Experimental Analysis of E-ARIMA for Workload Prediction in Cloud Computing Services

Krishan Kumar[1], K. Gangadhara Rao[2], Bobba Basaveswara Rao[3], Suneetha Bulla[4]
[1]Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, India, krishan0405@gmail.com
[2,3]Professor, Department of CSE, Acharya Nagarjuna University, Guntur, India. kancherla123@gmail.com
[4]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, 522502, AP, India. suneethabulla@gmail.com

Abstract: The exponential growth of data in cloud computing necessitates efficient resource management to maintain service availability and enhance runtime performance. This paper presents an experimental analysis of the E-ARIMA (Enhanced Auto Regressive Integrated Moving Average) model for workload prediction in cloud computing services. The proposed framework aims to address two critical technical issues: balancing large volumes of data across existing resources and optimally adjusting the number of resources to accommodate time-varying workloads. The E-ARIMA model, configured to predict future workloads and enable dynamic resource provisioning. This predictive capability helps mitigate the issues of over-provisioning and under-provisioning, ensuring optimal resource utilization in the cloud environment. To validate the effectiveness of the proposed framework, we conducted extensive experiments using AWS cloud services. The results demonstrate significant improvements in resource management, leading to enhanced service performance and reduced response times. The experimental analysis underscores the potential of E-ARIMA in facilitating efficient and adaptive resource management in cloud computing. This framework offers a robust solution for dynamic resource provisioning in cloud environments, addressing the challenges of data growth and variable workloads. The findings highlight the importance of predictive models like E-ARIMA in optimizing resource utilization and ensuring seamless service delivery in cloud computing

1. Introduction

The rapid growth of cloud computing has revolutionized the way computing resources are utilized, providing scalable and flexible solutions for various applications. With the increasing demand for cloud services, effective resource management has become a critical aspect to ensure high availability and optimal performance. As data continues to grow at an unprecedented rate, traditional methods of resource allocation and management are proving insufficient. Therefore, innovative approaches are necessary to dynamically adjust resources in response to fluctuating workloads and user demands [1]. In cloud environments, resource management involves two primary challenges: balancing a large volume of data across available resources and provisioning resources efficiently to adapt to time-varying workloads. Traditional resource management strategies, which are based on demand, availability, and scheduling, often lead to delayed response times and suboptimal resource utilization. This paper aims to address these challenges by proposing a novel framework that leverages workload prediction to enhance resource provisioning in AWS cloud services [3].

Workload prediction is a pivotal component in managing cloud resources dynamically. By accurately forecasting future workloads, cloud service providers can preemptively allocate resources, thereby avoiding the pitfalls of over-provisioning and under-provisioning. Over-provisioning results in unnecessary resource wastage, while under-provisioning can lead to performance degradation and unmet user demands. To tackle these issues, our framework incorporates the Enhanced Auto Regressive Integrated Moving Average (E-ARIMA) model, which is designed to predict workload patterns with high accuracy.

The E-ARIMA model extends the traditional ARIMA model by incorporating enhancements that improve its predictive capabilities in the context of cloud computing workloads. These enhancements include modifications to handle the non-stationary and volatile nature of cloud workloads, making the model more robust and reliable for dynamic resource provisioning. The integration of E-ARIMA into our resource management framework enables a proactive approach to resource allocation, ensuring that resources are available precisely when needed.

To validate the effectiveness of our proposed framework, we conducted extensive experimental analyses using AWS cloud services. AWS provides a versatile platform for testing and implementing cloud resource management strategies due to its wide range of services and scalability. Our experiments involved deploying the E-ARIMA model to predict workloads and dynamically allocate resources based on these predictions. The results demonstrated significant improvements in resource utilization and service performance, highlighting the practical benefits of our approach. The findings from our experiments underscore the potential of predictive models like E-ARIMA in optimizing cloud resource management. By enabling precise and timely resource provisioning, our framework enhances the overall efficiency and performance of cloud services. This proactive resource management approach not only improves user experience by reducing response times but also contributes to cost savings by minimizing resource wastage.

