Transactions on Computational and Scientific Methods | Vol. 4, No. 4, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Reinforcement Learning-Driven Dynamic Cache Management with Hotspot-Aware Replacement

Siyuan Dang[1*], Changfan Chen[2], Kok-Seng Chua[3]
[1]Stevens Institute of Technology, New Jersey, USA
[2]Johns Hopkins University, Baltimore, USA
[3]University of California, Los Angeles, Los Angeles, USA
[*]Corresponding Author: Siyuan Dang; sheboke93@gmail.com

Abstract: This paper focuses on the problem of cache management in dynamic environments. It proposes a hotspot-aware dynamic cache replacement strategy based on reinforcement learning. The goal is to improve the intelligence and efficiency of cache systems under complex access scenarios. The method models service nodes, content requests, and access features in the cache system as a state space. It designs a multi-dimensional state representation that incorporates access frequency, content popularity, recent access time, and content lifespan. A policy network is introduced to score and select each cache replacement action. Under the reinforcement learning framework, the model dynamically evaluates cache hit behavior through a reward function. The policy is optimized to maximize long-term benefits. To validate the effectiveness of the proposed approach, extensive experiments are conducted on a real-world YouTube Cache dataset. The evaluation covers several key metrics, including hit rate, system latency, misreplacement ratio, and traffic control. Sensitivity analyses are also performed across different cache capacities and parameter settings. The experimental results show that the proposed strategy demonstrates superior adaptability and robustness under various conditions. It significantly outperforms traditional replacement policies and existing learning-based methods. This study not only enhances the decision-making ability of cache replacement strategies in real-world complex environments but also highlights the potential of reinforcement learning mechanisms for structural optimization in caching systems.

Keywords: Dynamic cache management; policy network; hit rate optimization; intelligent scheduling

1. Introduction

In today's data-driven intelligent era, information systems are facing unprecedented pressure on data access. This is especially true in scenarios such as edge computing, the Internet of Things, and mobile networks. As a key mechanism to alleviate data access bottlenecks, the performance of caching systems directly affects overall system responsiveness and resource utilization. With the explosive growth of data volume and the high dynamism of user behavior, traditional cache replacement strategies are increasingly showing limitations. They struggle with hot data management, bursty requests, and non-stationary distributions [1], [2]. In resource-constrained environments, such as edge nodes and mobile devices, efficient and accurate content management within limited cache space has become a core challenge. It is essential for enhancing system intelligence and improving service quality.

Cache replacement strategies are critical scheduling components within caching systems. Their goal is to maximize cache hit rates, minimize response delays, and reduce system overhead during data replacement. In

previous studies, many strategies such as LRU (Least Recently Used), LFU (Least Frequently Used), and their variants have been widely applied in various caching architectures. However, these methods are typically based on static rules or heuristic algorithms [3]. They rely on fixed assumptions and lack adaptability to dynamic access patterns. For example, explosive growth in hot data, cyclical fluctuations in content popularity, and nonlinear changes in user interests often cause these strategies to fail or degrade in performance. Therefore, designing a cache replacement mechanism that can dynamically detect hotspot variations and respond intelligently is a key path toward smarter systems [4].

In recent years, reinforcement learning has emerged as a promising solution for intelligent cache scheduling. It is a sequential decision-making method based on agent-environment interaction [5]. Through reward mechanisms, it guides policy learning. This allows the system to explore optimal strategies without modeling access patterns explicitly. Over time, the system accumulates experience and improves its strategy. This learning paradigm is well-suited for dynamic environments. It shows strong adaptability and robustness, especially when facing unpredictable user request sequences. With reinforcement learning, caching systems can continuously optimize their behavior. Eventually, they evolve into intelligent systems that efficiently capture hot content and flexibly adjust replacement strategies [6].

Hotspot detection also plays a vital role in improving cache replacement effectiveness. In real-world applications, hot content often exhibits strong temporal and spatial locality. Its access frequency can change rapidly in short time spans. Accurate recognition and prediction of such trends can significantly enhance cache utilization. Traditional methods often rely on static thresholds or fixed-window statistics. These are insufficient to capture fine-grained behavior dynamics. By integrating reinforcement learning with hotspot detection mechanisms, it becomes possible to dynamically estimate content value. This grants the system adaptive insight and enables more user-aware and context-sensitive cache replacement decisions [7].

