TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/31 pp34–54
DOI: 10.26599/TST.2024.9010024
Volume 30, Number 1, February 2025

Application-Oriented Cloud Workload Prediction: A Survey and New Perspectives

Binbin Feng and Zhijun Ding*

Abstract: Workload prediction is critical in enabling proactive resource management of cloud applications. Accurate workload prediction is valuable for cloud users and providers as it can effectively guide many practices, such as performance assurance, cost reduction, and energy consumption optimization. However, cloud workload prediction is highly challenging due to the complexity and dynamics of workloads, and various solutions have been proposed to enhance the prediction behavior. This paper aims to provide an in-depth understanding and categorization of existing solutions through extensive literature reviews. Unlike existing surveys, for the first time, we comprehensively sort out and analyze the development landscape of workload prediction from a new perspective, i.e., application-oriented rather than prediction methodologies per se. Specifically, we first introduce the basic features of workload prediction, and then analyze and categorize existing efforts based on two significant characteristics of cloud applications: variability and heterogeneity. Furthermore, we also investigate how workload prediction is applied to resource management. Finally, open research opportunities in workload prediction are highlighted to foster further advancements.

Key words: cloud computing; workload prediction; resource management; artificial intelligence for IT operations (AIOps)

1 Introduction

With the rapid development of cloud computing, more and more applications have migrated or will migrate to cloud platforms[1]. Cloud platforms provide applications with powerful computing capabilities, flexible resource allocation, and a high degree of scalability[2], which ensure that applications can achieve better performance, cost, and energy efficiency[3]. In addition, cloud computing brings other advantages to applications, such as globalized deployment, high availability, and powerful data processing capabilities[4], which enable applications to meet user needs ably, provide faster and more stable services, and deliver seamless experiences to users across the globe[5].

To provide cloud applications and services that meet service level agreements (SLAs) to cloud subscribers, robust resource management methods must comprehensively optimize cloud applications' performance, cost, and energy consumption. However, resource management faces significant challenges due to the dynamics of cloud environments, the diversity of user requests and services, and the elastic provisioning of cloud resources[6]. These challenges are mainly

• Binbin Feng is with Key Laboratory of Embedded System and Service Computing, Ministry of Education, and also with Department of Computer Science and Technology, Tongji University, Shanghai 201804, China. E-mail: bining@tongji.edu.cn.
• Zhijun Ding is with Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Department of Computer Science and Technology, Tongji University, Shanghai 201804, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China. E-mail: dingzj@tongji.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2023-10-25; revised: 2024-01-19; accepted: 2024-01-24

reflected in the following aspects:

(1) **Long queuing time**: Due to improper resource allocation or traffic scheduling, user requests may have to wait in the queue for a more extended period before processing.

(2) **Performance unstable**: Due to the dynamics and resource sharing in cloud environments, the applications' performance may be unstable, resulting in a degraded user experience.

(3) **Resource competition**: Multiple applications or services may compete for the same resource, leading to resource bottlenecks, performance degradation, and service crashes.

(4) **Resource idle**: Due to rough or improper resource allocation strategy or inaccurate prediction, some resources may remain unused for longer, wasting resources.

(5) **High energy consumption**: Irrational resource management strategies may increase energy consumption and operational costs.

To solve the above problems, Fig. 1 shows a proactive framework for implementing artificial intelligence for IT operations (AIOps)[6, 7]. Specifically, **Monitoring** is to collect metrics, such as historical request and resource workloads, and quality of service (QoS) of cloud applications; **Analysis** is to predict future workloads and analyze in real-time whether SLAs are met; **Planning** is to make appropriate management decisions to avoid degradation of QoS, cost, and energy inefficiencies, etc.; and lastly, **Execution** is to realize specific operations based on methods such as capacity planning, deployment, scaling, scheduling, and migration with corresponding system tools. This framework serves the entire life cycle of applications. It is worth noting that workload prediction plays a vital role in this framework[8, 9].

However, there are still many challenges to achieving accurate workload prediction. From a general point of view, due to the dynamics of applications, cloud workloads are highly volatile and have variable patterns, which makes it difficult for traditional forecasting methods to predict workloads accurately. In addition, due to the diversity of applications, workload prediction needs to consider different applications' specific needs and features. Researchers have proposed many prediction solutions based on statistics, machine learning, deep learning, and reinforcement learning to solve the above

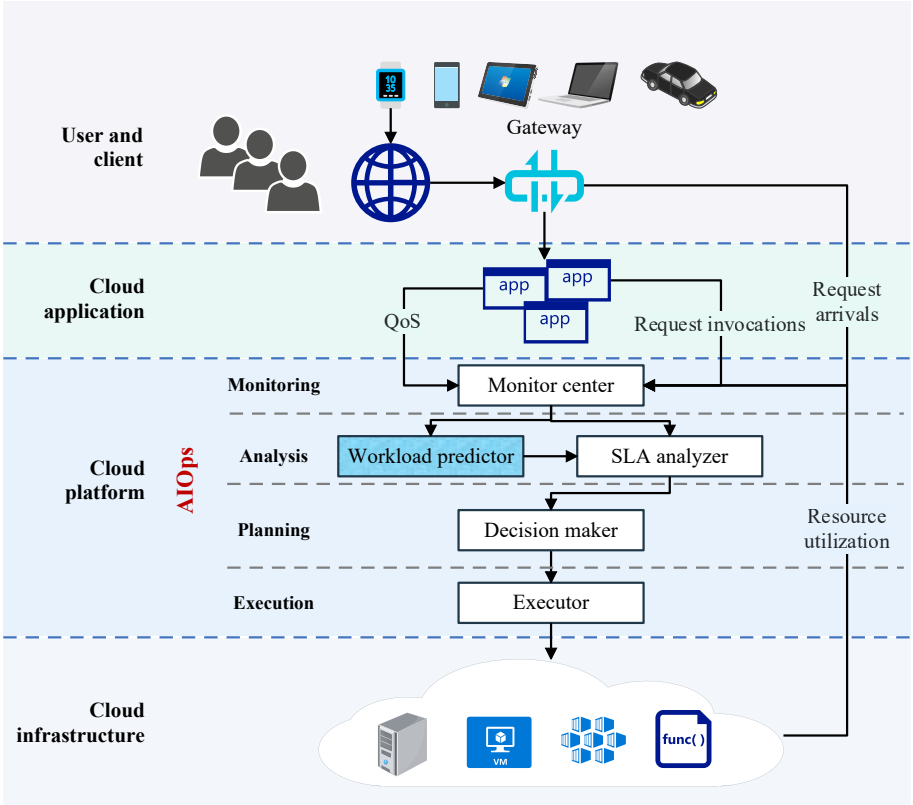

Fig. 1 Proactive application AIOps framework.

challenges.

This paper analyzes the latest workload prediction efforts and their techniques and motivations. Our core basis for categorization is "application-oriented", which means how application-specific (business software) characteristics affect workload changes of cloud applications. Therefore, we categorize these efforts from an application-oriented perspective. We describe how each attempts to predict workload changes and apply these results to the AIOps of applications. Based on an in-depth literature analysis, we propose open research opportunities to set the stage for future research. Specifically, the main contributions of this paper are as follows:

(1) We provide an overview of the basic features related to workload prediction research, including predicted targets, modeling techniques, evaluation metrics, and datasets.

(2) We analyze two characteristics of cloud applications, including variability and heterogeneity, and how application-specific characteristics affect their workload changes.

(3) We categorize recently published work on workload prediction based on the characteristics of cloud applications in conjunction with the research ideas, summarizing the research motivation, primary contributions, and core ideas.

(4) We present remaining research challenges and open opportunities to be addressed in workload prediction.

2 Basic Characteristic

This section introduces the basic features of workload prediction. It shows the predicted targets, modeling technologies, evaluation metrics, and datasets, as shown in Fig. 2, to give readers basic knowledge.

2.1 Predicted targets

Application prediction consists of many aspects, mainly including the business aspect and the resource aspect; the former mainly includes the request size, functional needs, QoS level, price, and SLA parameters

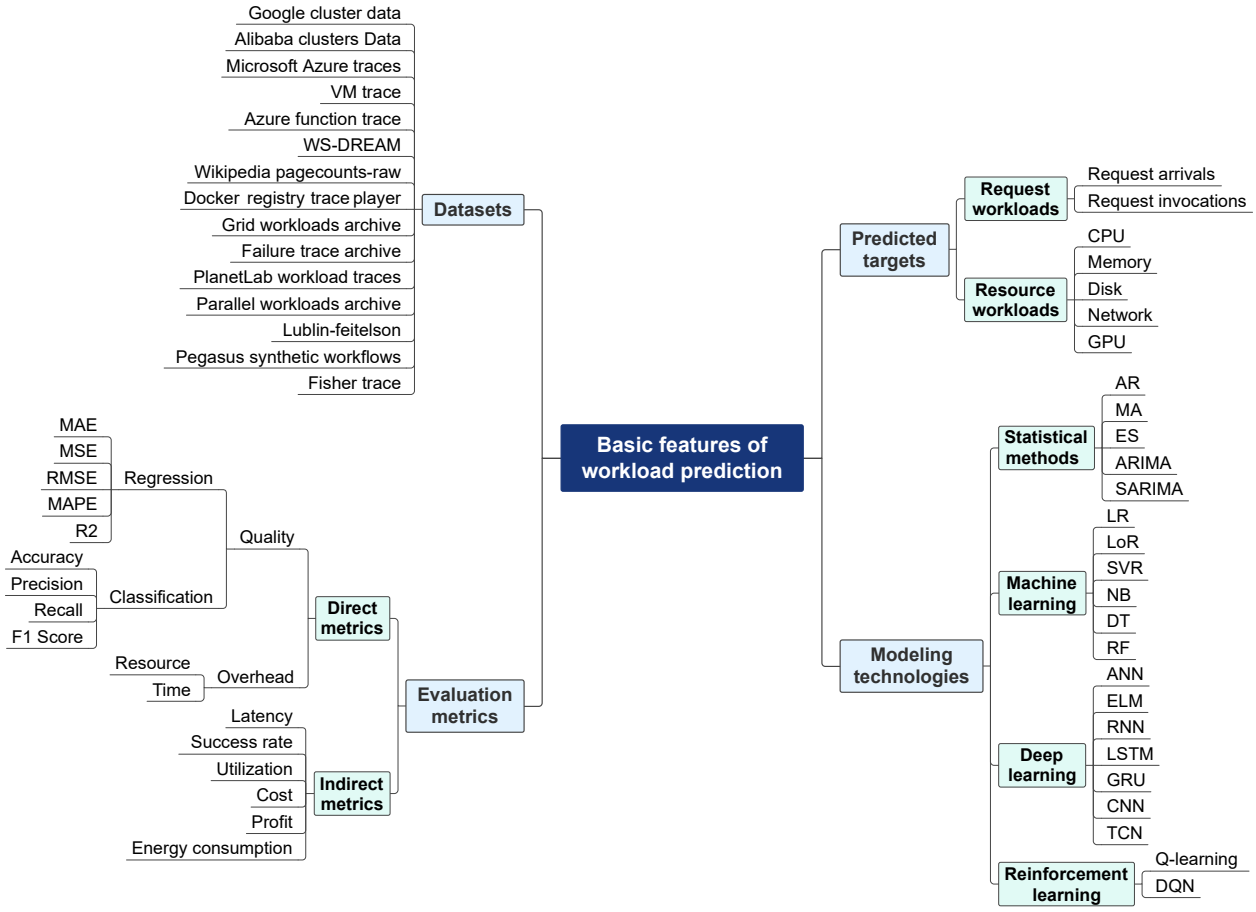

Fig. 2 Basic features of workload prediction.

related to the application business services, and the latter mainly includes the resource size, utilization rate, cost, and energy consumption related to the application resources. To deeply understand and optimize the runtime behavior of cloud applications, researchers usually carry out workload modeling and analysis from two dimensions, namely, request and resource workloads, to comprehensively depict the application's business and resource needs.

