Cluster Computing (2020) 23:2399–2424
https://doi.org/10.1007/s10586-019-03010-3

A survey and classification of the workload forecasting methods in cloud computing

Mohammad Masdari[1] · Afsane Khoshnevis[1]

Received: 8 July 2019 / Revised: 1 September 2019 / Accepted: 23 October 2019 / Published online: 5 December 2019

Abstract
Workload prediction is one of the important parts of proactive resource management and auto-scaling in cloud computing. Accurate prediction of workload in cloud computing is of high importance for improving cloud performance, mitigate energy consumptions, meeting the required quality of service (QoS) level, predicting the energy consumption of data centers (DCs), and improving the cloud service providers' scalability. However, in cloud computing context workload prediction is a challenging issue and various schemes using machine learning, data mining, and mathematical methods to deal with this issue. This scheme presents an extensive literature review of the workload prediction schemes proposed in the literature to improve resource management in the cloud DCs. It first provides the required knowledge regarding the workload prediction context and presents a taxonomy of the workload prediction schemes according to their applied prediction algorithm. Moreover, the main contributions of these schemes are illustrated and their major advantages and limitation are specified. At last, the open research opportunities in the workload prediction field are focused and the concluding remarks are presented.

Keywords SVM · ANN · SVR · Deep learning · Collaborative filtering · Ensemble

1 Introduction

Cloud computing is a promising technology aimed to bring various visualized resources, software, and platforms as services to its customers based on the pay-for-use model [1]. To provide high-performance cloud services for end-users, conducting resource management in cloud DCs is of high importance [2, 3] and it can decrease the energy consumption costs as well as CO_2 emissions [4, 5]. At general, resource management scheme can be classified as reactive and proactive categories which in the first case, when the workload increases/decreases to a predefined specific threshold, resource management will be conducted [6]. But, regarding the boot time of the VMs, the reactive method cannot deal with the sudden burst of the workload [7] and may result in service level agreement (SLA) violations. On the other hand, proactive methods solve this problem by predicting the future workload of DC by recognizing the possible resource usage patterns and provisioning the required resource. Consequently, by effective prediction, the performance degradation can be deterred and idle resources can be reduced to further improve the profit. However, conducting proactive resource management is not a trivial process and variable workload of the cloud-hosted services may lead to the following problems:

- Under-provisioning: The applications do not get enough resources to process all their requests and may cause SLAV.
- Over-provisioning: Virtual resources are assigned to the application more than needed, which incurs more cost to the customer. However, up to some level, over-provisioning is required to handle the fluctuation of workload up to some level.
- Oscillation: A combination of over-provisioning and under-provisioning problems happens as a result of auto-scaling.

✉ Afsane Khoshnevis
Koshnevisafsaneh@Gmail.com

Mohammad Masdari
M.Masdari@Iaurmia.ac.ir

[1] Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Consequently, accurate workload prediction is a crucial factor in conducting effective proactive resource management schemes and allocating on-demand resources to the user requests [8]. Thus, cloud resource management systems, on one hand, should be able to allocate the desired virtual resources in order to prevent performance loss and on the other hand should prevent the resource wastage by de-allocating the idle resources(auto-scaling) [9, 10]. To deal with these issues, as shown in Fig. 1, the cloud-hosted services should be monitored and their loads should be logged. Then, this historic load data can be processed and fed into a workload predictor to forecast future load. For this purpose, various resources such as CPU, memory, network bandwidth, and even I/O operations can be employed in the prediction process. Using this information, resource management and auto-scaling schemes can scale up/down the virtual resources as needed [11]. At general, cloud resources can be scaled horizontally and vertically which in the horizontal case [12], more VMs are provisioned as predicted to deal with the future loads and in the vertical case, the existing VMs' resources should be increased [7, 13, 14], but often operating systems do not allow such changes because of the security risks [15–18].

However, workload prediction in cloud computing is a challenging issue, since unlike HPC systems and Grid computing, cloud workloads have higher variance, are shorter, more interactive, and their average noise is almost 20 times of grid computing. In addition, since cloud resources are shared by several users or tasks they may suffer from some fluctuations and also new workload patterns can continuously emerge. Besides, non-stationarity workloads in cloud infrastructure, which their pattern change over time, make retraining of the prediction models more frequent and increases the overheads correspondingly. To solve these problems and regarding the importance of accurate workload prediction in the effective resource management of the cloud DCs, a significant deal of attention has been paid for load prediction by using various mathematical models and machine learning-based prediction algorithms [8, 19–24]. This article presents a thorough investigation of the state-of-the-art workload forecasting schemes, their applied techniques, and motivations to conduct them. It categorizes these schemes regarding their applied predicting method and describes how each framework tries to predict the future load and employs these results in resource management, auto-scaling, and scheduling. After conducting an in-depth analysis of the literature, open research issues in this context are provided which can lay the foundation of future studies.

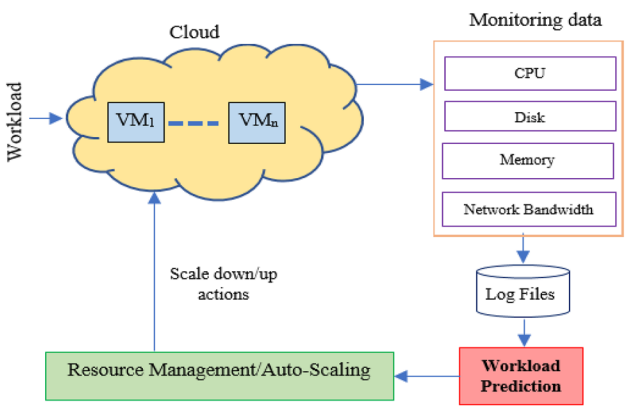

Fig. 1 Elasticity using workload prediction

To the best of our knowledge, this is the first article aimed to carry out a comprehensive study on the workload prediction schemes in the cloud computing context. The main contributions of this article are as follows:

- The background knowledge and existing challenges about the load prediction are presented.
- A classification of the recently published load prediction schemes is conducted according to their applied prediction algorithm. Also, the main contributions of each load prediction scheme are summarized and in each category of workload prediction schemes, their applied simulation factors, simulation environments, workloads, predicted factors are listed and compared.
- A critical discussion and a comprehensive comparison of the load prediction schemes are provided and their features are analyzed which can be useful in determining the future studies area.
- Illuminating the future researches challenges and open problem in the load prediction context.

The remaining of this article are organized as follows: Sect. 2 provides background concepts about load prediction and Sect. 3 presents the classification and overview of the literature. Also, Sect. 4 provides discussion and comparison results and presents the concluding issues and open research directions. Table 1 specifies the abbreviations applied in the rest of this article.

1.1 Workload prediction

Generally, the workload can be defined as all inputs requests which are sent from online interactions of the end-users with the cloud services or to batch-processed jobs. This section is trying to provide the main challenges, advantages and various details of the workload prediction in cloud DCs.

1.2 Motivations and objectives

Using workload prediction, dynamic resource management and proactive auto-scaling can achieve several important objectives. For instance, accurate forecast of the near future

Table 1 Abbreviations and Acronyms

Abbreviation	Description
ARIMA	Auto-regressive integrated moving average
ARMA	Autoregressive moving average
ANN	Artificial neural network
BPNN	Backpropagation neural network
DC	Data center
GA	Genetic algorithm
HMM	Hidden markov modeling
KNN	K-nearest neighbors
LR	Linear regression
LSTM	Long short-term memory
LTM	Long short-term memory
MAPE	Mean absolute percentage error
MAD	Mean absolute deviation
NRMSD	Normalized root mean square deviation
PSO	Particle swarm optimization
PM	Physical machine
QoS	Quality of service
RNN	Recurrent neural network
SLA	Service level agreement
SLAV	Service level agreement violation
SVM	Support vector machine
VM	Virtual machine

workload has a direct effect on the reducing response time, SLAV, over-provisioning, and under-provisioning problems. Effective handling of the workloads increases the scalability and throughput of the systems. Also, by preventing the over-provisioning of the virtual resources, the power consumption of the cloud DCs, cost, and the number of failed requests can be decreased, and customer satisfaction can be improved.

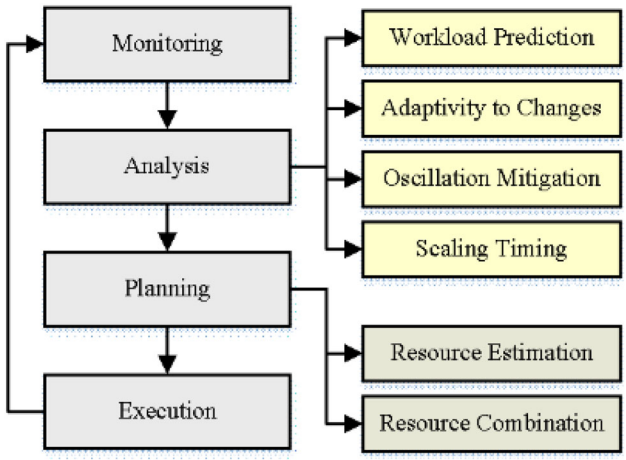

Fig. 2 MAPE loop

Figure 2 indicates the main steps of the auto-scaling process which should be executed in the cloud environment to provide elasticity and deal with the fluctuating workloads. As shown in this figure, these four steps known as MAPE loop, are monitoring, analysis, planning, and execution steps. In the monitoring step, auto-scaler should monitor the specified performance indicators to determine the need for scaling operations. In the analysis step, the auto-scaler determines whether it is necessary to perform scaling actions according to the monitored information.

To be more specific, the following issues should be considered in these items:

- Scaling timing: The auto-scaler should decide about the scaling action. It can reactively/proactively provision or de-provision the resources.
- Load prediction: if the auto-scaler is proactive, the load should be predicted accurately.
- Adaptiveness to changes: The auto-scaler should handle the changes and timely adapt its model and tunings to the new situation.
- Oscillation mitigation: scaling oscillation happens when the auto-scaler performs opposite scaling actions in a short period of time. Since this problem causes high resource wastage and SLAV, it should be prohibited.

The planning step estimates the total virtual resources which should be provisioned/de-provisioned in the next scaling action regarding constraints such as monetary cost to be more specific, the following operations will be performed in this section:

- Resource estimation: the planning step should be able to estimate how many resources are just enough to handle the current or incoming load. This is a difficult task as the auto-scaler requires to determine needed resources without being able to actually execute the scaling plan to observe the real application performance, and it has to take the specific application deployment model into account in this process.
- Resource combination: to provision resources, the auto-scaler can use vertical scaling or horizontal scaling. If horizontal scaling is employed, as the CSPs offer various types of VMs, the auto-scaler can choose one of them.

In the last step or the execution phase, the scaling plan should be executed to provision or de-provision the decided resources.

Fig. 3 Workload prediction challenges

1.3 Challenges

Figure 3 depicts the main challenges of the workload prediction in cloud computing DCs, which can be elaborated as follows [8]:

- Adaptability: The prediction model should be adaptable to the behavior changes of the hosted applications and must learn the applications dynamic behavior to decrease the prediction error. However, workload prediction schemes my fail when the workload data does not have any specific distributions.
- Proactive: Since the VM provisioning and migration are time-consuming, the prediction should be proactive. Thus, before the load burstiness occurs, the model should predict future demand so that the resource manager has enough time to provide the appropriate resources.
- Historic Data: An effective prediction model should investigate all effective resources and parameters on the workload behavior. It should consider the correlation between resources patterns extracted from historical data could show the application behavior in various dimensions and estimate the future behavior accurately. However, the proactive resource management schemes suffer from clod start problem, in which there is not required workload historic data to train the workload predictor.
- Complexity: To be efficient, time and space complexities of the prediction model should not be significant.
- Data Granularity: The initial phase for designing the prediction model is to determine which resources should be monitored. Then, the length of the sampling intervals should be defined, because the coarse-grained long-term sampling causes the model to lose the dynamism of the system while the short-term sampling, fine-grained, increases the cost of data collection and processing. It may include the details that are not useful and the model complexity increases to capture them.
- Pattern Length: Choosing the pattern length is a challenging issue and it should be selected to find the most popular patterns and the application behavior. In most prediction models, the pattern length is fixed and a sliding window is used to extract the patterns. Improper pattern length prevents the model to learn the specific patterns.

1.4 Workload type

Generally, the Cloud DCs workloads consist of a collection of diverse applications and services which have their own performance and resource requirements and by constraints specified in the form of SLAs.

The workloads can be classified according to their processing model, architectural structure, resource requirements, and non-functional requirements. In this context, regarding the processing model used by the workloads, online (interactive) and offline (batch) can be considered for them which have different behaviors, requirements, and impact on the resource management policies. For instance, an interactive workload can have short tasks, while the batch ones consist of resource-intensive and long tasks.

Also, cloud workloads can be classified according to their architectural structure expressed regarding the data flows and processing of each individual application. For example, multi-task applications can be structured by pipeline model, parallel model, and hybrid models. Furthermore, regarding the amount of applied resources workloads can be classified as I/O intensive, compute-intensive, and bandwidth sensitive. At general, network bandwidth is important for online interactive workloads, but storage and computing resources indicate batch workloads. Moreover, the resource requirements of some workloads may be stable, while as shown in Fig. 4, others may have specific temporal patterns such as periodic, bursting, growing and on/off. These patterns typically depend on the intrinsic characteristics of the applications, as well as on the workload intensity. A communication-intensive phase can be followed by a computation-intensive phase. The burstiness of the workload intensity in cloud DCs can increase resource demands and may have a negative impact on cloud performance.

