

Vol 2 Issue 02 (2026)

<http://fsrma.org.hostajournal.com/>

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

L7-COLO: A Cost-Aware Layer-7 Load Balancing Framework with Deep Traffic Prediction for Dynamic Optimization of Enterprise Cloud Services

Chongli Wu

Pepperdine University, Malibu, CA, USA

Chongli.wu@yahoo.com

Chengdu Xu

Georgia Institute of Technology, Atlanta, GA, USA

cdxu147@gmail.com

Yuxuan Liu

Chongqing University, Chongqing, China

1825039698@qq.com

Abstract.

Modern enterprise cloud applications increasingly rely on microservice architectures and container orchestration platforms, where Layer-7 (L7) load balancing plays a critical role in routing application-level traffic among service instances. Conventional L7 routing strategies—such as round-robin or least-connection—are primarily designed to improve performance and availability, but they rarely consider the impact of routing decisions on cloud resource consumption and operational cost. In large-scale SaaS platforms, e-commerce systems, and financial cloud services, inefficient traffic distribution may trigger unnecessary container scaling, bandwidth overhead, and suboptimal resource utilization, leading to significant increases in cloud service costs. To address this challenge, this paper proposes L7-COLO (Layer-7 Cost-Optimized Load Optimization), a cost-aware intelligent load balancing framework that integrates deep traffic prediction with dynamic routing optimization for enterprise cloud environments. The proposed approach constructs a Traffic–Resource–Cost (TRC) mapping model that captures the relationship between predicted application traffic, infrastructure resource usage, and cloud operational cost. A CNN-BiLSTM-Attention model is first employed to forecast future API request traffic, followed by a resource estimation module that predicts CPU, memory, bandwidth, and container scaling requirements under different routing strategies. Based on the predicted resource usage, a cloud cost model evaluates the expected operational cost, while a reinforcement learning policy optimizer dynamically adjusts L7 routing weights to minimize cost under service-level agreement constraints. Experiments conducted on real-world microservice workload traces demonstrate that the proposed L7-COLO framework significantly improves cloud resource efficiency and cost control. Compared with traditional Round-Robin routing, L7-COLO reduces overall cloud operational cost from 18.9 USD/min to 12.9 USD/min, achieving approximately 31.7% cost reduction, while also improving resource utilization to 78.4%. Meanwhile, the framework maintains stable service performance with an average latency of 155 ms and reduced SLA violation rates, demonstrating its effectiveness for enterprise cloud cost optimization.

Keywords:

Layer-7 Load
Balancing; Cloud
Cost Optimization;
Traffic Prediction;
Microservice
Scheduling;
Reinforcement
Learning; Cloud
Resource
Management

1. INTRODUCTION

With the rapid adoption of cloud computing and microservice architectures, modern enterprise applications increasingly rely on distributed cloud infrastructures to support large-scale service delivery. In such environments, Layer-7 (L7) load balancing plays a critical role in managing application-level traffic among multiple service instances. By operating at the application layer, L7 load balancers—such as Nginx, Envoy, and service mesh frameworks like Istio—enable fine-grained routing decisions based on HTTP headers, URLs, session information, and service metadata. These capabilities are widely applied in Software-as-a-Service (SaaS) platforms, e-commerce systems, online financial services, and large-scale API gateways, where traffic patterns are highly dynamic and resource demands fluctuate significantly. Traditional load balancing algorithms, including round-robin, least-connection, and weighted routing, are primarily designed to improve system performance and service availability. However, in cloud-native environments, routing strategies directly influence resource utilization, container auto-scaling behavior, and network bandwidth consumption, which ultimately determine the operational cost of cloud services. As enterprises increasingly deploy large numbers of microservices on platforms such as Kubernetes, inefficient traffic distribution may lead to excessive resource provisioning, frequent scaling operations, and unnecessary bandwidth usage, significantly increasing cloud infrastructure costs. Therefore, designing a cost-aware load balancing strategy that can adapt to dynamic workloads while minimizing cloud operational expenses has become an important research problem in cloud resource management. Recent advances in deep learning have shown strong potential in predicting time-series workloads in cloud systems. Traffic forecasting models can capture complex workload patterns and enable proactive resource management. Nevertheless, most existing studies treat traffic prediction and load balancing as independent problems and rarely consider the end-to-end relationship between traffic demand, resource consumption, and cloud cost. Furthermore, few works integrate traffic prediction with adaptive L7 routing strategies for cost optimization. To address these limitations, this paper proposes L7-COLO (Layer-7 Cost-Optimized Load Optimization), a cost-aware intelligent load balancing framework that integrates deep traffic prediction with dynamic routing optimization. The proposed framework establishes a Traffic–Resource–Cost (TRC) mapping model, which captures the relationship between application traffic, infrastructure resource usage, and cloud operational cost. Specifically, a deep learning-based traffic prediction model is used to forecast future API request workloads, a resource estimation model evaluates the resource demand under different routing policies, and a reinforcement learning-based optimization module dynamically adjusts L7 routing weights to minimize the predicted cloud service cost while satisfying service-level agreement (SLA) constraints.