The proposed framework represents a significant advancement in cloud resource management, addressing the critical challenges posed by the rapid growth of data and fluctuating workloads. The integration of the E-ARIMA model for workload prediction and dynamic resource provisioning provides a robust solution for maintaining high availability and optimal performance in cloud environments. This paper sets the foundation for future research and development in predictive resource management strategies, paving the way for more efficient and adaptive cloud computing services.

The rest of the paper is organized as follows. Section 2 describes the related work for the proposed approach. Section 3 experimental setup is described. Section 5 presents results with discussion. Finally, proposed work is concluded with future work.

2 Literature review

Several research papers on workload prediction have addressed the problem with different approaches [1][2][3][4]. These approaches can be broadly categorized based on the models used: statistical-based predictions and historical data-based predictions. Statistical-based workload predictions involve simple calculations such as addition, subtraction, and computing the mean of previous workloads. These methods are straightforward and provide quick estimates but may lack the sophistication needed to capture complex patterns in workload data. On the other hand, historical data-based prediction models analyze historical data to extract patterns and predict future workload instances. These models typically use more advanced techniques, including machine learning algorithms and time series analysis, to identify trends and seasonality in the data. By leveraging historical workload data, these models can provide more accurate and nuanced predictions, enabling better resource management in cloud environments.

Calheiros et, al. [5] used a classic and basic time-series forecasting method Autoregressive Moving Average (ARMA) mode to predict the workload. However, the limitation is that it may not work well for all types of workload traces. A Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) is combined with an Autoregressive Integrated Moving Average (ARIMA) for prediction [6]. Authors in [7] ARIMA combined with Radial Basis Function (RBF) for prediction purposes. Piacentini et, al. [8] used a combined approach of feed-forward Artificial Neural Network (ANN) and Support Vector Machine (SVM). Bao et al. [9] combined Relevance Vector Machine (RVM) with differential Empirical Mode Decomposition (EMD) for short-term predictions. Sharifan et al. [10] combines GARCH with the Support Vector Regression (SVR) algorithm for cloud computing workload prediction in mobile applications.

The popularity of Deep Learning (DL) approaches motivates to development of specific approaches for time-series analysis and prediction [11–14]. Hussain et al. [15] proposed a Recurrent Neural Network (RNN) based approach for time-series prediction. Gao et al. [16] a deep learning approach consists of a Deep Belief Network (DBF) and a regression layer that predicts VM workload Prediction. A simulation approach and a Genetic approach are combined for workload prediction [17]. To predict workload demand along with uncertainty a univariate and bivariate Bayesian deep learning approach is proposed by authors [18]. Setayesh et al. [19] proposed a multivariate attention and Gated (MAGD) Recurrent Unit-based deep learning approach for cloud workload forecasting.

Liu et al. [20] proposed a hierarchical framework that comprises two tiers one global tier for VM allocation and one local tier for power management. Deep Reinforcement Learning (DRL) technique is used to solve the global tier problem and LSTM-based workload prediction is used for the local tier. The experimental results show that the proposed framework significantly reduces energy consumption along with latency. Shahin [21] proposed a scaling technique to minimize the Slashdot problem, where the sudden arrival of traffic may not allow the auto-scale technique to scale. The paper predicts the required resources using LSTM-RNN, detects the Slashdot situation at an earlier stage, and performs suitable scalable action. The results show that the proposed approach reduces the Slashdot effect and improves SLA. White et al. [22] proposed an LSTM-based neural network to forecast QoS values. The authors used the frequency of service monitoring and changed it to increase the prediction accuracy.

Kumar et al. [23] proposed an approach that handles dynamic resource scaling using an LSTM network. Janardhana et al. [24] focus on forecasting of CPU usage of the machine. For this, the authors use time series of CPU usage. An LSTM network was proposed to forecast the CPU workload of the machine and results show significant improvement over the ARIMA model. Shan et al. [25] predicted the CPU and Bandwidth one time step ahead and compared it to a multi-time step ahead. The prediction was determined using RNN-BPNN, LSTM, and ARIMA models for comparison on different metrics. Bi et al. [26] proposed an integrated forecasting technique. The proposed approach comprises of Savitzky–Golay filter that eliminates noise and smooth out the non-stationary workload. After that it is combined with the LSTM model to predict time series, The results show better prediction results than BPNN. Shen et al. [27] , proposed LSTM-based workload prediction capable of handling long-term dependencies. To improve the long-term learning capabilities of LSTM, the author proposed a Bi-directional LSTM (BiLSTM), which consists of two LSTM RNN units that act as forward and backward LSTM. The main idea is to use one hidden layer that processes the data from a forward direction and models the impact of historical information. Another hidden layer is used to process the data from a backward direction and model the impact of future information. The proposed method can consider historical information and future information simultaneously and predict more effectively.