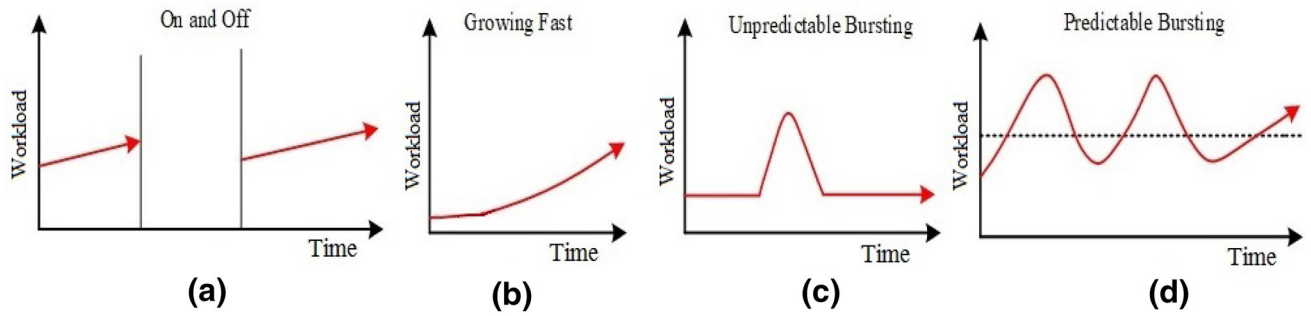

Fig. 4 Workload type

1.5 Datasets

Also, the workloads applied to evaluate the workload prediction approaches can be synthetic or real. Synthetic workloads are generated with workload generators, while real workloads can be achieved from benchmark datasets such as Google Cluster trace, NASA dataset, etc. or must be retrieved from real cloud platforms. Various datasets and workloads are used to evaluate the workload prediction approaches. Figure 5a depicts a host load from the Google traces and Fig. 5b indicates a trace load from the Auver-Grid dataset. Google workload contains over 40 million task events at minute resolution across about 12,000 hosts in 2011 over a 1 month period. These traces specify the resource and scheduling information of each task, such as scheduling class, event type, resource request, priority, resource usage rate, etc. Host load at a given time point is a total load of all running tasks on that host. Often the workload prediction schemes conduct seasonal and non-seasonal studies on the workload time series.

1.6 Evaluation factors

To evaluate the effectiveness of the workload prediction and analyze its impact on the resource the following metrics are used:

- Accuracy: The prediction models are mainly evaluated by the accuracy of their predicted results and whose outputs are closest to the actual values is the best. The deviation or error metrics measure the difference between the real behavior and the predicted behavior of the application the result of the prediction error, may result in problems such as under-provisioning and over-provisioning can. Figure 6 indicates some of the basic prediction error metrics utilized in the evaluation of the workload prediction approaches.
- Cost: prediction errors can lead to SLAV and low resources utilization. The cost metrics are employed to measure the cost resulted from the prediction error.
- Success: Success metrics specify how much the prediction method is able to forecast the future behavior of the application. Success Rate is defined as the ratio of the number of accurate estimations to the total number of estimations. The accurate prediction falls within some delta of the actual value.
- Profit: Profit metrics are applied to compute the profit of the CSP computed according to the revenue obtained from renting out the resources, preventing the SLAV and resources wastage.

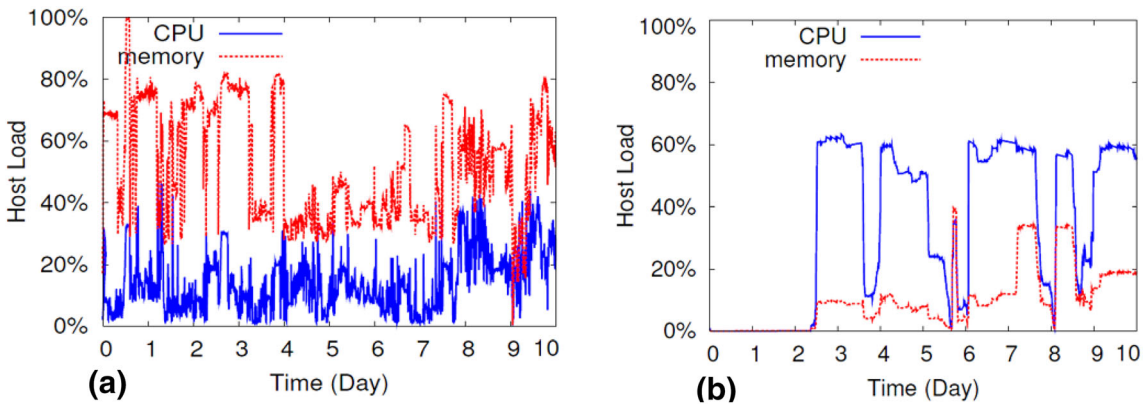

Fig. 5 A host load in two workload datasets

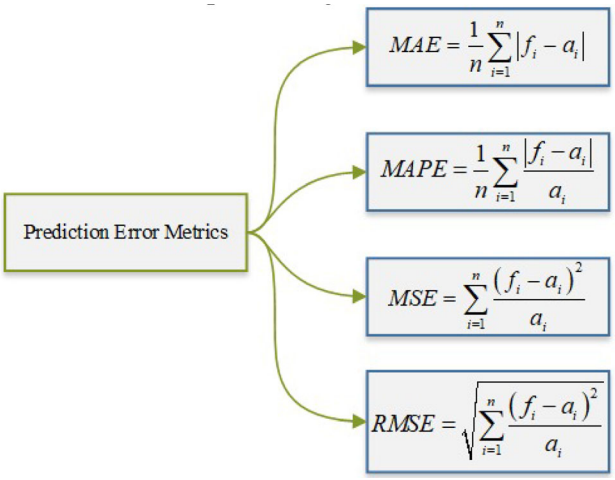

Fig. 6 Prediction error metrics

2 Load prediction schemes in cloud computing

A number of workload prediction schemes such as [25–50] have appeared in the literature. This section presents a classification of the proposed workload prediction frameworks and describes their main contributions and their utilized techniques for cloud workload prediction. Figure 7 depicts the classification of the workload prediction frameworks in the cloud computing environment according to their applied algorithms in the forecasting process. To be more specific, this section highlights the following issues about the investigated forecasting schemes:

- What are the main contributions of each workload prediction scheme?
- Which prediction algorithms are used to forecast the workload accurately?
- Which workload datasets are applied in each forecasting scheme?
- Which environments are used to evaluate each workload prediction scheme?
- Which evaluation factors are applied to assess the accuracy and effectiveness of each load forecasting scheme?
- Which resources are predicted by each scheme to recognize the incurred workload?

2.1 Regression-based schemes

This subsection is aimed to conduct a review on the regression-based workload prediction frameworks [51] designed for various cloud environments.

In [52], Antonescu et al. presented two predictive SLA-aware VM-scaling algorithms for dEIS systems for finding better scaling conditions using distributed applications derived from constant-load benchmarks, with SLA constraints. They used autoregressive predictive SLA-aware scaling to guarantee performance in the distributed cloud applications. As an advantage, the authors provide a comprehensive evaluation of their work regarding various metrics such as RMSD, execution time, number of VMs, and so on.

In [53] Yang et al. presented a linear regression model to estimate the load and applied it in an auto-scaling mechanism to scale virtual resources in real-time scaling and pre-scaling. They considered the pre-scaling using integer programming and introduced a greedy method for accurate forecasting which incurs a lower cost and SLAV.

2.1.1 ARIMA-based schemes

This subsection is aimed to conduct a review on the ARIMA-based load prediction frameworks such as [54, 55]. In [56], Li et al. presented ARIMA-DEC, a load prediction-based VM provisioning technique. This scheme employs an ARIMA-based load predictor with dynamic error compensation and applies it in TBAMP, a time-based cost-aware provisioning algorithm. ARIMA-DEC can reduce SLA default rate and TBAMP algorithm can save rental cost. TBAMP algorithm considers the cost of adjusted VMs and takes the cost of released VMs into account.

In [57], Kumar et al. tried to conduct a better forecast of the load to reduce the power cost. They compared forecast performance of the ARIMA, SARIMA (seasonal integrated ARMA), and ARFIMA (fractionally integrated ARMA) with the singular spectrum analysis method using CPU, RAM and network trace collected from Wikimedia Grid. They showed that increasing the input size does not necessarily provide better forecasting results, but the ARFIMA model suffers from high computation time when the input size increases.

In [58], Calheiros et al. provided a proactive approach for dynamic provisioning resource regarding forecasting performed with the ARIMA model. It applies a load analyzer component which provides its estimations to the other components to enable them to properly scale the resources. However, because of the limitations of the ARIMA model, this model is not able to predict the peak resource consumption.

In [59], Messias et al. tried to predict requests arriving in the next time period to prevent overloading. This problem becomes complicated when historical data is not available to be evaluated. They proposed a prediction approach using GA to aggregate time series-based forecasting models. The authors conducted workload prediction using the ARMA and ARIMA methods. They also applied the Holt-Winters

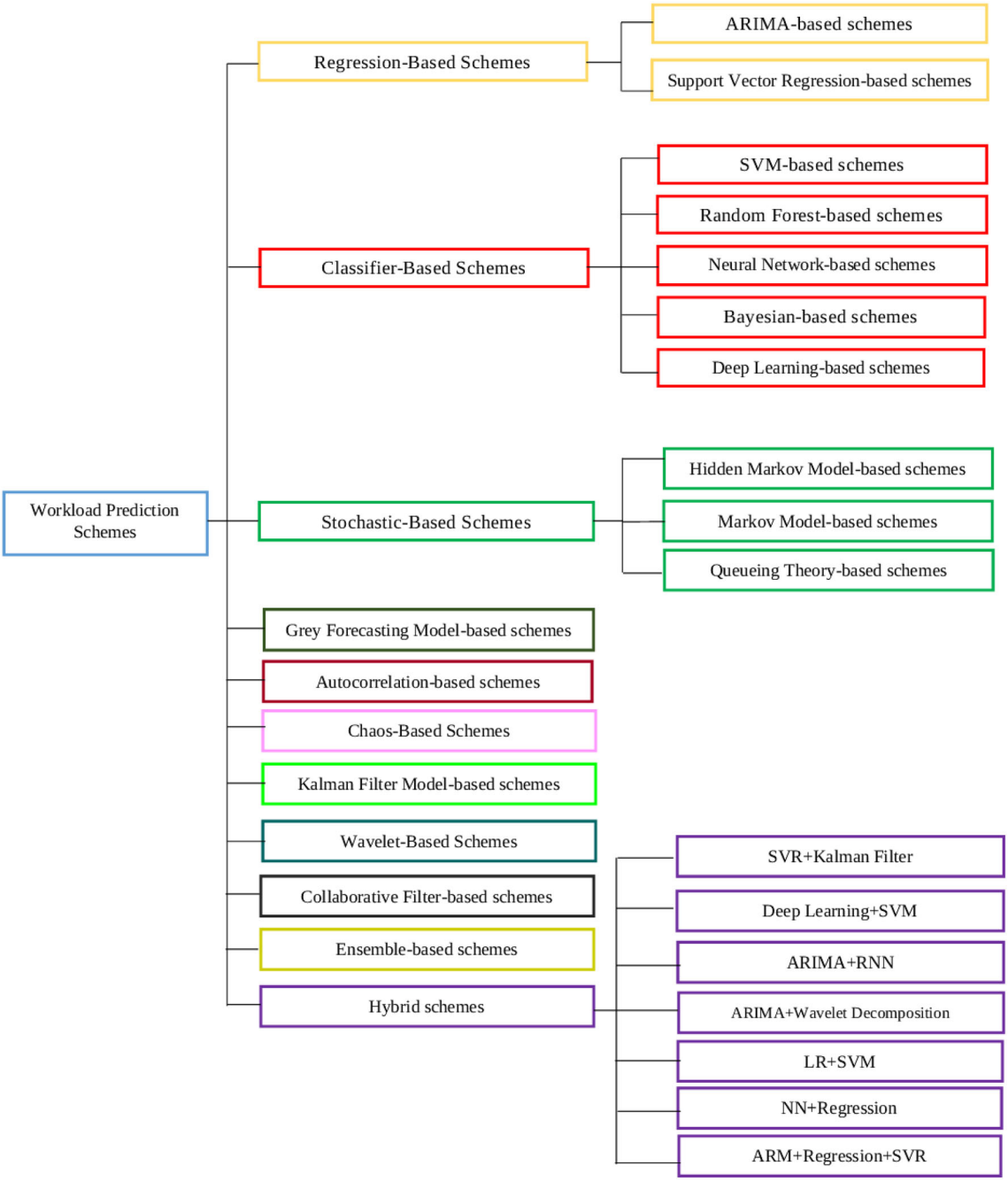

Fig. 7 Taxonomy of the load prediction schemes in cloud computing

approach to capture seasonality, but they do not provide a cost model to be optimized.

2.1.2 Support vector regression-based schemes

This subsection is aimed to conduct a review on the support vector regression or SVR-based load prediction frameworks which a number of them have been proposed in the workload prediction literature. For instance, in [60], Barati et al. provided TSVR, a tuned SVM-based approach which trains three SVR-based factors using the GA and PSO algorithms. It uses a chaotic sequence to improve prediction accuracy and prevented premature convergence by increasing the exploration and diversity in the search space. It also reduces the computational burden of generating random numbers in comparison to GA. In addition, kernel-based methods are applied to forecast memory and CPU loads. They performed simulation using Google cloud traces. Nevertheless, the TSVR takes a long time for tuning SVR parameters at the beginning of the algorithm.

