messages than heuristic approaches [2].

The ACES (Adaptive Control for Extreme-scale Stream processing systems) approach proposes a two-tiered optimization where global optimization determines time-averaged allocations, and a distributed resource controller uses adaptive control to ensure stability in the presence of burstiness [1]. This approach outperforms traditional approaches in terms of weighted throughput by over 20% in the limit of small buffers and over a wide range of burstiness levels, while maintaining end-to-end delay as little as a third of traditional approaches [1].

2. Fundamentals of Deep Reinforcement Learning for Data Pipelines

Deep Reinforcement Learning (deep RL) integrates the principles of reinforcement learning with deep neural networks, enabling agents to excel in diverse tasks [3]. According to Terven's overview, reinforcement learning is a paradigm of machine learning in which an agent learns an optimal behavior by interacting with an environment, receiving feedback in the form of rewards or penalties, and adapting its actions to maximize long-term returns [3]. The agent aims to maximize the expected cumulative reward, which can be written in the infinite-horizon setting as follows: E[∑(t=0 to ∞) $\gamma^t r_t$], where r_t is the reward received at time t, and $0 \leq \gamma < 1$ is a discount factor that balances the importance of immediate versus future rewards [3].

The RL framework consists of states, actions, rewards, policies, and value functions [3]. The state space represents the current condition of the system. In the context of data pipelines, as noted by Rafie et al., "Real-world problems usually have many features making it hard to model and describe the data" [4]. The action space encompasses all possible interventions the agent can take. Terven explains that policy gradient methods directly learn a parameterized policy $\pi(a|s,\theta)$ that maps state-to-action probabilities [3]. The reward function defines the optimization goals. A transformative breakthrough occurred when deep Q-networks (DQNs) demonstrated human-level performance on dozens of Atari 2600 video games using only raw pixel inputs and game scores as the sole training signals [3]. DQN addressed key challenges through two crucial stabilization techniques: experience replay and target network [3]. Experience replay stores transitions in a replay buffer and samples mini-batches randomly for training, breaking the strong correlations present in sequential observations [3]. The target network is a copy of the Q-network that is held fixed for a number of iterations and then periodically updated, which slows down changes in the target and reduces oscillations [3].

For data pipeline optimization challenges, Rafie et al. identify several limitations in traditional approaches: "Although the methods mentioned above can improve learning performance, however, they are involved with several limitations. For example, before starting the feature selection process, it is necessary to have access to the whole feature space. While in many real-world applications, such as a renowned microblogging and social networking service, features appear over time, and it is impossible to have all features at the beginning of the process" [4].

Soft actor-critic (SAC) is particularly relevant for continuous control tasks. As Terven notes, by optimizing not just for reward but also for high action entropy, SAC avoids collapsing to deterministic or overly narrow policies, substantially improving exploration [3]. In practical robotic scenarios, for example, navigating uneven terrain or manipulating objects under uncertainty, SAC's stochastic exploration allows the agent to discover robust strategies without extensive manual tuning [3].

Rafie et al. propose multi-objective approaches to feature selection that could be applicable to data pipelines: "The first objective function maximizes the relevancy criterion, while the second minimizes redundancy among the selected features" [4]. This approach is particularly valuable as "in contrast to most prior methods using an objective function, the Pareto set is used to select features with maximum relevance and minimal redundancy" [4].

According to Terven, three critical challenges exist in applying RL to real-world systems: sample efficiency, safety, interpretability, and multi-task learning [3]. For data pipelines, Rafie et al. note that "three critical conditions must satisfy each online multi-label streaming feature selection method; To begin, no domain knowledge of feature space should be required. Also, it must perform effective incremental updates in selected features. Furthermore, it should be accurate in each time instance for the classification performance to be acceptable" [4].

The application of DRL to data pipelines aligns with its broader use in resource management. As Terven notes, "In resource management scenarios, RL is used in distributed systems and cloud infrastructures. Data centers rely on RL to allocate computational resources, balance server loads, and regulate energy consumption" [3]. This makes DRL particularly suitable for optimizing data pipelines, where resources must be dynamically allocated in response to changing workloads and conditions.

