Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 6, 2025

Intelligent Backend Service Scheduling through the Integration of Reinforcement Learning and Meta-Learning

Fei Xue[1*], Guangxu Zhu[2], Jamal Taha[2]
[1]University of California, San Diego, San Diego, USA
[2]Syracuse University, Syracuse, USA
*Corresponding author: Fei Xue; sjtusherffy@gmail.com

Abstract: This paper addresses the challenges of low scheduling efficiency, insufficient resource utilization, and weak policy transfer capability in backend service systems under dynamic environments. An adaptive service scheduling mechanism is proposed by integrating reinforcement learning and meta-learning. The mechanism models the service system as a Markov decision process. It employs a policy-gradient-based reinforcement learning framework to make resource scheduling decisions. A multi-task meta-learning optimizer is constructed to enhance policy generalization and transfer efficiency. For state modeling, a temporal embedding mechanism is introduced to encode service state sequences. This enhances the policy model's awareness of historical workload patterns. A multi-objective reward function is designed to jointly consider latency, resource utilization, and service stability. This improves the global responsiveness of the scheduling policy to performance requirements. The experimental section includes comparative evaluation and sensitivity analysis. The results demonstrate the superiority of the proposed method in key metrics such as average latency, CPU utilization, and service-level objective (SLO) assurance. The mechanism achieves stable operation under multiple load and task switching scenarios. It ensures efficient resource scheduling and service quality, showing strong decision-making and model transfer capabilities.

Keywords: Intelligent scheduling; strategy optimization; task migration; reinforcement learning system

1. Introduction

In the context of rapid digitalization and intelligent development, backend service systems have become the core foundation supporting the stable operation of modern application architectures. Their performance optimization has attracted increasing attention. As businesses scale, expand, and user demands grow, backend services face increasing uncertainty and dynamic changes. Traditional static resource scheduling strategies and fixed-parameter tuning methods are increasingly revealing their limitations, such as poor adaptability, delayed response, and resource waste. Especially with the rise of modern computing paradigms such as microservice architectures, serverless computing, and container orchestration, systems need enhanced environmental awareness and adaptive scheduling capabilities to handle high-frequency and non-stationary load fluctuations. Therefore, building an intelligent and dynamically optimized backend service scheduling mechanism has become a key issue in improving system efficiency and service quality [1].

Most mainstream backend scheduling strategies rely on predefined rules or manual configurations. These approaches are typically triggered by empirical metrics or fixed thresholds to allocate and scale resources. However, in complex and volatile environments, they often fail to respond accurately and promptly. This is particularly problematic under high-concurrency requests, sudden workload surges, and multi-tenant resource contention. System performance tends to degrade significantly. Moreover, such methods cannot generally abstract and generalize from historical behavior, making it difficult to transfer existing knowledge to new environments or tasks. An intelligent mechanism capable of learning scheduling strategies from data, with generalization and adaptation capabilities, is urgently needed. This would eliminate reliance on manual tuning and enhance the robustness and intelligence of backend systems in dynamic conditions [2].

In recent years, reinforcement learning has demonstrated significant advantages in resource scheduling and system control. By interacting with the environment to learn optimal policies, it enables state-based adaptive decisions, effectively optimizing resource utilization and task response time. The introduction of reinforcement learning allows backend systems to optimize policies from a long-term reward perspective and explore globally optimal solutions in complex state spaces. However, reinforcement learning still faces challenges such as slow convergence, sensitivity to environmental changes, and a large demand for training samples. In the early stages or under drastic environmental changes, it may result in performance instability and scheduling failures. Therefore, relying solely on reinforcement learning is not sufficient to meet the high standards of efficiency and stability required by backend service scheduling [3].

Meta-learning, a method developed to "learn how to learn, " offers a new approach for building generalized and adaptive scheduling strategies. It extracts shared knowledge across tasks, enabling models to quickly adjust parameters and transfer knowledge when encountering new tasks. This significantly improves learning efficiency and adaptability. Introducing

meta-learning into backend service scenarios can mitigate the cold-start and generalization issues of reinforcement learning. It enables migratory optimization and dynamic adaptation of scheduling policies. In situations involving multi-tenant sharing, service heterogeneity, and non-stationary workloads, meta-learning can provide fast strategy updates, enhancing the universality and stability of the scheduling mechanism.