The work in [61], provided a decision-maker to handle the VM migration by estimating the load and combining it with predicted performance factors of the migration process. Thus, the migration can be started when the required resources are available and no performance degradation of applications happen. Figure 8 depicts the architecture of load prediction in this scheme.

Table 2 determines the datasets, simulation software, evaluation factors, and predicted factors applied in the evaluation of the regression-based schemes.

2.2 Classifier-based schemes

This part of the paper discusses the load forecasting approaches which have applied various types of classifiers for workload prediction.

2.2.1 SVM-based schemes

This subsection is aimed to conduct a review on the SVM-based load prediction frameworks designed for various cloud environments. For instance, in [62], Tong et al. proposed a feature periodical coefficient and some existed classification methods are implemented. Experiments on the real-world dataset invalidate the efficiency of the new proposed feature, which is in the most effective combinations of features, it boosts successful rate and decreases the MSE. The SVM method can achieve nearly the same performance as the Bayes methods and their performance is higher.

In [63], the authors presented WWSVM, a load prediction model using weighted wavelet SVM to estimate the PMs' load in the cloud DC. They used the wavelet transform as a kernel function in the SVM to assign a weight to the samples according to their importance and enhance the prediction accuracy. They have applied the PSO algorithm for parameter optimization and used the Google dataset to verify their approach. As shown in Fig. 9, this scheme consists of data preprocessing and load prediction phases, in which the first phase performs workload normalization and autocorrelation analysis. To validate the performance of this load prediction scheme, experiments are conducted using the Google dataset and chose CPU utilization in the load prediction process.

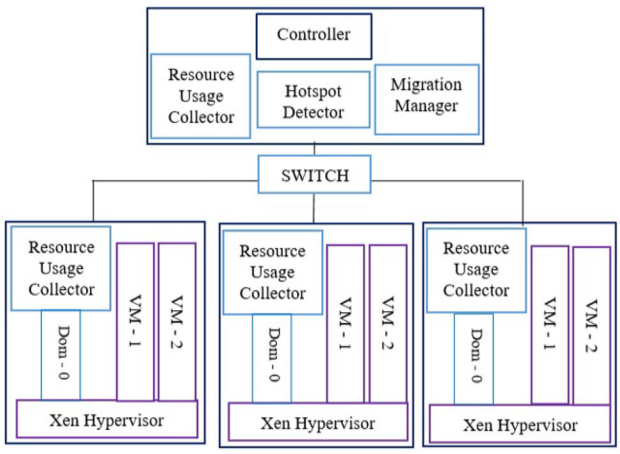

Fig. 8 Dynamic resource provisioning in [61]

In [64], Nikravesh et al. try to improve the prediction accuracy of auto-scaling using SVM and ANN classification. They indicated that prediction accuracy of SVM and ANN depends on their load pattern, but, SVM provides better prediction accuracy with periodic and increasing load patterns, while ANN has better results in forecasting unpredicted load patterns. They evaluate this scheme by using Amazon EC2.

2.2.2 Random forest-based schemes

In [65], Cetinski et al. provided AME-WPC, a model for workload forecasting in the DCs which improves the prediction accuracy. They handled load prediction using classification and regression methods and tested it with the random forest classifier. The architecture of this approach is depicted in Fig. 10. But, the influencing events in the workload fluctuations are not considered in this scheme.

2.2.3 Artificial neural network-based schemes

This subsection is aimed to conduct a review on the ANN-based workload estimation schemes [66–70] designed for cloud environments. For instance, in [71], Imam et al. employed a time delay ANN and a regression method to forecast jitter in the load. This regression model applies moderately to the trace, as evident by spline interpolation. Nevertheless, the analysis depicts more improvement in regression modeling techniques when dealing with such traces.

The work provided in [72], introduced POSITING, a forecasting model which conducts the sequential pattern mining, applies the correlation between various resources and finds applications' behavioral pattern. They investigated the capabilities of online learning for POSITING to provide reliable results, but it is not adaptable to the load variations. As an advantage, this scheme considers the correlation between different resources and extracts behavioral patterns of applications independently.

In [73], Kumar, et al. proposed a load prediction model using ANN and DE algorithm which is capable of learning proper mutation method and crossover rate. The simulations performed on NASA provided HTTP traces. As an advantage, this scheme avoids the risk of being trapped in

Table 2 Comparison of the regression-based schemes load estimation solutions

Schemes	Datasets/workloads						Simulators/environments		
	Google	Self-collected	NASA	EPC	Wikipedia	LANA	MATLAB	Amazon	CloudSim
[56]	✔							✔	
[57]		✔							
[58]		✔							✔
[59]	✔						✔		
[60]		✔						✔	
[61]		✔							✔
[55]		✔							✔

Schemes	Evaluation factors						Predicted factors				
	RMSD	MAPE	NRMSD	MAD	Cost	Exaction Time	CPU	Disk	I/O	Memory	Bandwidth
[56]				✔	✔		✔				✔
[57]						✔					
[58]	✔	✔	✔	✔		✔				✔	
[59]							✔		✔		
[60]						✔	✔			✔	
[61]		✔				✔	✔			✔	✔
[55]										✔	

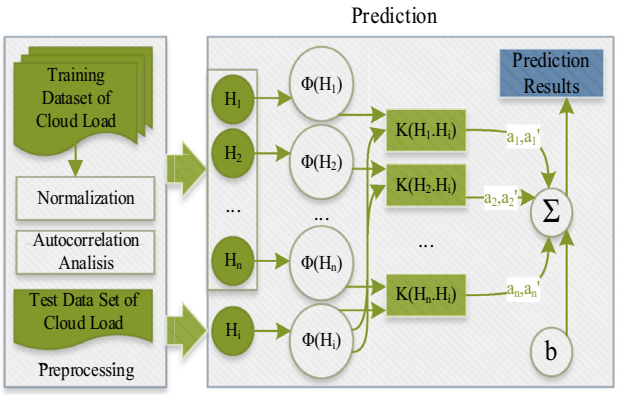

Fig. 9 The block diagram of the prediction model in [63]

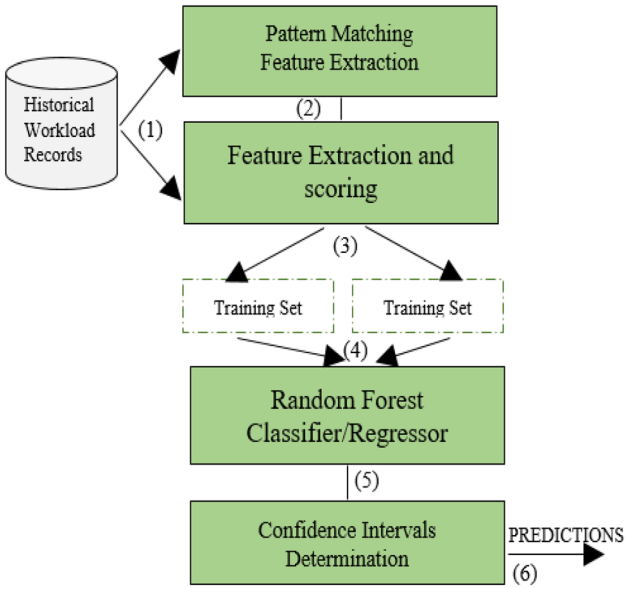

Fig. 10 Load forecasting in [65]

local optima. Figure 11 exhibits the ANN structure applied in this scheme.

In [74], Lu et al. introduce RVLBPNN, a load prediction model which uses the BPNN algorithm to exploit the relationships among the arriving loads. RVLBPNN improves prediction accuracy compared to the HMM and naive Bayes classifier-based models by a considerable margin. However, issues such as periodicity of the workload are not considered in this scheme.

In [75], Zhou et al. presented a solution for dynamic load-based on AHPGD and HHGA-RBF ANN which focuses on the load balancing of the allocation of user request tasks in a cloud. This load prediction model uses a hybrid hierarchical GA and the recursive least-squares method to train parameters of RBF ANNs. It is aggregated with the weighted round-robin algorithm and updates the weights of each node within the time period. They proposed three modules in their algorithm: node load information monitoring module, load prediction module, and request scheduling module. The architecture of this scheme is shown in Fig. 12.

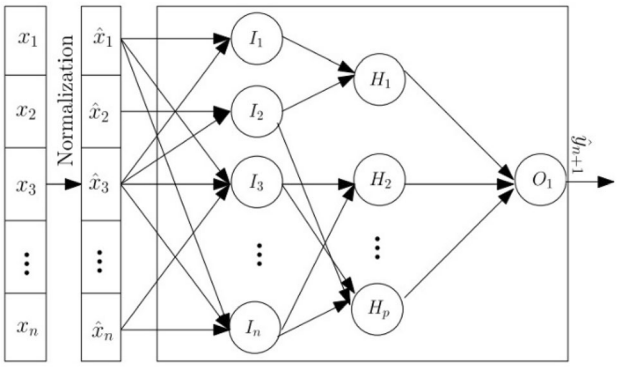

Fig. 11 Load predictor model in [73]

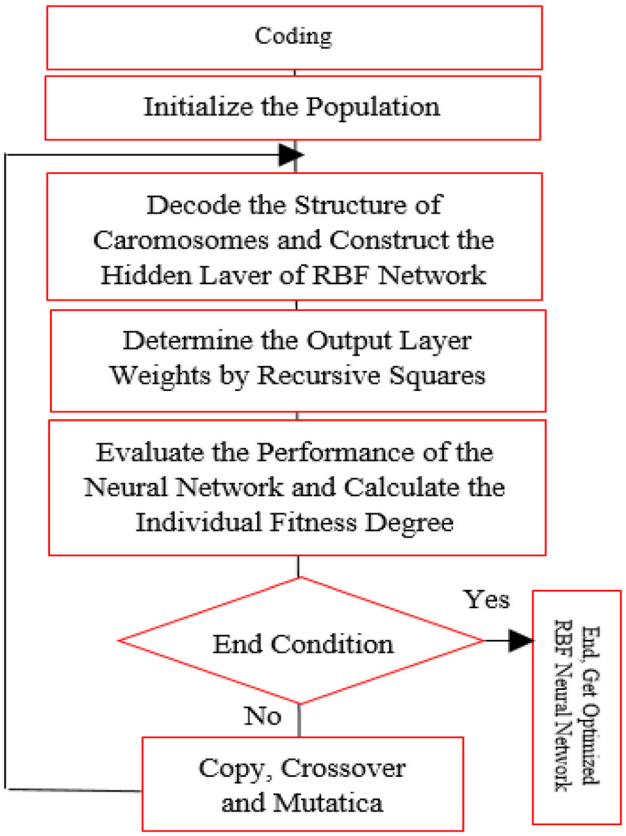

Fig. 12 RBF neural network training by HHGA in [75]

In, Imam, et al. presented a resource allocation scheme to support the increasing need for VMs. They used time delay ANN and regression techniques for load prediction. They utilized real load traces for performance evaluation to show that time delay ANN can predict the load in a cloud environment.

2.2.4 Bayesian-based schemes

This subsection is aimed to conduct a review on the Bayesian-based schemes [76–79] designed for load prediction frameworks various cloud environments.

In [80], Di et al. proposed a forecasting method to estimate load over long-term intervals and the average load in future time intervals, based on the Bayes model. They detected predictive features of the load to capture the predictability and host load pattern. They determined the effective combinations of these features for prediction. As an advantage, this scheme can detect the mean load for the future hours with high accuracy and low MSE, regardless of fluctuations.

In [81], Dietrich et al. provided a linear predictor for Least Mean Squares, a regression model system parameter identification. Load fluctuation is estimated via a linear-in-parameters model. This observation reduces the complexity of parameter estimation as the LMS learns the parameters of the model iteratively as the game progresses. However, the LMS cannot always outperform a hand-tuned PID controller.

In [82], Tian et al. Minimizing Content Reorganization and Tolerating Imperfect Workload Prediction for cloud-based Video-on-Demand Services Nguyen et al. try to reduce content reorganization and tolerate imperfect load forecasting. They presented a video-on-demand servicing system according to a pay-as-you-go cloud. They proposed a load absorber and designed a provisioning algorithm called Absorb Window. Load absorbers eliminate the bandwidth wastage and reduce the content reorganization. The architecture of this approach is depicted in Fig. 13.

2.2.5 Deep learning-based schemes

Deep learning approaches are suitable for long-term prediction of workloads and their performance can be further

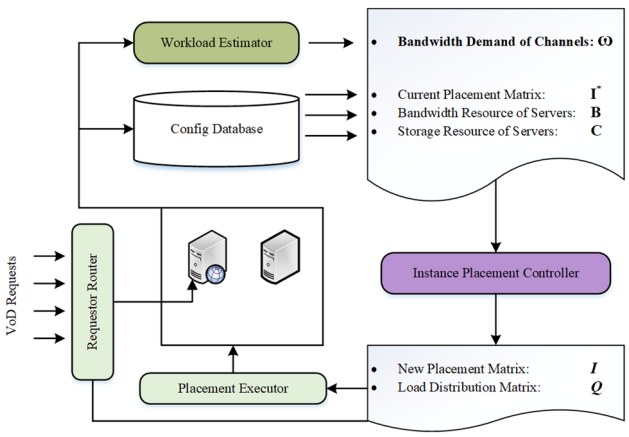

Fig. 13 Control loop in [82]

improved by increasing the size of training data and the depth of the model. A number of deep learning-based approaches are provided to forecast workload in the cloud DCs. For instance, in [83], Patel et al. tried to find a correlation among the workload of VMs regarding and predicted the workload of the next VMs with accuracy. In addition, they optimized granularity of training data, activation functions, and the number of layers. They have used predicted workload information for VM management and migration plan choice will be transferred to application provisioner which will receive the accepted user request and apply the suitable VM placement strategy to map the VM to PMs. They evaluated the effectiveness of their deep learning model using PlanetLab traces and showed that the LSTM can improve the performance of workload prediction while convolutional ANN gives a low performance. The architecture of this approach is depicted in Fig. 14. Their model receives the CPU utilization of VMs as input and forecast CPU utilization in the future.

