Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 4, 2026

PREDICTING SCHEDULING DELAYS IN CLOUD COMPUTING SYSTEMS USING MACHINE LEARNING

Ali Ahmad Siddiqui[*1], Syed Haider Abbas Naqi[2], Israr Ali[3], Muhammad Sohaib Naseem[4], Abdul Khaliq[5]

[*1,2,3]*Assistant Professor, FEST, Iqra University*
[4]*Senior Lecturer, Software Engineering department, Bahria University*
[5]*Senior Lecturer, CCSIS, IoBM*

[*1]alisiddiqui@iqra.edu.pk, [2]haider@iqra.edu.pk, [3]israr.ali@iqra,edu.pk,
[4]msohaibnaseem.bukc@bahria.edu.pk, [5]khaliq@iobm.edu.pk

DOI: https://doi.org/10.5281/zenodo.19438411

Keywords

Article History
Received: 11 February 2026
Accepted: 21 March 2026
Published: 06 April 2026

Corresponding Author: *
Ali Ahmad Siddiqui

Abstract

Properly allocating and scheduling resources in cloud computing systems helps achieve both performance *and cost goals. Poor scheduling can cause significant disruptions, which affect throughput, utilization of resources, and customer satisfaction. This study introduces a machine learning framework to predict scheduling delays for cloud instances.To predict scheduling delays for cloud instances, a complete dataset includes all requests and three types of supervised learning models (Random Forest, XGBoost, and Logistic Regression) have been evaluated. The dataset underwent many pre-processing steps, such as the elimination of leaks and the development of new features as well as the establishment of a classification target to identify which instance types had high-delays or low-delays when scheduled.*

Our results demonstrate that the ensemble-based models (Random Forest and XGBoost) performed better than the linear models because XGBoost correctly predicted scheduling arrangements 74.5% of the time, while also producing reasonable results in cases of class-imbalance. The combination of feature importance analysis and SHAP interpretation indicates that the total requested resource demand for CPUs, memory limit, and instance termination time are significant contributors to delays in scheduling.

As such, the approach proposed in this paper provides cloud service providers with the means to efficiently manage scheduling delays and thereby enhance the management of resources. Overall, this research illustrates that machine learning can accurately predict scheduling outcomes for cloud computing systems, thus contributing to the transition towards a more resource-efficient cloud computing environment.

1. Introduction

Cloud computing delivers software and database and server and storage and analytics and networking services through an internet-based system which operates at large-scale for both individual users and enterprise customers. The system gains widespread acknowledgment because it can achieve both cost savings and nonstop operational accessibility [1]. The Internet of Things (IoT) has become widely used in multiple fields which include intelligent transportation systems and healthcare management to generate

enormous quantities of data. The growing data volume has led organizations to depend more on cloud technologies which enable them to store and analyze data efficiently while minimizing costs and resource use.

Cloud environments function through their use of virtualized infrastructure which enables applications to operate on virtual machines (VMs). The VMs share access to CPU and memory and bandwidth resources. Applications need to run their processes concurrently, so they require scheduling systems which can identify the best sequence for their tasks. The scheduling process becomes unmanageable because organizations need to handle multiple applications at once. The solution to this problem has been established through cloud automation which uses artificial intelligence technology to automate scheduling and workload operations. Cloud automation develops intelligent virtualized environment systems which make instantaneous resource distribution and system operation decisions.

Researchers have investigated task scheduling in cloud computing through various studies which divide the field into traditional methods and intelligent systems. Traditional methods extend classical scheduling algorithms—First-In-First-Out (FIFO) Shortest Job First (SJF) Round Robin (RR) Min-Min and Max-Min—to develop scheduling solutions that operate effectively in cloud environments. The approaches present limitations because they cannot achieve simultaneous optimization of multiple system parameters which include makespan CPU utilization memory usage and bandwidth costs thus making them unsuitable for dealing with advanced cloud computing environments.

The intelligent approaches from reference seven to fifteen employ artificial intelligence methods which include fuzzy logic and particle swarm optimization and genetic algorithms to solve multi-objective optimization challenges. The methods provide benefits yet they function offline because they only optimize parameters after users submit their tasks. The system operates with extended execution times which makes it unfit for applications that require immediate response times such as IoT systems and big data analytics.

The researchers from the studies [16–22] investigated machine learning methods which included deep learning for automatic resource management in cloud computing systems. The methods use historical virtual machine resource consumption data to forecast upcoming workload patterns which leads to better resource distribution and avoids both under-provisioning and over-provisioning situations.