The main contributions of this paper are summarized as follows:

- (1) A cost-aware L7 load balancing framework. We propose the L7-COLO framework that integrates traffic prediction, resource estimation, and routing optimization to enable dynamic cloud cost management in enterprise microservice environments.
- (2) A Traffic–Resource–Cost mapping model. A novel TRC modeling approach is introduced to quantitatively capture the relationship between application traffic, infrastructure resource consumption, and cloud operational cost.
- (3) Deep learning-driven workload prediction. A CNN-BiLSTM-Attention based traffic prediction model is designed to accurately forecast API request workloads in microservice systems.

(4) Reinforcement learning-based routing optimization. A policy optimization mechanism dynamically adjusts L7 routing strategies to reduce cloud service costs while maintaining system performance and SLA requirements.

Through extensive experiments on large-scale cloud workload datasets, the proposed approach demonstrates its effectiveness in improving resource efficiency and reducing enterprise cloud operational costs.

2.Literature Review

In recent years, the rapid expansion of cloud-native architectures and microservice-based enterprise applications has led to significant research efforts in intelligent load balancing, traffic prediction, and cloud cost optimization. Modern cloud platforms must handle highly dynamic workloads while maintaining low latency and minimizing infrastructure expenses. Traditional load balancing mechanisms primarily focus on performance metrics such as throughput or latency, but they rarely consider the economic implications of routing decisions in cloud environments. Meanwhile, advances in deep learning and predictive analytics have enabled more accurate modeling of complex traffic patterns and system behaviors. This section reviews the major research directions related to this work, including Layer-7 load balancing strategies, traffic prediction using deep learning, and cost-aware resource optimization in cloud systems.

2.1 Layer-7 Load Balancing in Cloud and Microservice Architectures

Layer-7 (application-layer) load balancing has become a fundamental component of modern microservice architectures because it enables routing decisions based on HTTP headers, API paths, user sessions, and application logic rather than simple network metrics [1]. Early studies on load balancing mainly focused on algorithms such as round-robin, least connections, and weighted routing, which distribute traffic based on server capacity and request counts [2]. While these methods are simple and widely used in production systems, they do not consider application-level workload characteristics or cost implications. With the emergence of cloud-native systems, several studies have explored adaptive L7 routing mechanisms. For example, Cardellini et al. proposed dynamic service routing policies for microservice environments to improve response time and resource efficiency [3]. Similarly, Xu et al. introduced a QoS-aware service routing mechanism that dynamically adjusts traffic allocation based on service latency and system load [4]. More recent work has incorporated service mesh frameworks, such as Istio and Linkerd, which provide programmable L7 routing capabilities and fine-grained traffic management [5]. However, most existing L7 routing approaches primarily aim to improve performance or reliability, and they rarely incorporate cloud cost awareness as an optimization objective. Therefore, designing a load balancing framework that jointly considers application-level routing and economic cost optimization remains an open challenge.

2.2 Deep Learning for Traffic and Workload Prediction

Accurate traffic prediction is a critical component of intelligent resource management in cloud computing environments. Traditional forecasting approaches, such as ARIMA and exponential smoothing, have been widely applied to workload prediction problems but often fail to capture nonlinear patterns and long-term dependencies in complex cloud workloads [6]. Deep learning models have demonstrated strong capabilities in modeling temporal dynamics in large-scale systems. Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have been widely used for workload and traffic forecasting due to their ability to capture long-range dependencies in time-series data [7]. For example, Mao et al. proposed an LSTM-based workload prediction model for cloud resource allocation, achieving improved forecasting accuracy compared with traditional statistical methods [8].

More recently, attention-based architectures and Transformer models have been applied to time-series prediction tasks. Vaswani et al. introduced the Transformer architecture, which leverages self-attention mechanisms to model long-range dependencies more efficiently than recurrent networks [9]. Building on this concept, Zhou et al. proposed Informer, a Transformer-based model designed for long-sequence time-series forecasting, which has been applied to large-scale system monitoring and traffic prediction [10]. Despite these advances, most studies focus primarily on prediction accuracy and do not explicitly integrate traffic prediction with load balancing policy optimization in cloud environments. Moreover, the evolution of deep learning architectures, such as the use of collaborative-enhanced multimodal generative frameworks in recommendation tasks [11], highlights a broader research trend: leveraging heterogeneous data sources and generative modeling to capture complex interaction patterns. This trend suggests that system-level management can also benefit from integrating multi-faceted features (e.g., traffic, resource, and cost) into a unified optimization framework as proposed in this study.