.

3. Experimental Setup

The web workload involves real traces of web requests at cloud servers from Wikibooks [28]. This dataset contains page count files with hourly statistics, including the name of the requested page, the language of the accessed page, the number of requests in thousands, and the size of the returned content. To manage and aggregate this large amount of data from web servers into a centralized data storage system within the Hadoop ecosystem, Apache Flume is employed. Apache Flume is recognized as a standard, reliable, and robust tool for forwarding data from web servers to HDFS (Hadoop Distributed File System) storage at high speeds. this setup, a web server generates log data, which is then collected by a Flume agent. The agent buffers the collected data in a channel before forwarding it to a sink, which ultimately moves the data to centralized storage, such as HDFS. The Hadoop cluster for this system is set up on Amazon EC2 cloud environments, ensuring scalability and robustness for handling large volumes of web request data.

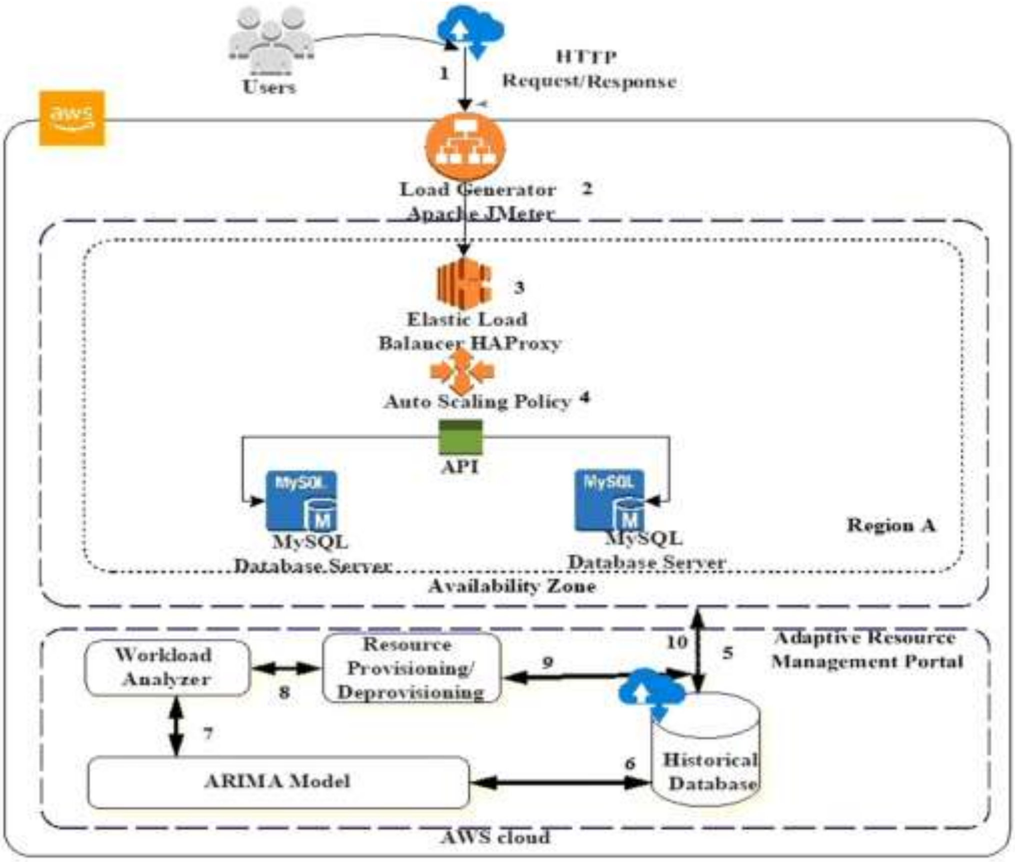

Fig. 1 Resource management portal with AWS cloud

Cloud applications require specific configurations, compositions, and deployment strategies to function optimally. To address these needs, Amazon Web Services (AWS) provides a robust framework for building experimental setups in public clouds, facilitating resource provisioning. The experimental setup for this study, depicted in Figure 6, includes various AWS services and tools: AWS EC2 (Elastic Compute Cloud) for scalable computing capacity, AWS instances, APIs, the Apache JMeter load generator, AWS Elastic Load Balancer HAProxy, MySQL database server, and CloudWatch alarms.

Amazon EC2 is a key component, offering scalable computing capacity within the AWS cloud. It enables the dynamic allocation of resources based on workload demands, ensuring that applications have the necessary compute power when needed. By integrating these components, the cloud framework can efficiently manage resources, handle variable workloads, and provide a resilient and scalable environment for cloud applications. This setup supports the dynamic and adaptive resource provisioning required for effective cloud resource management.

We can use Amazon EC2 to launch as many or as few virtual servers as needed, configure security and networking, and manage storage. Amazon EC2 enables scaling up or down to handle changes in requirements. An Amazon Machine Image (AMI) template is used to configure software, such as an operating system, an application server, and applications. AMIs launch instances, which are copies of the AMI running as virtual servers in the cloud. These launched instance types provide compute and memory facilities. HTTP or HTTPS requests that use the HTTP verbs GET or POST are redirected to the Apache Webserver with a MySQL database. User requests are linked with the load generator Apache JMeter, which is set up using Java programming. Forwarded requests from Apache JMeter are sent to the elastic load balancer HAProxy.

The requests are then forwarded to the application programming interface and the MySQL database server. The API, Apache JMeter, and database are set up on the AWS t2.micro instances. Amazon Elastic Block Store (Amazon EBS volumes) is used to store persistent data. An auto-scaling policy is applied based on CloudWatch alarms to increase or decrease the EC2 instance count. CloudWatch alarms analyze and observe specific metrics and parameters to trigger the alarm and specify thresholds for selected metrics and parameters. These metrics and parameters are analyzed using a prediction model to adapt to continuous changes in user workload and dynamically fulfill user requirements. The system checks the availability of resources in specified regions, calculates the capability of available VMs, and then allocates job requests to the resources. Once the number of jobs is allotted to the VMs, the current load is calculated. If a VM becomes overloaded or underloaded, resources will be added or released as required automatically. User requests, including the IP address of the source, time, request number, and location, were extracted over a period of one year. Apache JMeter is used to test the performance of static and dynamic resources and web dynamic applications. HTTP requests are sent to JMeter for simulation.

Apache JMeter is used to simulate a heavy load on a server, group of servers, or network to test its strength or analyze overall performance under different load types. It simulates web requests from the current and historical database. Apache JMeter forwards incoming requests to the Elastic Load Balancer HAProxy after simulation. The Elastic Load Balancer automatically distributes incoming application traffic across EC2 instances. It supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. It will add and remove compute resources as needed, without disrupting the overall flow of the workload.

The auto-scaling policy helps define rules for dynamically increasing or decreasing the number of EC2 instances based on CloudWatch alarms. Amazon's auto-scaling group policy defines the group rules and launches the instances. Instances will be provisioned or deprovisioned based on CPU utilization metrics. During each configuration, CloudWatch monitors CPU utilization, memory utilization, response time, throughput, disk read bytes, disk write bytes, network-in bytes, and network-out bytes. All configured parameters may be used in the future for other service provisioning in the cloud environment.