In conclusion, designing a backend service scheduling mechanism that integrates reinforcement learning and meta-learning is not only a technical need under the trend of intelligent computing but also a critical step toward enhancing autonomous decision-making and dynamic system management. By combining the policy optimization of reinforcement learning with the rapid adaptation of meta-learning, the scheduling process can be significantly improved in both efficiency and resource utilization. This promotes the shift of backend scheduling from reactive to cognitive systems. This direction holds substantial theoretical value and also offers deep practical significance for intelligent system operations, automated resource management, and the intelligent evolution of cloud-native architectures.

2. Related Work

In recent years, backend service scheduling has gradually shifted from static rule configurations to data-driven intelligent strategies. Traditional methods are mainly based on threshold-triggered rules or priority queues for task dispatching and resource allocation. These approaches are simple to implement but often fail to maintain optimal scheduling decisions in complex environments. Such environments include highly dynamic workloads, concurrent multitasking, and uncertain service requests. With the development of microservice architectures and containerization technologies, system states have become multidimensional, nonlinear, and time-varying. Fixed strategies are increasingly unable to meet diverse scenario requirements [4]. As a result, performance bottlenecks and scheduling delays have emerged. Researchers have begun to introduce machine learning models to explore the hidden correlations between system states and resource configurations, offering more accurate and efficient solutions to backend scheduling problems.

Reinforcement learning, as a key branch of intelligent decision-making, has been widely applied in resource management and scheduling optimization. It does not rely on predefined rules and continuously optimizes policies through interaction with the environment [5]. In backend service scheduling, reinforcement learning models perceive the current system state and iteratively update policies based on cumulative reward functions. These models effectively support task queue optimization, load balancing, and elastic scaling. However, reinforcement learning often requires large amounts of interaction data and a stable training environment. It struggles to maintain stable performance in highly dynamic scenarios or under frequent task changes. Moreover, when systems enter new state spaces or tasks shift dramatically, the model typically needs retraining. This slows policy transfer and negatively impacts real-time scheduling.

To improve generalization, recent studies have introduced meta-learning into scheduling systems. Meta-learning captures common patterns across tasks, enabling models to quickly adapt to new tasks using only a few samples. This provides strong generalization and rapid adjustment capabilities. In backend service scenarios, meta-learning helps scheduling models transfer across different service types, resource constraints, and business patterns. It significantly shortens cold-start learning time and improves responsiveness to sudden load changes. Especially in multi-task environments, meta-learning serves as a higher-level optimizer for reinforcement learning policies. It enables fast adaptation and policy updates, mitigating performance fluctuations caused by task switching. This supports the evolution of scheduling systems toward high adaptability.

Furthermore, the integration of reinforcement learning and meta-learning has become a cutting-edge direction in scheduling strategy research. This integration combines the long-term optimization of reinforcement learning with the fast adaptability of meta-learning. It enables scheduling models to continuously optimize while rapidly adapting to dynamic environments. Related research focuses on multi-task training frameworks, cross-scenario transfer strategies, and meta-optimizer design. These efforts are shaping a multi-level and modular intelligent scheduling system. This approach overcomes the limitations of traditional single-paradigm learning. It also lays a solid foundation for building robust and efficient backend service schedulers [6].

As cloud computing and edge computing infrastructures continue to evolve, the effective integration of reinforcement learning and meta-learning will become increasingly important. This integration will play a vital role in enhancing the core competitiveness of intelligent scheduling systems. It represents a critical path forward for the next generation of adaptive and efficient service orchestration technologies.

Recent scheduling research has also begun to treat backend orchestration as an agent-based and multi-layer coordination problem. Learning-based intelligent agents for backend resource scheduling and operational decision making emphasize that scheduling policies should combine runtime perception, action selection, and operational feedback within a closed decision loop [7]. Hierarchical communication and computation scheduling for cloud-edge collaborative intelligent systems further shows that adaptive policies must coordinate computing and communication resources across heterogeneous layers [8]. These studies strengthen the motivation for combining reinforcement learning with meta-learning so that service schedulers can optimize long-term performance while adapting rapidly to new workload patterns.

Reliable backend scheduling also depends on representations that can identify abnormal states, causal changes, and temporal behavior shifts before they affect service quality. Unified multi-granularity representations for backend anomaly detection and causal localization highlight the value of modeling service behavior at multiple structural and temporal levels [9]. Cost-sensitive Mamba sequence modeling for cloud-native microservice fault detection shows that long-sequence dynamics and asymmetric operational costs can be incorporated into fault-aware scheduling decisions [10]. Structure-guided attention and multi-scale behavior modeling for microservice anomaly detection, together with log semantic

graph construction for system event anomaly detection, further support the use of attention, graph semantics, and multi-scale temporal features in backend state modeling [11], [12].