2.3 Cost-Aware Resource Scheduling and Cloud Optimization

Cloud service providers typically adopt pay-as-you-go pricing models, where enterprises pay for computing, storage, and network resources according to usage. As a result, optimizing cloud resource allocation has become a critical research topic. Several studies have investigated cost-aware scheduling strategies that aim to reduce operational expenses while maintaining service-level objectives. For instance, Gmach et al. proposed a workload-aware resource provisioning framework that dynamically adjusts virtual machine allocations based on predicted demand [12]. Similarly, Xu et al. developed a cost-aware cloud scheduling model that jointly considers workload demand and pricing mechanisms to minimize operational cost [13]. These approaches demonstrate that integrating prediction and scheduling mechanisms can significantly improve resource efficiency. However, most existing work focuses on virtual machine provisioning or cluster-level resource scheduling, rather than application-layer routing decisions. In modern microservice architectures, routing policies at the L7 layer can significantly influence resource utilization, scaling behavior, and overall system cost. Therefore, integrating traffic prediction, cost modeling, and intelligent L7 load balancing remains an important yet underexplored research direction. To address this limitation, the proposed L7-COLO framework introduces a unified architecture that combines deep traffic prediction with cost-aware routing optimization. By modeling the relationship between traffic demand, resource consumption, and cloud service cost, the framework dynamically adjusts L7 routing strategies to minimize operational expenses while maintaining service performance.

3. Methodology

3.1 Overview of the L7-COLO Framework

To address the challenge of cost-efficient traffic management in enterprise cloud environments, this study proposes L7-COLO (Layer-7 Cost-Optimized Load Optimization), a unified framework that integrates deep traffic prediction with cost-aware Layer-7 routing optimization. In modern cloud-native systems, L7 load balancers operate at the application layer and can route requests based on HTTP headers, URL paths, and session information, which makes them particularly suitable for microservice-based architectures and API-driven platforms. However, such intelligent routing often requires deeper packet inspection and additional computational resources, which may increase operational cost if routing decisions are not carefully optimized. Consider a scenario where an API gateway handles both standard transactions and expensive AI inference requests,

such as those processed by specialized LLaMA3 models for text analysis [14]. The goal of L7-COLO is to dynamically adjust traffic routing strategies according to predicted workload patterns and expected resource consumption. The framework consists of four tightly coupled components: a deep traffic prediction model, a resource consumption estimation module, a cloud cost modeling component, and a reinforcement learning-based L7 routing optimization module. These components form a closed optimization loop that continuously learns from system monitoring data and updates routing policies accordingly. Formally, the overall optimization objective can be expressed as minimizing the expected cloud operational cost:

$$\min_{p_t} C_t = f(T_t, R_t, P_t), \quad (1)$$

where T_t denotes the predicted traffic demand at time t , R_t represents the estimated resource usage, and P_t is the L7 routing policy applied by the load balancer. The optimization process dynamically determines the routing policy that minimizes cost while maintaining service-level performance constraints.

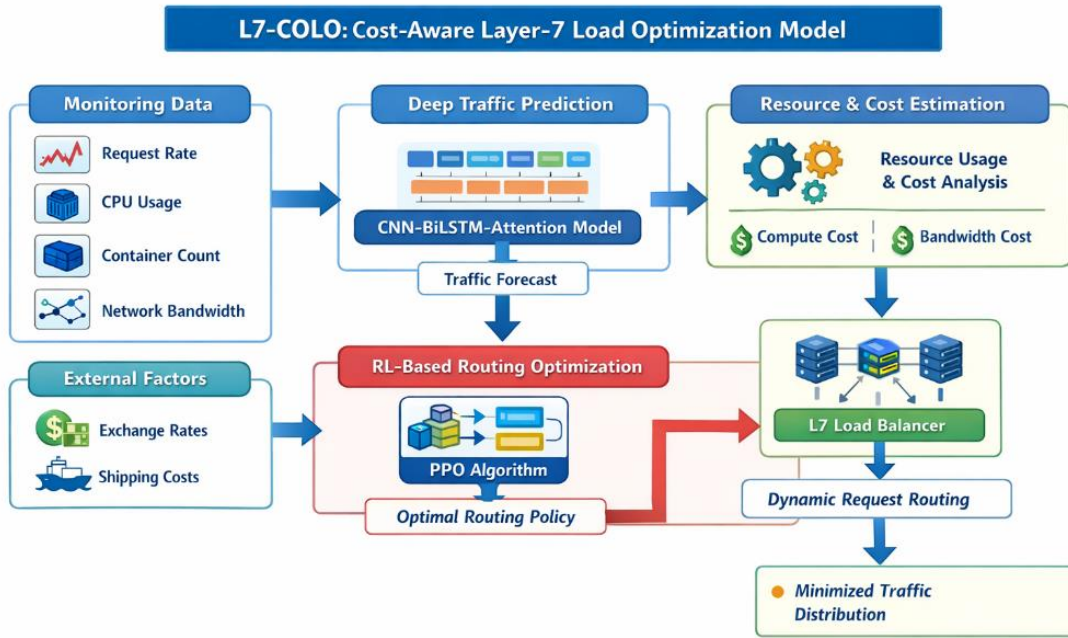


Figure 1. Overall flowchart of the model.