In [84], Gupta et al. applied multivariate LSTM models to forecast resource usage in the cloud DCs. They used the Google cluster traces and evaluated the LSTM model and bidirectional LSTM model with fractional difference-based methods. They indicated the LSTM model long-range dependencies in time series-based resource consumption data and produced better out of sample estimations. As an advantage, these multivariate extensions of LSTM and BLSTM models generate better estimations than univariate ones.

In [85], Zhang et al. introduced a deep learning model using the canonical polyadic decomposition to forecast the cloud load. They used the deep learning model to learn important features of the complex load data in VMs and applied the canonical polyadic decomposition to compress parameters to enhance the training efficiency. Table 3 determines the datasets, environments, evolution factors, and predicted factors applied in the classifier-based load forecasting schemes.

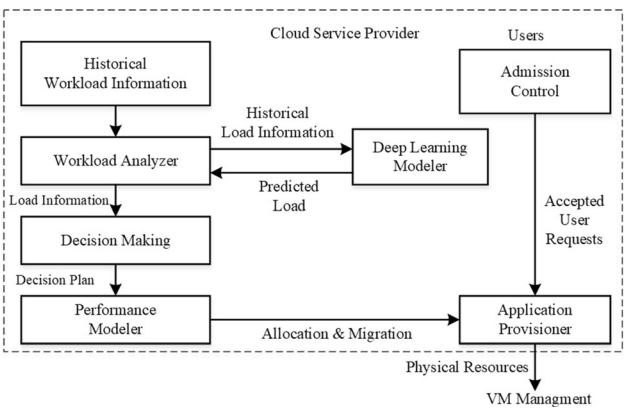

Fig. 14 Utilization-aware load forecasting in [83]

2.3 Stochastic-based workload prediction schemes

This subsection addresses the stochastic prediction schemes designed to estimate various loads in the cloud DCs using stochastic models.

2.3.1 Markov model-based schemes

A Markov chain is a mathematical tool to model a system during the time which experiences the transition from one state to another according to certain probabilistic rules. Markov chains can be classified as discrete-time and continuous-time Markov chains. Also, based on the number of previous states which they consider for deciding the next state, they can be classified as the first order and high order Markov chains. By definition, in the first-order Markov, each state only depends on its previous state, while in the high order Markov, each state depends on some of its predecessors. Markov chain models are successfully applied by various schemes such as [86, 87] to model the workload prediction. This subsection is aimed to conduct a review on the Markov chain-based schemes such as [88]. For example, in [89], Pacheco et al. studied the web load fluctuations to find how to achieve virtual resources in fluctuating traffic. They investigated the Markovian arrival processes or MAP and the related M/M/1 queueing model for performance forecasting of the deployed servers. MAPs are a special type of Markov models applied as a compact description of the time-varying characteristics of loads. MAPs can be used for heavy-tail distributions in HTTP traffic and can be applied within analytical queueing models to estimate system performance.

In [90], Shen et al. presented CloudScale, to automate resource scaling by using requests prediction and prediction error handling. It deals with scaling conflicts using migration. They used CloudScale on top of the Xen hypervisor and conducted simulations using the RUBiS benchmark driven by real Web server traces. As an advantage, this scheme employs DVFS for mitigating the energy usage regarding the SLA.

2.3.2 Hidden markov model-based schemes

HMM, or hidden Markov model is one of the most widely applied statistical Markov modeling tools for discrete-time series [91]. In contrast to the Markov chain models where all states are visible, an HMM uses hidden states which are unobservable. The HMM can be used to predict the future state of a stochastic variable. HMM are also used for workload prediction. For example, in [92], Khan et al. provide a co-clustering solution to find groups of VMs that

Table 3 Comparison of the classifier-based workload prediction schemes

Schemes	Datasets/workloads				Simulators/environments		
	Google	Self-collected	Planet Lab	LANA	MATLAB	Amazon EC2	CloudSim
[62]		✔					
[63]		✔					✔
[65]		✔					✔
[66]		✔			✔		
[67]	✔				✔		
[68]			✔			✔	
[69]		✔					✔
[70]	✔						✔
[71]		✔				✔	
[72]				✔			✔
[73]		✔				✔	
[74]		✔					✔
[75]				✔			✔
[76]		✔					✔

Schemes	Evaluation factors									Predicted factors				
	CPU	Cost	NMSE	RMSE	MSE	MAPE	Exaction Time	Error	Workload	CPU	Disk	I/O	Memory	Bandwidth
[62]						✔				✔				
[63]				✔		✔		✔				✔		✔
[65]			✔		✔				✔	✔	✔			
[66]						✔				✔			✔	
[67]			✔			✔	✔			✔	✔			
[68]									✔				✔	
[69]	✔													
[70]		✔								✔				
[71]									✔					
[72]									✔	✔		✔	✔	
[73]										✔				✔
[74]								✔						
[75]	✔									✔	✔		✔	✔
[76]	✔									✔				

have correlated load patterns and their activation periods. They introduced an HMM-based method to detect the temporal correlations in the VM clusters and to forecast fluctuation in their pattern.

In [56], Xu et al. tried to forecast and categorize the short-term cloud load using an HMM-based clustering approach. The Bayesian information criterion and Akaike information criterion are used to find the optimal HMM model size and cluster numbers. Trained HMMs are applied to detect the cluster that may possess the current load and with its data, a GA optimized Elman network is provided to forecast future load. Figure 15 depicts the block diagram of this forecasting scheme. However, they have not considered the correlation among the CPU, memory, and disk workloads.

2.3.3 Queuing model-based schemes

This subsection addresses schemes such as [93–95] which have used queueing models for workload prediction. For instance, in [96], Sahni et al. provide a heterogeneity-aware solution to handle the dynamic loads and keep the required QoS level. It conducts estimation using online resources profiling and workload history. It also provides the required resource configurations to achieve QoS at reduced cost and improved resource utilization. It captures the performance variation in the VMs and uses the request arrival pattern

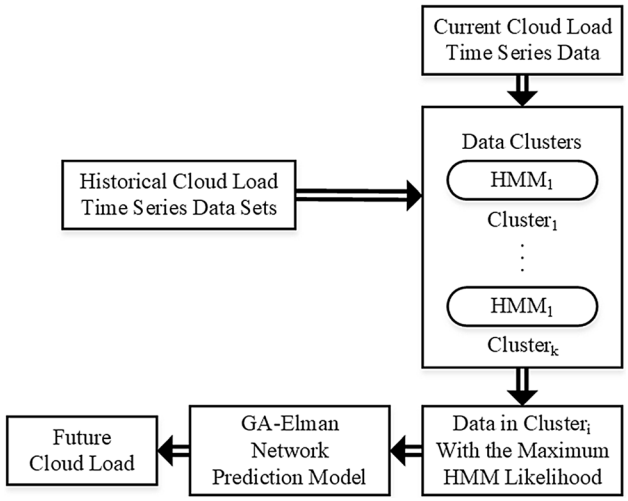

Fig. 15 Block diagram of the forecasting process in [56]

and the service rate to configure resources. However, this model only considers independent applications and does not support dependencies among the incoming requests.

The work in [97], provided a VM level resource auto-scaling scheme for a web application which can forecast its requests to determine optimal resource demand using queuing theory and multi-objective optimization. This scheme takes into account factors such as cost, latency, and SLAV factors in each time-unit re-assignment. They employed the Amazon cloud and evaluated their scheme using three real datasets.

Table 4 gives the datasets, simulation environments, and evaluation factors applied in the evaluation of the stochastic workload forecasting schemes.

2.4 Grey predicting-based schemes

The scheme in [98], presented a load predicting approach using grey predicting model to allocate VMs. The authors have used the time-dependent load in the same period in each day and forecasted whether the VM load tendency is towards increasing or decreasing? They have compared the forecasted value with the workload of the previous time period, and decide which VM in the PM should be migrated to have a balanced workload and less energy usage. Their experiments indicated that this scheme uses fewer data in the prediction process and can allocate the VMs resources with energy-saving. The architecture of this approach is depicted in Fig. 16.

2.5 Autocorrelation clustering-based schemes

The work in [99], Kluge et al. have employed autocorrelation clustering to predict the load of a periodic soft real-time application. Using this forecasting method, they tuned

Table 4 Comparison of the stochastic load prediction schemes

Schemes	Datasets/workloads					Simulators/environments			Evaluation factors		Predicted factors				
	Google	Self-collected	NASA	Saskatchewan	Wikipedia	MATLAB	CloudSim	Amazon EC2	MAPE	CPU Workload	CPU	Disk I/O	Memory	Bandwidth	Error
[92]			✓	✓				✓	✓		✓		✓	✓	
[56]		✓					✓				✓				
[89]		✓											✓		
[88]					✓				✓		✓				✓

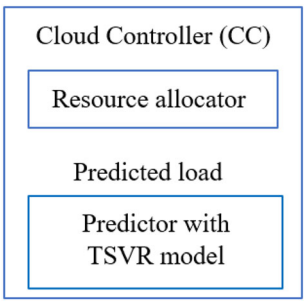

Fig. 16 Dynamic resource management in [98]

the processor performance to meet all deadlines. Nevertheless, they have not handled the numerical instabilities induced by the implicit rounding during the autocorrelation clustering algorithm execution.

2.6 Chaos-based schemes

In [100], Ardagna et al. applied capacity allocation techniques to coordinate multiple distributed resource controllers working in geographically distributed cloud sites. Capacity allocation solutions are integrated with a load redirection mechanism which forwards the incoming requests between various domains. The advantages include reducing the costs of the allocated VMs and meeting QoS constraints such as the average response time.

In [101], Qazi et al. presented PoWER that tries to predict the behavior of the cluster and distributes VMs in the cluster and turns off unused PMs for reducing power consumption. They have used chaos theory to make prediction indifferent to the loads' type and inherent cycles in them, and by conducting experiments indicated that their approach outperforms better than FFT-based time series method in load prediction.

2.7 Kalman filter model-based schemes

In [102], Hu et al. presented three models to estimate load using a Kalman filter model and put forward a pattern matching model to forecast the load. They applied its results to provide a new trigger strategy for cloud elasticity automatic scaling mechanism. This model improves the forecasting accuracy and reduces the automatic scaling delay, but it should be extended to support other workload prediction scenarios and improve its predicting accuracy. Table 5 determines the datasets, simulation software, evolution factors, and predicted factors applied in the outlined workload prediction schemes.

Table 5 Comparison of the grey predicting, autocorrelation, chaos, and Kalman filter-based workload prediction schemes

Schemes	Datasets/workloads			Simulator/environment			Evaluation factors			Predicted factors				
	Google	Self-collected	Wikipedia	CloudSim	Amazon EC2	MATLAB	MAPE	Cost	CPU	Error	Power Consumption	CPU	Disk	I/O	Memory	Bandwidth
[86]	↘				↘			↘		↘			↘			
[96]		↘				↘			↘						↘	
[80]	↘				↘		↘									↘
[98]	↘			↘							↘	↘				
[99]	↘							↘		↘		↘			↘	
[101]		↘					↘					↘	↘			
[100]	↘				↘				↘					↘		↘
[102]	↘				↘		↘		↘						↘	

2.8 Wavelet-based schemes

This part of the article tries to discuss the wavelet-based load estimation schemes such as [103–107] designed for cloud computing DCs. For example, in [108], Liu et al. proposed a VM migration solution which applies a time series-based load forecasting algorithm. They tuned the upper and lower bounds of load for hosts and predicted the tendency of their subsequent loads by creating a load time series using the cloud model. Afterward, they stipulated a VM load-aware migration WAM which chooses a source PM, a destination PM, and a VM on the source PM to be migrated. Also, in this scheme, the authors have considered CPU consumption as workload and applied the PlanetLab dataset and the CloudSim software for evaluation. The flowchart of this framework is provided in Fig. 17.

In [109], Lyu et al. introduced a forecasting method consisting of a forecast module, an adjustment module, and a collection module. The first module applies machine learning methods to enhance forecasting accuracy. As an advantage, they introduced an effective way of recognizing the dual-threshold load rate forecast mechanism to balance availability and profit. The architecture of this approach is depicted in Fig. 18.

In [110], Qazi et al. presented an efficient method to predict the cluster behavior based on its history and re-distribute VMs to free under-utilized PMs and turned them off to save power. They evaluated real loads and used a chaotic time series. Chaos theory with optimizations makes this framework indifferent to the loads' type and inherent cycles in them.

2.9 Collaborative filtering-based schemes

In [111], Duggan et al. presented a learning-based solution for load forecasting for analytical databases applied by different CSPs. Enabling load performance estimations that can be ported across hardware configurations it could help cloud users with their service-purchase decisions and CSPs in their provisioning decisions. This approach applies collaborative filtering to forecast lightweight load fingerprints that model the behavior of concurrent query loads for choosing hardware configurations.