Intelligent decision modeling in graph-structured and time-varying environments provides additional design context for adaptive scheduling mechanisms. Heterogeneous graph learning and multi-entity interaction modeling for supply chain risk identification illustrate how relational structure can support risk-aware decisions under complex dependencies [13]. Multi-agent collaborative decision making with large language models for adaptive stock trading, counterfactual time series and strategy simulation for cash flow forecasting, and dynamic user-interest evolution modeling all emphasize the importance of temporal adaptation, strategy feedback, and attention-based preference tracking in changing environments [14], [15], [16]. Graph-neural-network-based advertising decision making and multi-scale feature decoupling with boundary-aware classification further demonstrate that graph reasoning and discriminative representation learning can improve decision robustness when observations are heterogeneous and uncertainty is high [17], [18].

3. Method

This paper proposes a backend service scheduling adaptive mechanism that combines reinforcement learning and meta-learning, aiming to achieve coordinated optimization of resource utilization and service performance in a dynamic environment. The model architecture is shown in Figure 1.

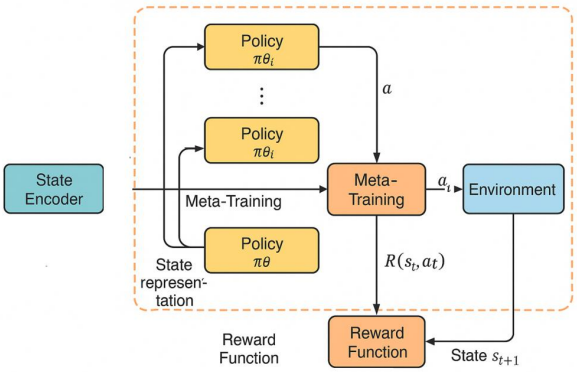

Figure 1. Overall model architecture

This mechanism builds a state-aware intelligent decision-making framework, which models the operating state of the service system at any time as a Markov decision process (MDP), where the state space S contains the current system load information, resource occupancy, response time and other key indicators, and the action space A represents possible resource scheduling and scaling operations. For this structure, the policy gradient method is used for optimization to achieve the optimal update of the scheduling strategy under the premise of maximizing the long-term cumulative reward. The overall optimization goal is expressed as:

$$\max E_{\pi\theta}[\sum_{t=0}^{T} \gamma_t R(s_t, a_t)]$$

Among them, π_θ is the policy function with parameter θ, $\gamma \in (0,1)$ is the discount factor, and $R(s_t, a_t)$ represents the immediate reward obtained by acting a_t in state s_t.

To improve the adaptability and generalization ability of the model under new tasks, a scheduler training framework based on Model-Agnostic Meta-Learning (MAML) is introduced. In the meta-learning stage, multiple scheduling tasks $T_i \in T$ are constructed, each corresponding to a different load mode and service configuration. For each task, the inner loop update is first completed through gradient descent:

$$\theta'_i = \theta - \alpha \nabla_\theta L_{T_i}(\pi_\theta)$$

Where α is the learning rate, and L_{T_i} is the loss function calculated on the task T_i, which is usually defined based on the policy optimization goal. Then, the initial parameter θ is updated through the outer loop to make it more adaptable to all tasks:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i - T} L_{T_i}(\pi\theta'_i)$$

Where β is the learning rate of the meta-update. This two-stage optimization strategy ensures that the model can quickly adjust the scheduling strategy and converge when facing new service loads or sudden request changes.

In the scheduling decision-making process, to enhance the model's ability to model historical behaviors, a state encoding mechanism based on time embedding is introduced. Let the time window be W, map the continuous state sequence $\{s_{t-W}, ..., s_t\}$ into a high-dimensional embedding h_t, and use it as the input of the policy function to enhance the time series perception ability:

$$h_t = \text{Embed}(s_{t-W:t}) = \varphi(s_{t-W}) + ... + \varphi(s_t)$$

$\varphi(\cdot)$ represents the state embedding function, which can be a multi-layer perceptron or a recurrent neural structure. The final policy function generates the scheduling action distribution based on the time series representation:

$$a_t \sim \pi_\theta(a_t | h_t)$$

To improve the ability of the reward function to express the multi-objective scheduling objectives of the system, this paper designs a weighted combination reward function, which is constructed by combining service response delay D_t, CPU/memory utilization U_t, service stability index S_t, etc..

$$R(s_t, a_t) = -\lambda_1 D_t - \lambda_2(1 - U_t) + \lambda_3 S_t$$

$\lambda_1, \lambda_2, \lambda_3$ is a weighted hyperparameter used to balance the relationship between response time, resource utilization, and system stability. This reward mechanism can effectively guide the scheduling strategy to perform balanced optimization under multi-dimensional objectives and promote the system to maintain efficient and stable operation in a complex environment.