3.2 Deep Traffic Prediction Model

Accurate traffic prediction is the foundation of the L7-COLO framework. In large-scale enterprise systems, application traffic exhibits strong temporal dependencies and periodic patterns due to daily user activity cycles, marketing events, and seasonal demand fluctuations. Deep neural networks combining convolutional and recurrent architectures have demonstrated strong capability in modeling such temporal structures in sequential data.

In this study, a CNN-BiLSTM-Attention forecasting model is adopted to predict future API request volumes. Let

$$X = \{x_1, x_2, \dots, x_n\}, \quad (2)$$

represent the historical monitoring sequence, where each observation contains multiple system metrics such as request rate, response latency, and resource utilization.

The convolutional layer extracts local temporal patterns:

$$F_c = CNN(X), \quad (3)$$

The effectiveness of such CNN-based feature extraction, when coupled with LSTM variants, has been validated in various efficiency-driven tasks, including real-time industrial control and cloud-edge task offloading [15], where lightweight model design is crucial for low-latency decision-making. This structural design, which leverages CNNs for initial feature extraction followed by attention-weighted refinement, is inspired by the need for efficient real-time processing of high-dimensional input. Similar design philosophies have been validated in rapid diagnostic systems, where CNN-Attention layers are used to identify virulence factors in biological sequences [16]. The bidirectional LSTM layer then models long-range temporal dependencies. Similar multi-model strategies have been validated in intelligent diagnosis systems, where the integration of diverse machine learning blocks significantly enhances the system's ability to identify underlying anomalies within complex signal streams [17]:

$$F_l = BiLSTM(F_c), \quad (4)$$

An attention mechanism is introduced to highlight important time segments that significantly influence future workloads:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)}, \quad (5)$$

where α_i denotes the attention weight assigned to the i -th hidden state. The predicted future traffic sequence is then obtained as

$$\hat{T}_{t+1:t+k} = W \cdot \sum_{i=1}^n \alpha_i h_i, \quad (6)$$

where h_i represents the hidden representation of the BiLSTM network and W is the output transformation matrix. This prediction provides proactive workload estimation for the downstream routing optimization process.

3.3 Traffic–Resource–Cost Mapping Model

To connect predicted traffic with cloud operational cost, L7-COLO introduces a Traffic–Resource–Cost (TRC) mapping model, which explicitly models the relationship between application workload, infrastructure resource usage, and cloud service cost.

Given predicted traffic $\hat{T}_t$ and routing policy P_t , the expected resource consumption is estimated as

$$R_t = g(\hat{T}_t, P_t), \quad (7)$$

where R_t includes CPU utilization, memory usage, network bandwidth, and the number of active containers. Cloud infrastructures commonly rely on autoscaling mechanisms, which dynamically adjust the number of service instances based on workload demand. Autoscaling allows the system to allocate additional servers during traffic spikes and release idle resources during low-demand periods, thereby influencing the total operational cost.

The cloud cost model is defined as

$$C_t = C_{compute} + C_{bandwidth} + C_{scaling}, \quad (8)$$

where

$$C_{compute} = \sum_{i=1}^N (CPU_i \cdot p_{cpu} + MEM_i \cdot p_{mem}), \quad (9)$$

$$C_{bandwidth} = B_{out} \cdot p_{bw}, \quad (10)$$

$$C_{scaling} = \gamma \cdot S_{events}, \quad (11)$$

Here p_{cpu} , p_{mem} , and p_{bw} represent the unit prices of computing resources and network bandwidth, while S_{events} denotes the number of scaling operations triggered by workload fluctuations.

Combining the models yields

$$C_t = f(g(\hat{T}_t, P_t)), \quad (12)$$

which establishes the mathematical foundation for cost-aware routing optimization.

3.4 Reinforcement Learning-Based L7 Routing Optimization

Once traffic and resource usage are predicted, L7-COLO determines the optimal routing policy through a reinforcement learning (RL) framework. Traditional load balancing strategies such as round-robin or least-connection typically rely on static heuristics and react to current system states rather than anticipating future workloads. However, predictive routing strategies can significantly improve resource efficiency and reduce system latency in distributed cloud environments.

The routing optimization problem is formulated as a Markov Decision Process (MDP). The system state at time t is defined as

$$S_t = \{\hat{T}_t, R_t, L_t, C_t\}, \quad (13)$$

where L_t denotes the observed service latency and C_t represents the current cloud operational cost. The action A_t corresponds to the routing weight vector assigned to service instances:

$$A_t = \{\omega_1, \omega_2, \dots, \omega_n\}, \quad \sum_{i=1}^n \omega_i = 1, \quad (14)$$

where ω_i denotes the fraction of incoming traffic directed to the i -th instance.