In scenarios such as intelligent networks, edge computing, and content delivery networks, caching systems face high-speed, low-latency, and high-concurrency demands. Under increasingly personalized user behaviors, improving service quality has become a critical issue. Reinforcement learning-based hotspot-aware cache replacement strategies enhance the system's ability to sense environmental changes. They continuously optimize replacement actions and support a shift from rule-driven to policy-driven caching. This paradigm shift promotes the development of intelligent caching systems. It also offers theoretical foundations and practical pathways for next-generation high-performance data scheduling. This direction holds great practical significance and broad application potential, particularly in building adaptive and low-latency data service infrastructures.

Beyond conventional reinforcement learning caching, recent cache-replacement research has increasingly combined predictive access modeling with policy learning. Competitive caching with machine-learned advice [8], multi-agent Deep-Q cache replacement for content delivery networks [9], Belady-inspired reinforcement learning with bypass and access-type analysis [10], and future-reference pattern learning for cache decisions [11] show that replacement policies can benefit from both global prediction and localized eviction control. These studies motivate the present hotspot-aware design, where access frequency, recency, content validity, and heat estimation are treated as policy inputs rather than isolated heuristic signals.

Broader intelligent-system studies further support this view. Transformer-based trend fusion for time-series forecasting [12] and interpretable graph-based anomaly detection for related-party transactions [13] illustrate how temporal dependencies and structural relationships can be modeled jointly for decision support. State-space temporal modeling for backend microservice anomaly detection [14], reinforcement learning-based elastic scaling optimization for microservices [15], failure detection with uncertainty and asymmetric risk optimization [16], and few-shot backend anomaly recognition with prior-enhanced representations [17] emphasize the need to combine state representation, uncertainty awareness, and adaptive policies in dynamic backend environments. Multi-scale temporal awareness in user-interest modeling [18] also provides a useful perspective for characterizing shifting content popularity in cache systems.

System-level representation methods provide additional methodological grounding for the proposed cache-control framework. Log semantic graph construction and CNN-based event anomaly detection [19], multi-view behavioral modeling for cloud resource prediction [20], graph neural network and large-language-model collaborative reasoning for enterprise ETL orchestration [21], distributed failure perception through operational-state representation learning [22], and constraint-aware virtual-machine resource matching [23] all highlight the value of integrating structural context, operational states, and resource constraints. These ideas align with the proposed policy network, which uses hotspot-aware state modeling and reward-driven replacement decisions to balance hit-rate improvement, latency reduction, and traffic control.

2. Proposed Approach

This study proposes a hotspot-aware dynamic cache replacement strategy based on reinforcement learning. The overall method structure revolves around the four core components of "state modeling - action design - reward definition - strategy update". First, to more accurately capture the dynamic behavior in the cache environment, the system state is modeled as a multidimensional feature vector, which contains indicators such as content access frequency, last access time, content validity period, and global heat estimation. Let the state of the i-th content in the current cache be $s_i = [f_i, t_i, \tau_i, h_i]$, where f represents the access frequency, t_i represents the time since the last access, τ_i is the remaining validity period of the content, and h_i is the global heat score. The overall system state is $S = \{s_1, s_2, .., s_n\}$, which is used to drive the state input of the policy network. The overall architecture of this model is shown in Figure 1.

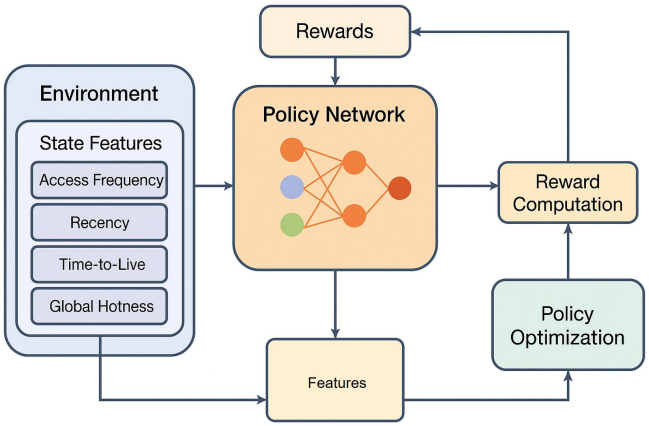


Figure 1. Overall model architecture

In terms of action design, each decision made by the reinforcement learning agent corresponds to a cache replacement action, specifically selecting an item from the current cache to replace. We define the action space A as the set of indices of the cache set, $A = \{1,2,..., n\}$, where n is the cache capacity. When a new content request arrives and the cache is full, the agent selects an item to replace based on the probability distribution $\pi(a|s; \theta)$ output by the policy network. This distribution is generated by the policy network based on the current state and represents the probability of replacing each cache item. The policy function is as follows:

$$\pi(a|s; \theta) = \frac{\exp(Q(s, a; \theta))}{\sum \exp(Q(s, a'; \theta))}$$

Where $Q(s, a; \theta)$ is the value evaluation function of the policy network for the state-action pair, and the parameter θ is the trainable weight of the network.

The reward function is designed to maximize cache hit rates and dynamically detect the value of hot content. For each content request, if the cache hits, the reward is set to a positive value, $r=+1$; otherwise, the

replacement fails, and the reward is set to a negative value, $r = -1$. Furthermore, a content popularity boosting reward, r_h, is introduced to enhance the ability to detect and capture sudden hot spots. The final instant reward function is:

$$r_t = \{1 + \lambda \cdot h_j, \text{ if hit on item j}$$

h_j represents the global popularity score of the requested content, and λ is an adjustable weight coefficient used to balance the basic hit reward and hotspot perception capability.

In the policy optimization process, this method uses a policy gradient-based optimization method to adjust the policy network parameters through the sampled state-action-reward sequence. The cumulative return is used as the objective function, which is defined as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

Where $\gamma \in (0,1)$ is a discount factor, which is used to balance long-term and short-term benefits. The update of policy parameters relies on the gradient ascent method, using the basic policy gradient formula in the REINFORCE algorithm as follows:

$$\nabla_\theta J(\theta) = E_\pi[\nabla_\theta \log \pi(a_t|s_t;\theta) \cdot G_t]$$

By continuously interacting with the environment and optimizing this objective function, the policy network autonomously learns optimal cache replacement behavior, maintaining efficient decision-making performance in a constantly changing access sequence. The overall strategy is guided by global heat modeling, using dynamic state as the basis for decision-making, and employing reinforcement learning mechanisms to achieve intelligent scheduling of the cache system.

3. Performance Evaluation

3.1 Dataset

This study employs the YouTube Cache Dataset to evaluate the proposed reinforcement learning-based cache replacement strategy. The dataset consists of real video request logs collected from YouTube edge servers in multiple regions. It includes millions of user request records, reflecting significant hotspot dynamics and diverse content types. Each entry contains fields such as request timestamp, content ID, video category, and request frequency. These features provide a comprehensive view of cache pressure and access distribution in real-world scenarios.

The dataset exhibits strong temporal correlation and non-stationary access patterns. Hot videos show intense fluctuations across different periods. This makes the dataset suitable for evaluating caching performance in scenarios involving bursty popular content and lifecycle-aware content management. Its large scale and multi-dimensional features also offer a rich sample space for state modeling and policy optimization in reinforcement learning.

In addition, the YouTube Cache Dataset is publicly available and realistic. It is widely used in caching research within edge computing and content delivery domains. The dataset supports reproducibility and comparative analysis. Its complex access behavior and long-tail distribution present both challenges and opportunities for intelligent caching strategies. It also helps validate the proposed method's applicability and robustness in real-world environments.

3.2 Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table1: Comparative experimental results

Model	Hit Rat	BMR	Latency Reduction	Traffic Reduction
Parrot [8]	72.6	0.142	18.3%	21.7%
DQN [9]	75.1	0.126	20.5%	24.0%
LRU-BaSE [10]	68.9	0.158	15.2%	18.1%
Stormbird [11]	76.4	0.121	22.6%	26.4%
Ours	79.8	0.107	25.9%	30.1%

As shown in the results of Table 1, the proposed hotspot-aware dynamic cache replacement strategy based on reinforcement learning outperforms existing mainstream methods across all evaluation metrics. In particular, the proposed method achieves a cache hit rate of 79.8 percent, which is significantly higher than that of traditional strategies such as LRU-BaSE at 68.9 percent and the representative reinforcement learning model DQN at 75.1 percent. This improvement indicates that the model can more effectively identify hot content in access patterns and make better decisions for cache retention and replacement.