The recent research results demonstrate that data-driven methods enable organizations to predict scheduling delays and improve resource use throughout their operations. The machine learning (ML) framework identifies historical workload patterns which enables organizations to predict future system performance. Machine learning-based predictive scheduling allows cloud service providers to preemptively distribute resources while eliminating operational bottlenecks and improving system reliability.

The current study uses machine learning models to create methods that forecast future scheduling delays. The research employs a complete process that involves data cleaning and creation of new data elements and testing three common machine learning methods which include Random Forest and XGBoost and Logistic Regression. We use interpretability techniques together with feature importance analysis and SHAP values to understand which factors cause scheduling delays to occur.

The study presents three main contributions through its work results. The first contribution involves creating a cloud workload dataset which has been cleaned and processed to remove data leakage while developing essential features that represent resource usage and system constraints. The second contribution assesses three supervised learning models which categorize instances into high-delay and low-delay groups. The third contribution uses interpretability techniques to extract valuable information for cloud resource management which connects predictive modeling with actual decision-making processes. Overall, this study demonstrates the effectiveness of machine learning as a robust tool for cloud resource optimization, enabling predictive insights

that enhance scheduling efficiency and improve system performance.

2. Related Work

The authors of Reference [7] examine how organizations in cloud environments balance two competing goals which require them to complete work tasks while decreasing their energy usage. They create a multi-objective optimization problem which uses dynamic voltage frequency scaling (DVFS) as its foundation. The solution to this problem uses nondominated sorting genetic algorithm (NSGA-II) to find optimal solutions which are then evaluated through an artificial neural network model that determines the best virtual machines (VMs) based on specific task requirements.

The research work presented in Reference [8] develops a multi-agent cloud monitoring system which enhances security measures and operational performance to achieve better task scheduling results while stopping unauthorized task execution and changes. The authors in Reference [24] introduce an advanced particle swarm optimization (PSO) method which reduces task completion time while enhancing resource management efficiency. The method maintains particle weight updates through all iterations while it creates unpredictable elements during the final phase to prevent solutions from settling into particular local maxima.

Reference [9] focuses on optimizing three conflicting objectives which include makespan and resource utilization and execution cost. The authors developed a multi-objective optimization problem which they solved using an epsilon-fuzzy dominance-based composite discrete artificial bee colony algorithm to achieve their Pareto-optimal solutions. In Reference [10] the scheduling process uses a Bayesian framework to integrate trust levels of cloud resources. The researchers developed a trust-based dynamic scheduling algorithm which reduces costs and execution time while maintaining secure task processing.

Reference [25] proposes a two-stage scheduling system which helps to decrease resource waste in cloud data centers. The first stage uses a Bayesian classifier to group tasks based on their historical data, and the second stage applies dynamic scheduling algorithms to distribute tasks among suitable virtual machines. The resource allocation framework in Reference [11] enables Infrastructure-as-a-Service (IaaS) providers to subcontract their processing requirements to external cloud services whenever their internal resources are depleted. The problem is solved through an integer programming model which uses self-adaptive learning PSO-based scheduling to achieve maximum provider profit while delivering superior Quality of Service (QoS) results.

The research presented in Reference [12] introduces a bio-inspired hybrid algorithm called GAACO which combines genetic algorithms and ant colony optimization to achieve better task scheduling results for IoT systems running on diverse cloud platforms. Reference [26] applies a game-theoretic framework to develop an energy-efficient task scheduling solution that includes a mathematical model which ensures balanced workload distribution during big data processing tasks. The research in Reference [27] develops a two-stage scheduling game framework which uses game theory to create a user-centered system. The first stage uses a Stackelberg game model which allows IaaS providers to lead while users follow their demand preferences. The second stage uses a differential game to enhance service ratings which helps providers build their competitive advantage. Reference [28] uses evolutionary game theory together with genetic algorithms to study the process of cloud federation formation. The approach explores the search space to identify optimal federation payoff solutions while evolutionary game dynamics maintain stable operations between different federated cloud systems.

The approaches show diverse methods which work effectively yet share a common restriction because they depend on offline optimization for their operations. The methods need to optimize scheduling parameters whenever tasks arrive which creates excessive computational demands that prevent their use in applications requiring immediate processing capabilities like real-time big data analytics. Machine learning (ML) has been

widely explored as a solution for resource management in cloud computing which includes workload prediction and virtual machine placement and resource scaling. Ensemble techniques such as Random Forest and gradient boosting have demonstrated strong performance in capturing nonlinear relationships and improving predictive accuracy.