Table 1 Chronological Evolution of Deep Reinforcement Learning Algorithms for Resource Management [3,4]

Algorithm	**Key Characteristics**	**Year Introduced**
DQN (Deep Q-Network)	Uses experience replay and target networks	2015
PPO (Proximal Policy Optimization)	Clips probability ratio to prevent large policy updates	2017
TRPO (Trust Region Policy Optimization)	Enforces constraint on policy change between updates	2015
SAC (Soft Actor-Critic)	Maximizes both reward and entropy for exploration	2018
DDPG (Deep Deterministic Policy Gradient)	Uses deterministic policy with target networks	2015
A3C (Asynchronous Advantage Actor-Critic)	Uses multiple workers to decorrelate experience	2016

3. Implementing DRL-Powered Pipeline Optimization

Implementing DRL for data pipeline optimization involves several key components that enable adaptive performance tuning for recommendation models. According to Nagrecha et al., their InTune system demonstrated that DRL-based optimization can increase data ingestion throughput by as much as 2.29X versus current state-of-the-art data pipeline optimizers while improving both CPU and GPU utilization [5]. This significant improvement highlights the effectiveness of reinforcement learning approaches for pipeline optimization.

The DRL agent is at the core of InTune, learning how to distribute CPU resources across a DLRM data pipeline to effectively parallelize data-loading and improve throughput. The system environment reflects various factors, including pipeline latency, free CPUs, free memory in bytes, model latency, DRAM-CPU bandwidth, and CPU processing speed [5]. The agent uses this information to determine appropriate resource allocation. As explained by Nagrecha et al., the reward function is based on pipeline throughput and memory usage, designed so that rewards approach zero as memory consumption nears 100%, thus preventing out-of-memory errors that frequently occur with other optimization approaches [5].

InTune's DRL agent uses a simple three-layer MLP architecture to minimize computational demands, requiring only about 200 FLOPs per iteration. This lightweight design ensures the agent doesn't interfere with the actual model training job [5]. The action space is designed to be incremental, allowing the agent to raise, maintain, or lower resource allocation for each pipeline stage by specified increments. This approach enables rapid convergence to an optimized solution within just a few minutes, even on complex real-world pipelines [5].

For IoT applications specifically, Mohammadi et al. note that traditional ML tools do not sufficiently address emerging analytic needs of IoT systems, particularly for streaming data that requires fast processing. Their survey emphasizes that IoT applications need different modern data analytics approaches according to the hierarchy of data generation and management [6]. They classify IoT analytics into big data analytics and streaming data analytics, with the latter requiring processing close to the source of data to remove unnecessary communication delays.

Mohammadi et al. also highlight that combining DRL with IoT enables more intelligent systems. They demonstrate that semi-supervised deep reinforcement learning can be applied to localization in smart campus environments, where the learning agent finds the best action to perform based on received signals from Bluetooth beacons [6]. Their experimental results show that the semi-supervised model consistently outperforms the supervised model in terms of rewards received and proximity to targets [6].

The implementation challenges for DRL in IoT contexts include the lack of large training datasets and preprocessing requirements. According to Mohammadi et al., most DL approaches require some preprocessing to yield good results, with image processing techniques working better when input data is normalized, scaled into specific ranges, or transformed into standard representations [6]. For IoT applications, preprocessing becomes more complex as the system deals with data from different sources that may have various formats and distributions while showing missing data [6].

Security and privacy preservation are also critical concerns for DRL implementations in IoT. Mohammadi et al. note that DL models must be enhanced with mechanisms to discover abnormal or invalid data, as they learn features from raw data and therefore can learn from invalid inputs. They suggest implementing a data monitoring DL model alongside the main model to address this issue [6].