Figure 3 shows the predicted targets and distribution of the existing work investigated.

2.1.1 Request workloads

The request workload usually consists of two main types of access requests: one is external access requests, which are requests initiated directly by end-users outside the cloud (e.g., through web browsers or mobile applications), including HTTP requests, API calls, etc. The second is internal system calls, which are requests generated by different components or services within the application system calling each other. For example, one may issue a call request to another microservice in a microservice application.

2.1.2 Resource workloads

General applications usually use only regular resources, including CPU, memory, disk, and network bandwidth. CPU is the core component that executes the application's instructions. Memory is temporary storage used to store data and code with the ability to read and write data quickly. The disk is a long-term storage device, including datasets, files, and user data. The network is a component used for application communication, supporting data receiving and sending between applications and enabling internal and external communication.

Besides these regular resources, some specialized tasks require new types of heterogeneous resources, such as GPUs, FPGAs, etc. GPUs are processors dedicated to graphics rendering for accelerating compute-intensive tasks such as deep learning. FPGAs are programmable logic devices for accelerating compute-intensive tasks like encryption and decryption.

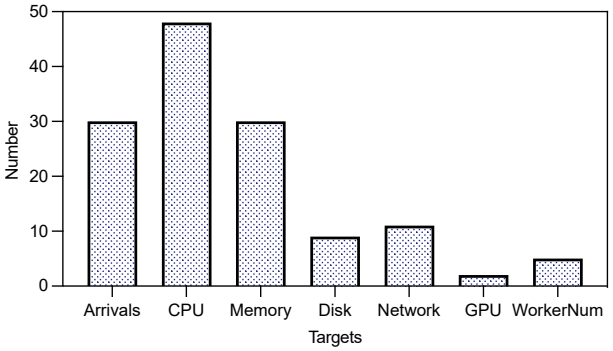

Fig. 3 Number of the schemes which predicted each target.

2.2 Modeling technologies

The basic techniques used in existing prediction solutions mainly include statistical methods, machine learning methods (ML), deep learning methods (DL), and reinforcement learning methods (RL).

Figure 4 shows the modeling technologies and distribution of the existing work investigated.

(1) **Statistical methods**: They use statistical principles and techniques to collect, analyze, and interpret data. General statistical methods for workload prediction include moving average (MA), autoregressive (AR), exponential smoothing (ES), autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA) methods, etc.

(2) **ML**: They automatically learn and improve methods from workload data, which include linear regression (LR), logistic regression (LoR), support vector machine (SVM), *K*-nearest neighbor (KNN), naive Bayes (NB), decision tree (DT), random forest (RF), etc.

(3) **DL**: They employ deep neural networks to automatically learn and extract features from large-scale, high-dimensional data, including artificial neural network (ANN), extreme learning machine (ELM), recurrent neural network (RNN), long short-term memory network (LSTM), gated recurrent unit network (GRU), convolutional neural network (CNN), and temporal convolutional neural network (TCN), etc.

(4) **RL**: They find the best prediction strategy by letting the model interact with the environment and continuously trial and error and learning, including Q-Learning, deep Q networks (DQN), etc.

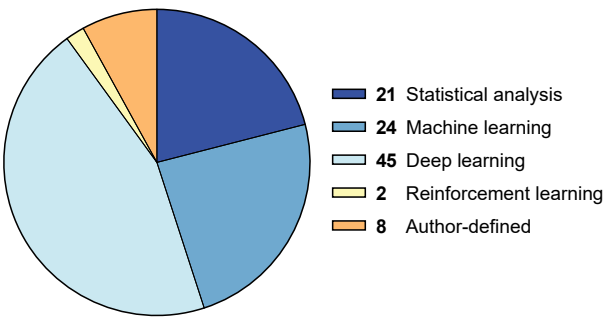

Fig. 4 Distribution of the schemes which employ each modeling technology.

2.3 Evaluation metrics

Figure 5 shows the main evaluation metrics and distribution of the existing work investigated.

2.3.1 Direct metrics

(1) Model quality evaluation: Suppose there are a total of m samples, where the estimated result of each sample is $\hat{y}_t$ and the actual result of each sample is y_t. Regression-based workload prediction models:

(a) **Mean absolute error (MAE)**: It is the average of the absolute value of the prediction error, which can accurately reflect the size of the actual prediction error. The smaller the MAE is, the better the quality of the model and the more accurate the prediction.

$$\text{MAE} = \frac{\sum_{t=1}^{m} |y_t - \hat{y}_t|}{m} \tag{1}$$

(b) **Mean square error (MSE)**: It is the average deviation between the predicted value and the true value. The smaller the MSE, the better the quality of the model and the more accurate the prediction.

$$\text{MSE} = \frac{\sum_{t=1}^{m} |y_t - \hat{y}_t|^2}{m} \tag{2}$$

(c) **Root mean square error (RMSE)**: It is the arithmetic square root of MSE. As with MSE, a smaller RMSE indicates better model quality and more accurate predictions.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{\sum_{t=1}^{m} |y_t - \hat{y}_t|^2}{m}} \tag{3}$$

(d) **Mean absolute percentage error (MAPE)**: It is a relative error measure that uses absolute values to keep the positive and negative errors from canceling one another out.

$$\text{MAPE} = \frac{1}{m} \sum_{t=1}^{m} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{4}$$

(e) **Coefficient of determination (R2)**: It is the ratio of the residual sum of squares to the total sum of squares. The closer R2 is to 1, the better the model fit.

$$R^2 = \frac{\sum_{t=1}^{m} |\hat{y}_t - \bar{y}_t|^2}{\sum_{t=1}^{m} |y_t - \bar{y}_t|^2} \tag{5}$$

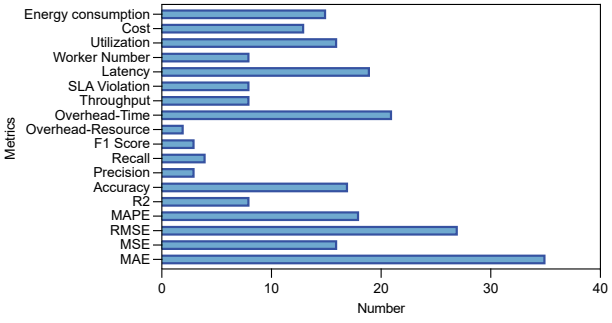

Fig. 5 Number of the schemes which employed each evaluation metric.

Classification-based workload prediction models: Let us first define the following four basic classification metrics:

(a) **True positive (TP)**: The number of samples whose truth is positive and whose prediction is also positive.

(b) **False positive (FP)**: The number of samples whose truth is negative and whose prediction is positive.

(c) **True negative (TN)**: The number of samples whose truth is negative and whose prediction is also negative.

(d) **False negative (FN)**: The number of samples whose truth is positive and whose prediction is negative.

We can obtain the following evaluation metrics:

(a) **Accuracy**: It is the ratio of the number of samples correctly predicted to the number of all samples. The higher the Accuracy, the better the overall ability of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

(b) **Precision**: It is the ratio of the number of samples correctly predicted as positive to the number of all samples predicted as positive. The higher the Precision, the better the model's reliability in predicting positive classes.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

(c) **Recall**: It is the ratio of the number of samples correctly predicted as positive to the number of all true positive samples. The higher the Recall, the better the model recognizes the positive class, and the fewer positive samples are missed.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

(d) **F1 score**: It is the reconciled mean of precision and recall. A higher F1 Score indicates that the model has better-balanced precision and recall.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

(2) Model overhead evaluation:

(a) **Resource overhead**: It includes the resource

consumption during the model generation, the storage management resource consumption after generation, and the resource consumption during the online model inference.

(b) **Time overhead**: It includes the time cost during the model generation and the time cost during the online model inference.

2.3.2 Indirect metrics

Since workload prediction models are often used for application resource management, related work also often uses some application AIOps metrics to indirectly reflect the effect of workload prediction, specifically:

(1) **Execution time**: It is also known as response time, which refers to the total time taken by a task or job to complete its execution.

(2) **Throughput**: It reflects the number of tasks processed successfully within a given period.

(3) **Success rate**: It reflects the percentage of tasks processed successfully within a given period.

(4) **SLA violation rate**: It reflects the percentage of tasks performed in violation of SLA requirements.

(5) **Resource utilization**: It reflects the allocated resource usage of applications.

(6) **Number of workers**: It reflects the number of active workers during application execution.

(7) **Cost**: It includes resource costs, violation costs, management costs, etc., reflecting the various costs involved in the running process of applications.

(8) **Profit**: It reflects the profit that the cloud provider earns by providing services to its subscribers.

(9) **Energy consumption**: It reflects the energy consumption generated during the application lifecycle, including static and dynamic energy consumption.

These metrics help us evaluate the effect of prediction models and optimize the AIOps of applications.

2.4 Datasets

Figure 6 shows the experiment ways and distribution of the existing work investigated. The following are the main public datasets:

(1) **Google cluster data**[10]: It traces data from Google's cluster management system (also known as Borg).

(2) **Alibaba cluster data**[11]: It traces data from the Alibaba production cluster and contains detailed information about the job/application.

(3) **Microsoft Azure traces**[12]: It traces data for Microsoft Azure systems, including virtual machine (VM) traces and Azure Function traces.

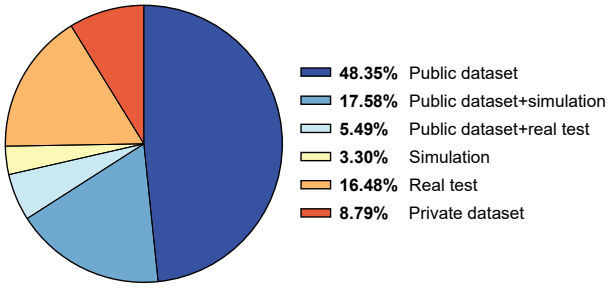

Fig. 6 Distribution of the schemes which employed each experiment way.

(4) **WS-DREAM**[13]: It maintains three datasets: (1) the QoS dataset, (2) the log dataset, and (3) the review dataset.