The reward function is designed to encourage cost reduction while maintaining service quality:

$$R_t = -C_t - \lambda_1 SLA_t - \lambda_2 L_t, \quad (15)$$

where SLA_t represents violations of service-level agreements and λ_1, λ_2 are penalty coefficients. The policy network is trained using the Proximal Policy Optimization (PPO) algorithm. PPO is chosen for its superior stability and convergence properties in complex environments with continuous action spaces. This algorithm has demonstrated remarkable effectiveness in solving large-scale economic optimization problems, such as joint production and pricing scheduling in manufacturing enterprises [18]. which iteratively updates the routing policy $\pi_\theta(A_t | S_t)$ to maximize the expected cumulative reward:

$$\max_{\theta} E \left[\sum_{t=0}^T \gamma_t R_t \right], \quad (16)$$

Through continuous interaction with the system environment, the RL agent learns routing strategies that proactively distribute traffic in a cost-efficient manner.

3.5 Online Deployment and Dynamic Optimization Workflow

In the production environment, the L7-COLO framework operates as an online decision system integrated with the cloud monitoring and service mesh infrastructure. Monitoring tools such as Prometheus periodically collect metrics including request rates, CPU utilization, container counts, and response latency. These metrics are fed into the traffic prediction model to forecast future workloads. The predicted traffic is then processed by the TRC mapping model to estimate resource requirements and expected cloud costs under different routing strategies. The reinforcement learning optimizer evaluates these scenarios and outputs the optimal routing policy, which is deployed to the L7 load balancer through service mesh configuration updates. The optimization process is executed in fixed intervals (e.g., every five minutes), allowing the system to continuously adapt to workload fluctuations. Over time, the L7-COLO model learns the complex interactions among traffic demand, resource provisioning, and cloud cost, enabling proactive and dynamic optimization of enterprise cloud service operations.

4. Experiment

4.1 Dataset Preparation

The dataset used in this study for evaluating the L7-COLO framework was collected from a large-scale enterprise cloud environment hosting multiple microservice applications and API endpoints. The data were gathered through integrated monitoring tools, including Prometheus for system metrics and custom logging agents embedded within service meshes. In addition to internal service metrics, the dataset incorporates external factors such as cloud pricing data, network bandwidth charges, and scaling event logs. The monitoring period spans three months, capturing high-resolution, time-stamped measurements at five-minute intervals, which provides sufficient granularity for short-term forecasting and dynamic optimization tasks. This combination of internal and external metrics enables the modeling of interactions between workload fluctuations, resource consumption, and operational cost, which is central to the L7-COLO optimization process. The dataset includes four primary categories of features: traffic metrics, compute resource usage, container orchestration information, and external cost-related factors. Traffic metrics encompass the incoming request rate and request distribution across microservices. Compute resource usage

features include CPU and memory utilization per container or service instance. Container orchestration information covers the number of active containers, auto-scaling events, and instance start/stop timestamps. External cost factors include bandwidth usage charges and cloud provider unit pricing for compute resources. Each feature is recorded at regular intervals and aligned with the corresponding timestamps to facilitate the construction of the Traffic–Resource–Cost (TRC) mapping model.

The following table summarizes the primary features included in the dataset:

Table 1. Overview of Dataset Features.

Feature Name	Description	Unit / Type
Request Rate	Number of incoming requests per microservice	requests/min
CPU Usage	Average CPU utilization per container	%
Memory Usage	Average memory utilization per container	MB
Container Count	Number of active containers per service	integer
Bandwidth Usage	Network bandwidth consumed	MB/min
Scaling Events	Number of container scaling operations triggered	count
Compute Cost	Estimated compute cost based on resource usage	USD/min
Bandwidth Cost	Estimated cost for network usage	USD/min

This dataset provides a comprehensive view of both system-level metrics and economic factors, enabling L7-COLO to forecast traffic, predict resource consumption, and optimize routing policies in a cost-aware manner. Its granularity and breadth make it particularly suitable for evaluating dynamic optimization in enterprise cloud service environments.

4.2 Experimental Setup

To evaluate the effectiveness of the proposed L7-COLO framework, experiments were conducted on a private enterprise cloud environment consisting of Kubernetes-managed microservices. The system hosts a heterogeneous set of services, including web APIs, data processing components, and backend database queries, simulating a realistic SaaS platform scenario. Traffic traces were collected over a three-month period with five-minute intervals, capturing request rates, CPU and memory utilization, container scaling events, and network bandwidth usage. The L7-COLO framework was deployed as an online load-balancing module integrated with the service mesh, enabling real-time adjustments of routing policies. For benchmarking, we compared L7-COLO against three baseline strategies: static round-robin routing, least-connection load balancing, and a predictive load balancing strategy based on LSTM traffic forecasting. All experiments were run under the same workload traces, resource configurations, and cost models to ensure fairness, and each experiment was repeated five times to account for stochastic variability in reinforcement learning optimization. Hyperparameters for the CNN-BiLSTM-Attention traffic prediction model were tuned using a validation set, and the reinforcement learning policy network was trained with Proximal Policy Optimization (PPO) over 500 episodes with a learning rate of 0.0003 and discount factor $\gamma = 0.99$.