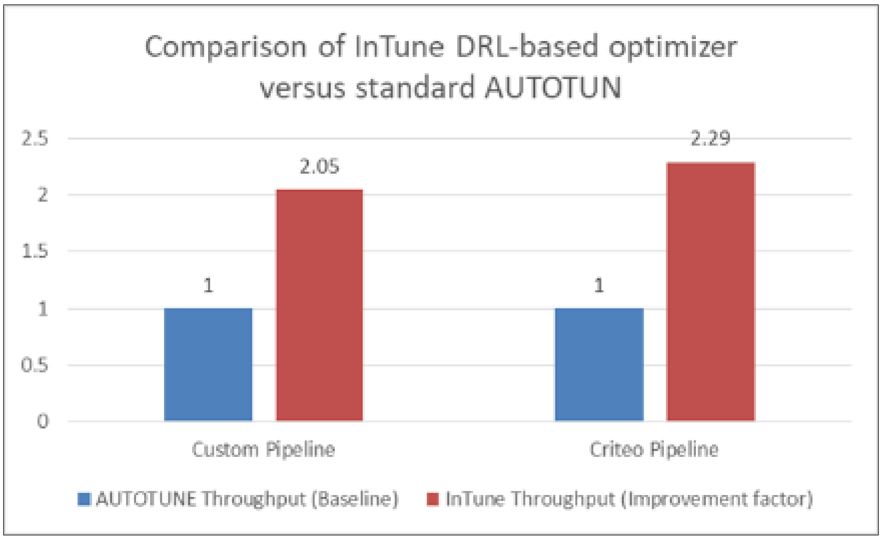

Figure 1 Improvements with InTune DRL-based optimizer over standard AUTOTUNE [5,6]

4. Benefits and Performance Improvements

Organizations implementing reinforcement learning for optimization can achieve significant benefits based on findings from the literature. According to Ogunfowora and Najjaran's comprehensive survey [7], reinforcement learning has seen substantial growth in maintenance planning applications, with an 80% increase in the number of RL and DRL-based publications for maintenance planning between 2019 and 2023.

The application of reinforcement learning techniques has demonstrated meaningful improvements in diverse optimization contexts. As documented in [7], maintenance activities typically consume 15%-40% of total production costs in factories. By leveraging condition monitoring data with reinforcement learning, organizations can develop smart maintenance planners that serve as precursors to achieving a smart factory [7]. These approaches help reduce machine failures, improve reliability, and reduce maintenance and production costs associated with unplanned downtime.

RL optimization has shown benefits in resource management in different contexts. According to Poloskei, "Since the public cloud providers serve on-demand invoicing, the reserved resources should be connected to the running tasks" [8]. This is particularly important because "The training process of a deep learning model takes some time" and "the training quality can often be efficiently increased by committing more resources, like attaching computation-intensive hyperparameter optimization measures" [8].

Intelligent workflow management translates to efficiency benefits, as demonstrated in Poloskei's research. MLOps approaches in cloud-native ecosystems leverage the cloud's full capabilities as cloud-native services, making operations more affordable and implementation more powerful [8]. A study conducted by Hummer et al. and cited in subsequent research indicates that "data handling uses 7% of the total execution time, but this time can be reduced due to parallelized computing procedures" [8]. This efficiency gain stems from the ability to specify workflows as a Directed Acyclic Graph (DAG) [8].

RL-powered approaches demonstrate superior performance compared to traditional implementations. As noted in [7], organizations that developed proper maintenance policies were able to "reduce the costs associated with planned and unplanned downtime of machines and maintenance costs." The authors also observed that agents using deep reinforcement learning for maintenance planning of wind turbines "outperformed the corrective, scheduled, and predictive maintenance strategies irrespective of the number of available maintenance crews because the agent learned to perform maintenance activities when the wind turbines are in a low power mode or demand is low" [7].

Beyond direct performance benefits, organizations gain operational efficiencies. According to Poloskei, "The MLOps approach concentrates on the modeling, eliminating the personnel and technology gap in the deployment" [8]. This approach helps address significant challenges, as "For a flourishing big data project, the organization should have analytics and information-technological know-how" [8]. The MLOps paradigm helps bridge these gaps by providing a structured approach to data pipeline design in cloud-native ecosystems, which, according to Poloskei's analysis, is "the recommended way for data pipeline design" [8].