(5) **Wikipedia pagecounts-raw**[14]: It is a trace of web requests made to Wikipedia servers, outages, and server issues that might affect the traces.

(6) **Docker registry trace player**[15]: It is used to replay anonymized production-level traces for a registry. The traces are from the IBM docker registry.

(7) **Grid workloads archive**[16]: It is a repository of utilization traces of several grids.

(8) **Failure trace archive**[17]: It is a repository of parallel and distributed system availability traces.

(9) **PlanetLab workload traces**[18]: It is a set of CPU utilization traces from PlanetLab VMs collected during 10 random days.

(10) **Parallel workloads archive**[19]: It is a collection of traces and models of workloads for high-performance computing (HPC) machines.

(11) **Lublin-Feitelson**[20]: It is a model for parallel tasks in supercomputers.

(12) **Pegasus synthetic workflows**[21]: It is the profiling data of 20 synthetic workflow applications, each with different size options.

(13) **Fisher**[22]: It is a collection of resource and performance metrics from a real Kubernetes system, recorded for 10 containers over 30 days.

3 Application-Oriented Workload Prediction

With the development of cloud computing, software engineering, big data, and AI, applications show profound evolutionary features in cloud platforms, including user behavior, software architecture, function, runtime, and system. Two major characteristics of cloud applications affect their workload changes:

(1) **Workload variability**: Due to the dynamic cloud

environments and elastic provisioning and sharing of application resources, the workloads of cloud applications exhibit significant volatility and pattern variability. The former refers to the fact that the inherent variance instability and noise perturbation of workloads, and the latter refers to the tendency of workloads to switch between different patterns over time.

(2) **Workload heterogeneity**: Due to the diversity and dynamics of applications and user behavior, which is mainly manifested in the heterogeneity of four aspects: workload type, software architecture, runtime, and function type. The design and implementation of workload prediction models must cater to these specific demands.

Based on these two significant features, in conjunction with the evolution of applications, researchers have conducted a series of explorations, as shown in Fig. 7.

3.1 Workload variability

3.1.1 High fluctuations

Unlike HPC systems and grid computing, cloud applications are more interactive and have a higher variance of workloads. Their average noise is almost 20 times that of grid computing[7]. As a result, workload variations of cloud applications are characterized by a high degree of volatility.

On the one hand, researchers have been dedicated to optimizing the models to enhance their robustness in modeling and prediction capabilities. This ensures that the models remain effective and reliable even when dealing with highly volatile workload sequences.

(1) Linear analysis: Researchers have proposed solutions with MA, AR, ES, ARIMA, and SARIMA based on a statistical analysis of historical data to fit statistical models. ARIMA combines the features of AR and MA to capture the autoregressive and moving average effects in the time series. SARIMA adds a seasonal factor to ARIMA, which can be adapted to more scenarios. Calheiros et al.[23] developed a cloud workload prediction module with ARIMA, enabling the prediction of application resource needs and allowing for proactive allocation and release of resources. Dhib et al.[24] presented a proactive dynamic VM allocation and deployment algorithm, which estimates the resource needs of requests by SARIMA, converts the deployment problem into a multidimensional knapsack problem, and derives the optimal mapping of allocated resources. Gupta and Kumar[25] demonstrated the effectiveness of ARIMA in forecasting the workloads of mid-term daily power systems. El-Kassabi et al.[26] proposed a multi-strategy framework that monitors, predicts, and adjusts workflows with ARIMA in dynamic cloud environments.

(2) Nonlinear analysis: With the development of AI technology, machine learning and deep learning methods with more robust modeling capabilities are applied to workload prediction. Liu et al.[27] proposed a cascaded shallow model based on SVM for workload prediction of network devices. Bala and Chana[28]

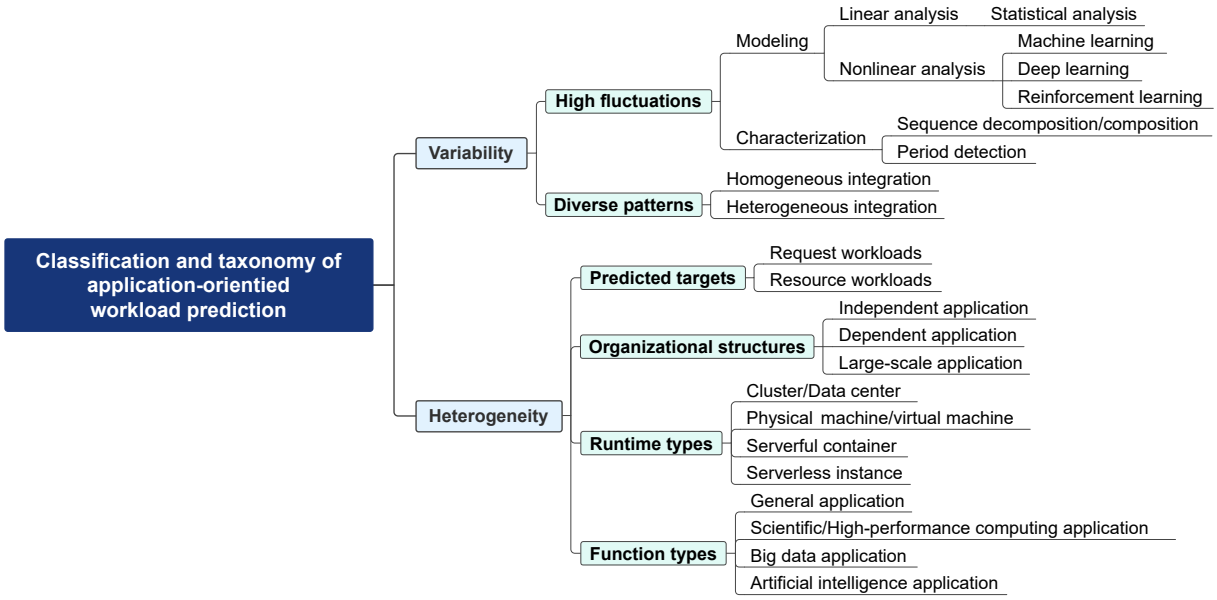

Fig. 7 Classification and taxonomy of application-oriented workload prediction.

tested machine learning algorithms such as KNN, ANN, RF, and SVM, which confirmed that RF has the highest prediction accuracy. Baig[29] introduced an adaptive model selector method that dynamically identifies the most suitable prediction method from a set of trained models. Through validation on three publicly available datasets, the authors determined that the RF exhibited superior performance as a workload predictor. The RF's exceptional performance can be attributed to two key properties: its capability to capture nonlinear relationships within the data and its ability to handle datasets with a large number of features. Lu et al.[30] proposed a rand variable learning rate backpropagation neural network to predict request arrivals in large-scale data centers. Li et al.[31] proposed a workload prediction method based on improved LSTM, which generates a new RNN architecture by splicing BiLSTM and GridLSTM, which can extract more intricate and evolving features. Yadav et al.[32] proposed a deep learning method based on LSTM to efficiently extract nonlinear features of workload variations to predict server workloads. In addition, reinforcement learning techniques have also been introduced into workload prediction. Ahamed et al.[33] utilized deep Q-learning for federated cloud workload prediction, which is a model that extracts latent patterns and optimizes VM resource allocation.

On the other hand, researchers have explored the optimization of workload prediction solutions with significant data noise in workload characterization. These studies focus on data preprocessing techniques, such as time series decomposition and period detection, to eliminate high-frequency noise from workload data or extract regular features of workload variations. By building models based on preprocessed data, the adverse effects of data noise and high heteroskedasticity on model accuracy are mitigated, resulting in improved prediction performance. Tian et al.[34] proposed a prediction method using least squares support vector machine (LSSVM) for the trend component and ARIMA for the stochastic component, which achieved better prediction results. Jeddi and Sharifian[35] decomposed the workload sequence into three layers based on wavelet transform and predicted the different components separately using grouping methods for data handling (GMDH). Kumar and Singh[36] proposed an ELM-based workload prediction model for the trend, seasonal, and stochastic components obtained from the seasonal additive decomposition method. Yazdanian and Sharifian[37] proposed an integrated workload prediction model with generative adversarial network (GAN) and LSTM, which decomposes the original sequence into constituent components with different frequency bands and then learns and predicts each component separately.

3.1.2 Diverse patterns

New workload patterns always emerge with the evolution of user behavior, applications, and environments. In addition, non-stationary workloads present different patterns that change over time, regenerating models more frequently and increasing overhead accordingly.

Therefore, integrated prediction methods have been gradually applied to cloud workload prediction, mainly because cloud workload patterns are usually diverse and irregular, and no single predictor can perform well in all workload patterns[38]. Cao et al.[39] proposed a two-layer model structure comprising an optimizer layer responsible for the ongoing merging and elimination of predictors and an integration layer responsible for generating final prediction results. Kaur et al.[40] proposed a prediction method REAP that integrates feature selection and eight machine learning methods to attain superior prediction accuracy. Von Krannichfeldt et al.[41] proposed an online integrated learning approach for workload prediction, which merges batch learning with online learning. Each batch comprises 12 local models, and their outputs are synthesized by an online regression model. Kim et al.[38] built an integrated model for workload forecasting consisting of 21 local predictors and determined their weights using a support vector regression model. Lalitha Devi and Valli[42] proposed a hybrid model based on ARIMA and ANN to predict CPU and memory utilization. ARIMA detects the linear component, and ANN analyzes the nonlinear component using the residuals derived from ARIMA. The predicted values are combined with the previous data to generate a new time series, which is then fed into a Savitzky-Golay filter to remove any inaccuracies. Bao et al.[43] proposed an integrated framework for cloud workload prediction based on adaptive pattern mining, which employs a clustering-based sampling method to generate training samples for various patterns, each of which generates a dedicated prediction model based on LSTM. An error-based weight aggregation method is further proposed to predict future workloads.

3.2 Workload heterogeneity

3.2.1 Predicted targets

(1) Request workloads: The attributes and size of request workloads are significantly heterogeneous, varying by function type, user behavior, and other factors. Li et al.[44] proposed a workload forecasting model based on ANN. Based on analyzing the statistical features of request workloads, on the one hand, the domain knowledge providing the extended structural information of workload changes is embedded into ANN for linear regression; on the other hand, the regularization with noise is combined to improve the generalization ability. Gandhi et al.[45] correctly allocated data center resources by analyzing historical request traces to identify long-term workload patterns. It then dynamically allocates capacity through two strategies: proactive resource provisioning handles estimated base workloads on a coarse time scale, while reactive resource provisioning handles excess workloads on a finer time scale.

(2) Resource workloads: Resource workloads exhibit diversity, encompassing various aspects such as CPU, memory, disk, network bandwidth, GPU, and more. These workloads are affected by the request workloads and the complex dynamics of the cloud environment. Desire et al.[46] investigated the resource workload prediction for cloud games. Each resource workload is computed based on the proposed Fractional Rider Deep LSTM network, which is used to achieve proactive resource allocation. Ullah et al.[47] proposed a multivariate time series-based workload prediction framework for multi-attribute resource allocation and established a BiLSTM model to predict the supply and utilization of multiple resources. Kaim et al.[48] developed a deep learning-based workload prediction model that applies an attention mechanism with BiLSTM and CNN to learn multivariate resource workload variations through the designed input block, feature selection block, and sequence learning block.