In [112], Zhang et al. provided a prediction-based scaling solution which uses collaborative filtering with a pattern matching technique. It enhances reactive rule-based scalability techniques and provides a method to link SLA according to lower-level metrics from the infrastructure. Nevertheless, for fine-tuning of this approach, more infrastructure metrics should be considered.

Table 6 determines the dataset, simulation software environment, and the factors predicted and evaluated the

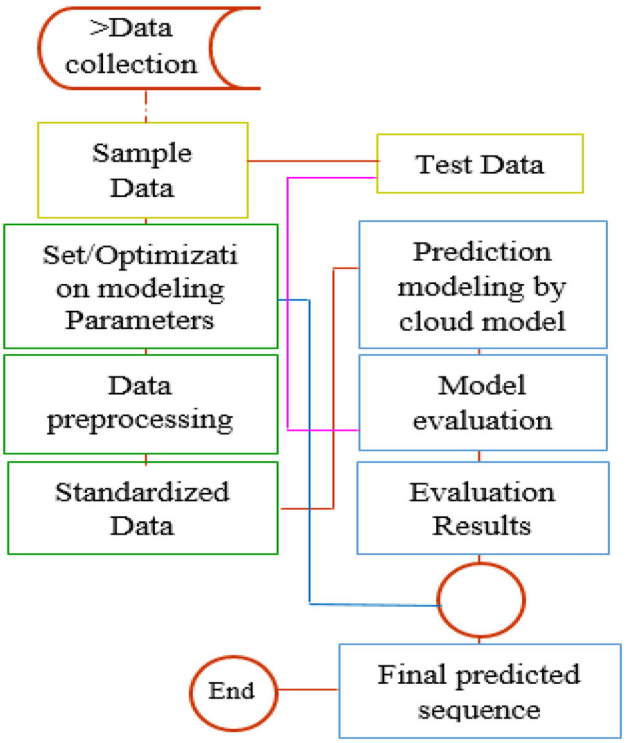

Fig. 17 Load forecasting in [108]

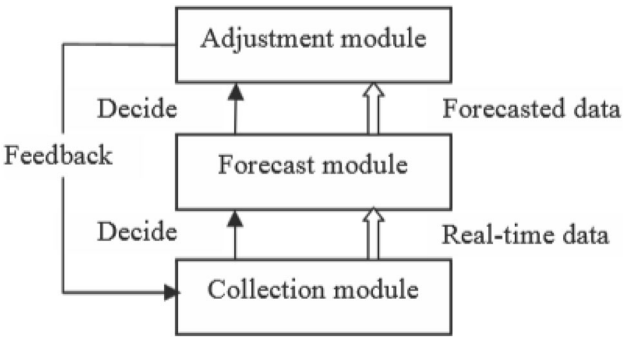

Fig. 18 Workload predicting architecture in [109]

wavelet, collaborative filtering-based schemes-based workload prediction schemes.

2.10 Ensemble-based schemes

Even though some of the previously discussed workload prediction schemes have applied a single prediction method, their accuracy may not be as required and also the prediction length may not be increased. To mitigate these problems several ensemble-based load forecasting frameworks have been proposed in the literature which this subsection is aimed to review them.

For example, in [113], Cao et al. introduced propose an ensemble method which uses multiple models to increase

Table 6 Properties of the wavelet and collaborative filtering-based load forecasting approaches

Datasets/workloads			Simulators/environments				Evaluation factors					Predicted factors				
Schemes	Google	Self-collected	Planet Lab	MATLAB	Amazon EC2	CloudSim	C ++	MAPE	Workload	CPU	Error	Exaction Time	CPU	Disk	I/O	Memory	Bandwidth
[87]		↘				↘				↘						↘	
[108]		↘						↘					↘				
[108]	↘								↘				↘	↘			
[108]			↘			↘						↘	↘	↘		↘	
[108]		↘		↘								↘				↘	↘
[108]	↘						↘						↘				
[108]		↘						↘			↘		↘				
[108]	↘				↘											↘	
[108]	↘					↘		↘					↘			↘	
[108]	↘				↘				↘				↘				
[108]	↘					↘					↘		↘		↘		
[108]	↘												↘				↘

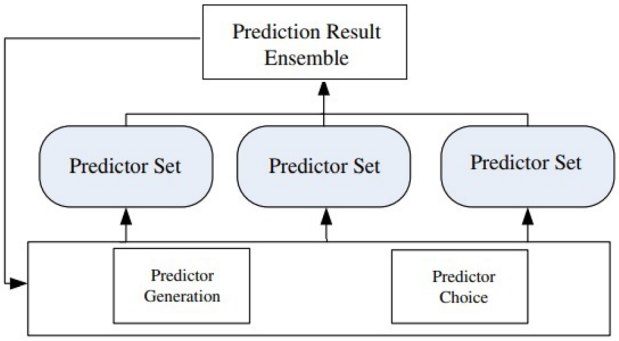

Fig. 19 Ensemble-based forecasting in [113]

performance and CPU load forecasting. They apply a two-layer ensemble model which consists of predictor and ensemble layers. The predictor optimization layer applies new predictor instances and removes those ones with poor performance. The ensemble layer produces the final forecasting based on the results of multiple predictor instances and can provide feedback to the predictor optimization layer, which helps it to adopt appropriate optimization strategies. In this scheme, predictor replacement is used regarding the performance evaluation for maintaining the performance of a predictor set. Then, the poorest predictor should be removed and another predictor should be added. The architecture of this approach is depicted in Fig. 19.

The work in [114], provided a prediction method to enhance accuracy in the auto-scalers using an ensemble-based load forecasting approach. They evaluated several predicting models for in predicting various load patterns. This ensemble technique is implemented using three real-world loads. They trained each model in real-time and aggregated the forecasted results based on the weights computed using inverse errors of the fitted values for the training data. However, further work is needed to identify the optimum input window size to maximize accuracy while meeting the temporal restrictions on calculating the forecasts in real-time.

In [115], Singh et al. tried to reduce PMs' power usage, cooling, and CO_2 emissions to improve the sustainability of the cloud infrastructure. They used load forecasting techniques that guide in identifying servers, time intervals, and other critical parameters needed in the cloud DCs. This scheme is able to deal with non-stationarity workloads and by updating its learning parameters, avoids re-training of its prediction models. Furthermore, they applied Weighted Majority and Simulatable Experts to deal with the extensive non-stationarity and massive online streaming data.

In [116], Sommer et al. proposed PRUF, an ensemble-based forecasting module to predict future utilization of VMs. They proposed a proactive VM migration policy using predictive overload detection and performed a study in the CloudSim. The architecture of this approach is

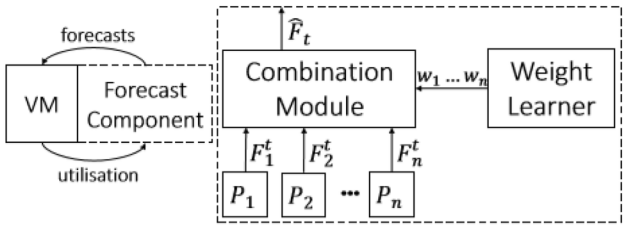

Fig. 20 Forecasting architecture in [116]

depicted in Fig. 20. Table 7 compares the properties of ensemble-based methods.

2.11 Hybrid load prediction schemes

This subsection attempts to discuss the load forecasting designed using a combination of the before mentioned predicting methods.

2.11.1 SVR + kalman filter

In [117], Hu et al. presented KSwSVR, a multi-step-ahead load predicting method, which integrates SVR and Kalman smoother. Public trace is applied to verify its forecasting accuracy, stability, and adaptability. CPU allocation experiment indicated that the KSwSVR can reduce resources usage while meeting SLA requirements. In this scheme, the Kalman smoother is employed to reduce the noise of resources usage data, caused by measurement errors.

2.11.2 Deep learning + SVM

In [118], Tarsa et al. used hierarchical sparse coding, which is a form of deep learning to model user-driven loads using on-chip hardware performance counters. They predicted periods of low instruction throughput, which frequency and voltage can be scaled to reclaim power. Using a multi-layer coding structure, this method codes counter values features learned from data and passes them to an SVM classifier where they act as signatures for predicting future load states.

2.11.3 ARIMA + RNN

In [119], Janardhanan et al. focused on the time series predicting of CPU usage in DCs using LSTM network and evaluated it against the ARIMA model.

2.11.4 ARIMA + wavelet decomposition

In [120], Bi et al. introduced a hybrid method which uses wavelet decomposition and ARIMA to forecast the future

Table 7 Comparison of the ensemble-based schemes workload prediction schemes

Schemes	Datasets/workloads					Simulators/environments			Evaluation factors		Predicted factors			
	Google	Self-collected	NASA	WorldCup98	Wikipedia	MATLAB	CloudSim	C ++	CPU	Workload	CPU	Disk	I/O	Memory	Bandwidth
[114]					↘	↘			↘		↘			↘	
[115]		↘				↘				↘	↘				
[116]	↘						↘		↘					↘	

load. It tries to smooth the task time series by using SavitzkyGolay filtering and decomposes it into multiple components via wavelet decomposition. Their forecasting results are reconstructed via wavelet reduction to estimate the number of arriving tasks. However, better data smoothing algorithms can be used to further improve the prediction accuracy of this scheme.

2.11.5 LR + SVM

In [121], Liu et al. proposed an adaptive approach for load forecasting, which classifies load into various classes assigned for various forecasting models regarding the load features and assigns various prediction models regarding the workload features. They transformed the load classification problem into a task assignment problem using a mixed 0–1 integer programming model and provided an online solution for it. For prediction, they have used linear regression and SVM which is good at the prediction of nonlinear data. They applied the Google cluster trace to evaluate this approach. The architecture of this solution is exhibited in Fig. 21. As an advantage, this approach improves the platform cumulative relative forecasting errors.

2.11.6 ANN + regression

In [122], Tang et al. introduced MLWNN which applies linear regression and wavelet ANN to forecast short-term load. They provide a heuristic power-aware job scheduling with a load forecasting method and employed the error backpropagation algorithm to train a three-layered feed-forward WNN model and get a minimum error. The authors presented a job scheduling approach, which includes a resource management method based on the MLWNN workload prediction. They conducted their experiments using CloudSim software and indicated that their approach can reduce power usage and increase resources utilization.

In [123], Gandhi et al. tried to improve resources allocation in cloud DCs to reduce SLAV and power usage. They employed a predictive resource provisioning method, which deals with load estimation at coarse time scales and reactive provisioning to deal with any excess of load at finer time scales. The combination of predictive and reactive provisioning achieves an improvement in meeting SLA, conserving power, and reducing provisioning costs. The architecture of this scheme is shown in Fig. 22.

2.11.7 ARM + regression + SVR

In [124], Guo et al. proposed NUP, a hybrid forecasting method, which uses the load type to switch forecasting

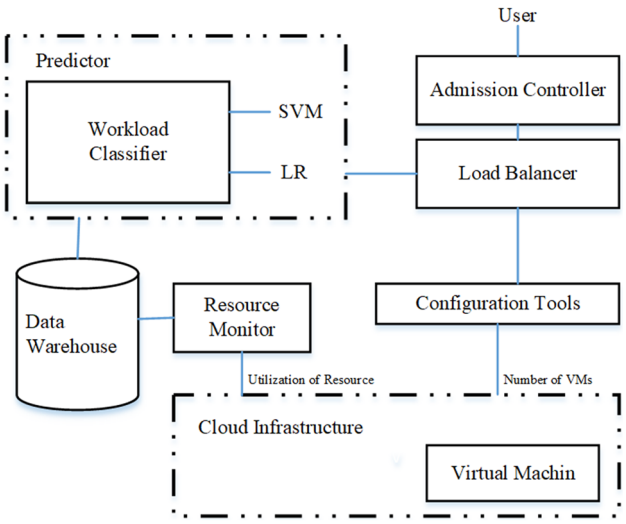

Fig. 21 Load forecasting in [121]

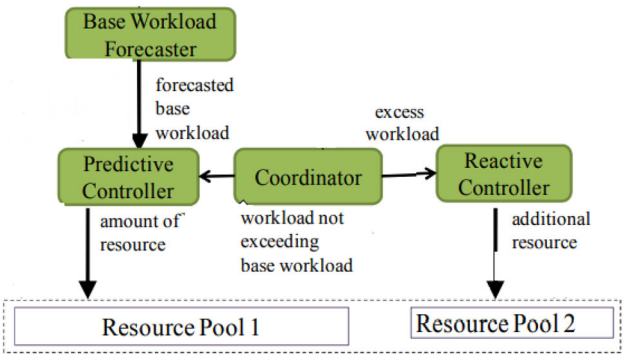

Fig. 22 Hybrid resource provisioning in [123]

algorithms. It used autocorrelation coefficients and Hurst exponents of loads to determine the loads belong to the period or the trend. NUP applies linear regression and similarities among periods to replace missing data of trend and period loads. It uses linear regression and ARMA to predict the trend and SVR to forecast the period.

Table 8 determines the datasets, simulation software, evaluation factors, and the prediction factors considered in the hybrid and ensemble-based load forecasting schemes.

3 Discussion

This subsection provides an extensive comparison of the workload forecasting approaches designed for the various cloud environments and its results can illuminate the future research directions. It mainly analyzes the following issues about these schemes:

- Publication year of the published schemes in the workload prediction schemes.