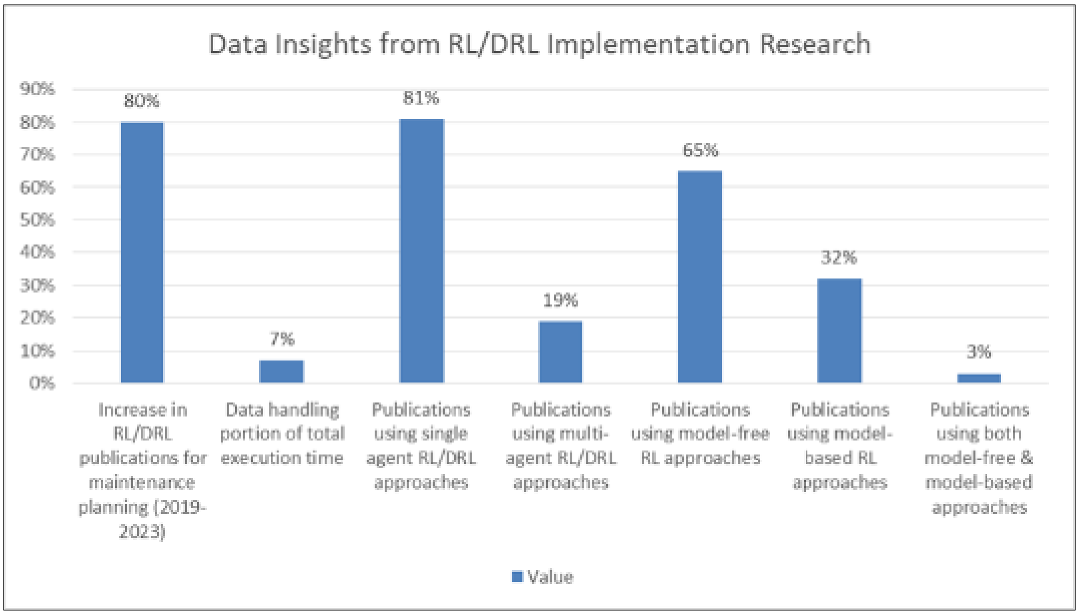

Figure 2 Data Insights from RL/DRL Implementation Research [7,8]

5. Industry Applications and Case Studies

DRL-powered pipeline optimization is delivering transformative results across numerous data-intensive industries. In healthcare, reinforcement learning applications have shown remarkable potential. As documented in Al-Hamadani et al.'s comprehensive review, reinforcement learning has been effectively applied in both healthcare and robotics domains [9]. For robotics applications, reinforcement learning addresses the challenges of robotic grasping and manipulation in unstructured and dynamic environments, which remain critical problems due to the variability and complexity of the real world [9]. Traditional machine learning approaches often struggle to handle the diversity of objects in terms of size, weight, texture, transparency, and fragility. Consequently, reinforcement learning has emerged as a solution, allowing robots to learn through trial and error and adapt to various situations [9].

In the healthcare sector, reinforcement learning techniques have been applied to cell growth problems, an area of increasing interest due to its significance in optimizing cell culture conditions, advancing drug discovery, and enhancing understanding of cellular behavior [9]. Studies have shown applications in modeling cell movement, particularly in the early stage of C. elegans embryogenesis, where deep reinforcement learning was combined with agent-based modeling frameworks to model basic cell behaviors, including cell fate, division, and migration [9].

For data-intensive computing infrastructure, thermal management represents a critical optimization challenge that directly impacts both performance and energy efficiency. Zhang et al. developed a deep reinforcement learning approach for data center thermal management that demonstrated significant potential [10]. Their comprehensive evaluation showed that actor-critic, off-policy, and model-based algorithms outperformed other approaches in terms of optimality, robustness, and transferability [10]. These implementations were able to reduce constraint violations and achieve approximately 8.84% power savings in certain scenarios compared to default controllers [10].

Zhang et al. noted that while DRL techniques show promise, deploying these algorithms in real-world systems presents challenges as they are sensitive to specific hyperparameters, reward functions, and work scenarios [10]. Their experiments revealed that algorithms can be very sensitive to several techniques and hyperparameters, such as state preprocessing, learning rate, and network architecture [10]. The study identified that constraint violations and sample efficiency are areas that still require improvement before widespread real-world implementation [10].