3.2.2 Organizational structures

With the evolution of applications from monolithic architecture to service-oriented architecture and further to microservice architecture, researchers explore how workload prediction models can incorporate the structural characteristics of applications.

(1) Independent analysis: Researchers focus only on the workload variation of the application itself for modeling. An and Zhou[49] proposed a resource need prediction method with the generic application and workload models, which takes parameterized cloud applications, workload variation, and resource profiles as inputs to derive the corresponding resource demands. Wang et al.[50] proposed an online resource prediction model for clouds that uses the trend degree to categorize workload waveforms and samples based on scalable windows. A best error gradient-boosted regression algorithm is presented for generating the prediction model to predict resource usage based on request workloads. Feng et al.[6] proposed an ensemble workload prediction model considering adaptive sliding window and temporal locality integration. The former considers the workload trend correlation, temporal correlation, and random fluctuations to maximize the prediction accuracy with low overhead. The latter proposes the concept of temporal locality for local predictor behavior, and model integration is achieved by designing a multi-class regression weighting algorithm. Matoussi and Hamrouni[51] proposed a workload prediction method considering temporal locality to predict request arrivals, which controls the computation time by dynamically setting the window size and achieves the prediction by dynamically assigning weights to different data points.

(2) Dependent analysis: Facing distributed or hierarchical applications, researchers design the prediction methods by analyzing workload changes of different components in one application. Khorsand et al.[52] proposed a hybrid resource allocation method for multi-tier applications. Based on the MAPE-k loop, SVR is used to predict the number of requests per tier, and the planner determines when and how much VMs are allocated to a specific layer, thus realizing the proactive resource allocation. Ding et al.[53] proposed an integrated prediction model based on transfer learning and online learning. Container-oriented predictors for common and individual changes are constructed, respectively, and the integrated model is obtained based on a dynamic weighting strategy, ensuring the model's availability, adaptability, and versatility. Feng and Ding[54] proposed an end-to-end workload prediction method based on deep learning and creatively put forward the concept of workload group behavior. It also proposes a container correlation calculation algorithm to guide the representation of workload group behavior and to model the relationship between the evolution of workload group behavior and future workload changes through a custom deep

network. Li et al.[55] proposed a multi-view edge workload prediction method ELASTIC based on a cloud-edge collaboration paradigm. A learnable aggregation layer captures the correlation between sites at the global phase to reduce the time overhead. A disaggregation layer combines the intra-site correlation and inter-site correlation to optimize the forecasting accuracy at the local stage.

(3) Large-scale analysis: Large-scale cloud applications may have thousands of instances. Balancing prediction accuracy and model overhead has become a serious challenge for large-scale workload prediction. Lee et al.[56] proposed a feature selection method to reduce the inference time of the prediction model. 12 features were filtered out from 87 features by statistical techniques, and it was proved that the model trained with 12 features had high accuracy and low time overhead than the model trained with 87 features. To meet the real-time demands of large-scale application resource management, Tang[57] proposed a parallel improved LSTM algorithm, which analyzes the correlation and dependence of historical workload data, constructs a two-dimensional LSTM model, and achieves dependency and weight parallelization. Chen et al.[58] proposed a periodicity-based parallel time series prediction algorithm, which designs a data compression and abstraction algorithm to handle massive datasets. The periodic pattern recognition of multi-layer time series is realized based on Fourier spectrum analysis. Huang et al.[59] proposed a multi-scale attention-based deep clustering method for large-scale workloads, which can cluster workloads with pattern changes and amplitude differences by extracting workload features at different time scales based on a multi-scale attention mechanism. Ruta et al.[60] proposed a three-layer BiLSTM model for large-scale workload forecasting of network devices, which establishes a single model to cover all devices and takes their historical workloads as input.

3.2.3 Runtime types

(1) Cluster/Data center granularity: Researchers collect workload data with coarse-grained management and subsequently conduct workload prediction for clusters/data centers. Kumar and Singh[61] developed a request workload forecasting model for data centers based on ANN and an adaptive differential evolution method. It learns and extracts workload patterns from historical data and uses evolutionary methods to train the model to minimize the impact of initial scheme selection. Kumar et al.[62] proposed a workload forecasting framework for data centers. The biphasic adaptive differential evolution learning algorithm is introduced to improve the network learning process, allowing adaptive and enhanced pattern learning and improving the model's prediction accuracy and convergence speed. Saxena and Singh[63] proposed an improved adaptive differential evolution (AADE) learning algorithm with three-dimensional adaptive ability and applied it to train the neural network for data center workload prediction. The AADE training algorithm adaptively improves neuronal connections and helps learn traces of workloads by correlating patterns extracted from historical data. Singh et al.[64] proposed a data center workload prediction model based on an evolutionary quantum neural network for the first time, which uses the computational efficiency of quantum computing to encode workload information into qubits and spread the information through the network to improve the prediction accuracy. It also uses an adaptive differential evolution algorithm to optimize the weights of qubit networks. Patel and Bedi[65] proposed a multivariable deep learning framework for workload prediction in data centers and designed a deep learning method based on multiple attention and GRU to improve prediction accuracy and reduce complexity. Karthikeyan et al.[66] proposed a tree hierarchical deep convolution neural network based on the herd optimization algorithm used for cloud data center workload forecasting. It uses the kernel correlation method to preprocess historical data, then carries on workload prediction, and uses a herd optimization algorithm to optimize the model parameters.

(2) PM/VM granularity: With the improvement of AIOps technology, the granularity of application management becomes finer, and researchers begin to pay attention to the workload prediction of **PM/VM granularity**. Nashold and Krishnan[67] proposed a neural network to predict VM's CPU usage, considering two distinct models on short-term and long-term time scales: SARIMA and LSTM. It is verified that SARIMA performs better than LSTM in long-term tasks but worse in short-term tasks, proving that LSTM is more robust. Kumar et al.[68] proposed an independent cloud server workload prediction method, which proposes prediction error feedback to enable the prediction model to learn from its recent prediction pattern and to better learn network weights based on

the blackhole algorithm. Ouhame et al.[69] proposed a CNN-LSTM model for predicting the measurement of VM resource usage, including CPU, memory, and network usage. The linear correlation among the multivariable data is filtered by the vector autoregression method, then the residual data are entered into the CNN layer to extract the complex features of each VM usage metric, and then the time information is modeled and predicted by LSTM. Dogani et al.[70] proposed a multi-step hybrid prediction method for VM workload forecasting, which uses statistical analysis to build training sets, then uses CNN to extract hidden spatial features among all related variables, and finally uses GRU network and attention mechanism to extract time correlation features.

(3) Container granularity: With the continuous improvement of virtualization technology, many applications have changed from the traditional PM/VM-based deployment to the serverful container-based deployment, so researchers began to promote the workload prediction problem of container granularity. Zhang et al.[71] proposed a hybrid model based on triple exponential smoothing and LSTM for Docker container workload forecasting, which can capture short-term and long-term dependencies in resource series and smooth resource utilization data. Two single models are combined based on MAPE error to improve the prediction accuracy. Chen and Wang[72] proposed an adaptive short-term workload forecasting algorithm, using principal component analysis to extract the main types of container demands and perform outlier detection and replacement to generate a more stationary sequence. Then, the short-term prediction method is used to select the prediction method with higher accuracy adaptively, and the weighting factor is adjusted dynamically. Xie et al.[73] proposed a hybrid model of ARIMA and triple exponential smoothing, which is responsible for mining and predicting linear and nonlinear relationships in container resource workload series, respectively. Tang et al.[22] designed a container workload prediction model in which the metric selection module provides effective input features for the prediction model, and the neural network training module uses BiLSTM to generate the prediction model to predict workloads.

(4) Serverless instance granularity: Serverless computing promotes the development of application features, whose features such as function granularity, event trigger, and support for scaling to 0 and keeping alive, bringing new research challenges to workload prediction. Roy et al.[74] proposed a probability-based workload prediction and warm-up model for serverless applications, which predicts whether a function will be called and concurrency within a specific time interval based on Fast Fourier transform and guides function warm-up and preservation. Bhattacharjee et al.[75] proposed a deep learning prediction service system for serverless applications, which quickly predicts workloads by identifying different trends, then formulates an optimization problem, and heuristically allocates computing resources through horizontal and vertical scaling. Zhao et al.[76] proposed an incremental learning predictor, which uses the function's spatial-temporal overlap codes and profiles to improve prediction accuracy through the end-to-end call path. Wei and Gao[77] studied the serverless workload prediction problem. The optimal characterization is carried out from the perspective of the characterization process to improve the prediction accuracy, and then the results and complexity of different prediction models are compared to explore the comparative advantages between the different methods, which provides a reference for the AIOps of serverless applications. Roy et al.[78] proposed a framework that employs the hot start mechanism for warming up the components of the workflows by decoupling the runtime environment from the component function code to mitigate cold start overhead and optimize the service time and service cost jointly. Poppe et al.[79] focused on reducing resource availability latency by predicting suspend/resume patterns and proactively resuming resources for each database in a serverless computing model. In addition, avoid resources that occupy short idle times to free the backend from ineffective suspend/resume workflow applications.

3.2.4 Function types

In addition to general applications, there are some types of applications that have special requirements.

(1) Scientific/High-performance computing application: It usually needs to deal with large amounts of data and complex computational tasks. Therefore, workload prediction and AIOps techniques pay more attention to the amount of general resources, such as CPU, memory, disk, and network and task computing. da Silva et al.[80] proposed a prediction method that automatically characterizes workflow task requirements, estimates task runtime, disk space, and peak memory consumption based on the input data

size, and finds correlations between these parameters. Tanash et al.[81] created a supervised ML-based workload prediction model, which adopts different typical ML algorithms to develop predictive analytic functions for Slurm to predict the amount of memory resources and the running time of each job. Newaz and Molla[82] studied the memory need prediction of various applications in a Titan supercomputer system. The maximum memory usage of jobs is predicted by identifying specific features of users and applications. Finally, a comprehensive resource workload prediction model is constructed based on RF and XGBoost.

(2) Big data computing application: It is data-oriented and mainly processes and analyzes large amounts of data, such as IoT, social media, and e-commerce data. Therefore, workload prediction and AIOps techniques pay more attention to the amount of resources, such as disk IO, network bandwidth, and data processing and transmission. Burrell et al.[83] proposed a workload prediction method to avoid unnecessary transmission of workload metrics. In addition, ML-based algorithms, including LR and LSTM, predict CPU, memory, and network utilization. Ruan et al.[84] presented a DL-based storage workload prediction method for data-intensive applications, which consists of four phases: workload acquisition, data pre-processing, time-series prediction, and data post-processing. Among them, the time series prediction stage is implemented based on LSTM.