For the Block Miss Ratio (BMR), the proposed method reduces it to 0.107, which is lower than other approaches. This suggests that the model has stronger stability and consistency, maintaining high hit performance even under rapid content updates and shifting request patterns. In highly non-stationary environments, traditional strategies often suffer from frequent misplacement due to their reliance on static rules. In contrast, the proposed method avoids this issue through dynamic policy learning.

In terms of latency, the proposed strategy also performs well on the Latency Reduction metric, achieving 25.9 percent. This is notably higher than Parrot at 18.3 percent and DQN at 20.5 percent. These results further confirm that reinforcement learning enables the model to make faster and more responsive scheduling decisions based on the current environment. As a result, the system delay caused by remote data access is significantly reduced. This advantage is especially important for latency-sensitive scenarios such as edge computing and content delivery networks.

Moreover, the proposed method achieves a 30.1 percent reduction in overall traffic, which is the best among all compared methods. This reflects its superior ability to manage high-frequency data accurately. Taken together, the results show that the proposed strategy not only excels in individual metrics but also maintains a high level of consistency in overall system performance, resource efficiency, and adaptability to dynamic environments. These findings validate the effectiveness and practicality of the reinforcement learning-based hotspot-aware mechanism for cache scheduling.

This paper also gives an analysis of the impact of learning rate on cache hit rate, and the experimental results are shown in Figure 2.

The experimental results in the figure show that different learning rate settings have a significant impact on cache hit rate. This indicates that the convergence speed and policy stability of the reinforcement learning strategy during training are largely controlled by the learning rate. Among the five learning rate settings shown, the value of 1e-3 yields the highest hit rate, reaching nearly 0.80. This suggests that it achieves a good balance between exploration and exploitation during policy optimization, enabling the model to learn effective rules for judging the value of cached content.

When the learning rate is too small, such as 1e-4, the hit rate is relatively low, reaching only 0.765. This result reflects that the learning process may suffer from slow convergence and insufficient updates. As a consequence, the model fails to capture the evolving trends of hotspot content promptly, reducing the

responsiveness of the caching system. In addition, local optima in the cache replacement policy may be difficult to overcome due to the limited update step.

On the other hand, when the learning rate is too large, such as 1e-2, the hit rate drops to the lowest point, only 0.742. An excessively large step size can cause violent oscillations in policy parameters during training. This results in instability or even divergence in the learned policy. The model then struggles to maintain consistent replacement behavior in dynamic environments, leading to frequent misplacement and missed hotspot content, which harms overall performance.

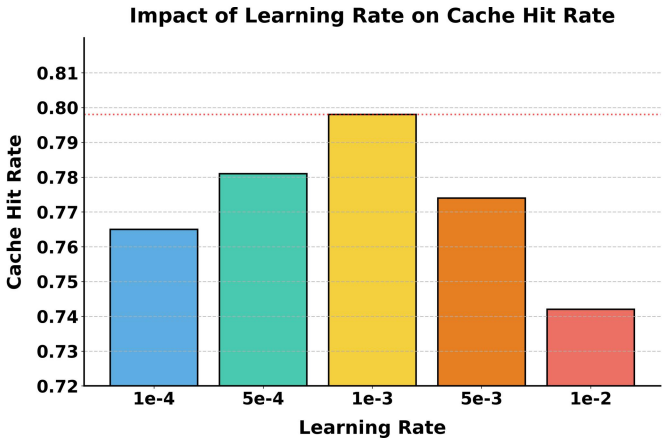


Figure 2. Analysis of the impact of learning rate on cache hit rate

In summary, the learning rate is a critical hyperparameter that directly affects the training outcomes of the reinforcement learning-based cache replacement strategy. A well-chosen learning rate, such as 1e-3, can accelerate policy convergence and improve the model's generalization ability in complex access sequences. This experiment confirms the importance of hyperparameter tuning in reinforcement learning methods and provides practical guidance for model deployment and transfer.

This paper also presents an evaluation of the strategy adaptability under cache capacity changes, and the experimental results are shown in Figure 3.