The research conducted by Zhang et al. incorporated a comprehensive four-dimensional analysis of DRL applications in data centers, examining algorithms, tasks, system dynamics, and knowledge transfer [10]. This structured approach enabled detailed evaluation of various DRL algorithms for dynamic thermal management deployment using both analytical and numerical methods [10]. Their findings emphasize the importance of qualitative and quantitative evaluation metrics for comprehensive analysis, including stability, robustness, sample efficiency, safety, asymptotic performance, asymptotic improvement, and jumpstart [10].

These advancements demonstrate how DRL-powered optimization is transforming data processing across diverse industries, though challenges remain in achieving optimal implementation in real-world environments.

Table 2 Reinforcement Learning Performance Across Industrial Applications [9,10]

Algorithm	Performance Metric	Value
PPO and SAC	Success Rate	100%
YOLO and SAC	Success Rate (Building Blocks)	95%
QMIX-PSA	Success Rate (Metal Workpieces)	82%
	Success Rate (Daily Items)	83%
SAC	Success Rate	80%
PPO		70%

6. Conclusion

Deep Reinforcement Learning has established itself as a powerful paradigm for optimizing data pipelines across numerous domains. The integration of neural networks with traditional reinforcement learning principles creates systems capable of learning optimal resource allocation strategies through interaction with complex environments. From healthcare applications that model cell growth and movement to data center thermal management systems that reduce power consumption while maintaining operational parameters, DRL demonstrates versatility and effectiveness. The technology shows particular strength in handling the dynamic, unpredictable nature of modern data processing environments, where traditional methods frequently falter. While implementation challenges persist, including sensitivity to hyperparameters and reward function design, the trajectory of advancement points toward increasingly robust solutions. Actor-critic architectures, off-policy learning, and model-based frameworks have demonstrated superior performance characteristics across multiple metrics. As these technologies mature, organizations can expect continued improvements in operational efficiency, resource utilization, and system performance. The future of data pipeline optimization likely involves increasingly sophisticated DRL implementations that combine the strengths of various algorithmic methods while mitigating their respective challenges, ultimately delivering more intelligent and responsive data processing ecosystems across industries.

References

[1] Lisa Amini et al., "Adaptive Control of Extreme-scale Stream Processing Systems", microsoft.com, 2006, [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/jain06extreme.pdf

[2] Chandrakanth Lekkala, "Leveraging Reinforcement Learning for Autonomous Data Pipeline Optimization and Management", IJSR, 2023, [Online]. Available: https://www.ijsr.net/archive/v12i5/SR24531190901.pdf

[3] Juan Terven, "Deep Reinforcement Learning: A Chronological Overview and Methods", MDPI, Feb. 2025, [Online]. Available: https://www.mdpi.com/2673-2688/6/3/46

[4] Azar Rafie et al., "A Multi-Objective online streaming Multi-Label feature selection using mutual information", ScienceDirect, 2023, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417422024472

[5] Kabir Nagrecha et al., "InTune: Reinforcement Learning-based Data Pipeline Optimization for Deep Recommendation Models", ACM Digital Library, 2023, [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3604915.3608778

[6] Mehdi Mohammadi et al., "Deep Learning for IoT Big Data and Streaming Analytics: A Survey", arXiv, 2018, [Online]. Available: https://arxiv.org/pdf/1712.04301

[7] Oluwaseyi Ogunfowora, and Homayoun Najjarana, "Reinforcement and Deep Reinforcement Learning-based Solutions for Machine Maintenance Planning, Scheduling Policies, and Optimization", arXiv, 2023, [Online]. Available: https://arxiv.org/pdf/2307.03860

[8] Istvan Poloskei, "MLOps approach in the cloud-native data pipeline design", ResearchGate, 2021, [Online]. Available: https://www.researchgate.net/publication/350775603_MLOps_approach_in_the_cloud-native_data_pipeline_design

[9] Mokhaled N A Al-Hamadani et al., "Reinforcement Learning Algorithms and Applications in Healthcare and Robotics: A Comprehensive and Systematic Review", National Library of Medicine, 2024, [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11053800/

[10] Qingang Zhang et al., "Deep reinforcement learning towards real-world dynamic thermal management of data centers", ScienceDirect, 2023, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0306261922018189