The traditional scheduling methods, which rely on fixed heuristics and established rules, experience challenges when trying to operate in changing cloud environments. The use of queueing theory for scheduling delay analysis has been established, yet the method relies on basic assumptions, which fail to model the intricate resource interactions present in multiple system resources. More recently, ML-based predictive scheduling techniques have gained attention. Regression-based models have been applied to estimate job completion times and resource utilization. The existing approaches mainly target continuous prediction tasks, which leads to an oversight of high-delay event classification that plays a vital role in Service Level Agreement (SLA) management. The ensemble learning algorithms Random Forest and XGBoost provide multiple benefits, which include their ability to model feature interactions and their capability to handle noisy data, while also enabling users to understand feature importance through their output.

The process of deploying machine learning models in cloud environments requires interpretability as an essential requirement. The SHAP framework which stands for SHapley Additive exPlanations provides system administrators with tools to analyze individual instances through its explanation system. The transparent system enables resource management to be executed in advance while it builds confidence in systems that use models for making decisions.

3. Dataset and Problem Formulation

3.1 Dataset Description

The study uses a dataset which comes from a public cloud instance workload repository that includes 23871 cloud instance requests. The instance records include various features which show resource requests and resource limits and the times when instances were created and scheduled and deleted and the maximum instances allowed per node. The dataset provides a rich foundation for analyzing factors influencing scheduling delays.

3.2 Data Preprocessing

To guarantee strong model capabilities while preventing data protection breaches, we implemented two methods which involve data processing. The first method required us to delete all identifier columns from the system which included instance_sn and role and app_name because these fields contained unique identifiers that would show actual scheduling results. The second method required us to process time data by converting creation_time and scheduled_time and deletion_time into numeric values while we used median imputation to fill in the missing data. The calculation of schedule delay involves determining the time difference between scheduled_time and creation_time which results in the target variable schedule_delay. The study eliminated all instances that had negative delays or incomplete scheduling information. The study developed new system features through engineering work which included the following system dynamics features: CPU-memory ratio; Disk-CPU ratio; Total demand which represents the sum of CPU and memory and disk and RDMA requests; Memory pressure and disk pressure ratios; Maximum instances per node as system load indicator The preprocessing of the dataset resulted in 12 numeric features and 12,390 clean samples which become available for supervised learning.

3.3 Problem Formulation

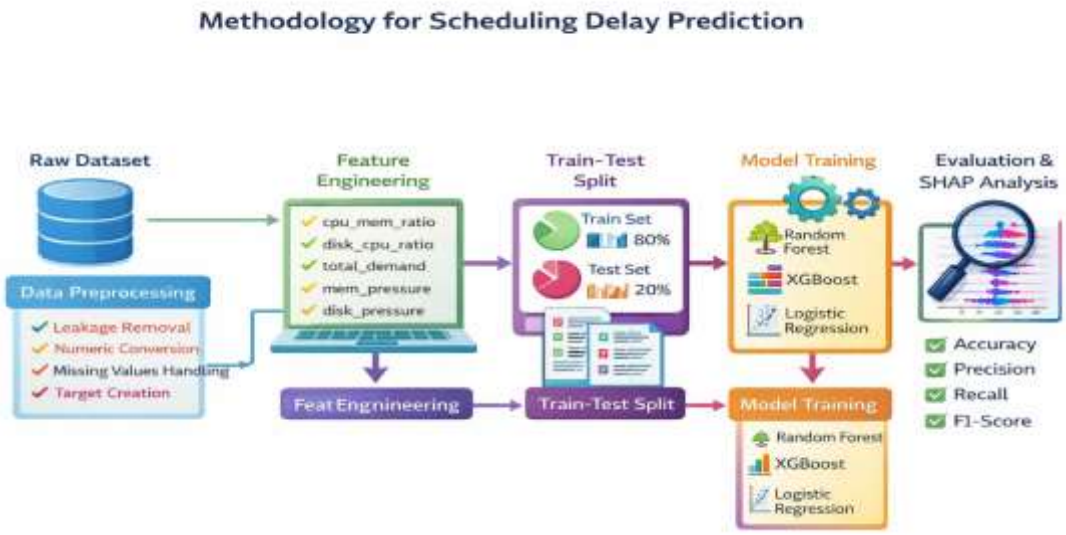
The prediction task was formulated as a **binary classification problem**:

- **Class 0:** Low-delay instances (schedule delay ≤ median)
- **Class 1:** High-delay instances (schedule delay > median)

This setup allows us to focus on identifying instances at risk of excessive delays, which is most relevant for operational decision-making.