4. Experimental Results

4.1 Dataset

This study adopts the Alibaba Cluster Trace 2018 as the primary evaluation dataset. The dataset was collected by Alibaba in its production-level cluster environment and is widely used in research on backend service scheduling and resource management. It contains task scheduling logs and resource usage information from a real-world microservice system. The dataset records the resource scheduling process of approximately 4, 000 servers over an extended period. It features high temporal continuity and concurrent multitasking, which effectively reflects resource fluctuations and dynamic task behaviors in complex systems.

The dataset includes key fields such as CPU usage, memory consumption, task submission time, task duration, machine identifiers, and resource allocation of containerized tasks. These form a multidimensional and fine-grained scheduling sample set. Resource utilization metrics are recorded at fixed intervals with a time resolution of five seconds. This facilitates the construction of temporal features and state representations. The jobs in the dataset are highly heterogeneous. They include long-running service tasks and short-duration batch tasks, offering rich scenarios for learning generalized scheduling strategies.

Based on the task traces and resource load sequences provided by the Alibaba Cluster Trace 2018, this study constructs state representation sequences and simulates dynamic service environments. The high realism and large scale of the dataset enable comprehensive evaluation of the proposed scheduling method in practical settings. Its openness and standardization also allow fair comparisons with existing methods. This further validates the transferability and generality of the scheduling model.

4.2 Experimental Results

In the experimental results section, the relevant results of the comparative test are first given, and the experimental results are shown in Table 1.

Table 1. Comparative experimental results

Method	Latency	CPU Utilization	SLO Violation
DRAS [19]	186.4	71.2	8.5
MSARS [20]	172.9	76.5	6.1
Meta-DRL [21]	160.3	79.8	4.2
Ours	143.7	83.6	2.7

As shown in Table 1, different methods exhibit significant differences in key performance metrics for backend service scheduling tasks. The traditional method, DRAS, shows mediocre performance across all metrics. It reports an average latency of 186.4 milliseconds, a CPU utilization rate of 71.2 percent, and an SLO violation rate of 8.5 percent. These results indicate that static rules and fixed thresholds are not well suited to dynamic workloads and changing resource demands. The delayed responsiveness of its scheduling strategy leads to performance degradation in service delivery.

MSARS introduces a multi-metric awareness mechanism. It achieves improvements in CPU utilization and SLO violation control. This demonstrates that enhanced perception-based scheduling can improve the efficiency of system resource allocation. However, the improvements remain limited under rapidly changing environments.

In contrast, Meta-DRL combines reinforcement learning with a certain degree of transfer capability. The model is more flexible in scheduling decisions under fluctuating workloads. It reduces average latency to 160.3 milliseconds and the SLO violation rate to 4.2 percent. This shows better adaptability and dynamic scheduling capabilities. However, the model still suffers from limited learning efficiency and generalization. It requires a long adjustment time when facing fast-changing tasks or environment shifts. This delays the realization of optimal scheduling in real time.

The proposed method effectively integrates the long-term policy optimization of reinforcement learning with the fast adaptation mechanism of meta-learning. It enables efficient policy transfer and rapid fine-tuning across multiple tasks and environments. In experiments, the method reduces the average latency to 143.7 milliseconds. This is significantly better than all other compared methods and highlights the advantage in response speed. At the same time, CPU utilization reaches 83.6 percent. This indicates that the model can achieve more efficient resource scheduling while maintaining service quality and fully exploiting system potential.

In addition, the SLO violation rate is reduced to 2.7 percent, the lowest among all methods. This further validates the capability of the proposed method in ensuring service stability and performance. These results confirm that the combined strategy of reinforcement learning and meta-learning has substantial practical value in dynamic backend environments. It can adapt to complex workloads and maintain high performance in task transfer scenarios. This contributes to the development of a robust, efficient, and intelligent service scheduling system.