Table 8 Comparison of the hybrid and ensemble-based schemes load prediction approaches

Datasets/workloads					Simulators/environments			Evaluation factors			Predicted factors				
Schemes	Google	Self-collected	Saskatchewan	NASA	MATLAB	AmazonEC2	CloudSim	MSE	Cost	CPU	Power	CPU	Disk	I/O	Memory	Bandwidth
[117]			↘	↘							↘	↘	↘		↘	
[118]			↘	↘						↘		↘				
[120]			↘	↘	↘					↘		↘			↘	
[119]		↘							↘							
[124]	↘					↘		↘				↘		↘	↘	↘
[121]		↘					↘					↘	↘		↘	
[123]		↘					↘	↘				↘			↘	

- Simulator software and environments applied to analyze the outlined schemes.
- Factors applied to compare the proposed frameworks and exhibit their effectiveness.
- Datasets and workloads employed in the investigated prediction schemes.
- The number of the load prediction schemes which have applied each forecasting method.
- The number of schemes which have predicted various resources in their predictions.

Figure 23 depicts the publication year of the outlined load forecasting schemes. As shown in this figure many workload predictions schemes have been recently proposed to deal with this problem and this context is an active research field.

Furthermore, Fig. 24 exhibits the datasets employed in the studied schemes and specifies the number of solutions which apply each dataset. As shown in this figure, the main datasets applied in this context are Google and NASA datasets and the self-collected data by the authors from real environments.

Figure 25 depicts the experiment factors applied in the evaluation of the load prediction schemes and the number of schemes which have applied each evaluation factor. As it is shown in this figure, factors such as CPU load, cost, and execution time are mostly employed by the studied schemes. Figure 26 shows the number of loads predicting schemes designed and proposed using the prediction methods outlined before. As shown in this figure, ANN and wavelet transform is used by more load forecasting schemes. Figure 27 indicates the factors which forecasted by the workload prediction schemes. On the other hand, some of the schemes recognize the load with the increase of CPU consumption while the others may consider other factors such as memory, bandwidth, and the even the disk I/O. As shown in this figure, CPU consumption is a critical factor considered by more forecasting schemes to detect workload.

Figure 28 shows the factors predicted by the workload prediction approaches to forecast the workload. As shown in this figure, only a few schemes have considered three or four factors in their predictions and in future studies, this issue can be further investigated to better predict the workload and prevent resource wastage. Figure 29 reflects the environments and simulators utilized in the evaluation of the investigated schemes and determines the number of approaches which have used each kind of simulators. As shown in this figure, CloudSim is the most popular simulator software applied in this context.

Figure 30 exhibits the number of the scheme which has applied only one dataset and number of schemes that have used two datasets in their simulation and verification

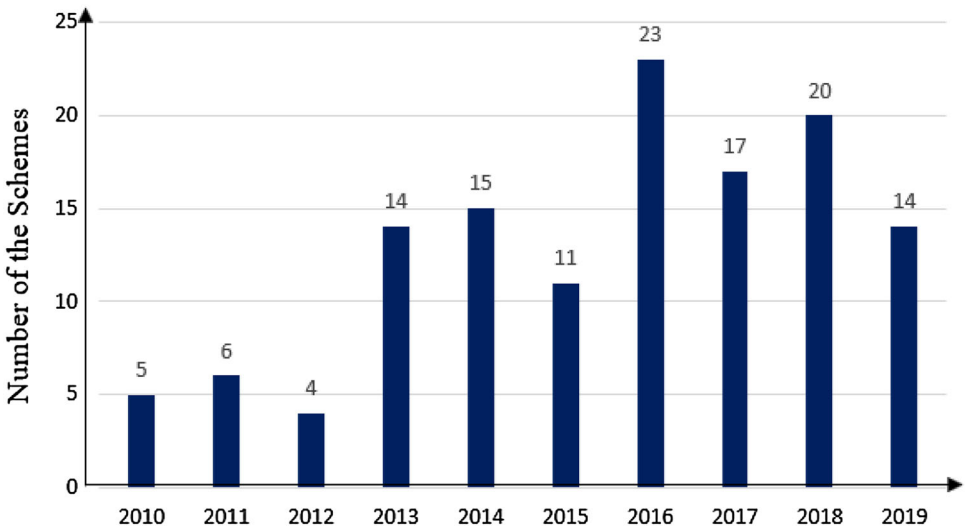

Fig. 23 Publication year of the outlined load forecasting schemes

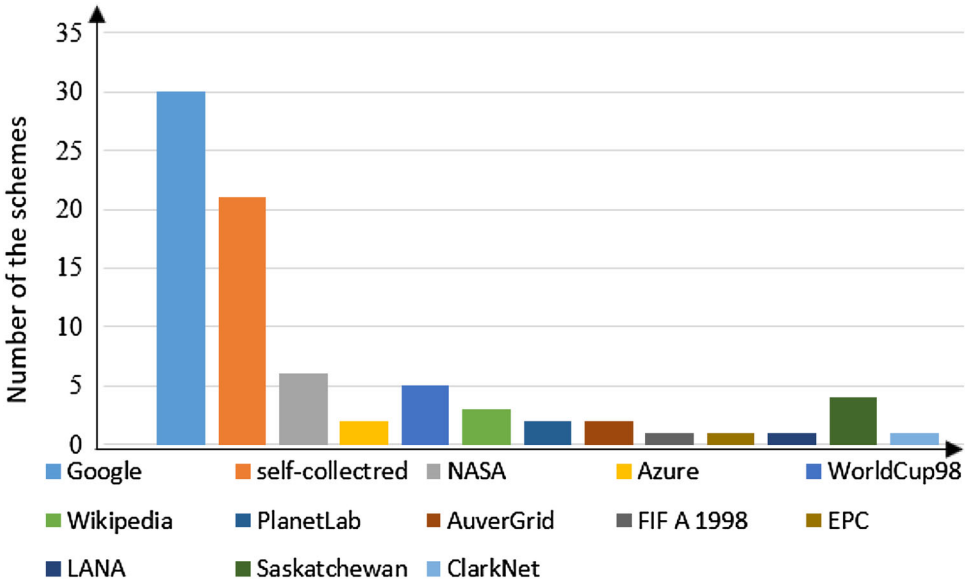

Fig. 24 Applied datasets in the workload forecasting schemes

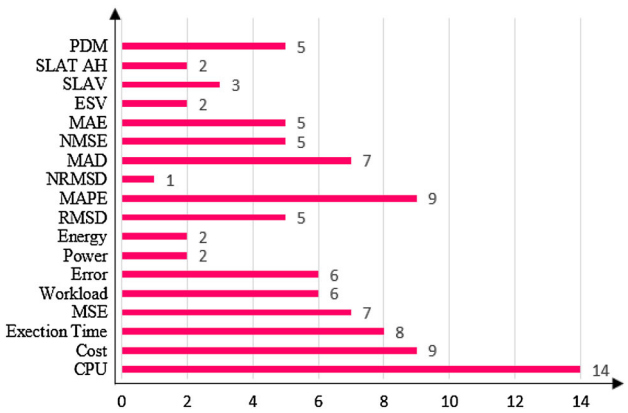

Fig. 25 Simulation factors applied in the load prediction

process. As depicted in this figure, only a few schemes have applied two datasets; consequently, in future researches and studies the workload forecasting schemes can evaluate their approaches using multiple datasets to further ensure of their approach's accuracy.

4 Conclusion

The main objective of the cloud computing paradigm is to provide various virtual remote resources and service to its customers. In this context, providing the guaranteed QoS, increasing throughput, and return on investment are of the features which can be achieved by effective resource management in cloud DCs. Future workload prediction in cloud DCs is an essential step in proper resource

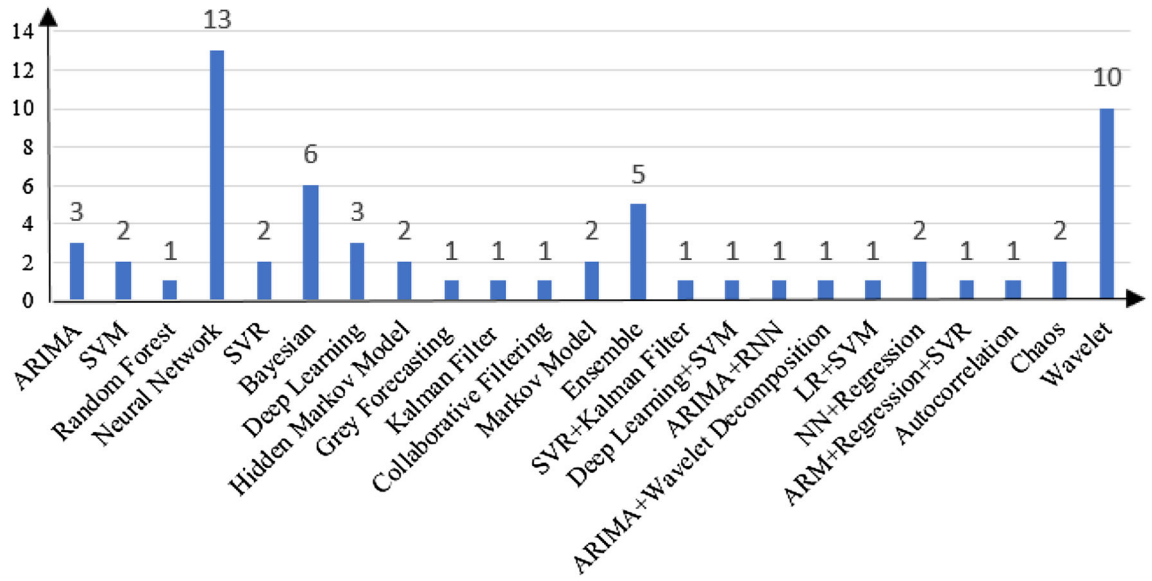

Fig. 26 Applied algorithms in the load predicting methods

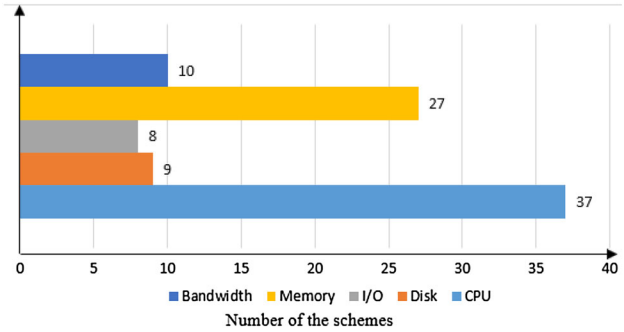

Fig. 27 Number of the scheme which forecasted each factor

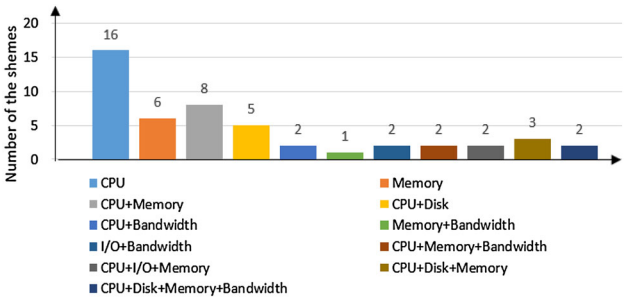

Fig. 28 Factors predicted in the load forecasting schemes

management and auto-scaling approaches which aids cloud service providers in provisioning/de-provisioning virtual resources. However, prediction errors can cause problems such as under-provisioning or over-provisioning, which the former reduces the cloud performance and leads to SLA violations and the latter leads to the resource wastage problem.

Regarding the importance of the accurate load prediction based on the historical workload data and handling

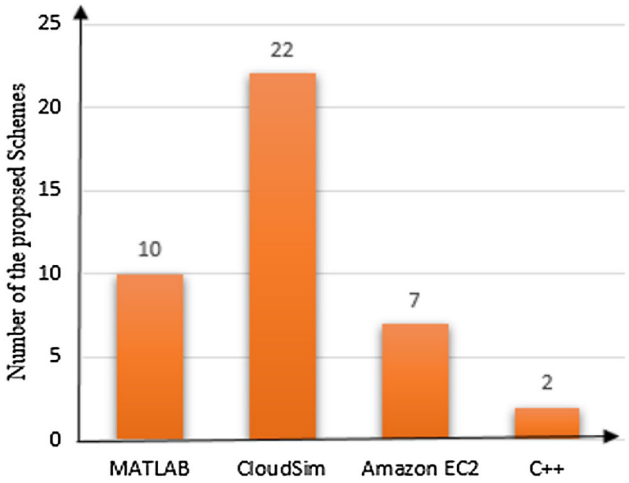

Fig. 29 Environments and simulators

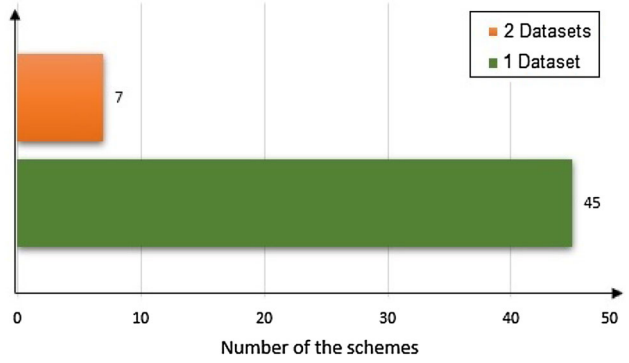

Fig. 30 Datasets

issues such as workload fluctuation and Slashdot effects, various load prediction schemes are provided in the

literature. This paper first presents the basic concepts and challenges in the workload prediction process. Then, it delivers a taxonomy and survey of the investigated load forecasting approaches and describes their main contributions and applied algorithms to conduct predictions. Furthermore, features such as the applied workload datasets, simulation factors, predicted factors and simulators employed by each forecasting scheme is illuminated. Furthermore, an extensive analysis of the workload forecasting schemes is provided which can be useful for future studies and researches. In the future studies the following issues can be further investigated:

- Exploring other machine learning techniques to further improve the workload prediction's performance.
- Providing better load forecasting schemes to recognizing more realistic and complex request patterns which may happen in real life.
- Defining new workload prediction metrics, for example on the lags in burst predictions. Also, since the cost of prediction errors in the cloud environment is not symmetric, defining better evaluation metrics should be considered on this issue.
- Regarding the suitability of the non-linear prediction models to predict time series with seasonal variations, they can be used for optimizing processes with longer time horizons.
- Investigating the resource management algorithms to utilize the achieved forecasting results.
- Integrating the load prediction schemes with the intrusion detection schemes to recognize the DDoS attacks from the Slashdot effects.
- Creating lightweight workload prediction schemes to be applied in the recently emerging technologies such as IoT, cloudlets, fog computing, and mobile edge computing which have limited and fewer resources than the cloud DCs.
- One of the important directions for future researches is the integration of the autoscaling schemes with the IDS and IPS systems to better handle the DDoS attacks and Yo–Yo attacks. Generally, autoscaling systems convert the DDoS attacks to EDoS attacks to deal with malicious behaviors. Recognizing the DDoS workload from the users' workload is an open issue which should be dealt with in the future researches [55, 125–127].