4. Methodology

4.1 Model Selection



We selected three supervised learning models:

1. **Random Forest Classifier:** An ensemble of decision trees using bagging to reduce variance and improve robustness to noise.
2. **XGBoost Classifier:** Gradient boosting model that captures nonlinear interactions between features, widely used in predictive modeling for high-dimensional datasets.
3. **Logistic Regression:** Baseline linear classifier for comparison, highlighting limitations of simple models in complex environments.

4.2 Training and Evaluation

The dataset was split into **80% training** and **20% testing**, maintaining class distribution. Hyperparameters were set as:

- Random Forest: 300 trees, no maximum depth
- XGBoost: 300 estimators, learning rate 0.1, logloss evaluation
- Logistic Regression: max iterations 1000

Models were evaluated using **accuracy, precision, recall, and F1-score**, focusing particularly on the high-delay class due to its operational significance.

4.3 Feature Interpretation

Feature importance was derived using:

- **Random Forest Gini importance**
- **SHAP values**: for instance-level and global interpretability

SHAP analysis allowed identification of features most responsible for predicting high-delay instances, providing actionable insights for cloud administrators.

5. Results

5.1 Classification Performance

Table 1: Model Performance on Test Set

Model	Accuracy	Precision (High-delay)	Recall (High-delay)	F1-Score (High-delay)
Random Forest	0.7187	0.57	0.55	0.56
XGBoost	0.7449	0.65	0.47	0.55
Logistic Regression	0.6848	0.61	0.10	0.18

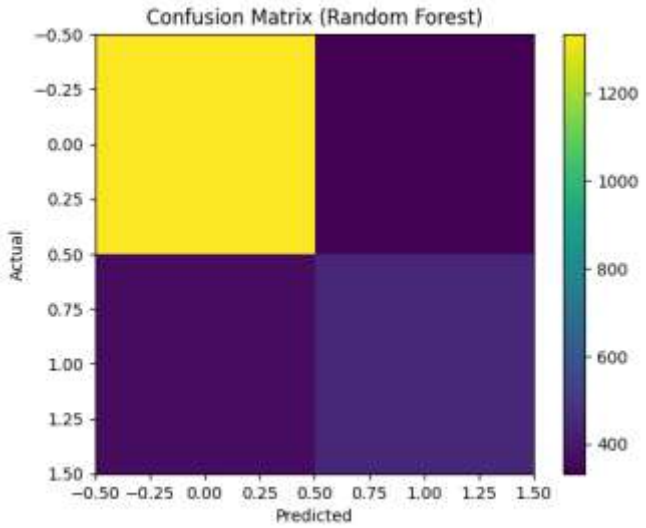


Figure 1: Confusion matrix of Random Forest classifier

Observations:

- XGBoost achieved the highest overall accuracy (74.5%) and balanced class-wise performance.
- Random Forest demonstrated reliable predictive capability with a moderate trade-off between precision and recall.
- Logistic Regression performed poorly for high-delay instances, highlighting the need for nonlinear modeling approaches.

5.2 Feature Importance Analysis

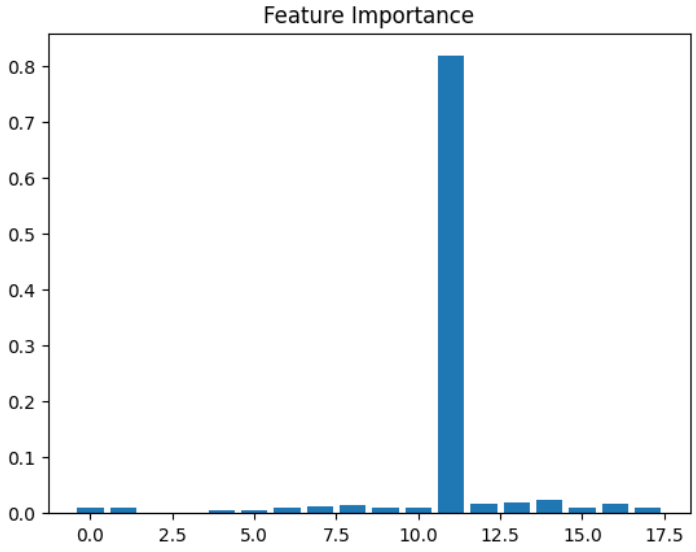


Figure 2: Random Forest feature importance

Key findings:

- **CPU request** and **total demand** are dominant predictors.
- Memory request and limits significantly contribute, consistent with system resource bottlenecks.
- Deletion time and max instances per node have lower but non-negligible influence.

