Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 8, 2024

Transferable Load Forecasting and Scheduling via Meta-Learned Task Representations

Tao Yang
Illinois Institute of Technology, Chicago, USA
tyang28@hawk.iit.edu

Abstract: This paper addresses the problems of low prediction accuracy and poor generalization of scheduling strategies in multi-tenant cloud environments. It proposes a meta-learning-based method for cross-scenario load prediction and adaptive scheduling. The method consists of two core modules: a task-aware representation embedding mechanism and a meta-optimized scheduling strategy. First, a task-level representation learning model is constructed to extract transferable structural features from historical load sequences. This improves the model's ability to understand heterogeneous tasks. Then, a scheduling policy generator is designed based on a meta-learning framework. It optimizes the initialization of policy parameters through multi-task training, enabling the scheduler to quickly adapt and efficiently allocate resources when new tasks arrive. Comprehensive experiments are conducted on a real-world cloud workload dataset. The results show that the proposed method outperforms existing representative approaches in terms of prediction error, scheduling violation rate, and response latency. It demonstrates good generalization and stability, and effectively enhances resource utilization and service quality in cloud platforms.

Keywords: Cloud computing scheduling, task modeling, meta-optimization, load prediction

1.Introduction

With the rapid development of cloud computing technologies, the demand for computing resources has become increasingly dynamic and complex. Cloud platforms have become a critical infrastructure for supporting compute-intensive tasks. From high-concurrency transaction processing in e-commerce to dynamic load scheduling in AI inference services, cloud infrastructure is responsible for meeting the growing needs of elastic computing[1]. A core capability of cloud services lies in the on-demand allocation of computing, storage, and network resources, which enables efficient resource utilization and cost control. However, in practical applications, cloud resource demands often exhibit high temporal correlation, burstiness, and diversity. These characteristics make traditional static or experience-driven approaches to load prediction and scheduling insufficient in complex and volatile environments. This can result in delayed resource allocation, increased service latency, and SLA violations. Therefore, developing more intelligent, flexible, and generalizable load prediction and scheduling mechanisms has become a key challenge in current cloud computing research[2].

Most existing cloud load prediction methods are designed for single scenarios or fixed-tenant environments. These models typically rely on large volumes of homogeneous data and cannot quickly adapt to new environments or tasks[3]. This limitation is particularly evident in multi-tenant cloud platforms. Different tenants often have significantly different business models, workload characteristics, and resource usage patterns, making it difficult for a single prediction model to generalize well across scenarios. Meanwhile, the cloud environment is highly dynamic, with frequent changes in task types and tenant requests[4]. Models trained offline quickly become obsolete. In this context, there is an urgent need to explore intelligent modeling methods with strong transferability and high learning efficiency. Such methods should enable rapid adaptation and enhance the resilience and service quality of cloud platforms in complex environments[5].

Meta-learning, also known as "learning to learn," has emerged as a promising approach in recent years. It offers natural advantages such as fast transfer and few-shot adaptation. This makes it a suitable theoretical foundation for addressing multi-scenario cloud load prediction problems. By learning shared knowledge from multiple tasks, meta-learning models can quickly adjust their parameters when encountering new tasks[6]. This enables fast prediction and scheduling decisions, significantly reducing adaptation time. Compared to traditional deep learning models, meta-learning demonstrates stronger robustness and generalization in the face of data distribution shifts and frequent task switching. Introducing meta-learning into cloud load modeling can potentially overcome the limitations of current methods, such as high retraining cost, long latency, and poor adaptability. It enables a shift from static response to dynamic learning in resource prediction and scheduling[7].

In cloud systems, the accuracy of load prediction directly impacts the effectiveness of resource scheduling. Underestimation may lead to degraded service performance and increased request queuing delays. Overestimation may cause resource waste and higher operational costs. In scenarios involving multiple tenants or applications running simultaneously, the uncertainty of workloads increases significantly. This further complicates the design of scheduling strategies[8]. Traditional approaches often rely on rule-based or historical average methods, which fail to dynamically capture hidden patterns in complex business behaviors. In contrast, load prediction models based on meta-learning can infer the resource demand characteristics of current tasks using training experience from historical tasks. They can quickly adapt across tenants and business scenarios, providing strong support for building intelligent scheduling strategies.

In summary, research on meta-learning-based cross-scenario cloud load prediction and adaptive scheduling has significant theoretical and practical value. On one hand, it offers effective solutions to the poor adaptability of current scheduling systems in dynamic environments. On the other hand, it lays a foundation for building future cloud platforms with self-aware, self-adaptive, and self-optimizing capabilities. As cloud computing evolves toward larger scale, more complex architectures, and higher intelligence, predictive and scheduling algorithms with rapid generalization capabilities will become essential to delivering high-quality cloud services. This research will not only improve resource utilization and service stability in cloud platforms but also provide key algorithmic and technical support for the evolution of intelligent cloud infrastructure.

2. Related work

2.1 Research on Load Forecasting Method

In cloud computing systems, load prediction is a critical prerequisite for resource management and task scheduling. It has long been a key research direction in the intelligent control of cloud platforms. Traditional load prediction methods often rely on time series analysis and statistical modeling. Common techniques include sliding windows and autoregressive moving average models to fit historical data[9]. These methods are simple in structure and easy to implement. However, they show significant limitations when dealing with highly bursty workloads, lack clear periodicity, or are influenced by multiple dimensions. This is especially true in multi-tenant or complex business scenarios, where prediction errors increase noticeably. As a result, such methods fail to meet the dual requirements of real-time response and high accuracy in modern cloud platforms[10].