(3) Artificial intelligence applications: It usually deals with large amounts of data and complex AI models. It is diverse and complex, which requires heterogeneous resource support, such as GPUs. Li et al.[85] proposed a cluster scheduler to solve the imbalanced resource utilization between inference and training clusters, which introduces capacity lending, where idle inference servers are lent to the training cluster, and uses LSTM to predict the usage of inference resources and can proactively recycle resources. Gu et al.[86] proposed a GPU resource management platform for DL jobs with intelligent resource estimation and scheduling, in which the proposed resource estimation method analyzes the model's hyperparameters and the job's parameters based on RF.

4 Integration with Resource Management

Figure 8 illustrates the main aspects of workload forecasting that serve resource management. On the one hand, it supports long-term decisions, including capacity planning, application deployment, and dynamic migration; on the other hand, it supports short-term decisions, including request scheduling, resource allocation, and elastic scaling.

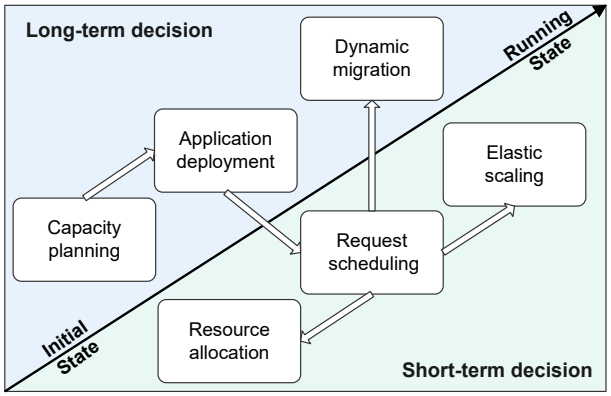

Fig. 8 Workload prediction for resource management.

4.1 Proactive capacity planning

Planning cloud infrastructure capacity for applications is a critical problem that could lead to significant service improvements, cost savings, and environmental sustainability. Capacity planning is usually long-term, which is the process of procuring machines that form the data center or cluster capacity. Liu et al.[87] introduced a workload-based model to characterize the diversified application scenarios and plan geo-distributed data centers in fast-developing economies, which transforms the problem into a quadratic programming problem and better captures the quickly changing nature of demand composition. Andreadis et al.[88] presented a capacity planning system for cloud data centers that introduces the notion of portfolios of scenarios, allowing the exploration of heterogeneous possible topologies and resources, as well as horizontal and vertical resource scaling. Newell et al.[89] presented a region-scale resource allowance system, which introduces the concept of reservation, i.e., a guaranteed amount of server capacity that functions as a logical cluster and abstracts away the complexity of data centers, hardware heterogeneity, and workload characteristics and takes a two-level approach to plan resource capacity for all data centers in a region. Le et al.[90] proposed a framework that deals with the inter-dependency of capacity planning and operational management for sustainable data centers, which provides a capacity planning decision to construct, expand, and operate the data center annually.

4.2 Proactive application deployment

Application deployment enables application instances to be placed on the cloud infrastructure, which usually determines the initial location of applications. Proactive deployment helps to prevent co-located applications from entering undesirable states, such as resource contention and waste. Zhong et al.[91] proposed a containerized task deployment algorithm that deploys co-location containers to optimize server resource utilization. It adopts the K-means algorithm for task classification among the historical traces, with reference to workload dimensions such as CPU usage, memory consumption, disk storage, and network bandwidth. Ray et al.[92] presented a reinforcement learning-based proactive mechanism for microservice placement and migration, which can prefetch and pre-provision microservices to be used in the near future by considering the microservice dependency structure while meeting the needs of previously invoked services. Li et al.[93] proposed an energy-efficient proactive caching solution for video streaming applications, which takes advantage of the available historical data to estimate the actual distribution of user requests and forms a stochastic mixed-integer programming scheme for caching and delivery scheduling. Bae and Park[94] introduced a framework for proactive service caching by taking a deep learning approach, which utilizes the ConvLSTM model for accurately predicting service popularity over time to guide the service caching on the distributed infrastructure.

4.3 Proactive request scheduling

Widespread cloud infrastructure allows applications to serve user requests from all over the globe. Combined with user behavior, request workloads exhibit significant daily cycles, holiday cycles, and intermittent surges. If the traffic spikes or plummets in a short period, it may have a considerable impact on the application service. Therefore, effective traffic sensing can guide the proactive scheduling of user requests. Kumar et al.[95] proposed an autonomous workload prediction and resource scheduling framework for industrial IoT requests. Based on the MAPE loop, an autoencoder-based deep learning model is used to predict the workloads, and the crow search optimization algorithm is used to schedule workloads to the best fog nodes. Tang et al.[96] proposed a parallel job scheduling algorithm based on workload prediction, which realizes the parallelism of the workload prediction model through 2DLSTM and implements the workload-aware job scheduling to balance computational demands and compute nodes. Niknafs et al.[97] proposed a resource manager incorporating multi-step workload prediction for heterogeneous embedded platforms. Mixed integer linear programming and heuristic scheduling are employed to meet task deadlines and minimize energy usage. Das et al.[98] proposed a spatio-temporal query framework for scheduling spatio-temporal query requests within user-supplied deadlines and budgets. It generates query parse trees, identifies geospatial services, constructs service chains, and predicts resource demands. Finally, cooperative game theory selects an appropriate query execution scheme. Fei et al.[99] proposed a method to achieve elastic task scheduling using data clustering. The number of tasks in each class cluster is predicted using ARIMA to provide a reference for resource provisioning. Then, the proposed energy-efficient resource allocation method dynamically provides resources for the tasks in each cluster. Genez et al.[100] proposed a mechanism to deal with uncertainty in available cloud bandwidth values to minimize underestimating performance and cost. A multiple linear regression method is used to compute a reduction factor as input to the hybrid scheduling program to guide decisions.

4.4 Proactive resource allocation

Resource allocation is essentially short-term capacity planning, which is the process of provisioning and allocating resources from the capacity already installed. Allocating the proper resources for applications in runtime is a key issue as cloud computing is an on-demand allocation and pay-as-you-go model. Wang et al.[101] presented a microservice elastic scheduling approach ESMS that integrates task scheduling with the instance and VM auto-scaling. It uses a statistically-based policy to allocate resources for streaming workloads and heuristically evaluates the workflow performance of different configurations. Wen et al.[102] proposed a fine-grained dynamic resource allocation method for workflow applications, which allocates resources for functions at runtime by analyzing the memory size of each function step in a workflow and considering inter- and intra-function parallelism. Safaryan et al.[103] designed a memory

allocation optimization model SLAM for serverless workflow applications. It uses distributed tracing to identify relationships between functions and estimate the workflow execution time for different memory configurations. Based on cost or time objectives, SLAM quickly searches the optimal memory configuration for each application. Feng et al. [104] proposed a resource configuration estimation method for heterogeneous workflow applications, which creates an integrated multi-task expert classifier to analyze individual and common resource usage patterns, optimizing allocation accuracy and efficiency.

4.5 Proactive elastic scaling

Elasticity is a key feature provided by the cloud computing model for applications[105], where horizontal and vertical scaling and variants of their combinations are common implementation operations. Proactive elastic scaling enables increasing and decreasing resources in advance, reducing the impact of resource scaling time and ensuring a high quality of service and cost efficiency. Singh et al.[106] proposed a workload prediction model for automatically scaling application resources. Multi-class classification based on support vector machines is used to predict future workloads so that sufficient VMs can be booted in advance. Abdullah et al.[107] proposed a burst-aware auto-scaling approach that uses workload prediction to detect bursts in dynamic workloads and automatically allocate resources to microservices, reducing response time to avoid violating service goals. Razzaq et al.[108] proposed a hybrid auto-scaled service cloud model for the smart campus system that automatically detects and manages service bursts and carries out the workload prediction and auto-scaling employing an ensemble algorithm. Zhao et al.[109] proposed a proactive optimization method based on Kubernetes auto-scaling strategy, which combines empirical modal decomposition and ARIMA to predict workloads and elastically scale the Pod instances ahead of time to reduce the response delay. Yan et al.[110] proposed a Kubernetes-based scaling system, which includes a workload prediction algorithm with an attention mechanism based on BiLSTM and a reinforcement learning method to achieve reactive and proactive scaling. Iqbal et al.[111] propose a proactive web application scaling method, where the web application access logs are analyzed by an unsupervised learning approach to capture workload characteristics, and the proactive resource auto-scaling is realized based on the predicted workload patterns.

4.6 Proactive dynamic migration

Application migration is essentially dynamic application deployment, which dynamically adjusts the mapping between application instances and the infrastructure. Therefore, it needs to implement the transition of the application from the source location to the target location. Ali Khan et al.[112] proposed a platform-independent, centralized, workload-aware resource manager that supports multiple resource types with predictors guiding the scheduler and coordinator to achieve workload-aware resource allocation and migration decisions. Liu et al.[113] developed a container consolidation solution that jointly exploits current and predicted CPU utilization based on LR. The solution is divided into three phases: 1) overutilized or underutilized physical machine detection, 2) selecting containers as migration objects, and 3) determining migration destinations. Both Tamilarasi and Akila[114] and Biswas et al.[115] investigated VM migration methods based on load balancing, which estimates server workloads. Once the estimated workloads are unbalanced, VM migration is triggered. Zeng et al.[116] proposed an energy-efficient VM consolidation framework for cloud data centers. Based on the RL algorithm, it selects the most influential VMs to alleviate the workloads of the overloaded hosts and then uses predictive-aware RL to find suitable target hosts. Pushpalatha and Ramesh[117] proposed a workload prediction-based VM migration strategy to improve energy efficiency. Neural Network is utilized for workload prediction, and VM migration is performed based on the proposed Harris Hawks Spider Monkey Optimization, where the decision-making process considers power, workload, and resource parameters.