References

1. Masdari, M., ValiKardan, S., Shahi, Z., Azar, S.I.: Towards workflow scheduling in cloud computing: a comprehensive analysis. J. Netw. Comput. Appl. **66**, 64–82 (2016)
2. Masdari, M., Nabavi, S.S., Ahmadi, V.: An overview of virtual machine placement schemes in cloud computing. J. Netw. Comput. Appl. **66**, 106–127 (2016)
3. González-Martínez, J.A., Bote-Lorenzo, M.L., Gómez-Sánchez, E., Cano-Parra, R.: Cloud computing and education: a state-of-the-art survey. Comput. Educ. **80**, 132–151 (2015)
4. Masdari, M., Salehi, F., Jalali, M., Bidaki, M.: A survey of PSO-based scheduling algorithms in cloud computing. J. Netw. Syst. Manage. **25**(1), 122–158 (2017)
5. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: issues and challenges. J. Grid Comput. **14**(2), 217–264 (2016)
6. Coutinho, E.F., de Carvalho Sousa, F.R., Rego, P.A.L., Gomes, D.G., de Souza, J.N.: Elasticity in cloud computing: a survey. Ann. Telecommun. **70**(7–8), 289–309 (2015)
7. Lorido-Botran, T., Miguel-Alonso, J., Lozano, J.A.: A review of auto-scaling techniques for elastic applications in cloud environments. J. Grid Comput. **12**(4), 559–592 (2014)
8. Amiri, M., Mohammad-Khanli, L.: Survey on prediction models of applications for resources provisioning in cloud. J. Netw. Comput. Appl. **82**, 93–113 (2017)
9. Singh, S., Chana, I.: QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. (CSUR) **48**(3), 42 (2016)
10. Kaur, T., Chana, I.: Energy efficiency techniques in cloud computing: a survey and taxonomy. ACM Comput. Surv. (CSUR) **48**(2), 22 (2015)
11. Dougherty, B., White, J., Schmidt, D.C.: Model-driven auto-scaling of green cloud computing infrastructure. Future Gener. Comput. Syst. **28**(2), 371–378 (2012)
12. Qu, C., Calheiros, R.N., Buyya, R.: Auto-scaling web applications in clouds: a taxonomy and survey. ACM Comput. Surv. (CSUR) **51**(4), 73 (2018)
13. Netto MA, Cardonha C, Cunha RL, Assuncao MD (2014) Evaluating auto-scaling strategies for cloud computing environments. In 2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems, IEEE, p 187–196
14. de Assunção, M.D., Cardonha, C.H., Netto, M.A., Cunha, R.L.: Impact of user patience on auto-scaling resource capacity for cloud services. Future Gener. Comput. Syst. **55**, 41–50 (2016)
15. Qu, C., Calheiros, R.N., Buyya, R.: Auto-scaling web applications in clouds: a taxonomy and survey. ACM Comput. Surv. **51**(4), 73 (2016)
16. Turowski M, Lenk A (2015) Vertical scaling capability of OpenStack. Service-Oriented Computing-ICSOC 2014 Workshops. Springer, Cham, p 351–362
17. Cai, Z., Li, Q., Li, X.: Elasticsim: a toolkit for simulating workflows with cloud resource runtime auto-scaling and stochastic task execution times. J. Grid Comput. **15**(2), 257–272 (2017)
18. Armant, V., De Cauwer, M., Brown, K.N., O'Sullivan, B.: Semi-online task assignment policies for workload consolidation in cloud computing systems. Future Gener. Comput. Syst. **82**, 89–103 (2018)
19. Kardani-Moghaddam, S., Buyya, R., Ramamohanarao, K.: Performance anomaly detection using isolation-trees in heterogeneous workloads of web applications in computing clouds. Concurr. Comput. (2019). https://doi.org/10.1002/cpe.5306
20. Bajaj S (2018) Current drift in energy efficiency cloud computing: new provocations, workload prediction, consolidation, and resource over commitment. In critical research on Scalability and security issues in virtual cloud environments: IGI Global, Pennsylvania, p 283–303
21. Li, L., Feng, M., Jin, L., Chen, S., Ma, L., Gao, J.: Domain knowledge embedding regularization neural networks for

workload prediction and analysis in cloud computing. J. Inf. Technol. Res. (JITR) **11**(4), 137–154 (2018)
22. Guo, M., Guan, Q., Ke, W.: Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload. IEEE Access **6**, 15178–15191 (2018)
23. Pagán, J., Zapater, M., Ayala, J.L.: Power transmission and workload balancing policies in eHealth mobile cloud computing scenarios. Future Gener. Comput. Syst. **78**, 587–601 (2018)
24. Deng, R., Lu, R., Lai, C., Luan, T.H., Liang, H.: Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet Things J. **3**(6), 1171–1181 (2016)
25. Zhong, C., Yuan, X.: Intelligent elastic scheduling algorithms for PaaS cloud platform based on load prediction. 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 1500–1503. IEEE, New Jersey (2019)
26. Youssef, F., El Habib, B.L., Hamza, R., El Houssine, L., Ahmed, E., Hanoune, M.: A new conception of load balancing in cloud computing using tasks classification levels. Int. J. Cloud Appl. Comput. (IJCAC) **8**(4), 118–133 (2018)
27. Stergiou, C., Psannis, K.E., Kim, B.-G., Gupta, B.: Secure integration of IoT and cloud computing. Future Gener. Comput. Syst. **78**, 964–975 (2018)
28. Stergiou, C., Psannis, K.E., Gupta, B.B., Ishibashi, Y.: Security, privacy & efficiency of sustainable cloud computing for big data & IoT. Sustain. Comput. **19**, 174–184 (2018)
29. Sonkar, S., Kharat, M.: Load prediction analysis based on virtual machine execution time using optimal sequencing algorithm in cloud federated environment. Int. J. Inf. Technol. **11**(2), 265–275 (2019)
30. Singh, P., Gupta, P., Jyoti, K.: Tasm: technocrat arima and svr model for workload prediction of web applications in cloud. Clus. Comput. **22**(2), 619–633 (2019)
31. Sharma, P., Sengupta, J., Suri, P.: Survey of intrusion detection techniques and architectures in cloud computing. IJHPCN **13**(2), 184–198 (2019)
32. Rahhali, H., Hanoune, M.: A new conception of load balancing in cloud computing using Hybrid heuristic algorithm. Int. J. Comput. Sci. Issues (IJCSI) **15**(6), 1–8 (2018)
33. Qaddoum, K.S., El Emam, N.N., Abualhaj, M.A.: Elastic neural network method for load prediction in cloud computing grid. Int. J. Electr. Comput. Eng. **9**(2), 1201 (2019)
34. Prassanna J, Venkataraman N Adaptive regressive holt–winters workload prediction and firefly optimized lottery scheduling for load balancing in cloud. Wireless Networks
35. Patel, D., Gupta, R.K., Pateriya, R.: Energy-aware prediction-based load balancing approach with VM migration for the cloud environment. Data engineering and applications, pp. 59–74. Springer, Singapore (2019)
36. Nguyen, H.M., Kalra, G., Kim, D.: Host load prediction in cloud computing using long short-term memory encoder–decoder. J. Supercomput. (2019). https://doi.org/10.1007/s11227-019-02967-7
37. Nguyen, H.M., Kalra, G., Jun, T.J., Woo, S., Kim, D.: ESNemble: an echo state network-based ensemble for workload prediction and resource allocation of Web applications in the cloud. J. Supercomput. **75**(10), 6303–6323 (2019)
38. Li, L., Wang, Y., Jin, L., Zhang, X., Qin, H.: Two-stage adaptive classification cloud workload prediction based on neural networks. Int. J. Grid High Perform. Comput. (IJGHPC) **11**(2), 1–23 (2019)
39. Kirchoff DF, Xavier M, Mastella J, De Rose CA (2019) A preliminary study of machine learning workload prediction techniques for cloud applications. In 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, New Jersey, p 222–227
40. Gupta, B., Agrawal, D.P., Yamaguchi, S.: Handbook of research on modern cryptographic solutions for computer and cyber security. IGI global, Pennsylvania (2016)
41. Bhagavathiperumal, S., Goyal, M.: Dynamic provisioning of cloud resources based on workload prediction. Computing and Network Sustainability., pp. 41–49. Springer, Singapore (2019)
42. Amiri, M., Mohammad-Khanli, L., Mirandola, R.: A new efficient approach for extracting the closed episodes for workload prediction in cloud. Computing (2019). https://doi.org/10.1007/s00607-019-00734-3
43. Zhang, H., Jiang, G., Yoshihira, K., Chen, H., Saxena, A.: Intelligent workload factoring for a hybrid cloud computing model. 2009 Congress on Services-I, pp. 701–708. IEEE, New Jersey (2009)
44. Di S, Wang CL (2013) Minimization of cloud task execution length with workload prediction errors. In 20th Annual International Conference on High Performance Computing. IEEE, New Jersey, p 69–78
45. Khoshkbarforoushha, A., Ranjan, R., Gaire, R., Abbasnejad, E., Wang, L., Zomaya, A.Y.: Distribution based workload modelling of continuous queries in clouds. IEEE Trans. Emerg. Topics Comput. **5**(1), 120–133 (2017)
46. Wang P, Fang W, Guo B, Bao H (2017) Apply petri nets to human performance and workload prediction under multitask. In International Conference on Applied Human Factors and Ergonomics. Springer, Cham. p 395–405
47. Reeba PJ, Shaji R, Jayan J (2016) A secure virtual machine migration using processor workload prediction method for cloud environment. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE, p 1–6
48. Baldan, F.J., Ramirez-Gallego, S., Bergmeir, C., Benitez-Sanchez, J.M., Herrera, F.: A forecasting methodology for workload forecasting in cloud systems. IEEE Trans. Cloud Comput. **6**(4), 929–941 (2016)
49. Singh N, Rao S (2012) Online ensemble learning approach for server workload prediction in large datacenters. In 2012 11th International Conference on Machine Learning and Applications. IEEE, p 68–71
50. Hagan, M.T., Behr, S.M.: The time series approach to short term load forecasting. IEEE Trans. Power Syst. **2**(3), 785–791 (1987)
51. Babu, K.R., Samuel, P.: Interference aware prediction mechanism for auto scaling in cloud. Comput. Electr. Eng. **69**, 351–363 (2017)
52. Antonescu, A.-F., Braun, T.: Simulation of SLA-based VM-scaling algorithms for cloud-distributed applications. Future Gener. Comput. Syst. **54**, 260–273 (2016)
53. Yang J, Liu C, Shang Y, Mao Z, Chen J (2013) Workload predicting-based automatic scaling in service clouds. In Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on IEEE. p 810–815
54. Bao J, Lu Z, Wu J, Zhang S, Zhong Y (2014) Implementing a novel load-aware auto scale scheme for private cloud resource management platform. In Network Operations and Management Symposium (NOMS), 2014 IEEE. IEEE, p 1–4
55. Khorsand, R., Ghobaei-Arani, M., Ramezanpour, M.: WITHDRAWN: a fuzzy auto-scaling approach using workload prediction for MMOG application in a cloud environment. Elsevier, Amsterdam (2018)
56. Li S, Wang Y, Qiu X, Wang D, Wang L (2013) A workload prediction-based multi-vm provisioning mechanism in cloud computing. In 2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, p 1–6