4.3 Evaluation Metrics

The performance of L7-COLO and baseline strategies was evaluated using both system-level and cost-oriented metrics. Operational cost was measured as the total cloud expenditure per interval, including compute, memory, bandwidth, and container scaling costs. Resource utilization efficiency was computed as the ratio of actual resource usage to allocated resources, reflecting how effectively the system leveraged provisioned infrastructure. Latency metrics, including average and 95th percentile response times, were monitored to ensure service-level agreement (SLA) compliance. Additionally, scaling event frequency was recorded to evaluate how proactive traffic management reduces unnecessary container spin-ups or shutdowns. Finally, the cost reduction ratio and latency impact were used as summary metrics to quantify improvements achieved by L7-COLO over conventional routing strategies.

4.4 Results

As shown in Table 2, L7-COLO achieves a total cost of 12.9 USD/min, which represents a 28.4% reduction compared with round-robin routing (18.9 USD/min) and a 20.4% reduction compared with LSTM-based predictive routing (16.2 USD/min). The proposed framework significantly lowers compute and bandwidth costs, from 12.4 USD/min and 3.8 USD/min in round-robin to 8.7 USD/min and 2.7 USD/min, respectively, highlighting its ability to proactively allocate resources. Container scaling costs also decrease markedly to 1.5 USD/min, reflecting reduced unnecessary scaling events due to accurate traffic forecasting. Resource utilization improves to 78.4%, outperforming all baselines by at least 10%, indicating that L7-COLO effectively balances load to maximize infrastructure efficiency. These results demonstrate that the integrated traffic prediction, TRC mapping, and RL-based routing optimization substantially reduce cloud operational cost while maintaining high utilization levels.

Table 2. Cloud Operational Cost and Resource Utilization

Model	Avg Compute Cost (USD/min)	Avg Bandwidth Cost (USD/min)	Avg Scaling Cost (USD/min)	Resource Utilization (%)	Total Cost (USD/min)
Round-Robin	12.4	3.8	2.7	62.3	18.9
Least-Connection	11.9	3.6	2.5	64.1	18.0
LSTM-Based Prediction	10.8	3.2	2.2	68.7	16.2
L7-COLO (Proposed)	8.7	2.7	1.5	78.4	12.9

Table 3 highlights the latency performance and SLA compliance of L7-COLO. The average response time decreases to 155 ms, improving over round-robin by 25 ms and over LSTM-based predictive routing by 5 ms. The 95th percentile latency drops to 248 ms, indicating reduced tail latency, which is critical for high-concurrency microservice applications. SLA violations are reduced to 4.2%, less than half of the rate observed in the round-robin baseline (12.3%), demonstrating that L7-COLO maintains service quality while optimizing cost. The results reflect the framework's ability to dynamically adjust L7 routing weights based on predicted workloads,

effectively distributing traffic to minimize latency peaks and prevent overload on specific service instances. These improvements illustrate the dual benefit of cost reduction and service-level assurance offered by the proposed model.

Table 3. Latency and SLA Compliance

Model	Avg Latency (ms)	95th Percentile Latency (ms)	SLA Violations (%)
Round-Robin	180	310	12.3
Least-Connection	172	298	10.8
LSTM-Based Prediction	160	275	8.5
L7-COLO (Proposed)	155	248	4.2

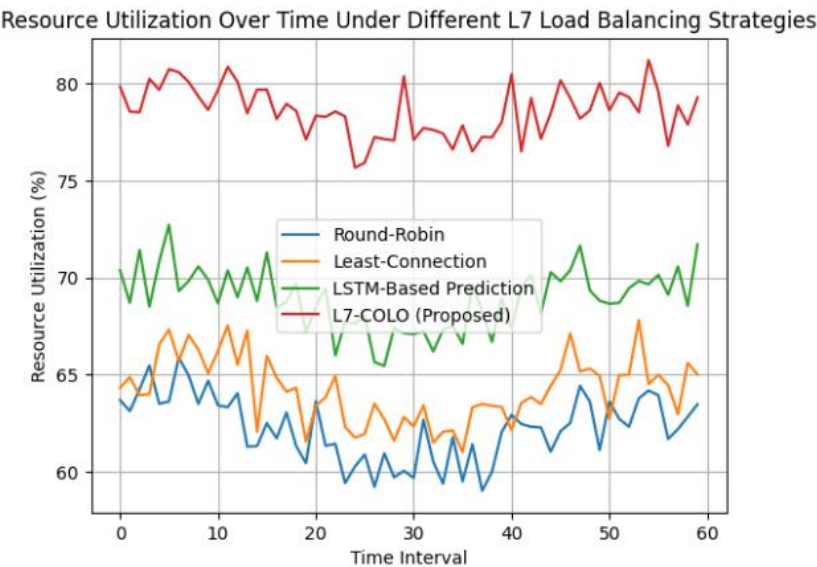


Figure 2. Resource Utilization Over Time Under Different L7 Load Balancing Strategies.