5 Future Direction

5.1 Large-scale workload prediction

Figure 9 illustrates six future directions of workload prediction research. With the popularity of cloud computing, the size and complexity of applications are increasing dramatically. Large-scale workload prediction focuses on designing workload prediction models and mechanisms to balance workload prediction accuracy and overhead for large-scale

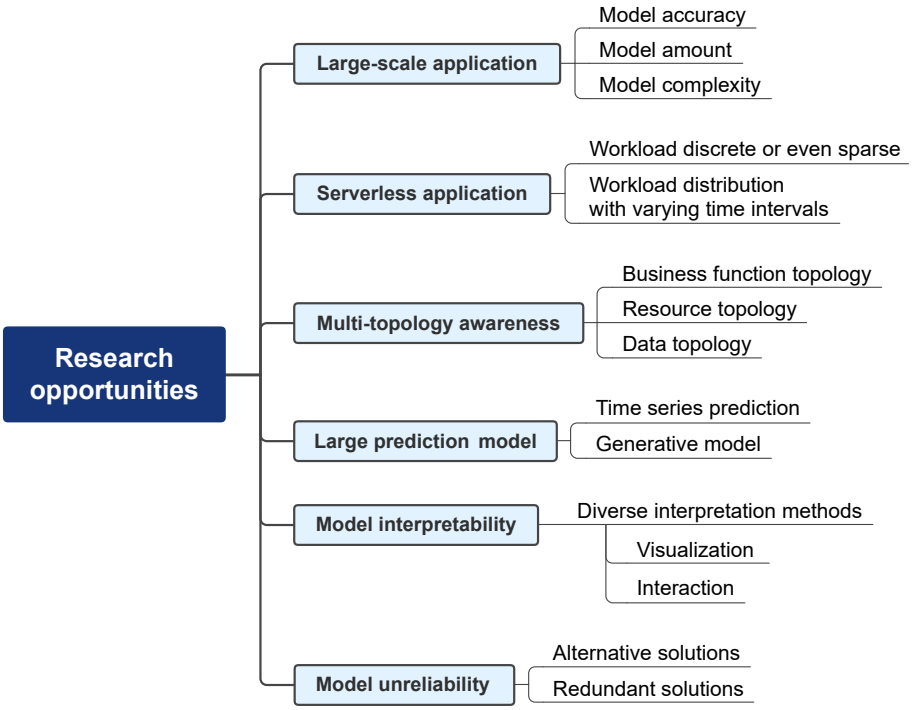

Fig. 9 Research opportunities of workload prediction.

applications. The main difficulties faced by large-scale workload prediction are reflected in (1) more prediction objects, large-scale application instances bring about large-scale workload prediction objects, resulting in the need for a large number of prediction models, huge computation and storage overheads, which greatly increase the management costs; (2) the contradiction between prediction accuracy and model overheads, where the pursuit of accuracy leads to an increase in model parameters, complexity, and number, which leads to a huge overhead, while the pursuit of overhead implies the adoption of simpler, lightweight, and parallelized models, which often makes it difficult to guarantee accuracy.

The research on large-scale workload prediction is still in development, and there is still much room for optimization. How to optimize the model number, complexity, and accuracy to improve the applicability of models to large-scale scenarios and have strong scalability is still an issue worth exploring further.

5.2 Workload prediction for serverless instances

Serverless computing provides developers with a platform without managing the underlying infrastructure. However, due to its event-driven nature, the workloads of serverless applications are highly dynamic and uncertain, which is mainly manifested in the fact that the workload data is discrete or even sparse, and the workload data may not be collected at equal intervals, which poses a tougher challenge for workload prediction.

Facing these challenges, simple analysis based only on historical workload values may be difficult. Therefore, how to effectively characterize and extract the workload patterns of serverless applications and enable the prediction models to achieve accurate awareness of uncertainty is an issue worthy of research.

5.3 Multi-topology guides workload prediction

With the development of cloud-native technologies, applications developed in a microservice architecture and deployed in a containerized distributed manner have specific business function topologies and resource topologies. In addition, the role of data as a driver for applications is becoming more significant. Therefore, a complex application may be influenced by a combination of business function topology, resource topology, and data topology rather than just the individual behavior of each object.

Therefore, how to achieve the effective perception of heterogeneous topologies to guide workload prediction for complex applications is also a worthwhile research question. Notably, graph neural networks (GNNs) have shown strong performance in several domains, especially when dealing with complex structured data.

Considering the topological dependencies of complex applications, modeling techniques such as GNN can provide new perspectives for workload prediction.

5.4 Large models applied to workload prediction

Recently, large language models (LLMs) have come into practical use due to their powerful generative and inference capabilities. They are built using self-attention mechanisms and deep learning techniques such as Transformer, allowing the models to learn more complex linguistic features and patterns. Although generative large models have been widely used in the natural language domain, it is worth noting that workload prediction, one of the key inference tasks in the AIOps of cloud applications, has not yet yielded a large model that addresses this specific task.

It is also worth studying how to extend generative large models to workload prediction scenarios by leveraging their powerful inference capabilities to enhance the AIOps level of application services in small and medium-sized enterprises.

5.5 Model interpretability

More and more machine learning models are being applied to workload prediction solutions. However, as machine learning models become more complex, they are often viewed as "black boxes". This makes it difficult to understand and interpret their predictions. Lack of model interpretability can lead to poor decision-making and resource allocation, affecting system performance and stability. Therefore, cloud providers and subscribers expect predictions to be interpretable to better understand, optimize, and leverage models.

How to utilize diverse means to enhance the interpretability of the results of complex workload prediction models is still an issue worthy of research. Future research could explore how to combine different interpretation methods and how to leverage new techniques, such as visualization and interactive analytics, to improve the interpretability of models.

5.6 Model unreliability

Due to the complexity of the cloud workload prediction problem, even the state-of-the-art prediction models cannot guarantee that the prediction results will always be reliable. In such cases, workload prediction models can cause unacceptably severe problems, leading to application degradation in terms of performance, cost, and energy consumption metrics, or even system crashes and security issues.

Therefore, it is necessary to consider how feedback and adaptive methods can be utilized to evaluate the effectiveness of workload prediction models and make timely adjustments and updates. Moreover, while continuously optimizing the prediction capacity of models, it becomes crucial to take measures to ensure the acceptability of resource management in the worst-case scenario. A robust resource management strategy should consider alternative or redundant solutions so that the resource management system avoids being overly sensitive to the workload prediction results as much as possible and has contingency capabilities.

6 Conclusion

This paper comprehensively reviews workload prediction solutions and proactive resource management solutions for cloud applications. The basic features of workload prediction are discussed, including predicted targets, modeling techniques, evaluation metrics, and datasets. Further, a classification and taxonomy methodology for application-oriented workload prediction solutions is proposed, considering the characteristics of cloud applications. In addition, a classification and taxonomy methodology of proactive resource management solutions for cloud applications is presented. Finally, we extensively analyze six directions related to workload prediction for cloud applications to develop more research ideas for readers.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62372330).

References

[1] C. Cui, B. He, C. Yu, J. Xiao, C. Li, D. Fan, S. Li, L. Mi, Z. Cao, S. Yang, et al., Astrocloud: A distributed cloud computing and application platform for astronomy, arXiv preprint arXiv:1701.05641, 2017.

[2] M. G. Avram, Advantages and challenges of adopting cloud computing from an enterprise perspective, *Procedia Technol.*, vol. 12, pp. 529–534, 2014.

[3] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, Comparison of several cloud computing platforms, in *Proc. Second Int. Symp. on Information Science and Engineering*, Shanghai, China, 2009, pp. 23–27.

[4] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, Cloud computing: Opportunities and challenges, *IEEE*

Potentials, vol. 33, no. 1, pp. 34–36, 2014.
[5] J. Viega, Cloud computing and the common man, *Computer*, vol. 42, no. 8, pp. 106–108, 2009.
[6] B. Feng, Z. Ding, and C. Jiang, FAST: A forecasting model with adaptive sliding window and time locality integration for dynamic cloud workloads, *IEEE Trans. Serv. Comput*., vol. 16, no. 2, pp. 1184–1197, 2023.
[7] M. Masdari and A. Khoshnevis, A survey and classification of the workload forecasting methods in cloud computing, *Clust. Comput.*, vol. 23, no. 4, pp. 2399–2424, 2020.
[8] D. Saxena, J. Kumar, A. K. Singh, and S. Schmid, Performance analysis of machine learning centered workload prediction models for cloud, *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1313–1330, 2023.
[9] S. Kashyap and A. Singh, Prediction-based scheduling techniques for cloud data center's workload: a systematic review, *Clust. Comput.*, vol. 26, no. 5, pp. 3209–3235, 2023.
[10] Google cluster trace, https://github.com/google/cluster-data, 2024.
[11] Alibaba/clusterdata, https://github. com/alibaba/ clusterdata, 2024.
[12] AzurePublicDataset, https://github. com/Azure/Azure-PublicDataset, 2024.
[13] WS-DREAM: Towards Open Datasets and Source Code for Web Service Research, http://wsdream.github.io/, 2024.
[14] Analytics/archive/data/pagecounts-raw, https://wikitech. wikimedia.org/w/index.php?title=Analytics/Archive/Data /Pagecounts-raw&oldid= 1912925, 2024.
[15] A. Anwar, M. Mohamed, V. Tarasov, M. Littley, L. Rupprecht, Y. Cheng, N. Zhao, D. Skourtis, A. S. Warke, H. Ludwig et al., Improving docker registry design based on production workload analysis, in *Proc. 16th USENIX Conf. File and Storage Technologies*, Oakland, CA, USA, 2018, pp. 265–278.
[16] The Grid Workloads Archive, http://gwa.ewi.tudelft.nl/, 2024.
[17] B. Javadi, D. Kondo, A. Iosup, and D. Epema, The failure trace archive: Enabling the comparison of failure measurements and models of distributed systems, *J. Parallel Distrib. Comput.*, vol. 73, no. 8, pp. 1208–1223, 2013.
[18] K. Park and V. S. Pai, CoMon, *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006.
[19] D. G. Feitelson, D. Tsafrir, and D. Krakov, Experience with using the Parallel Workloads Archive, *J. Parallel Distrib. Comput.*, vol. 74, no. 10, pp. 2967–2982, 2014.
[20] U. Lublin and D. G. Feitelson, The workload on parallel supercomputers: Modeling the characteristics of rigid jobs, *J. Parallel Distrib. Comput.*, vol. 63, no. 11, pp. 1105–1122, 2003.
[21] C. Goble, S. Soiland-Reyes, F. Bacall, S. Owen, A.Williams, I. Eguinoa, B. Droesbeke, S. Leo, L. Pireddu, L. Rodr´ıguez-Navas, et al., Implementing fair digital objects in the eosc-life workflow collaborator, *Zenodo*, 2021.
[22] X. Tang, Q. Liu, Y. Dong, J. Han, and Z. Zhang, Fisher: An efficient container load prediction model with deep neural network in clouds, in *Proc. IEEE Intl. Conf. on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications* (*ISPA/IUCC/BDCloud/SocialCom/SustainCom*), Melbourne, Australia, 2018, pp. 199–206.
[23] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, Workload prediction using ARIMA model and its impact on cloud applications' QoS, *IEEE Trans. Cloud Comput.*, vol. 3, no. 4, pp. 449–458, 2015.
[24] E. Dhib, K. Boussetta, N. Zangar, and N. Tabbane, Cost, energy, and response delay awareness-solution for cloud resources management: Proposition of a predictive dynamic algorithm for VMs allocation over a distributed cloud infrastructure, *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 4, pp. 2119–2129, 2022.
[25] A. Gupta and A. Kumar, Mid Term Daily Load Forecasting using ARIMA, Wavelet-ARIMA and Machine Learning, in *Proc. IEEE Int. Conf. Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe* (*EEEIC / I&CPS Europe*), Madrid, Spain, 2020, pp. 1–5.
[26] H. El-Kassabi, M. A. Serhani, R. Dssouli, N. Al-Qirim, and I. Taleb, Cloud workflow resource shortage prediction and fulfillment using multiple adaptation strategies, in *Proc. IEEE 11th Int. Conf. Cloud Computing* (*CLOUD*), San Francisco, CA, USA, 2018, pp. 974–977.
[27] C. Liu Shallow, Deep, ensemble models for network device workload forecasting, in *Proc. 2020 Federated Conf. Computer Science and Information Systems, Annals of Computer Science and Information Systems*, Virtual Event, 2020, pp. 101–104.
[28] A. Bala and I. Chana, Prediction-based proactive load balancing approach through VM migration, *Eng. Comput.*, vol. 32, no. 4, pp. 581–592, 2016.
[29] S. U. R. Baig, W. Iqbal, J. L. Berral, A. Erradi, and D. Carrera, Adaptive prediction models for data center resources utilization estimation, *IEEE Trans. Netw. Serv. Manage.*, vol. 16, no. 4, pp. 1681–1693, 2019.
[30] Y. Lu, J. Panneerselvam, L. Liu, and Y. Wu, RVLBPNN: A workload forecasting model for smart cloud computing, *Sci. Program*., vol. 2016, p. 5635673, 2016.
[31] S. Li, J. Bi, H. Yuan, M. Zhou, and J. Zhang, Improved LSTM-based prediction method for highly variable workload and resources in clouds, in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics* (*SMC*), Toronto, Canada, 2020, pp. 1206–1211.
[32] M. P. Yadav, N. Pal, and D. K. Yadav, Workload prediction over cloud server using time series data, in *Proc. 11th Int. Conf. Cloud Computing, Data Science & Engineering* (*Confluence*), Noida, India, 2021, pp. 267–272.
[33] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. Basuhail, and K. Jambi, Deep reinforcement learning for

workload prediction in federated cloud environments, *Sensors*, vol. 23, no. 15, pp. 6911, 2023.