57. Kumar AS, Mazumdar S (2016) Forecasting HPC workload using ARMA models and SSA. In 2016 International Conference on Information Technology (ICIT). IEEE, p 294–297
58. Calheiros, R.N., Masoumi, E., Ranjan, R., Buyya, R.: Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Trans. Cloud Comput. **3**(4), 449–458 (2015)
59. Messias, V.R., Estrella, J.C., Ehlers, R., Santana, M.J., Santana, R.C., Reiff-Marganiec, S.: Combining time series prediction models using genetic algorithm to autoscaling Web applications hosted in the cloud infrastructure. Neural Comput. Appl. **27**(8), 2383–2406 (2016)
60. Barati, M., Sharifian, S.: A hybrid heuristic-based tuned support vector regression model for cloud load prediction. J. Supercomput. **71**(11), 4235–4259 (2015)
61. Raghunath, B.R., Annappa, B.: Virtual machine migration triggering using application workload prediction. Procedia Comput. Sci. **54**, 167–176 (2015)
62. Tong, J.J., Hai-hong, E., Song, M.N., Song, J.D.: Host load prediction in cloud based on classification methods. J. China Univ. Posts Telecommun. **21**(4), 40–46 (2014)
63. Zhong, W., Zhuang, Y., Sun, J., Gu, J.: A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine. Appl. Intell. **48**(11), 4072–4083 (2018)
64. Nikravesh AY, Ajila SA, Lung CH (2015) Towards an autonomic auto-scaling prediction system for cloud resource provisioning. In Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. IEEE Press, p 35–45
65. Cetinski, K., Juric, M.B.: AME-WPC: advanced model for efficient workload prediction in the cloud. J. Netw. Compu. Appl. **55**, 191–201 (2015)
66. Nehru EI, Venkatalakshmi B, Balalcrishnant R, Nithya R (2013) Neural load prediction technique for power optimization in cloud management system. In 2013 IEEE Conference on Information & Communication Technologies. IEEE, p 541–544
67. Nguyen HM, Woo S, Im J, Jun T, Kim D (2016) A workload prediction approach using models stacking based on recurrent neural network and autoencoder. In 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, p 929–936
68. Wamba GM, Li Y, Orgerie AC, Beldiceanu N, Menaud JM (2017) Cloud workload prediction and generation models. In 2017 29th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). IEEE, p 89–96
69. Yu Y, Jindal V, Yen IL, Bastani F (2016) Integrating clustering and learning for improved workload prediction in the cloud. In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD). IEEE, p 876–879
70. Song, B., Yu, Y., Zhou, Y., Wang, Z., Du, S.: Host load prediction with long short-term memory in cloud computing. J. Supercomput. **74**(12), 6554–6568 (2018)
71. Imam MT, Miskhat SF, Rahman RM, Amin MA (2011) Neural network and regression based processor load prediction for efficient scaling of Grid and Cloud resources. In 14th International Conference on Computer and Information Technology (ICCIT 2011). IEEE, p 333–338
72. Yang, Q., Zhou, Y., Yu, Y., Yuan, J., Xing, X., Du, S.: Multi-step-ahead host load prediction using autoencoder and echo state networks in cloud computing. J. Supercomput. **71**(8), 3037–3053 (2015)
73. Kumar, J., Singh, A.K.: Workload prediction in cloud using artificial neural network and adaptive differential evolution. Future Gener. Comput. Syst. **81**, 41–52 (2018)
74. Lu, Y., Panneerselvam, J., Liu, L.: Wu Y (2016) Rvlbpnn: a workload forecasting model for smart cloud computing. Sci. Prog. **2016**, 9 (2016)
75. Zhou, X., et al.: Load balancing prediction method of cloud storage based on analytic hierarchy process and hybrid hierarchical genetic algorithm. SpringerPlus **5**(1), 1989 (2016)
76. Kousiouris, G., Cucinotta, T., Varvarigou, T.: The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks. J. Syst. Softw. **84**(8), 1270–1291 (2011)
77. Ramezani F, Naderpour M (2017) A fuzzy virtual machine workload prediction method for cloud environments. In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, p 1–6
78. Yang, J., et al.: A cost-aware auto-scaling approach using the workload prediction in service clouds. Inf. Syst. Front. **16**(1), 7–18 (2014)
79. Liu B, Lin Y, Chen Y (2016) Quantitative workload analysis and prediction using Google cluster traces. In Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on, 2016. p 935–940
80. Di S, Kondo D, Cirne W (2012) Host load prediction in a Google compute cloud with a Bayesian model. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, p. 21
81. Dietrich B, Nunna S, Goswami D, Chakraborty S, Gries M (2010) LMS-based low-complexity game workload prediction for DVFS. In 2010 IEEE International Conference on Computer Design. IEEE, p 417–424
82. Tian, C., et al.: Minimizing content reorganization and tolerating imperfect workload prediction for cloud-based video-on-demand services. IEEE Trans. Serv. Comput. **9**(6), 926–939 (2016)
83. Patel, Y.S., Misra, R.: Performance comparison of deep VM workload prediction approaches for cloud. In Progress in Computing, Analytics and Networking, pp. 149–160. Springer, Singapore (2018)
84. Gupta S, Dinesh DA (2017) Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks. In 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, p 1–6
85. Zhang, Q., Yang, L.T., Yan, Z., Chen, Z., Li, P.: An efficient deep learning model to predict cloud workload for industry informatics. IEEE Trans. Industr. Inf. **14**(7), 3170–3178 (2018)
86. Gong Z, Gu X, Wilkes J (2010) Press: predictive elastic resource scaling for cloud systems. In Network and Service Management (CNSM), 2010 International Conference on, 2010. p 9–16
87. Jv, B.B., Dharma, D.: HAS: hybrid auto-scaler for resource scaling in cloud environment. J. Parallel Distrib. Comput. **120**, 1–15 (2018)
88. Panneerselvam J, Liu L, Antonopoulos N, Bo Y (2014) Workload analysis for the scope of user demand prediction model evaluations in cloud environments. In Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. IEEE Computer Society, p 883–889
89. Pacheco-Sanchez S, Casale G, Scotney B, McClean S, Parr G, Dawson S (2011) Markovian workload characterization for qos prediction in the cloud. In 2011 IEEE 4th International Conference on Cloud Computing. p 147–154
90. Shen Z, Subbiah S, Gu X, Wilkes J (2011) Cloudscale: elastic resource scaling for multi-tenant cloud systems. In Proceedings of the 2nd ACM Symposium on Cloud Computing. p. 5

91. Chen, X.: Decentralized computation offloading game for mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. **26**(4), 974–983 (2014)
92. Khan A, Yan X, Tao S, Anerousis N (2012) Workload characterization and prediction in the cloud: A multiple time series approach. In 2012 IEEE Network Operations and Management Symposium. p 1287–1294
93. Guo Y, Stolyar A, Walid A (2018) Online vm auto-scaling algorithms for application hosting in a cloud. IEEE Transactions on Cloud Computing
94. Gandhi, A., Dube, P., Karve, A., Kochut, A., Zhang, L.: Model-driven optimal resource scaling in cloud. Softw. Syst. Model. **17**(2), 509–526 (2017)
95. Vondra, T., Šedivý, J.: Cloud autoscaling simulation based on queueing network model. Simul. Model. Pract. Theory **70**, 83–100 (2017)
96. Sahni, J., Vidyarthi, D.P.: Heterogeneity-aware adaptive auto-scaling heuristic for improved QoS and resource usage in cloud environments. Computing **99**(4), 351–381 (2017)
97. Jiang J, Lu J, Zhang G, Long G (2013) Optimal cloud resource auto-scaling for web applications. In Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, 2013. p 58–65
98. Jheng JJ, Tseng FH, Chao HC, Chou LD (2014) A novel VM workload prediction using Grey Forecasting model in cloud data center. In The International Conference on Information Networking 2014 (ICOIN2014). p 40–45
99. Kluge F, Uhrig S, Mische J, Satzger B, Ungerer T (2010) Dynamic workload prediction for soft real-time applications. In 2010 10th IEEE International Conference on Computer and Information Technology. p 1841–1848
100. Ardagna D, Casolari S, Panicucci B (2011) Flexible distributed capacity allocation and load redirect algorithms for cloud systems. In 2011 IEEE 4th International Conference on Cloud Computing. p 163–170
101. Qazi K, Li Y, Sohn A (2013) PoWER: prediction of workload for energy efficient relocation of virtual machines. In Proceedings of the 4th annual Symposium on Cloud Computing, 2013: ACM, p. 31
102. Hu Y, Deng B, Peng F, Wang D (2016) Workload prediction for cloud computing elasticity mechanism. In 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). p 244–249
103. Ghorbani M, Wang Y, Xue Y, Pedram M, Bogdan P (2014) Prediction and control of bursty cloud workloads: a fractal framework. In Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis. ACM, p. 12
104. Cortez E, Bonde A, Muzio A, Russinovich M, Fontoura M, Bianchini R (2017) Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In Proceedings of the 26th Symposium on Operating Systems Principles. p 153–167
105. Ganapathi A, Chen Y, Fox A, Katz R, Patterson D (2010) Statistics-driven workload modeling for the cloud. In 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010). p 87–92
106. Nguyen HM, Kim SH, Le DT, Heo S, Im J, Kim D (2015) Epcloud flow: load prediction and migration optimizations for epc network on cloud. In 2015 IEEE 8th International Conference on Cloud Computing. p 981–984
107. Prevost JJ, Nagothu K, Jamshidi M, Kelley B (2014) Optimal calculation overhead for energy efficient cloud workload prediction. In 2014 World Automation Congress (WAC), 2014: IEEE, p 741–747
108. Liu, Y., Gong, B., Xing, C., Jian, Y.: A virtual machine migration strategy based on time series workload prediction using cloud model. Math. Probl. Eng. **2014**, 11 (2014)
109. Lyu H, Li P, Yan R, Masood A, Sheng B, Luo Y (2016) Load forecast of resource scheduler in cloud architecture. In 2016 International Conference on Progress in Informatics and Computing (PIC). p 508–512
110. Qazi K, Li Y, Sohn A (2014) Workload prediction of virtual machines for harnessing data center resources. In 2014 IEEE 7th International Conference on Cloud Computing, 2014: IEEE, p 522–529
111. Duggan J, Chi Y, Hacigümüş H, Zhu S, Cetintemel U (2013) Packing light: portable workload performance prediction for the cloud. In 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW). p 258–265
112. Zhang L, Zhang Y, Jamshidi P, Xu L, Pahl C (2014) Workload patterns for quality-driven dynamic cloud service configuration and auto-scaling. In Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014: IEEE Computer Society. p 156–165
113. Cao, J., Fu, J., Li, M., Chen, J.: CPU load prediction for cloud environment based on a dynamic ensemble model. Software **44**(7), 793–804 (2014)
114. Shariffdeen R, Munasinghe D, Bhathiya H, Bandara U, Bandara HD (2016) Adaptive workload prediction for proactive auto scaling in PaaS systems. In 2016 2nd International Conference on Cloud Computing Technologies and Applications (Cloud-Tech), 2016: IEEE, p 22–29
115. Singh, N., Rao, S.: Ensemble learning for large-scale workload prediction. IEEE Trans. Emerg. Topics Comput. **2**(2), 149–165 (2014)
116. Sommer M, Klink M, Tomforde S, Hähner J (2016) Predictive load balancing in cloud computing environments based on ensemble forecasting. In 2016 IEEE International Conference on Autonomic Computing (ICAC), 2016: IEEE, p 300–307
117. Hu R, Jiang J, Liu G, Wang L (2013) KSwSVR: a new load forecasting method for efficient resources provisioning in cloud. In 2013 IEEE International Conference on Services Computing, 2013: IEEE, p 120–127
118. Tarsa SJ, Kumar AP, Kung H (2014) Workload prediction for adaptive power scaling using deep learning. In 2014 IEEE International Conference on IC Design & Technology, 2014: IEEE, p 1–5
119. Janardhanan D, Barrett E (2017) CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models. In 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017: IEEE, p 55–60
120. Bi J, Zhang L, Yuan H, Zhou M (2018) Hybrid task prediction based on wavelet decomposition and ARIMA model in cloud data center. In 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018: IEEE, p 1–6
121. Liu, C., Liu, C., Shang, Y., Chen, S., Cheng, B., Chen, J.: An adaptive prediction approach based on workload pattern discrimination in the cloud. J. Netw. Comput. Appl. **80**, 35–44 (2017)
122. Tang, X., Liao, X., Zheng, J., Yang, X.: Energy efficient job scheduling with workload prediction on cloud data center. Clus. Comput. **21**(3), 1581–1593 (2018)
123. Gandhi A, Chen Y, Gmach D, Arlitt M, Marwah M (2011) Minimizing data center SLA violations and power consumption via hybrid resource provisioning. In 2011 International Green Computing Conference and Workshops, 2011: IEEE, p 1–8
124. Guo J, Wu J, Na J, Zhang B (2017) A type-aware workload prediction strategy for non-stationary cloud service. In 2017

IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA), 2017: IEEE, p 98–103
125. Ahn, Y.W., Cheng, A.M., Baek, J., Jo, M., Chen, H.-H.: An auto-scaling mechanism for virtual resources to support mobile, pervasive, real-time healthcare applications in cloud computing. IEEE Netw. **27**(5), 62–68 (2013)
126. Shahin AA (2017) Automatic cloud resource scaling algorithm based on long short-term memory recurrent neural network. arXiv preprint arXiv:1701.03295
127. Ali-Eldin A, Tordsson J, Elmroth E, Kihl M (2013) Workload classification for efficient auto-scaling of cloud resources. Tech. Rep.

Mohammad Masdari received his B.Tech. degree in Computer Software Engineering from Islamic Azad University, Qazvin Branch, Iran, in 2001, and M.Tech. degree in Computer Software Engineering from Islamic Azad University, South Tehran Branch, Tehran, Iran, in 2003. He received his Ph.D. degree in Computer Software Engineering from Islamic Azad University, Science and research branch, Tehran, Iran, in 2014. Since 2003, he worked a faculty member of Islamic Azad University, Urmia branch, Iran. Presently he is an Assistant Professor in the Department of Computer Engineering of Islamic Azad University, Urmia branch, Iran. His research interests include Distributed Systems and Network Security.

Afsane Khoshnevis Was born in 1992. She received the B.Sc. in Computer Software Engineering from Islamic Azad University, Urmia branch, Iran, in 2015, and M.Sc. in Computer Science in 2018.