To overcome the limited nonlinear modeling capabilities of traditional methods, researchers have increasingly adopted machine learning and deep learning techniques for cloud load modeling. Typical approaches include using regression models such as support vector regression and random forests to capture resource usage trends. Other methods apply recurrent neural networks (RNN), long short-term memory networks (LSTM), or Transformer architectures to learn temporal dependencies and multi-scale patterns. Some studies also integrate multi-source data, including task characteristics, network traffic, and system logs, to build multimodal inputs that improve prediction accuracy[11]. These approaches achieve good results in certain specific scenarios. However, they often suffer from poor generalization, difficulty in model transfer, and slow adaptation to new environments. In situations with frequent task switching or sparse data, deep models typically require retraining. This makes them unsuitable for deployment across diverse scenarios[12].

In addition, some studies have explored the use of attention mechanisms and graph neural networks in load prediction. These models aim to enhance the ability to capture changes in workload structure and resource dependencies among tenants. Nevertheless, most existing methods still rely heavily on well-labeled training data[13,14]. Their designs are often optimized for specific platforms or business scenarios, lacking transferability across environments. This limitation is especially critical in complex settings such as multi-tenant cloud platforms and edge-cloud collaborative computing[15]. Under these conditions, prediction model performance often drops sharply when the environment changes or data distribution shifts. Therefore, designing a prediction method with strong task generalization and fast scene adaptation has become a key development direction in this field. It also provides the theoretical foundation and practical motivation for the meta-learning-based approach proposed in this study.

2.2 Meta-Learning

As a learning paradigm designed to improve model generalization and fast adaptation, meta-learning has shown broad application prospects in recent years[16.17]. It has been applied in various fields such as computer vision, natural language processing, and recommendation systems. The core idea of meta-learning is to abstract knowledge from training across many tasks. This allows the model to learn efficiently when facing a new task, using only a few samples or training iterations[18]. The introduction of meta-learning breaks the dependence of traditional supervised learning on large amounts of labeled data and long training cycles. It is especially suitable for scenarios where tasks change frequently and data distributions vary significantly. In cloud computing systems, the high heterogeneity across tenants, business scenarios, and application types makes traditional models difficult to generalize. Therefore, meta-learning offers a natural advantage in this context[19].

In the area of resource management and system scheduling, initial research on meta-learning has focused on optimizing task-scheduling strategies and learning resource allocation decisions. Most methods adopt model-agnostic meta-learning frameworks. These frameworks train an initialization model across multiple scheduling tasks, enabling fast adaptation to new load types, resource constraints, or quality of service objectives[20,21]. In these studies, meta-learning not only improves the convergence speed and adaptability of scheduling systems but also provides a certain level of generalization over complex policy spaces. Some research has further combined meta-learning with reinforcement learning. This approach trains transferable policy networks, allowing scheduling systems to migrate strategies across environments and reduce the need for explicit environment modeling. These efforts pave the way for deploying general and efficient agents in dynamic cloud scheduling[22].

Despite its promising performance in system optimization, research on meta-learning for cloud load prediction remains limited. Existing methods tend to focus on scheduling policy optimization while paying less attention to the upstream task of load modeling[23]. At the same time, cloud load prediction often exhibits key characteristics such as low task density, large differences across scenarios, and frequent data updates. These are typical features of few-shot and multi-task learning problems, which align well with meta-learning. Applying meta-learning to load prediction can address the generalization limitations of traditional methods. It can also significantly improve modeling efficiency and response speed in new environments. In the future, exploring how to integrate meta-learning with time series modeling and graph-based modeling will be an important direction. This will enhance the model's ability to perceive and adapt to complex system behaviors.

3. Method

This study addresses the challenges of poor generalization and slow adaptation in load prediction and scheduling for multi-tenant and multi-scenario cloud platforms. It proposes a meta-learning-based method for cross-scenario cloud load prediction and adaptive scheduling. The core innovation of this method lies in two components. First, a Task-aware Representation Embedding (TRE) mechanism is designed. It extracts transferable structural features from historical workload data across scenarios, enhancing the model's representation ability and initialization quality for new tasks. Second, a Meta-Optimized Scheduling Policy (MOSP) framework is introduced. It performs meta-training of scheduling strategies under various resource constraints and SLA requirements. This enables the scheduling model to generate near-optimal decisions quickly when facing new workload patterns. The architecture of the overall model is illustrated in Figure 1.