6 Results and discussion

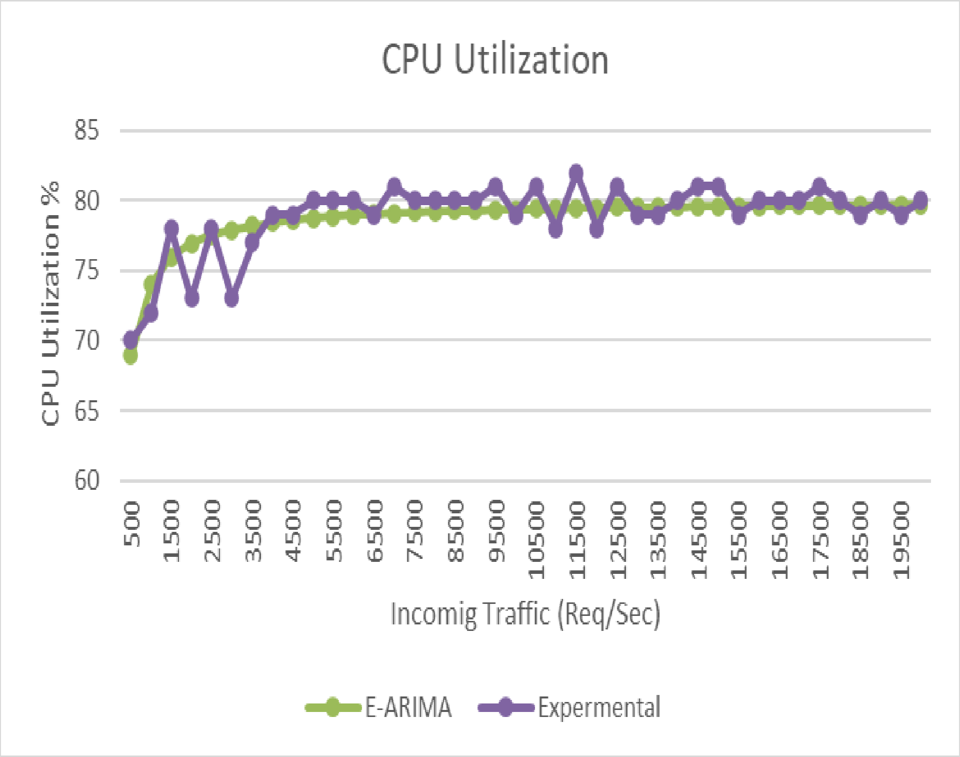

Fig2: CPU utilization in EARIMA and Experimental

CPU utilization within an optimal range to prevent both underutilization and overloading of resources. Underutilized instances indicate wasted resources, leading to higher costs without corresponding performance benefits, whereas overloaded instances can degrade performance, leading to increased response times and potential service interruptions. The E-ARIMA model played a crucial role in predicting workload patterns, enabling proactive resource provisioning. By accurately forecasting future CPU demands, the framework dynamically adjusted the number of active EC2 instances to match the predicted workload.