[34] Z. Tian, S. Li, Y. Wang, and Y. Sha, A prediction method based on wavelet transform and multiple models fusion for chaotic time series, *Chaos Solitons Fractals*, vol. 98, pp. 158–172, 2017.

[35] S. Jeddi and S. Sharifian, A hybrid wavelet decomposer and GMDH-ELM ensemble model for Network function virtualization workload forecasting in cloud computing, *Appl. Soft Comput.*, vol. 88, p. 105940, 2020.

[36] J. Kumar and A. K. Singh, Decomposition based cloud resource demand prediction using extreme learning machines, *J. Netw. Syst. Manag.*, vol. 28, no. 4, pp. 1775–1793, 2020.

[37] P. Yazdanian and S. Sharifian, E2LG: A multiscale ensemble of LSTM/GAN deep learning architecture for multistep-ahead cloud workload prediction, *J. Supercomput*., vol. 77, no. 10, pp. 11052–11082, 2021.

[38] I. K. Kim, W. Wang, Y. Qi, and M. Humphrey, Forecasting cloud application workloads With*CloudInsight*for predictive resource management, *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1848–1863, 2022.

[39] J. Cao, J. Fu, M. Li, and J. Chen, CPU load prediction for cloud environment based on a dynamic ensemble model, *Softw. Pract. Exp*., vol. 44, no. 7, pp. 793–804, 2014.

[40] G. Kaur, A. Bala, and I. Chana, An intelligent regressive ensemble approach for predicting resource usage in cloud computing, *J. Parallel Distrib. Comput.*, vol. 123, pp. 1–12, 2019.

[41] L. Von Krannichfeldt, Y. Wang, and G. Hug, Online ensemble learning for load forecasting, *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 545–548, 2021.

[42] K. Lalitha Devi and S. Valli, Time series-based workload prediction using the statistical hybrid model for the cloud environment, *Computing*, vol. 105, no. 2, pp. 353–374, 2023.

[43] L. Bao, J. Yang, Z. Zhang, W. Liu, J. Chen, and C. Wu, On accurate prediction of cloud workloads with adaptive pattern mining, *J. Supercomput.*, vol. 79, no. 1, pp. 160–187, 2023.

[44] L. Li, M. Feng, L. Jin, S. Chen, L. Ma, and J. Gao, Domain knowledge embedding regularization neural networks for workload prediction and analysis in cloud computing, *J. Inf. Technol. Res.*, vol. 11, no. 4, pp. 137–154, 2018.

[45] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah, Minimizing data center SLA violations and power consumption via hybrid resource provisioning, in *Proc. Int. Green Computing Conf. and Workshops*, Orlando, FL, USA, 2011, pp. 1–8.

[46] K. K. D´esir´e, K. A. Francis, K. H. Kouassi, E. Dhib, N. Tabbane, and O. Asseu, "Fractional rider deep long short term memory network for workload predictionbased distributed resource allocation using spark in cloud gaming, *Engineering*, vol. 13, no. 03, pp. 135–157, 2021.

[47] F. Ullah, M. Bilal, and S.-K. Yoon, Intelligent time-series forecasting framework for non-linear dynamic workload and resource prediction in cloud, *Comput. Netw.*, vol. 225, p. 109653, 2023.

[48] A. Kaim, S. Singh, and Y. S. Patel, Ensemble CNN attention-based BiLSTM deep learning architecture for multivariate cloud workload prediction, in *Proc. 24th Int. Conf. Distributed Computing and Networking*, Kharagpur, India, 2023, pp. 342–348.

[49] C. An and J. T. Zhou, Resource demand forecasting approach based on generic cloud workload model, in *Proc. IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation* (*SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*), Guangzhou, China, 2018, pp. 554–563.

[50] X. Wang, J. Cao, D. Yang, Z. Qin, and R. Buyya, Online cloud resource prediction via scalable window waveform sampling on classified workloads, *Future Gener. Comput. Syst.*, vol. 117, pp. 338–358, 2021.

[51] W. Matoussi and T. Hamrouni, A new temporal locality-based workload prediction approach for SaaS services in a cloud environment, *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 7, pp. 3973–3987, 2022.

[52] R. Khorsand, M. Ghobaei-Arani, and M. Ramezanpour, FAHP approach for autonomic resource provisioning of multitier applications in cloud computing environments, *Softw. Pract. Exp*., vol. 48, no. 12, pp. 2147–2173, 2018.

[53] Z. Ding, B. Feng, and C. Jiang, COIN: A container workload prediction model focusing on common and individual changes in workloads, *IEEE Trans. Parallel Distrib. Syst*., vol. 33, no. 12, pp. 4738–4751, 2022.

[54] B. Feng and Z. Ding, GROUP: An end-to-end multi-step-ahead workload prediction approach focusing on workload group behavior, in *Proc. ACM Web Conf. 2023*, Austin, TX, USA, 2023, pp. 3098–3108.

[55] Y. Li, H. Yuan, Z. Fu, X. Ma, M. Xu, and S. Wang, ELASTIC: Edge workload forecasting based on collaborative cloud-edge deep learning, in *Proc. ACM Web Conf. 2023*, Austin, TX, USA, 2023, pp. 3056–3066.

[56] C. Lee, M. Song, K. Min, E. Ha, J. Lee, and W. Kim, Optimization of cloud computing workload prediction model with domain-based feature selection method, in *Proc. Int. Conf. Artificial Intelligence in Information and Communication* (*ICAIIC*), Bali, Indonesia, 2023, pp. 868–871.

[57] X. Tang, Large-scale computing systems workload prediction using parallel improved LSTM neural network, *IEEE Access*, vol. 7, pp. 40525–40533, 2019.

[58] J. Chen, K. Li, H. Rong, K. Bilal, K. Li, and P. S. Yu, A periodicity-based parallel time series prediction algorithm in cloud computing environments, *Inf. Sci. Int. J.*, vol. 496, no. C, pp. 506–537, 2019.

[59] J. Huang, C. Xiao, W. Wu, Y. Yin, and H. Chang, MADC: Multi-scale attention-based deep clustering for workload prediction, in *Proc. IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing &*

Communications, Social Computing & Networking (*ISPA/BDCloud/SocialCom/SustainCom*), New York City, NY, USA, 2021, pp. 316–323.

[60] D. Ruta, L. Cen, and Q. H. Vu, Deep Bi-directional LSTM networks for device workload forecasting, in *Proc. 2020 Federated Conf. Computer Science and Information Systems, Annals of Computer Science and Information Systems*, 2020, pp. 115–118.

[61] J. Kumar and A. K. Singh, Workload prediction in cloud using artificial neural network and adaptive differential evolution, *Future Gener. Comput. Syst.*, vol. 81, no. C, pp. 41–52, 2018.

[62] J. Kumar, D. Saxena, A. K. Singh, and A. Mohan, BiPhase adaptive learning-based neural network model for cloud datacenter workload forecasting, *Soft Comput. A Fusion Found. Methodol. Appl.*, vol. 24, no. 19, pp. 14593–14610, 2020.

[63] D. Saxena and A. K. Singh, Auto-adaptive learning-based workload forecasting in dynamic cloud environment, *Int. J. Comput. Appl.*, vol. 44, no. 6, pp. 541–551, 2022.

[64] A. K. Singh, D. Saxena, J. Kumar, and V. Gupta, A quantum approach towards the adaptive prediction of cloud workloads, *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 12, pp. 2893–2905, 2021.

[65] Y. S. Patel and J. Bedi, MAG-D: A multivariate attention network based approach for cloud workload forecasting, *Future Gener. Comput. Syst*., vol. 142, no. C, pp. 376–392, 2023.

[66] R. Karthikeyan, V. Balamurugan, R. Cyriac, and B. Sundaravadivazhagan, $COSCO_2$: AI-augmented evolutionary algorithm based workload prediction framework for sustainable cloud data centers, *Trans. Emerg. Telecommun. Technol*., vol. 34, no. 1, p. e4652.2023.

[67] L. Nashold and R. Krishnan, Using lstm and sarima models to forecast cluster cpu usage, arXiv preprint arXiv:2007.08092, 2020.

[68] J. Kumar, A. K. Singh, and R. Buyya, Self directed learning based workload forecasting model for cloud resource management, *Inf. Sci.*, vol. 543, pp. 345–366, 2021.

[69] S. Ouhame, Y. Hadi, and A. Ullah, An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model, *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10043–10055, 2021.

[70] J. Dogani, F. Khunjush, M. R. Mahmoudi, and M. Seydali, Multivariate workload and resource prediction in cloud computing using CNN and GRU by attention mechanism, *J. Supercomput.*, vol. 79, no. 3, pp. 3437–3470, 2023.

[71] L. Zhang, Y. Xie, M. Jin, P. Zhou, G. Xu, Y. Wu, D. Feng, and D. Long, A novel hybrid model for docker container workload prediction, *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 3, pp. 2726–2743, 2023.

[72] J. Chen and Y. Wang, An adaptive short-term prediction algorithm for resource demands in cloud computing, *IEEE Access*, vol. 8, pp. 53915–53930, 2020.

[73] Y. Xie, M. Jin, Z. Zou, G. Xu, D. Feng, W. Liu, and D. Long, Real-time prediction of docker container resource load based on a hybrid model of ARIMA and triple exponential smoothing, *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1386–1401, 2022.

[74] R. B. Roy, T. Patel, and D. Tiwari, IceBreaker: Warming serverless functions better with heterogeneity, in *Proc. 27th ACM Int. Conf. Architectural Support for Programming Languages and Operating Systems*, Lausanne, Switzerland, 2022, pp. 753–767.