5.3 SHAP Interpretation

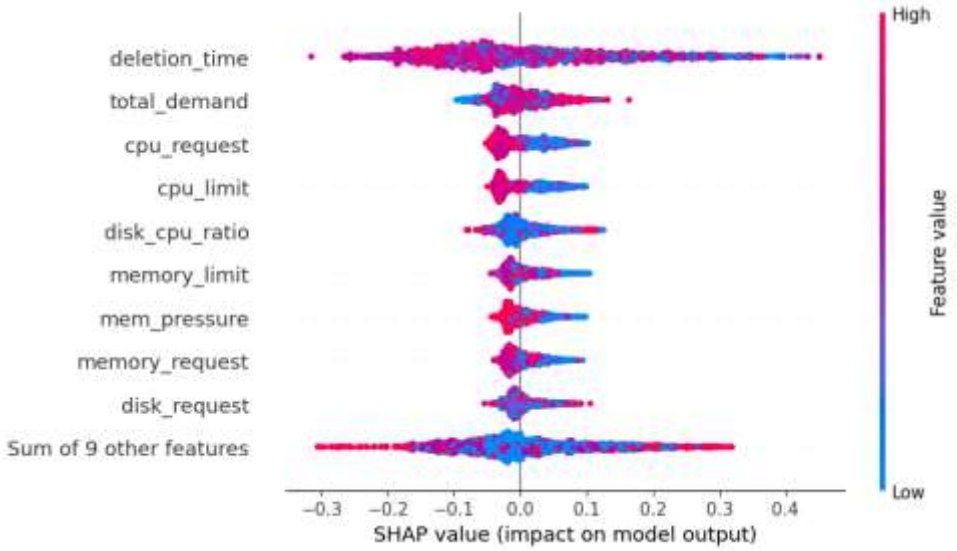


Figure 3: SHAP beeswarm plot for high-delay class (see uploaded figure).

Insight

- High CPU and total demand figures indicate a higher likelihood of scheduling delays.
- Secondary contributors to scheduling delays are memory and disk pressures; therefore, the two resources are indicative of resource contention.
- Individual interpretation of each prediction by SHAP helps in the making of proactive decisions.

6. Discussion

This conclusion provides a number of critical insights:

1. Ensemble models perform better than linear models for predicting scheduling delay events, highlighting the value of modeling the nonlinear interactions between features.
2. Feature engineering is important in practice, with some derived features (e.g., CPU memory ratio, total demand) significantly contributing to improvements in model performance.
3. Through the use of SHAP analysis, model users can take appropriate actions based on insights from this analysis, encouraging system administrators to prioritize the allocation of resources toward CPU-bound workloads as a method of reducing high-delay event occurrences.
4. Class imbalance continues to present a challenge, where high delay instances are more difficult to predict; thus, further improvements can be gained through advanced techniques (e.g.; SMOTE; cost-sensitive learning).
5. The operational business implications of these insights include: Through proactive scheduling utilizing model predictions, optimizing resource utilization and eliminating bottlenecks will improve overall Service Level Agreement (SLA) compliance.

7. Conclusion

This study proposes a complete machine-learning system to predict the occurrence of scheduling delays in Cloud Computing Systems. The study also confirms through analysis that Random Forest and XGBoost ensemble methods can predict instances of high delayed occurrence because we processed a large dataset, developed good feature sets, and tested three different types of supervised learning techniques. The study found that XGBoost produced the best accuracy with 74.50% of predicted correct values.

The feature importance analysis and SHAP value analysis both indicated that CPU Request, total resource utilization and memory constraint are the primary causal factors of scheduling delays Cloud Resource Management will benefit from the research by providing a framework for supporting

predictive scheduling along with optimizing Cloud Resource Allocation.

Future research directions for this study include investigating Deep Learning Architectures to create more accurate predictions of real-time workloads as well as investigating new methods for addressing class imbalance.