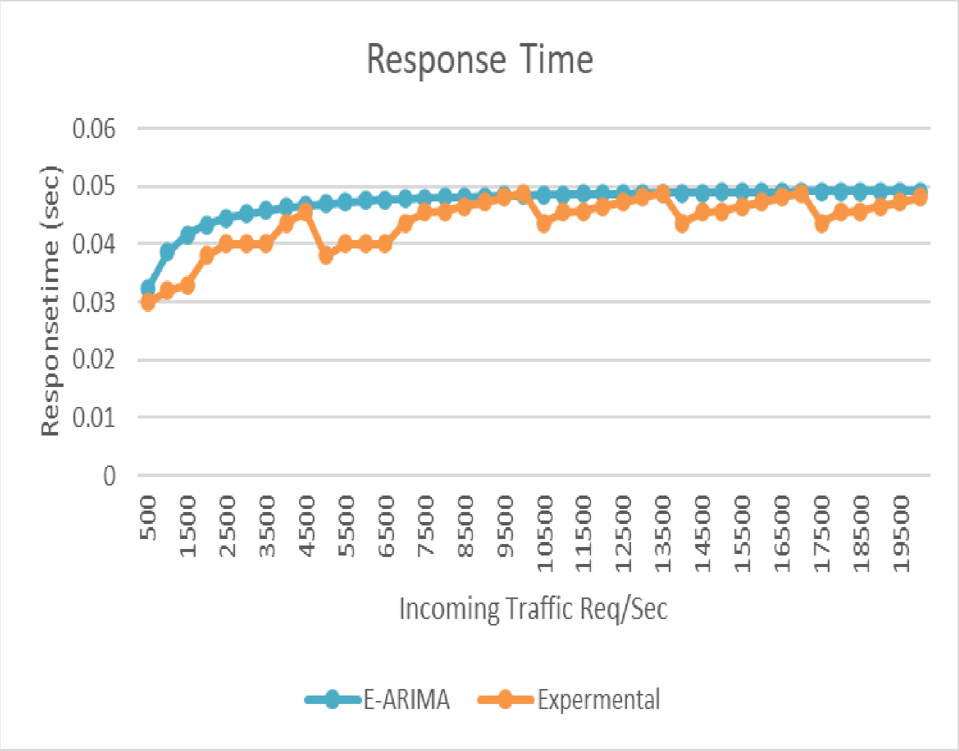

Fig2: Response Time in EARIMA and Experimental

In the experiment, response times were monitored using Apache JMeter, which simulated web requests and measured the time taken for each request to be processed and responded to by the system. The adaptive resource provisioning strategy aimed to minimize response times by ensuring that adequate computing resources were available to handle incoming workloads. The integration of the elastic load balancer HAProxy helped distribute incoming traffic evenly across the available EC2 instances, preventing any single instance from becoming a bottleneck. By dynamically scaling resources based on workload predictions, the framework effectively reduced response times, ensuring a smooth and responsive user experience even during peak load periods.

7 Conclusion

This paper proposed and evaluated a framework for dynamic and adaptive resource provisioning in AWS cloud services, leveraging the E-ARIMA model for workload prediction. Our experimental analysis demonstrated the effectiveness of the E-ARIMA model in forecasting workloads and enabling proactive resource management, addressing the challenges of over-provisioning and under-provisioning. By utilizing Apache Flume for efficient data collection and aggregation, and deploying a comprehensive AWS framework including EC2 instances, elastic load balancers, and CloudWatch alarms. E-ARIMA can significantly improve CPU utilization, reduce response times, and optimize costs. By continuously monitoring these metrics and dynamically adjusting resources, the proposed framework ensures a balanced and efficient cloud infrastructure capable of handling varying workloads while maintaining high performance and cost-efficiency. the proposed framework represents a significant advancement in cloud resource management, offering a practical solution to the challenges posed by rapidly growing data and variable workloads. This research lays the groundwork for further exploration and development of predictive resource management strategies, paving the way for more efficient and adaptive cloud computing services.

References:

1. L. Ruan, Y. Bai, S. Li, S. He, L. Xiao, Workload time series prediction in storagesystems: a deep learning based approach, Clust. Comput. (2021).
2. H.M. Nguyen, G. Kalra, D. Kim, Host load prediction in cloud computing using long short-term memory encoder–decoder, J. Supercomput. 75 (11) (2019) 7592–7605.
3. P.A. Dinda, D.R. Hallaron, Host load prediction using linear models, Cluster Comput. 3 (4) (2000) 265–280.