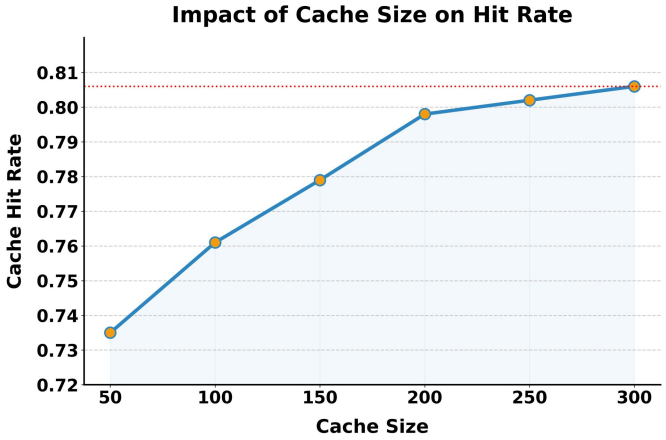


Figure 3. Evaluating policy adaptability under cache capacity changes

The experimental results in the figure show that as the cache size gradually increases, the cache hit rate rises steadily and significantly. This indicates that the proposed reinforcement learning strategy has good adaptability to capacity changes. When the cache size increases from 50 to 300, the hit rate improves from approximately 0.735 to 0.806. This increase of over 7 percent demonstrates that the model effectively utilizes the available space. It learns better replacement strategies to enhance caching efficiency.

When the cache size is small, the system is constrained by limited space. The policy must frequently replace content and requires a strong capability to identify hotspot data. In this case, the dynamic policy of the reinforcement learning model tends to retain high-value content. This prevents hit loss caused by blind replacement. As a result, the proposed strategy outperforms traditional methods in resource-constrained environments. This flexible policy generation mechanism is particularly valuable in scenarios with limited cache, such as edge computing and mobile devices.

As the cache size increases to a moderate range, for example between 150 and 250, the hit rate continues to improve, but at a slower rate. This suggests that with more storage space, the model can better establish statistical patterns of content usage and access probability. This leads to more stable and reliable replacement decisions. At this stage, performance gains are not only due to increased capacity but also to the model's ability to learn content importance and understand global system states.

Finally, when the cache size reaches 300, the hit rate approaches 0.81, and policy performance tends to saturate. These experimental results confirm that the proposed reinforcement learning-based replacement strategy maintains strong adaptability across different cache sizes. It demonstrates the ability to learn compact strategies under limited capacity and to express richer policies in larger environments. This provides both theoretical support and practical evidence for building adaptive and generalizable caching systems.

4. Conclusion

This study focuses on the challenges of hotspot awareness and policy adaptability in dynamic cache management. It proposes a reinforcement learning-based cache replacement strategy. By modeling multi-dimensional states such as access frequency, temporal features, and content popularity, the method enables intelligent decision-making through a policy network. Unlike traditional heuristic algorithms that rely on fixed rules, the proposed strategy continuously optimizes replacement decisions in highly dynamic and non-stationary environments. This leads to significant improvements in cache hit rate, response speed, and traffic reduction. The model demonstrates strong robustness and generalization. With fine-grained state modeling and a carefully designed reward mechanism, it effectively captures sudden hotspot events and content trend shifts, offering a viable path for building intelligent and adaptive caching systems.

Experimental results verify the advantages of the proposed method from multiple perspectives. Sensitivity analyses under varying cache sizes, learning rate settings, and access patterns confirm its performance stability and resource efficiency. Compared with several representative baseline strategies, the proposed method achieves noticeable improvements in key metrics such as hit rate, system latency, and traffic control. These results validate the feasibility and potential of reinforcement learning frameworks for cache scheduling. The method is also highly scalable and suitable for various application scenarios, including edge computing, content delivery networks (CDNs), and on-device caching. It helps address the challenges of large-scale, high-concurrency, and low-latency data services.

At the application level, this research lays a technical foundation for developing cache systems with long-term autonomous learning and continuous evolution. It addresses the limitations of traditional systems that suffer from rigid rules and delayed responses to shifting hotspots. With reinforcement learning, cache replacement no longer depends on static parameters. Instead, decisions are dynamically generated based on environmental states. This enables stronger adaptability to changes in user behavior and content popularity. The shift from rule-driven to policy-driven paradigms will drive the development of caching systems toward greater intelligence, customization, and efficiency.