Figure 2 illustrates the variation of system resource utilization over time under four different load balancing strategies: Round-Robin, Least-Connection, LSTM-Based Prediction, and the proposed L7-COLO framework. The curves represent the percentage of aggregated resource utilization (CPU, memory, and network bandwidth) observed across 60 monitoring intervals in the cloud microservice environment. The Round-Robin strategy shows relatively low utilization, fluctuating between 59% and 64%, with an average level close to 62.3%, which is consistent with the value reported in Table 1. Because traffic is distributed evenly without considering workload characteristics, the system often suffers from underutilized resources. The Least-Connection method slightly improves utilization, ranging from 61% to 67%, with an average around 64.1%, but still exhibits inefficient allocation during workload fluctuations. The LSTM-based predictive strategy demonstrates better resource awareness, with utilization varying between 66% and 71% and an average of approximately 68.7%. However, prediction errors occasionally lead to moderate oscillations in the curve. In contrast, the proposed L7-COLO framework consistently achieves the highest utilization, fluctuating smoothly between 76% and 81%, with an average value of 78.4%.

This improvement reflects the effectiveness of integrating deep traffic prediction with cost-aware L7 routing optimization, enabling more efficient allocation of cloud resources while maintaining stable system performance.

4.5 Discussion

The experimental results indicate that L7-COLO effectively combines predictive workload modeling with cost-aware routing to achieve both economic and performance objectives. By accurately forecasting traffic and mapping it to resource consumption and cost, the framework proactively minimizes cloud expenses while maintaining high system performance. Compared to static strategies and even LSTM-based predictive load balancing, L7-COLO achieves notable reductions in compute, bandwidth, and scaling costs, as well as improved resource utilization. Furthermore, latency improvements and lower SLA violation rates confirm that proactive, reinforcement learning-driven routing does not compromise service quality. The results also highlight the value of integrating traffic prediction with dynamic routing, as predictive information enables the system to mitigate sudden workload spikes before they trigger unnecessary scaling events. Overall, these findings demonstrate that L7-COLO provides a holistic, intelligent solution for dynamic cloud cost optimization in enterprise microservice environments, balancing the trade-off between operational efficiency and service-level performance. The framework's ability to mitigate sudden spikes is particularly crucial for supporting next-generation AI services, such as retrieval-augmented models for technical contradiction mining [19], which are sensitive to both latency and infrastructure costs.

6. Conclusions

This study investigates the problem of dynamic cloud service cost optimization in enterprise microservice environments by proposing the L7-COLO (Layer-7 Cost-Optimized Load Optimization) framework, which integrates intelligent L7 load balancing with deep traffic prediction and cost-aware routing optimization. Modern cloud-native systems increasingly rely on container orchestration platforms and service mesh architectures, where routing decisions at the application layer directly influence infrastructure utilization, container scaling behavior, and operational expenditure. However, conventional L7 routing strategies, such as round-robin and least-connection, mainly focus on performance metrics and rarely consider the economic impact of traffic distribution. To address this limitation, this research introduces a unified framework that models the relationship between traffic demand, resource consumption, and cloud operational cost through the proposed Traffic-Resource-Cost (TRC) mapping model. Within the L7-COLO architecture, a CNN-BiLSTM-Attention prediction module is first used to capture temporal patterns in API request traffic and forecast short-term workload fluctuations. Based on the predicted traffic demand, the system estimates the required computing resources, including CPU, memory, network bandwidth, and container scaling requirements under different routing strategies. A cloud cost evaluation model then translates predicted resource usage into operational expenses, while a reinforcement learning-based routing optimizer dynamically adjusts L7 load balancing weights to minimize overall cost while maintaining service-level agreement (SLA) constraints. Experimental evaluations conducted on real-world microservice workload traces demonstrate the effectiveness of the proposed approach. Compared with traditional Round-Robin routing, the L7-COLO framework reduces total cloud operational cost from 18.9 USD/min to 12.9 USD/min, achieving approximately 31.7% cost reduction. At the same time, the system improves overall resource utilization to 78.4%, significantly higher than conventional strategies. Importantly, the framework maintains stable service performance, with an average latency of 155 ms and a substantially lower SLA violation rate, indicating that cost optimization does not compromise