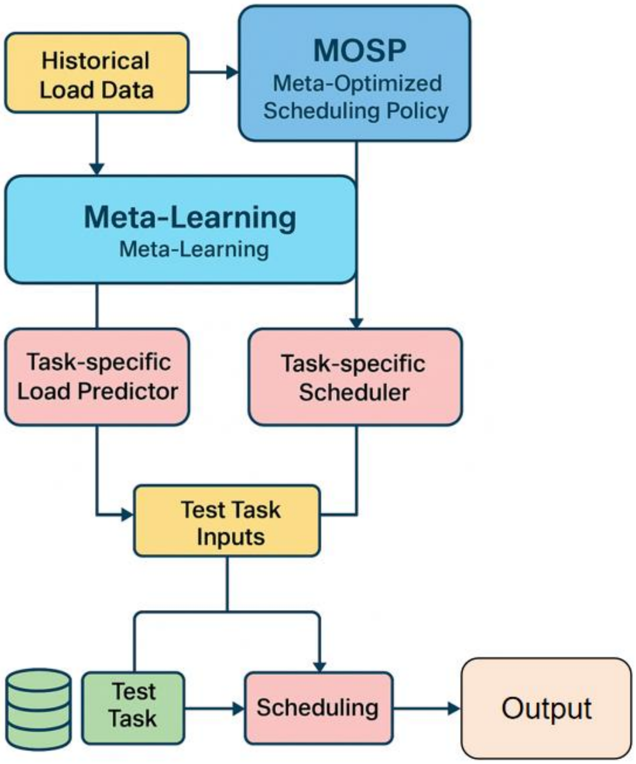

Figure 1. Overall model architecture diagram

3.1 Task-aware Representation Embedding

In multi-tenant cloud computing scenarios, the heterogeneity of task workloads is prominent. Different tasks often exhibit diverse resource usage patterns, temporal dynamics, and operational priorities. Traditional unified modeling approaches, which treat all tasks with a shared representation space or rely on handcrafted features, are insufficient to capture the complex, implicit structural differences that exist between tasks. This lack of fine-grained differentiation leads to degraded model performance when applied to new or unseen tasks, especially under varying resource demands and workload behaviors. To address this challenge, it is essential to design a representation mechanism that can encode task-specific load features while preserving the shared knowledge across tasks.

To enhance the model's ability to generalize across different tasks, this study introduces the Task-aware Representation Embedding (TRE) mechanism. TRE is designed to model and abstract the structured load behavior of tasks by learning informative task-level representations from historical load sequences. It captures both temporal patterns and statistical characteristics of resource consumption, enabling the meta-learning model to condition its parameter updates on these embeddings. By integrating TRE into the learning pipeline, the system can construct discriminative and transferable representations that reflect the unique load semantics of each task. This facilitates faster and more accurate adaptation when the model is exposed to new task types. The detailed module architecture of TRE is illustrated in Figure 2.

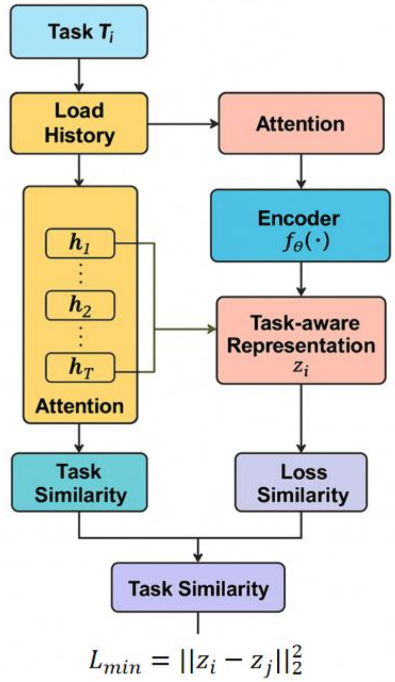

Figure 2. TRE module architecture

First, the load history sequence of a given task T_i is represented as $X_i = \{x_1, x_2, ..., x_T\}$, where $x_t \in R^d$ represents the resource usage vector at the t-th time step. We use a differentiable encoder $f_\theta(\cdot)$ to map this sequence into a fixed-length task representation vector:

$$z_i = f_\theta(X_i)$$

The representation $z_i \in R^h$ serves as a high-level structural summary of the task, which contains key information such as load pattern, fluctuation characteristics, and periodicity.

In order to enhance the perception of key timing features, we introduce an attention mechanism to perform weighted aggregation on the payload sequence. Given the intermediate representation h_t of each time step, its attention weight is calculated as follows:

$$a_t = \frac{\exp(w^T h_t)}{\sum_{k=1}^{T} \exp(w^T h_k)}$$

The final embedding is represented as:

$$z_i = \sum_{t=1}^{T} a_t h_t$$

This attention aggregation method enables the model to automatically focus on the key change points in the load sequence and improve the fine-grained expression capability of task modeling.

In order to ensure that similar tasks have similar representation structures in the embedding space, we introduce a consistency regularization term for task representation. For two similar tasks T_i, T_j , we minimize the distance between their embedding vectors:

$$L_{sim} = || z_i - z_j ||_2^2$$

This regularization term guides the model to learn a discriminative representation space, which helps to improve the initialization performance of the meta-learning model on new tasks.

Finally, the task representation z_i will be used as one of the input features of the meta-learning model to generate parameters or quickly adapt the scheduler or predictor during the task initialization phase. By introducing the task-aware embedding mechanism, the TRE module effectively introduces the structural information between different tasks into the training process, providing more discriminative and generalized support for subsequent scheduling strategy learning.

3.2 Meta-Optimized Scheduling Policy

In a dynamic and ever-changing cloud computing environment, scheduling strategies must quickly adapt to different tasks, resource conditions, and service level objectives. To address this need, this study proposes a meta-optimized scheduling policy mechanism (Meta-Optimized Scheduling Policy, MOSP). This module takes task-aware representations as input. It learns from scheduling optimization experiences across multiple historical tasks to build a scheduling strategy generator with generalization capability. The core idea of this method is to optimize the initialization of scheduling policy parameters within a meta-learning framework. This allows the model to converge rapidly to a near-optimal strategy for new tasks without requiring extensive iterations. The detailed module architecture of SADAN is illustrated in Figure 3.