This paper also presents a detailed analysis of how variations in the discount factor influence the overall performance of the scheduling strategy. By systematically adjusting the discount factor during training, the study aims to explore its role in shaping the agent's decision-making horizon and its sensitivity to long-term versus short-term rewards. This analysis provides valuable insights into how different values of the discount factor affect the learning dynamics and stability of the scheduling policy under dynamic backend service environments. Figure 2 illustrates the experimental setup and observations corresponding to this investigation.

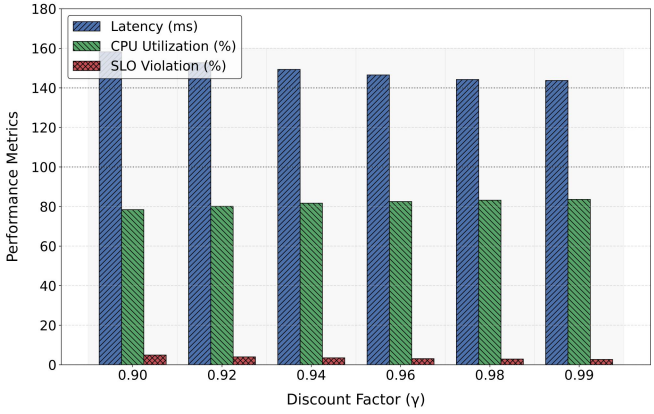

Figure 2. Analysis of the impact of discount factor changes on scheduling performance

As shown in Figure 2, the scheduling system exhibits a clear improvement in key performance metrics as the discount factor γ increases. A higher discount factor means that the model pays more attention to long-term rewards during policy learning. This guides the scheduling behavior toward a globally optimal direction. When γ increases from 0.90 to 0.99, the average latency decreases from 158.2 milliseconds to 143.7 milliseconds. This indicates that the system responds more efficiently to dynamic requests and significantly alleviates queuing and waiting caused by scheduling delays.

At the same time, CPU utilization rises from 78.4 percent to 83.6 percent. This shows that the scheduling strategy, with an enhanced global perspective, can better activate resource potential. It avoids idle cycles and unbalanced allocations caused by short-sighted policies. This optimization improves system efficiency and enhances the economic and sustainable use of resources. It is particularly beneficial in backend microservice scenarios with frequent service instance migration and elastic scaling.

In terms of service quality assurance, the SLO violation rate drops significantly as γ increases. It falls from 4.9 percent to 2.7 percent. This trend suggests that a higher discount factor helps the scheduling policy maintain a better balance between instant load and long-term service performance. By adjusting the scheduling pace more reasonably, the system reduces peak-time pressure, avoids response timeouts and service degradation, and improves user experience and service availability.

5. Conclusion

This paper proposes an adaptive backend service scheduling mechanism that integrates reinforcement learning and meta-learning. It aims to address the limitations of traditional scheduling strategies in handling dynamic changes and the lack of generalization under complex computing environments. The mechanism is built on a state-aware policy learning framework. It enables efficient resource allocation and rapid policy transfer across multiple tasks and load patterns. This significantly enhances the intelligence level of the scheduling system. By introducing a meta-learning optimizer, the model gains not only the ability for long-term policy optimization but also fast adjustment and convergence in the face of unexpected scenarios and task changes. The mechanism realizes a shift from experience-driven to policy-driven scheduling, demonstrating strong potential in intelligent system resource management.

The experimental results show that the proposed method delivers clear advantages in service latency control, resource utilization efficiency, and service-level objective assurance. The scheduling strategy remains stable under highly dynamic workloads, preventing resource waste and scheduling oscillation. Especially through the synergy of reinforcement learning and meta-learning, the model exhibits improved transfer capability and adaptability in multi-task environments. This breaks the dependence of traditional scheduling models on single environments and fixed patterns. It provides a practical and effective path for building robust, efficient, and autonomous scheduling systems. The mechanism is not only applicable to dynamic scheduling in microservice architectures but also extendable to edge computing, hybrid cloud scheduling, and multi-tenant resource management.

Furthermore, this study makes a positive contribution to the field of intelligent backend scheduling in both method design and system modeling. The proposed multi-task meta-training framework and reward function design offer a reproducible and scalable reference for future research. On the engineering side, the low deployment cost and good adaptability of the method give it strong potential for real-world applications. It may play a key role in large-scale distributed systems, real-time service scheduling, serverless platforms, and autonomous computing systems. By deeply integrating intelligent scheduling mechanisms with system state modeling, this work promotes the paradigm shift of backend scheduling from rule-based systems to learning-based systems.