Looking forward, the proposed approach can be extended to more complex network architectures. These include cross-node cooperative caching, multimodal content-aware scheduling, and joint optimization of transmission and caching. It is especially relevant in next-generation edge intelligence, vehicular networks, and satellite communication systems, where resources are limited and environments are highly dynamic. By integrating large-model-driven content understanding and prediction mechanisms, the strategy may achieve

better foresight and interpretability. Future work may explore its stability and deployment efficiency in larger-scale, higher-dimensional environments. This will provide theoretical support and practical solutions for advancing next-generation intelligent network infrastructures.

References

[1] H. Zhou, M. Erol-Kantarci, and H. V. Poor, "Learning from peers: Deep transfer reinforcement learning for joint radio and cache resource allocation in 5G RAN slicing," IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 4, pp. 1925-1941, 2022.

[2] S. Liu, C. Zheng, Y. Huang, et al., "Distributed reinforcement learning for privacy-preserving dynamic edge caching," IEEE Journal on Selected Areas in Communications, vol. 40, no. 3, pp. 749-760, 2022.

[3] Q. Wu, Y. Zhao, Q. Fan, et al., "Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning," IEEE Journal of Selected Topics in Signal Processing, vol. 17, no. 1, pp. 66-81, 2022.

[4] Y. Zhou, F. Wang, Z. Shi, et al., "An end-to-end automatic cache replacement policy using deep reinforcement learning," in Proc. International Conference on Automated Planning and Scheduling, vol. 32, 2022, pp. 537-545.

[5] S. Chen, Z. Yao, X. Jiang, et al., "Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks," IEEE Transactions on Communications, vol. 69, no. 4, pp. 2441-2456, 2020.

[6] X. Xu, F. Wu, M. Bilal, X. Xia, W. Dou, L. Yao et al., "XRL-SHAP-Cache: An Explainable Reinforcement Learning Approach for Intelligent Edge Service Caching in Content Delivery Networks," Science China Information Sciences, vol. 67, 2024.

[7] S. Weerasinghe, A. Zaslavsky, S. W. Loke, A. Abken and A. Hassani, "Reinforcement Learning Based Approaches to Adaptive Context Caching in Distributed Context Management Systems," arXiv preprint, arXiv:2212.11709, 2022.

[8] T. Lykouris and S. Vassilvitskii, "Competitive caching with machine learned advice," Journal of the ACM, vol. 68, no. 4, pp. 1-25, 2021.

[9] J. K. Dassanayake, M. Wang, M. Z. Hameed, et al., "Multi-agent Deep-Q network-based cache replacement policy for content delivery networks," Future Internet, vol. 16, no. 8, p. 292, 2024.

[10] H. J. Yoo, J. H. Kim, and T. H. Han, "RL-based cache replacement: A modern interpretation of Belady's algorithm with bypass mechanism and access type analysis," IEEE Access, vol. 11, pp. 145238-145253, 2023.

[11] H. Choi and S. Park, "Learning future reference patterns for efficient cache replacement decisions," IEEE Access, vol. 10, pp. 25922-25934, 2022.

[12] S. Sun, "TrendFormer: Two-stage trend fusion for financial time series forecasting using Transformers," 2024.

[13] C. Chen, "Interpretable graph-based anomaly detection for related-party transactions in auditing systems," 2024.

[14] Y. Xue, "State-space temporal modeling and feature representation learning for anomaly detection in backend microservice systems," 2024.

[15] X. Li and Z. Song, "Reinforcement learning-based elastic scaling policy optimization for microservices," 2024.

[16] Y. Liu, Z. Zhang, and S. Liang, "Intelligent backend failure detection with uncertainty quantification and asymmetric risk optimization," 2024.

[17] Y. Sun, "Prior enhanced representation learning and adaptive recognition method for backend anomaly detection under weakly labeled and few-shot conditions," 2024.

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

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

[20] X. Zhang and Z. Fang, "Deep learning-based multi-view behavioral modeling and trend regression for cloud resource prediction," 2024.

[21] Y. Zhang and L. Steven, "Automatic orchestration of enterprise ETL processes via graph neural network and large language model collaborative reasoning," 2024.

[22] W. Huang, R. Zhan, and Z. Huang, "Data-driven failure perception for distributed systems through operational state representation learning," 2024.

[23] J. Kou and E. Pei, "Constraint-aware resource matching for virtual machine placement in cloud environments," 2024.