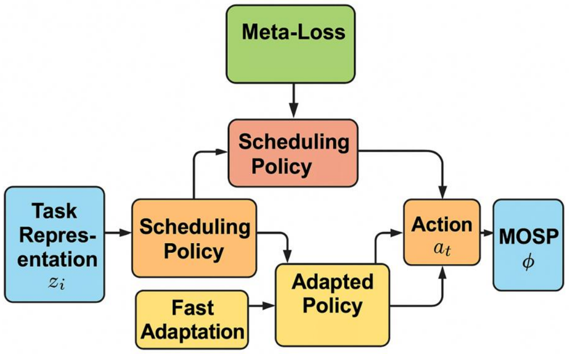

Figure 3. MOSP module architecture

Assume that the task representation vector of a task T_i is z_i , the system state is s_t , the scheduling strategy is a parameterized function $\pi_\phi(s_t, z_i)$, and its output is the resource allocation decision a_t at the current moment. That is:

$$a_t = \pi_{\phi}(s_t, z_t)$$

This strategy realizes personalized generation and dynamic adaptation of scheduling strategies by considering the joint input of task representation and system status.

In order to obtain policy parameter initialization with good generalization ability in the training phase, this method adopts a meta-optimization mechanism based on gradient update. Assuming that the loss function of the task T_i in a scheduling process is $L_{T_i}(\phi)$, its policy update is:

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

Where α is the inner learning rate, and ϕ'_i represents the strategy parameters after a quick adaptation to task T_i.

The meta-optimization goal is to minimize the total loss after fast updates on multiple tasks and optimize the initial parameter ϕ. Its objective function is:

$$\min_{\phi} \sum_i L_{T_i}(\phi'_i) = \sum_i L_{T_i}(\phi - \alpha \nabla_{\phi} L_{T_i}(\phi))$$

This objective function reflects the model's ability to quickly adapt to future tasks and is a direct reflection of the strategy's generalization ability.

Finally, in the actual deployment phase, for a newly arrived task T_{new}, the scheduler can quickly generate resource allocation decisions through a small amount of fine-tuning or directly based on the initialization strategy B. The scheduling process is:

$$a_t^{new} = \pi_{\phi'}(s_t, z_{new})$$

This design effectively embeds task information into the scheduling strategy generation process, realizes the rapid migration and dynamic optimality of the scheduling strategy, and takes into account resource utilization and service quality requirements while ensuring system stability.

4. Experimental Results

4.1 Dataset

This study uses the Google Cluster Trace dataset as the primary data source for load modeling and scheduling policy learning. The dataset was collected from a large-scale distributed computing cluster operating in a real production environment. It contains detailed records of resource requests, task scheduling logs, and machine status information. With a long time span and rich dimensions, it is one of the most widely used public datasets for studying resource management and scheduling in cloud computing.

Google Cluster Trace covers the scheduling and execution of tens of thousands of tasks across thousands of servers. It includes data on the usage and request amounts for resources such as CPU, memory, disk, and network. It also provides metadata such as job start time, end time, priority, and failure reasons. The dataset features high load variability and task heterogeneity. It effectively simulates complex scenarios involving multi-tenant and multi-application workloads in real cloud platforms. It is suitable for task-level load modeling, predictive analysis, and scheduling policy research.

In this study, task behavior traces from the dataset are used to construct time series inputs. These inputs help extract workload patterns and generate task representations. They also support the scheduling module in modeling the dynamic changes of task resource demands. The dataset's realism and scale provide strong data support for verifying model generalization and evaluating policy transfer mechanisms.

4.2 Experimental setup

To validate the effectiveness of the proposed method in cross-task load prediction and scheduling adaptation, the experiments use task execution logs from the Google Cluster Trace dataset as the primary data source. After preprocessing, a multi-task time series sample set is constructed. Tasks are divided into multiple scenarios based on task types and resource request characteristics. All tasks are split in chronological order into training, validation, and test sets. These are used for meta-training, fast adaptation, and generalization performance evaluation, respectively.

During model training, the task representation module uses a standard multilayer perceptron (MLP) as the encoder. The scheduling policy is built using a two-layer policy network. Optimization is performed using the Adam optimizer. All experiments are conducted on a server with 32 GB of memory and an NVIDIA V100 GPU. Model hyperparameters are selected through grid search on the validation set. Table 1 summarizes the key experimental parameter settings. The detailed settings are shown in Table 1.

Table 1: Experimental setup and hyperparameter configuration

Parameter	Value
Dataset source	Google Cluster Trace
Input sequence length	50
Encoder structure	2-layer MLP, 128 hidden dim
Policy network structure	2-layer MLP, ReLU
Optimizer	Adam
Learning Rate	0.001
Inner layer update steps (K)	5
Meta-learning outer batch size	8 tasks/batch
Experimental Platform	NVIDIA V100, 32GB RAM

4.3 Experimental Results

1) Comparative experimental results

This paper first gives the comparative experimental results, as shown in Table 2.

Table2: Comparative Results

Method	MAE	Scheduling Default Rate	Average scheduling delay
Ours	0.083	2.4%	53.7
MAML-Scheduler[24]	0.097	4.2%	61.5
DeepRM[25]	0.112	5.9%	74.2
K-argued[26]	0.095	3.8%	59.1

The comparison results in the table show that the proposed meta-learning-based scheduling method demonstrates significant advantages across all metrics. This validates its generalization and fast adaptation capabilities in complex and dynamic cloud environments. Compared to traditional methods, the proposed approach effectively captures structural differences between tasks. It can also quickly adjust prediction and scheduling strategies when facing new tasks, improving overall system scheduling efficiency and stability.