4. W. Zhong, Y. Zhuang, J.J. Sun, J. Gu, A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine, Appl. Intell. 48 (11) (2018) 4072–4083.
5. R.N. Calheiros, E. Masoum, R. Ranjan, R. Buyya, Workload prediction using ARIMA model and its impact on cloud applications QoS, IEEE Trans. Cloud Comput. 3 (2015) 449–458.
6. Z. Tan, J. Zhang, J. Wang, J. Xu, Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models, Appl. Energy 87 (2010) 3606–3610.
7. M. Shafe-Khah, M.P. Moghaddam, M.K. Sheikh-El-Eslami, Price forecasting of day-ahead electricity markets using a hybrid forecast method, Energy Convers. Manag. 52 (2011) 2165–2169.
8. M. Piacentini, F. Rinaldi, Path loss prediction in urban environment using learning machines and dimensionality reduction techniques, Comput. Manag. Sci. 8 (2011) 371–385.
9. Y. Bao, H. Wang, B. Wang, Short-term wind power prediction using differential EMD and relevance vector machine, Neural Comput. Appl. 25 (2014) 283–289.
10. S. Sharifan, M. Barati, An ensemble multiscale wavelet-GARCH hybrid SVR algorithm for mobile cloud computing workload prediction, Int. J. Mach. Learn.
Cybern. 10 (2019) 3285–3300.
11. A. Mozo, B. Ordozgoiti, S. Gómez-Canaval, Forecasting short-term data center network traffic load with convolutional neural networks, PLoS One (2018).
12. M. Amiri, L. Mohammad-Khanli, R. Mirandola, An online learning model based on episode mining for workload prediction in cloud, Future Gener. Comput. Syst. 87 (2018) 83–101.
13. H. Wang, G. Li, G. Wang, J. Peng, H. Jiang, Y. Liu, Deep learning based ensemble approach for probabilistic wind power forecasting, Appl. Energy 188 (2018) 56–70.
14. H. Assem, S. Ghariba, G. Makrai, P. Johnston, L. Gill, F. F. Pilla, Urban water fow and water level prediction based on deep learning, in: Lecture Notes Computer Science (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), in: LNAI, vol. 10536, 2017, pp. 317–329.
15. A.J. Hussain, D. Al-Jumeily, H. Al-Askar, N. Radi, Regularized dynamic selforganized neural network inspired by the immune algorithm for financial time series prediction, Neurocomputing 188 (2016) 23–30
16. L. Ruan, Y. Bai, S. Li, S. He, L. Xiao, Workload time series prediction in storage systems: a deep learning based approach, Clust. Comput. (2021).
17. H.M. Nguyen, G. Kalra, D. Kim, Host load prediction in cloud computing using long short-term memory encoder–decoder, J. Supercomput. 75 (11) (2019) 7592–7605.
18. P.A. Dinda, D.R. Hallaron, Host load prediction using linear models, Cluster Comput. 3 (4) (2000) 265–280.