[75] A. Bhattacharjee, A. D. Chhokra, Z. Kang, H. Sun, A. Gokhale, and G. Karsai, BARISTA: Efficient and scalable serverless serving system for deep learning prediction services, in *Proc. IEEE Int. Conf. Cloud Engineerin*g (*IC2E*), Prague, Czech Republic, 2019, pp. 23–33.

[76] L. Zhao, Y. Yang, Y. Li, X. Zhou, and K. Li, Understanding, predicting and scheduling serverless workloads under partial interference, in *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, St. Louis, MS, USA, 2021.

[77] J. Wei and M. Gao, Workload prediction of serverless computing, in *Proc. 2021 5th Int. Conf. Deep Learning Technologies* (*ICDLT*), Qingdao, China, 2021, pp. 93–99.

[78] R. B. Roy, T. Patel, and D. Tiwari, DayDream: Executing dynamic scientific workflows on serverless platforms with hot starts, in *Proc. SC22*: *Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, Dallas, TX, USA, 2022, pp. 1–18.

[79] O. Poppe, Q. Guo, W. Lang, P. Arora, M. Oslake, S. Xu, and A. Kalhan, Moneyball, *Proc. VLDB Endow.*, vol. 15, no. 6, pp. 1279–1287, 2022.

[80] R. F. da Silva, G. Juve, M. Rynge, E. Deelman, and M. Livny, Online task resource consumption prediction for scientific workflows, *Parallel Process. Lett.*, vol. 25, no. 3, p. 1541003, 2015.

[81] M. Tanash, B. Dunn, D. Andresen, W. Hsu, H. Yang, and A. Okanlawon, Improving HPC system performance by predicting job resources via supervised machine learning, in *Proc. Practice and Experience in Advanced Research Computing on Rise of the Machines* (*Learning*), Chicago, IL, USA, 2019, pp. 1–8.

[82] M. N. Newaz and M. A. Mollah, Memory usage prediction of HPC workloads using feature engineering and machine learning, in *Proc. Int. Conf. High Performance Computing in Asia-Pacific Region*, Singapore, 2023, pp. 64–74.

[83] D. Burrell, X. Chatziliadis, E. T. Zacharatou, S. Zeuch, and V. Markl, Workload prediction for iot data management systems, *BTW 2023*, 2023.

[84] L. Ruan, Y. Bai, S. Li, S. He, and L. Xiao, Workload time series prediction in storage systems: a deep learning based approach, *Clust. Comput.*, vol. 26, no. 1, pp. 25–35, 2023.

[85] J. Li, H. Xu, Y. Zhu, Z. Liu, C. Guo, and C. Wang, Lyra: Elastic scheduling for deep learning clusters, in *Proc. Eighteenth European Conf. Computer Systems*, Rome Italy, 2023, pp. 835–850.

[86] R. Gu, Y. Chen, S. Liu, H. Dai, G. Chen, K. Zhang, Y. Che, and Y. Huang, Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed GPU clusters, *IEEE Trans. Parallel Distrib. Syst*., p. 1, 2021.

[87] R. Liu, W. Sun, and W. Hu, Workload based geo-distributed data center planning in fast developing economies, *IEEE Access*, vol. 8, pp. 224269–224282, 2020.

[88] G. Andreadis, F. Mastenbroek, V. van Beek, and A. Iosup, Capelin: Data-driven compute capacity procurement for cloud datacenters using portfolios of scenarios, *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 1, pp. 26–39, 2022.

[89] A. Newell, D. Skarlatos, J. Fan, P. Kumar, M. Khutornenko, M. Pundir, Y. Zhang, M. Zhang, Y. Liu, L. Le, et al., RAS: Continuously optimized region-wide datacenter resource allocation, in *Proc. ACM SIGOPS 28th Symp. on Operating Systems Principles*, Virtual Event, Germany, 2021, pp. 505–520.

[90] T. N. Le, Z. Liu, Y. Chen, and C. Bash, Joint capacity planning and operational management for sustainable data centers and demand response, in *Proc. Seventh Int. Conf. on Future Energy Systems*, New York, NY, USA, 2016, pp. 1–12.

[91] Z. Zhong, J. He, M. A. Rodriguez, S. Erfani, R. Kotagiri, and R. Buyya, Heterogeneous task co-location in containerized cloud computing environments, in *Proc. IEEE 23rd Int. Symp. on Real-Time Distributed Computing* (*ISORC*), Nashville, TN, USA, 2020, pp. 79–88.

[92] K. Ray, A. Banerjee, and N. C. Narendra, Proactive microservice placement and migration for mobile edge computing, in *Proc. IEEE/ACM Symp. on Edge Computing* (*SEC*), San Jose, CA, USA, 2020, pp. 28–41

[93] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, and M. Pan, Energy-efficient proactive caching for adaptive video streaming via data-driven optimization, *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5549–5561, 2020.

[94] H. Bae and J. Park, Proactive service caching in a MEC system by using spatio-temporal correlation among MEC servers, *Appl. Sci.*, vol. 13, no. 22, p. 12509, 2023.

[95] M. Kumar, A. Kishor, J. K. Samariya, and A. Y. Zomaya, An autonomic workload prediction and resource allocation framework for fog-enabled industrial IoT, *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9513–9522, 2023.

[96] X. Tang, Y. Liu, T. Deng, Z. Zeng, H. Huang, Q. Wei, X. Li, and L. Yang, A job scheduling algorithm based on parallel workload prediction on computational grid, *J. Parallel Distrib. Comput.*, vol. 171, no. C, pp. 88–97, 2023.

[97] M. Niknafs, P. Eles, and Z. Peng, Runtime resource management with multiple-step-ahead workload prediction, *ACM Trans. Embed. Comput. Syst*., vol. 22, no. 4, p. 71,

[98] J. Das, S. Ghosh, S. K. Ghosh, and R. Buyya, LYRIC: Deadline and budget aware spatio-temporal query processing in cloud, *IEEE Trans. Serv. Comput*., vol. 15, no. 5, pp. 2869–2882, 2022.

[99] B. Fei, X. Zhu, D. Liu, J. Chen, W. Bao, and L. Liu, Elastic resource provisioning using data clustering in cloud service platform, *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1578–1591, 2022.

[100] T. A. L. Genez, L. F. Bittencourt, N. L. S. da Fonseca, and E. R. M. Madeira, Estimation of the available bandwidth in inter-cloud links for task scheduling in hybrid clouds, *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 62–74, 2019.

[101] S. Wang, Z. Ding, and C. Jiang, Elastic scheduling for microservice applications in clouds, *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 98–115, 2021.

[102] Z. Wen, Y. Wang, and F. Liu, StepConf: SLO-aware dynamic resource configuration for serverless function workflows, in *Proc. IEEE INFOCOM 2022 - IEEE Conf. Computer Communications*, London, UK, 2022, pp. 1868–1877.

[103] G. Safaryan, A. Jindal, M. Chadha, and M. Gerndt, SLAM: SLO-aware memory optimization for serverless applications, in *Proc. IEEE 15th Int. Conf. Cloud Computing* (*CLOUD*), Barcelona, Spain, 2022, pp. 30–39.

[104] B. Feng, Z. Ding, X. Zhou, and C. Jiang, Heterogeneity-aware proactive elastic resource allocation for serverless applications, *IEEE Trans. Serv. Comput*., pp. 1–14, 2024.

[105] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes, and J. N. de Souza, Elasticity in cloud computing: A survey, *Ann. Telecommun. Ann. Des Télécommunications*, vol. 70, no. 7, pp. 289–309, 2015.

[106] S. T. Singh, M. Tiwari, and A. S. Dhar, Machine learning based workload prediction for auto-scaling cloud applications, in *Proc. OPJU Int. Technology Conf. Emerging Technologies for Sustainable Development* (*OTCON*), Raigarh, India, 2023, pp. 1–6.

[107] M. Abdullah, W. Iqbal, J. L. Berral, J. Polo, and D. Carrera, Burst-aware predictive autoscaling for containerized microservices, *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1448–1460, 2022.

[108] M. A. Razzaq, J. A. Mahar, M. Ahmad, N. Saher, A. Mehmood, and G. S. Choi, Hybrid auto-scaled service-cloud-based predictive workload modeling and analysis for smart campus system, *IEEE Access*, vol. 9, pp. 42081–42089, 2021.

[109] A. Zhao, Q. Huang, Y. Huang, L. Zou, Z. Chen, and J. Song, Research on resource prediction model based on Kubernetes container auto-scaling technology, *IOP Conf. Ser.*: *Mater. Sci. Eng.*, vol. 569, no. 5, p. 052092, 2019.

[110] M. Yan, X. Liang, Z. Lu, J. Wu, and W. Zhang, HANSEL: Adaptive horizontal scaling of microservices using Bi-LSTM, *Appl. Soft Comput*., vol. 105, p. 107216, 2021.

[111] W. Iqbal, A. Erradi, and A. Mahmood, Dynamic workload patterns prediction for proactive auto-scaling of web applications, *J. Netw. Comput. Appl.*, vol. 124, pp. 94–107, 2018.

[112] A. Ali Khan, M. Zakarya, I. U. Rahman, R. Khan, and R.

Buyya, HeporCloud: An energy and performance efficient resource orchestrator for hybrid heterogeneous cloud computing environments, *J. Netw. Comput. Appl.*, vol. 173, p. 102869, 2021.

[113] J. Liu, S. Wang, A. Zhou, J. Xu, and F. Yang, SLA-driven container consolidation with usage prediction for green cloud computing, *Front. Comput. Sci. Sel. Publ. Chin. Univ.*, vol. 14, no. 1, pp. 42–52, 2020.

[114] P. Tamilarasi and D. Akila, Prediction based load balancing and VM migration in big data cloud environment, in *Proc. 2nd Int. Conf. Computation, Automation and Knowledge Management* (*ICCAKM*), Dubai, United Arab Emirates, 2021, pp. 123–127.

[115] N. K. Biswas, S. Banerjee, U. Biswas, and U. Ghosh, An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing, *Sustain. Energy Technol. Assess.*, vol. 45, p. 101087, 2021.

[116] J. Zeng, D. Ding, X. K. Kang, H. Xie, and Q. Yin, Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center, *IEEE Trans. Parallel Distrib. Syst.*, p. 1, 2022.

[117] R. Pushpalatha and B. Ramesh, Workload prediction based virtual machine migration and optimal switching strategy for cloud power management, *Wirel. Pers. Commun.*, vol. 123, no. 1, pp. 761–784, 2022.

Zhijun Ding received the PhD degree from Tongji University, Shanghai, China, in 2007. Currently he is a professor with Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interests include formal method, Petri nets, services computing, and workflow. He has published more than 100 papers in domestic and international academic journals and conference proceedings.

Binbin Feng received the BS degree from Shandong Jianzhu University, Jinan, China, in 2020. He is pursuing the PhD degree in Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests include service computing and microservices.