In terms of load prediction accuracy, traditional deep learning models have certain abilities for temporal modeling. However, they often perform inconsistently when data distribution changes due to the lack of cross-task knowledge transfer. In contrast, the proposed method introduces task-aware representations and a meta-learning framework. It leverages shared structures across historical tasks to generate high-quality load representations for new tasks. This reduces prediction errors and improves the model's robustness to task heterogeneity.

Regarding scheduling violation rates, reinforcement learning methods can perform well in single scenarios. However, they often lack adaptability during task switching across multiple scenarios. This may lead to imbalanced resource allocation or violations of service level agreements (SLAs). In comparison, the proposed meta-optimized scheduling strategy dynamically adjusts scheduling behavior based on task embedding vectors. This enables policy transfer and fast adaptation, reducing system risks and improving resource utilization efficiency.

Finally, in terms of average scheduling delay, traditional models typically require longer computation and adjustment times. This makes it difficult to meet real-time response requirements. The proposed method injects meta-learning capabilities at the policy initialization stage. This allows the model to schedule tasks without repeated trial-and-error or retraining, reducing the response time from task input to scheduling execution. It highlights the practical advantage of the method in complex and dynamic environments. These results demonstrate the importance and value of designing scheduling strategies with task generalization and fast adaptation capabilities for cloud computing scenarios.

2) Ablation Experiment Results

This paper also further gives the results of the ablation experiment, and the experimental results are shown in Table 3.

Table 3: Ablation Experiment Results

Method	MAE	Scheduling Default Rate	Average scheduling delay
Baseline	0.106	5.7%	71.9
+TRE	0.094	4.3%	63.4
+MOSP	0.089	3.6%	59.2
Ours	0.083	2.4%	53.7

As shown in the ablation results in Table 3, the Task-aware Representation Embedding (TRE) module and the Meta-Optimized Scheduling Policy (MOSP) each play important roles in improving overall system performance. The base model, without any of these modules, struggles to capture structural differences between tasks and load fluctuations. This leads to poor prediction accuracy and unstable scheduling performance. These results indicate that in multi-task and multi-scenario cloud environments, shallow representations and single-strategy approaches cannot meet the high demands for generalization and adaptability.

After introducing the TRE module, the model gains a stronger ability to represent task-specific load features. By abstracting historical load sequences into structured representations, task-specific information is effectively captured. This significantly improves prediction accuracy and enhances the model's sensitivity to input variations. The inclusion of TRE allows the model to generate differentiated predictive representations based on task context, improving both load awareness and the stability of the scheduling foundation.

With the additional integration of the MOSP module, the scheduling system gains fast adaptation and optimization capabilities. MOSP uses a meta-learning approach to initialize and update the scheduling policy parameters. This allows the model to generate near-optimal strategies with fewer adjustment steps when facing new load patterns or resource constraints. The mechanism reduces reliance on long-term training and large datasets while maintaining high resource utilization and stable task execution.

Finally, when TRE and MOSP are combined, the model achieves the best performance in both prediction and scheduling. This demonstrates the overall advantage of constructing an end-to-end task-aware and policy-transferable framework. The results further confirm the practical value and broad application potential of the proposed method in complex cloud computing environments.

3) Cross-scenario prediction capability evaluation under different task types

This paper further provides an evaluation of cross-scenario prediction capabilities under different task types, aiming to investigate the model's adaptability and generalization in heterogeneous workload environments. In real-world cloud computing systems, tasks often belong to various categories, such as compute-intensive, storage-intensive, and hybrid types, each exhibiting distinct resource consumption patterns and temporal dynamics. Evaluating prediction performance across these diverse task types is essential to understand how well the proposed method can transfer learned knowledge from one scenario to another. This evaluation helps assess whether the model can maintain stable predictive behavior when faced with significant variations in workload characteristics.

To conduct this evaluation, a comprehensive experimental design is employed, where the model is tested on a range of task types that differ in both structure and behavior. The purpose is to validate the model's ability to construct meaningful and transferable representations that are not limited to specific scenarios. Such analysis allows for a deeper examination of how effectively the model generalizes across different operational contexts and task demands. The results of this evaluation are illustrated in Figure 4, which provides a visual representation of the model's performance across multiple task categories.

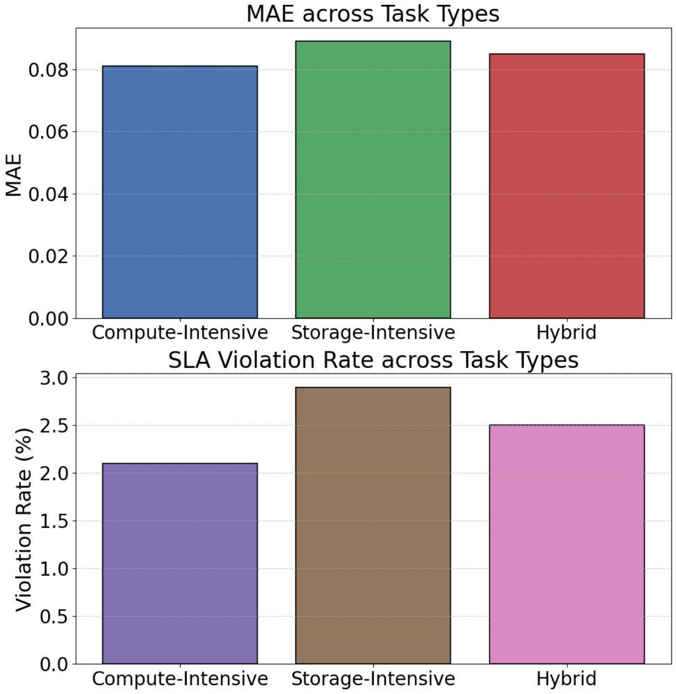

Figure 4. Cross-scenario prediction capability evaluation under different task types

As shown in the results of Figure 4, the proposed method demonstrates stable prediction and scheduling performance across different types of tasks. This reflects strong cross-scenario generalization ability. For compute-intensive, storage-intensive, and hybrid tasks, the model accurately captures the differences in load patterns. This indicates that the task-aware representation mechanism effectively improves the modeling quality of heterogeneous task features, providing a solid foundation for subsequent scheduling.