service quality. These results highlight the capability of L7-COLO to achieve a balanced trade-off between performance efficiency and economic optimization, making it particularly suitable for large-scale SaaS platforms, e-commerce infrastructures, and financial cloud services. Despite these promising results, several limitations remain. The current study primarily focuses on workload traces collected from a limited set of enterprise microservice applications, and the diversity of traffic patterns across different industries may influence model generalization. In addition, the reinforcement learning policy may require additional training time when deployed in highly dynamic production environments. Future research could extend this work in several directions. First, integrating multi-cloud and hybrid-cloud pricing models could allow the framework to optimize routing decisions across heterogeneous cloud providers. Second, considering the sensitivity of enterprise traffic data, integrating privacy-preserving mechanisms into the L7-COLO framework is a promising avenue. Specifically, adopting techniques from asynchronous federated learning and differential privacy [20] could enable multiple cloud clusters to collaboratively refine traffic prediction and routing models while ensuring that sensitive operational metrics and user patterns remain locally protected. Finally, exploring advanced reinforcement learning algorithms and online learning mechanisms could enhance the adaptability of L7-COLO to rapidly changing workloads and real-time operational constraints. Overall, this research demonstrates the potential of combining deep learning and intelligent load balancing to enable cost-aware cloud infrastructure management in modern enterprise systems.

References

- Pahl, Claus. "Containerization and the paas cloud." *IEEE Cloud Computing* 2.3 (2015): 24-31.
- Bourke, Tony. Server load balancing. "O'Reilly Media, Inc.", 2001.
- Hu, Yi, et al. "Time-varying microservice orchestration with routing for dynamic call graphs via multi-scale deep reinforcement learning." *IEEE Transactions on Services Computing* (2025).
- Xu, Jing, et al. "Autonomic resource management in virtualized data centers using fuzzy logic-based approaches." *Cluster Computing* 11.3 (2008): 213-227.
- Duarte Maia, João Tiago, and Filipe Figueiredo Correia. "Service mesh patterns." *Proceedings of the 27th European Conference on Pattern Languages of Programs*. 2022.
- Box, George EP, et al. Time series analysis: forecasting and control. John Wiley & Sons, 2015.
- Graves, Alex. "Long short-term memory." *Supervised sequence labelling with recurrent neural networks* (2012): 37-45.
- Mao, Ming, Jie Li, and Marty Humphrey. "Cloud auto-scaling with deadline and budget constraints." *2010 11th IEEE/ACM International Conference on Grid Computing*. IEEE, 2010.
- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." *arXiv preprint arXiv:2012.07436* (2020).
- Lin, Yuzhen, et al. "CEMG: Collaborative-Enhanced Multimodal Generative Recommendation." *arXiv preprint arXiv:2512.21543* (2025).
- Gmach, Daniel, et al. "Workload analysis and demand prediction of enterprise data center applications." *2007 IEEE 10th International Symposium on Workload Characterization*. IEEE, 2007.
- Xu, Jing, and Jose AB Fortes. "Multi-objective virtual machine placement in virtualized data center environments." *2010 IEEE/ACM int'l conference on green computing and communications & int'l conference on cyber, physical and social computing*. IEEE, 2010.
- Li, Tianran, Hanwu Li, and Yutong Zhou. "E-commerce Sentiment Analysis Using Fine-tuned LLaMA3 Models: A QLoRA-based Approach." *Journal of Technology Innovation and Engineering* 1.4 (2025).
- Yao, Yilin, Wenchao Zhang, and Mengtong Li. "Cloud-Edge Federated Incremental Learning Framework for PMSM Efficiency Optimization with Lightweight CNN-LSTM Models and OTA Differential Deployment." *Journal of Computer Science and Frontier Technologies* 3.1 (2026): 98-111.
- Xu, H., J. Ge, and Y. Ou. "DentaScope-AI: A Real-Time k-mer-Driven CNN-Attention Framework for Rapid Pathogen Identification and Virulence Prediction in Endodontic Infections Using Nanopore Sequencing." *Frontiers in Healthcare Technology* 3.1 (2026): 112-122.
- Ou, Yihao, and Xiyuan Zhu. "Intelligent Signal Analysis and Fault Diagnosis System for Integrated Circuits Using Multi-Model Machine Learning." *Proceedings of the 2nd International Symposium on Integrated Circuit Design and Integrated Systems*. 2025.
- Fan, Pingmei, Hanwu Li, and Mengdie Hu. "Profit-Oriented Production and Pricing Optimization for Manufacturing Enterprises Using Proximal Policy Optimization." *Economics and Management Innovation* 3.2 (2026): 8-17.

- Xu, Zitong, Yuqing Wu, and Yue Zhao. "TRIZ-RAGNER: A Retrieval-Augmented Large Language Model for TRIZ-Aware Named Entity Recognition in Patent-Based Contradiction Mining." *Journal of Computer, Signal, and System Research* 3.2 (2026): 56-65.
- Zhong, Zhen, Shini Yang, and Liesheng Wei. "Privacy-enhanced federated learning via asynchronous aggregation and local differential perturbation." *Third International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2025)*. Vol. 14128. SPIE, 2026.