8. Figures and Tables

- **Figure 1:** Confusion matrix of Random Forest classifier
- **Figure 2:** Random Forest feature importance
- **Figure 3:** SHAP beeswarm plot for high-delay class
- **Table 1:** Classification performance of Random Forest, XGBoost, and Logistic Regression

REFERENCES

1. Wahab OA, Bentahar J, Otrok H, Mourad A. Resource-aware detection and defense system against multi-type attacks in the cloud: repeated Bayesian stackelberg game. IEEE Trans Depend Secure Comput. 2019. https://doi.org/10.1109/TDSC.2019.2907946.
2. Rjoub G, Bentahar J, Wahab OA. BigTrustScheduling: trust-aware big data task scheduling approach in cloud computing environments. Future Generat Comput Syst. 2020;110:1079–1097. https://doi.org/10.1016/j.future.2019.11.019.
3. Singh S, Chana I. QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput Surv. 2016;48(3):42.
4. Ghomi EJ, Rahmani AM, Qader NN. Load-balancing algorithms in cloud computing: a survey. J Netw Comput Appl. 2017;88:50-71.
5. Bhoi U, Ramanuj PN. Enhanced max-min task scheduling algorithm in cloud computing. Int J Appl Innovat Eng Manag. 2013;2(4):259-264.
6. Chen H, Wang F, Helian N, Akanmu G. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. Paper presented at: Proceedings of the National Conference on Parallel Computing Technologies; 2013:1-8.
7. Sofia AS, GaneshKumar P. Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II. J Netw Syst Manag. 2018;26(2):463-485.
8. Grzonka D, Jakobik A, Kołodziej J, Pllana S. Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security. Future Generat Comput Syst. 2018;86:1106-1117.
9. Gomathi B, Krishnasamy K, Balaji BS. Epsilon-fuzzy dominance sort-based composite discrete artificial bee colony optimisation for multi-objective cloud task scheduling problem. Int J Bus Intell Data Mining. 2018;13(1-3):247-266.
10. Wang W, Zeng G, Tang D, Yao J. Cloud-DLS: dynamic trusted scheduling for cloud computing. Exp Syst Appl. 2012;39(3):2321-2329.
11. Zuo X, Zhang G, Tan W. Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IAAS cloud. IEEE Trans Automat Sci Eng. 2014;11(2):564-573.
12. Basu S, Karuppiah M, Selvakumar K, et al. An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment. Future Generat Comput Syst. 2018;88:254–261.
13. Bataineh AS, Mizouni R, Bentahar J, El Barachi M. Toward monetizing personal data: a two-sided market analysis. Future Generat Comput Syst. 2020;111:435–459
14. Rjoub G, Bentahar J. Cloud task scheduling based on swarm intelligence and machine learning. In: Younas M, Aleksy M, Bentahar J, eds. Proceedings of the 5th IEEE International Conference on Future

Internet of Things and Cloud, FiCloud. Prague, Czech Republic: IEEE; 2017:272-279.
15. Peng Z, Lin J, Cui D, Li Q, He J. A multi-objective trade-off framework for cloud resource scheduling based on the deep Q-network algorithm. Cluster Comput. 2020. https://doi.org/10.1007/s10586-019-03042-9.
16. Barrett E, Howley E, Duggan J. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. Concurr Comput Pract Exp. 2013;25(12):1656-1674.
17. Song B, Yu Y, Zhou Y, Wang Z, Du S. Host load prediction with long short-term memory in cloud computing. J Supercomput. 2018;74(12):6554-6568.
18. Liu N, Li Z, Xu J, et al. A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. Paper presented at: Proceedings of the IEEE 37th International Conference on Distributed Computing Systems; 2017:372-382.
19. Qiu F, Zhang B, Guo J. A deep learning approach for VM workload prediction in the cloud. In: Chen Y, ed. 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), . Shanghai: IEEE/ACIS; 2016:319-324. http://doi.org/10. 1109/SNPD.2016.7515919.
20. Wahab OA, Kara N, Edstrom C, Lemieux Y. MAPLE: a machine learning approach for efficient placement and adjustment of virtual network functions. J Netw Comput Appl. 2019;142:37-50.
21. Wahab OA, Cohen R, Bentahar J, Otrok H, Mourad A, Rjoub G. An endorsement-based trust bootstrapping approach for newcomer cloud services. Inf Sci. 2020;527:159–175. https://doi.org/10.1016/j.ins.2020.03.102.
22. Rohoden K, Estrada R, Otrok H, Dziong Z. Stable femtocells cluster formation and resource allocation based on cooperative game theory. Comput Commun. 2019;134:30-41