In terms of prediction error, although task types differ in time scale and fluctuation range, the model maintains consistent error levels across all categories. This suggests that the model learns transferable and shared structural features. It shows that the meta-learning framework can extract common knowledge from historical tasks and adapt quickly to new ones. This addresses the failure of traditional methods during the initialization phase of unseen tasks.

The trend of scheduling violation rates further shows that the model maintains low scheduling errors even for complex tasks such as storage-intensive or hybrid ones. This demonstrates the stability of the scheduling strategy when facing resource bottlenecks or concurrent tasks. This advantage is attributed to the MOSP module, which optimizes policy initialization. It enables the model to quickly generate effective scheduling behavior in new environments, reducing resource allocation bias and SLA violation risks.

Overall, the experiment confirms the generality and adaptability of the proposed method across different task structures. It provides additional evidence for the practicality and robustness of the task-aware and meta-optimized scheduling mechanism in real cloud environments. It also suggests a direction for future work, which is to further improve performance under extreme task types and enhance the model's generality and integration capability.

4) The generalization ability test of the model under different load fluctuation modes

This paper also presents a test of the generalization ability of the model under different load fluctuation modes, and the experimental results are shown in Figure 5.

As shown in the experimental results in Figure 5, the proposed method maintains stable prediction and scheduling performance under different load fluctuation patterns. This indicates strong generalization capability. When facing stable, periodic, and bursty load changes, the overall performance variation remains within a controllable range. This suggests that the task representation and scheduling policy can adapt to diverse runtime environments.

For load prediction, the model achieves consistent accuracy under stable and periodic loads. This reflects that the task-aware embedding mechanism can extract key temporal structure features and model regular patterns effectively. Although performance slightly drops under bursty loads due to their unpredictability, the model still maintains a competitive level. This shows its ability to withstand sudden changes and make rapid adjustments.

In terms of scheduling performance, the increase in scheduling violation risk under bursty load patterns indicates more severe resource allocation challenges. However, the advantage of the MOSP module in policy initialization and fast adaptation keeps the overall violation rate low. In particular, under periodic and stable loads, the scheduling policy dynamically adjusts based on task history. This helps ensure compliance with service-level agreements.

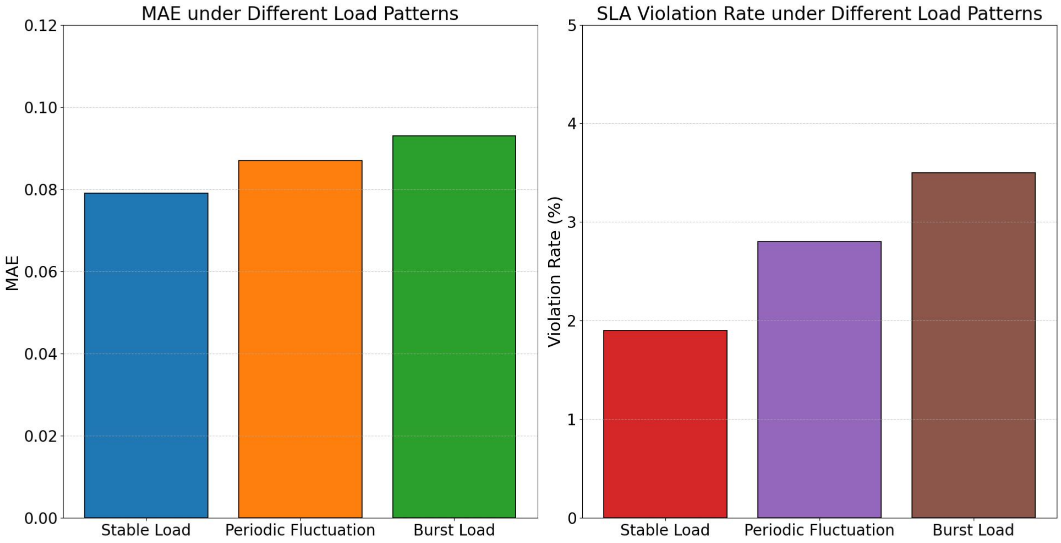

Figure 5. The generalization ability test of the model under different load fluctuation modes

In summary, this experiment further confirms the robustness and practicality of the proposed method in handling diverse load fluctuations. It highlights the importance of meta-learning strategies and task-aware mechanisms in enhancing model generalization in cloud computing environments. The ability to adapt well to changing conditions is a core feature required for building intelligent cloud scheduling systems suitable for real-world deployment.

5) Resource fairness scheduling experiment in a multi-tenant environment

This paper also presents a resource fairness scheduling experiment in a multi-tenant environment, aiming to evaluate the ability of the proposed method to maintain balanced resource allocation among multiple tenants. In practical cloud computing platforms, ensuring fairness is a critical aspect of system performance, especially when diverse workloads from different tenants compete for shared computing resources. Fair scheduling not only prevents resource monopolization by a single tenant but also contributes to overall system stability and service quality. Therefore, this experiment is designed to analyze how effectively the scheduling strategy distributes resources under concurrent and dynamic task demands from multiple tenants.