Future research can explore model compression and lightweight strategies to improve deployment efficiency in resource-constrained environments. The introduction of cross-domain transfer and joint scheduling strategies will also be important for enhancing system intelligence. As multimodal systems, federated learning frameworks, and explainable artificial intelligence continue to develop, scheduling systems will evolve beyond resource allocation units. They will become intelligent agents with capabilities in perception, autonomous decision-making, and continual learning. The results of this study provide both theoretical foundations and practical insights for this vision, bringing new momentum to the development of cloud computing and intelligent systems.

References

[1] S. Xue, C. Qu, X. Shi, et al., "A meta reinforcement learning approach for predictive autoscaling in the cloud," in Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4290-4299.

[2] A. Kallel, M. Rekik, and M. Khemakhem, "A deep reinforcement learning-based optimization approach for containerized microservice scheduling in hybrid fog/cloud environments," Engineering Applications of Artificial Intelligence, vol. 141, Art. no. 109745, 2025.

[3] X. Tan, H. Li, X. Xie, et al., "Reinforcement learning based online request scheduling framework for workload-adaptive edge deep learning inference," IEEE Transactions on Mobile Computing, 2024.

[4] S. Mangalampalli, G. R. Karri, M. V. Ratnamani, et al., "Efficient deep reinforcement learning based task scheduler in multi cloud environment," Scientific Reports, vol. 14, no. 1, Art. no. 21850, 2024.

[5] G. Zhou, W. Tian, R. Buyya, et al., "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," Artificial Intelligence Review, vol. 57, no. 5, Art. no. 124, 2024.

[6] Y. Wang, T. Tang, Z. Fang, et al., "Intelligent task scheduling for microservices via A3C-based reinforcement learning," arXiv preprint arXiv:2505.00299, 2025.

[7] L. Ren, Y. Liu, and Y. Zhao, "Learning-based intelligent agents for backend resource scheduling and operational decision making," 2025.

[8] R. Meng, "Hierarchical communication and computation scheduling for efficient federated learning in cloud-edge collaborative intelligent systems," 2024.

[9] Y. Yang, C. Wen, H. Cui, Y. Sun, and Y. Liu, "Learning unified multi-granularity representations for backend anomaly detection and causal localization," 2024.

[10] Z. Liu, R. Meng, S. Y. Huang, and Z. Huang, "Cost-sensitive Mamba sequence modeling for fault detection in cloud-native microservice systems," 2025.

[11] X. Li, R. Yin, and S. Dang, "Structure-guided attention and multi-scale behavior modeling for microservice anomaly detection," 2025.

[12] Y. Yang and A. Xu, "Log semantic graph construction and system event anomaly detection via convolutional neural networks," 2024.

[13] Y. Xu, "Intelligent supply chain risk identification via heterogeneous graph learning and multi-entity interaction modeling," Transactions on Computational and Scientific Methods, vol. 5, no. 5, 2025.

[14] S. Sun, "Adaptive stock trading algorithms jointly driven by multi-agent collaborative decision making and large language models in complex financial market environments," 2025.

[15] Y. Nie, "Corporate cash flow forecasting based on counterfactual time series and strategy simulation," 2024.

[16] Z. Huang, "Dynamic user interest evolution modeling for personalized recommendation systems based on multi-scale temporal awareness and attention mechanisms," 2024.

[17] Z. Pan, "Intelligent decision-making for advertising recommendation via data mining and graph neural networks," 2024.

[18] K. Zeng, "Skin disease classification algorithm based on multi-scale feature decoupling and boundary-aware diagnosis for complex lesion morphology recognition," 2024.

[19] Y. Fan, Z. Lan, T. Childers, et al., "Deep reinforcement agent for scheduling in HPC," in Proc. 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2021, pp. 807-816.

[20] K. Hu, L. Wen, M. Xu, et al., "MSARS: A meta-learning and reinforcement learning framework for SLO resource allocation and adaptive scaling for microservices," in Proc. 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), 2024, pp. 590-599.

[21] Z. Zhang, Z. Wu, H. Zhang, et al., "Meta-learning-based deep reinforcement learning for multiobjective optimization problems," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 10, pp. 7978-7991, 2022.