19. W. Zhong, Y. Zhuang, J.J. Sun, J. Gu, A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine, Appl. Intell. 48 (11) (2018) 4072–4083.
20. R.N. Calheiros, E. Masoum, R. Ranjan, R. Buyya, Workload prediction using ARIMA model and its impact on cloud applications QoS, IEEE Trans. Cloud Comput. 3 (2015) 449–458.
21. Z. Tan, J. Zhang, J. Wang, J. Xu, Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models, Appl. Energy 87 (2010) 3606–3610.
22. M. Shafe-Khah, M.P. Moghaddam, M.K. Sheikh-El-Eslami, Price forecasting ofb day-ahead electricity markets using a hybrid forecast method, Energy Convers. Manag. 52 (2011) 2165–2169.
23. M. Piacentini, F. Rinaldi, Path loss prediction in urban environment using learning machines and dimensionality reduction techniques, Comput. Manag. Sci. 8 (2011) 371–385.
24. Y. Bao, H. Wang, B. Wang, Short-term wind power prediction using differential EMD and relevance vector machine, Neural Comput. Appl. 25 (2014) 283–289.
25. S. Sharifan, M. Barati, An ensemble multiscale wavelet-GARCH hybrid SVR algorithm for mobile cloud computing workload prediction, Int. J. Mach. Learn. Cybern. 10 (2019) 3285–3300.
26. A. Mozo, B. Ordozgoiti, S. Gómez-Canaval, Forecasting short-term data center network traffic load with convolutional neural networks, PLoS One (2018).
27. M. Amiri, L. Mohammad-Khanli, R. Mirandola, An online learning model based on episode mining for workload prediction in cloud, Future Gener. Comput. Syst. 87 (2018) 83–101.
28. H. Wang, G. Li, G. Wang, J. Peng, H. Jiang, Y. Liu, Deep learning based ensemble approach for probabilistic wind power forecasting, Appl. Energy 188 (2018) 56–70.
29. H. Assem, S. Ghariba, G. Makrai, P. Johnston, L. Gill, F. F. Pilla, Urban water fow and water level prediction based on deep learning, in: Lecture Notes Computer Science (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), in: LNAI, vol. 10536, 2017, pp. 317–329.
30. A.J. Hussain, D. Al-Jumeily, H. Al-Askar, N. Radi, Regularized dynamic selforganized neural network inspired by the immune algorithm for financial time series prediction, Neurocomputing 188 (2016) 23–30