The experiment involves simulating a multi-tenant environment where several tenants submit tasks simultaneously, each with different load characteristics and resource requirements. The scheduling system must allocate CPU, memory, and other critical resources while maintaining fairness across tenants. To assess this, fairness metrics are introduced to quantify the degree of balance in resource usage. This setup allows for the observation of whether the scheduling strategy can provide equitable access to system resources, regardless of the variability in task types or arrival patterns. The detailed outcomes of this experiment are visually presented in Figure 6, which illustrates the resource usage trends and fairness dynamics over time.

As shown in the experimental results of Figure 6, the proposed scheduling method achieves balanced resource utilization in a multi-tenant environment. This reflects good scheduling fairness. The resource usage curves of the three tenants remain close to the overall trend. This indicates that the scheduling strategy effectively prevents resource skew or long-term occupation of critical system resources by any single tenant. Such balanced allocation is a direct result of the joint optimization of task representation and scheduling policy.

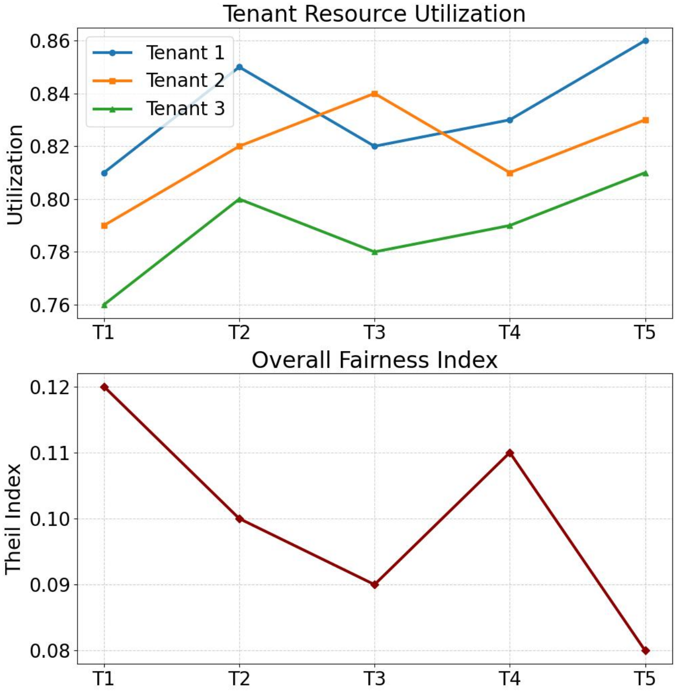

Figure 6. Resource fairness scheduling experiment in a multi-tenant environment

Over time, the fluctuations in resource usage among tenants remain within a reasonable range. This shows that the proposed method provides strong tenant isolation and consistent responsiveness in dynamic systems. Even when the structure of resource demand changes, the scheduler uses learned task representations to allocate resources in an orderly manner. This avoids issues such as resource contention and scheduling starvation and improves system availability and stability.

In terms of resource fairness metrics, the trend of the Theil index further reflects the system's ability to regulate scheduling fairness. As time progresses, the index gradually decreases, indicating that resource allocation among tenants becomes more balanced. The system continuously optimizes the scheduling strategy during operation and achieves fair scheduling across tasks. This result verifies the self-adjustment mechanism of the MOSP module during policy transfer and its sustained ability to enhance system-level fairness.

Overall, the results show that the proposed scheduling method performs well not only in single-task environments but also maintains consistent and fair resource usage in complex multi-tenant scenarios. The model can dynamically perceive task load features and adjust the resource allocation strategy. This provides a solid algorithmic foundation for building efficient and fair cloud resource management frameworks.

5. Conclusion

This paper presents a meta-learning-based cross-scenario resource control method for cloud environments with multiple tenants. It addresses the limitations of traditional approaches in task transfer, model generalization, and fast adaptation. The proposed method introduces innovations at both the task modeling and scheduling strategy levels. Specifically, it designs a task-aware representation embedding mechanism and a meta-optimized scheduling policy module. Experimental results show that the proposed method achieves high stability, low error, and fast response across various load scenarios and task types, demonstrating its practicality and scalability in dynamic and complex systems.

The task-aware representation embedding mechanism structurally models load sequences. This enables the model to extract transferable and discriminative task features. It effectively solves the problems of information loss and poor generalization in multi-task modeling. Meanwhile, the meta-optimized scheduling strategy uses shared knowledge from historical tasks to initialize and rapidly fine-tune scheduling parameters. This allows the model to generate high-quality scheduling strategies quickly under different task demands and resource constraints. This end-to-end generalization mechanism removes the reliance of traditional schedulers on specific scenarios and offers a practical path toward building general and intelligent scheduling systems.

From an application perspective, the proposed method holds significant theoretical value and broad practical potential. It can be widely applied to elastic cloud services, serverless computing frameworks, and AI inference deployment platforms. As enterprise applications increasingly demand intelligent resource management, adaptability, and service stability, scheduling algorithms with task transfer learning capabilities will become key technologies for improving cloud service quality. In addition, the method has strong transferability to emerging scenarios such as edge computing, hybrid clouds, and green computing. It lays a solid algorithmic foundation for the next generation of intelligent and collaborative resource scheduling systems.

6. Future work

Future research may explore finer-grained task representation learning methods to improve model robustness under extreme workload fluctuations. Integrating federated learning, multi-agent scheduling, and system feedback mechanisms could help build more interactive, autonomous, and secure intelligent resource management frameworks. By extending the meta-learning approach proposed in this paper to broader system optimization tasks, it is possible to accelerate the development of intelligent cloud platforms. This will support a shift from automation to self-evolving systems aimed at sustainable and efficient operations.

References

[1] Zhang K, Guo W, Feng J, et al. Load forecasting method based on improved deep learning in cloud computing environment[J]. Scientific Programming, 2021, 2021(1): 3250732.

[2] Ramamoorthi V. Optimizing Cloud Load Forecasting with a CNN-BiLSTM Hybrid Model[J]. International Journal of Intelligent Automation and Computing, 2022, 5(2): 79-91.

[3] Xu, M., Song, C., Wu, H., Gill, S. S., Ye, K., & Xu, C. (2022). esDNN: deep neural network based multivariate workload prediction in cloud computing environments. ACM Transactions on Internet Technology (TOIT), 22(3), 1-24.

[4] Peng H, Wen W S, Tseng M L, et al. A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy[J]. Soft Computing, 2021, 25(15): 10205-10220.

[5] Patel E, Kushwaha D S. A hybrid CNN-LSTM model for predicting server load in cloud computing[J]. The Journal of Supercomputing, 2022, 78(8): 1-30.

[6] Rotib H W, Nappu M B, Tahir Z, et al. Electric load forecasting for Internet of Things smart home using hybrid PCA and ARIMA algorithm[J]. International Journal of Electrical and Electronic Engineering & Telecommunications, 2021, 10(6): 369-376.

[7] Saxena D, Singh A K. Workload forecasting and resource management models based on machine learning for cloud computing environments[J]. arXiv preprint arXiv:2106.15112, 2021.

[8] Ouhame S, Hadi Y, Ullah A. An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model[J]. Neural Computing and Applications, 2021, 33(16): 10043-10055.

[9] Kulkarni M, Deshpande P, Nalbalwar S, et al. Cloud computing based workload prediction using cluster machine learning approach[C]//International Conference on Computing in Engineering & Technology. Singapore: Springer Nature Singapore, 2022: 591-601.

[10] Ahamed Z, Khemakhem M, Eassa F, et al. Technical study of deep learning in cloud computing for accurate workload prediction[J]. Electronics, 2023, 12(3): 650.

[11] Khan T, Tian W, Ilager S, et al. Workload forecasting and energy state estimation in cloud data centres: ML-centric approach[J]. Future Generation Computer Systems, 2022, 128: 320-332.

[12] Oprea S V, Bâra A. An Edge-Fog-Cloud computing architecture for IoT and smart metering data[J]. Peer-to-Peer Networking and Applications, 2023, 16(2): 818-845.

[13] Husnoo M A, Anwar A, Hosseinzadeh N, et al. Fedrep: Towards horizontal federated load forecasting for retail energy providers[C]//2022 IEEE PES 14th Asia-Pacific Power and Energy Engineering Conference (APPEEC). IEEE, 2022: 1-6.

[14] Farrag T A, Elattar E E. Optimized deep stacked long short-term memory network for long-term load forecasting[J]. IEEE Access, 2021, 9: 68511-68522.

[15] Bacanin N, Stoean C, Zivkovic M, et al. On the benefits of using metaheuristics in the hyperparameter tuning of deep learning models for energy load forecasting[J]. Energies, 2023, 16(3): 1434.

[16] Quansah, P. K., & Tenkorang, E. K. A. (2023). Short-term load forecasting using A particle-swarm optimized multi-head attention-augmented CNN-LSTM network. arXiv preprint arXiv:2309.03694.

[17] Yin, W. (2020). Meta-learning for few-shot natural language processing: A survey. arXiv preprint arXiv:2007.09604.

[18] Tian Y, Zhao X, Huang W. Meta-learning approaches for learning-to-learn in deep learning: A survey[J]. Neurocomputing, 2022, 494: 203-223.

[19] Huisman M, Van Rijn J N, Plaat A. A survey of deep meta-learning[J]. Artificial Intelligence Review, 2021, 54(6): 4483-4541.

[20] Chi Z, Gu L, Liu H, et al. Metafscil: A meta-learning approach for few-shot class incremental learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 14166-14175.

[21] Baik S, Choi J, Kim H, et al. Meta-learning with task-adaptive loss function for few-shot learning[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 9465-9474.

[22] Lake B M, Baroni M. Human-like systematic generalization through a meta-learning neural network[J]. Nature, 2023, 623(7985): 115-121.

[23] Kim K K, Kim M, Pyun K, et al. A substrate-less nanomesh receptor with meta-learning for rapid hand task recognition[J]. Nature Electronics, 2023, 6(1): 64-75.

[24] Yao H, Wang Y, Wei Y, et al. Meta-learning with an adaptive task scheduler[J]. Advances in Neural Information Processing Systems, 2021, 34: 7497-7509.

[25] Chen W, Xu Y, Wu X. Deep reinforcement learning for multi-resource multi-machine job scheduling[J]. arXiv preprint arXiv:1711.07440, 2017.

[26] Dogani J, Khunjush F, Seydali M. K-agrued: A container autoscaling technique for cloud-based web applications in kubernetes using attention-based gru encoder-decoder[J]. Journal of Grid Computing, 2022, 20(4): 40.