World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:17, No:3, 2023

Distributed System Computing Resource Scheduling Algorithm Based on Deep Reinforcement Learning

Yitao Lei, Xingxiang Zhai, Burra Venkata Durga Kumar

***Abstract*—**As the quantity and complexity of computing in large-scale software systems increase, distributed system computing becomes increasingly important. The distributed system realizes high-performance computing by collaboration between different computing resources. If there are no efficient resource scheduling resources, the abuse of distributed computing may cause resource waste and high costs. However, resource scheduling is usually an NP-hard problem, so we cannot find a general solution. However, some optimization algorithms exist like genetic algorithm, ant colony optimization, etc. The large scale of distributed systems makes this traditional optimization algorithm challenging to work with. Heuristic and machine learning algorithms are usually applied in this situation to ease the computing load. As a result, we do a review of traditional resource scheduling optimization algorithms and try to introduce a deep reinforcement learning method that utilizes the perceptual ability of neural networks and the decision-making ability of reinforcement learning. Using the machine learning method, we try to find important factors that influence the performance of distributed system computing and help the distributed system do an efficient computing resource scheduling. This paper surveys the application of deep reinforcement learning on distributed system computing resource scheduling. The research proposes a deep reinforcement learning method that uses a recurrent neural network to optimize the resource scheduling. The paper concludes the challenges and improvement directions for Deep Reinforcement Learning-based resource scheduling algorithms.

***Keywords*—**Resource scheduling, deep reinforcement learning, distributed system, artificial intelligence.

I. Introduction

DISTRIBUTED computing is a type of computing system that utilizes multiple computing elements to process large amounts of data. A distributed system is a seemingly coherent system consisting of many computing elements [1]. Distributed computing applies distributed systems in large-scale computing. It divides the data to be processed into many small parts, computed by different distributed system components, and finally integrates the results. However, unreasonable computing resource scheduling cannot take advantage of the full performance of distributed computing system. To fully use the system resources, developing an effective distributed computing system resource scheduling algorithm is necessary.

Since most resource scheduling problems are NP-hard, there are few studies focusing on finding a standard solution that fits all situations. Traditionally there are distributed computing resource scheduling algorithms like rotation method, weight method, least connection method, least missing method, and fastest response method. These algorithms manage computing resources with some single principle like priority and response time. However, many more elements influence the performance of the distributed system: Network latency, data communication efficiency, compute node processing power, task segmentation, inability to budget processing time, task turbulence, and so on. However, some advanced algorithms like genetic algorithm, ant colony motivation, and other heuristic algorithms are applied in distributed system computing resource scheduling. Table I shows some representative heuristic algorithms and their characteristics. However, it is difficult for traditional heuristic algorithms to process large, complex, and dynamic resource scheduling [2]. Some serious problems exist in these kinds of algorithms that cannot suit different situations. Things turn on when Deep Learning is developed. The deep learning algorithms can consider many variables. Since a limited amount of data can show the varieties and results, traditional deep learning algorithms of supervised and unsupervised learning are difficult to implement. Under this situation, we can combine the traditional deep learning neural network with reinforcement learning methods. Deep learning has a good perceptual ability, while reinforcement learning can make good decisions. In this article, we will propose a deep reinforcement learning method to realize computing resource scheduling in distributed systems considering many factors like response time, task processing time, network delay, and so on.

In recent years, deep reinforcement learning has greatly affected the Artificial Intelligence area. Some applications of deep reinforcement learning like Go, Atari, and StarCraft have given people a deep impression. Deep reinforcement learning has been applied in many areas of our daily life. In most cases, reinforcement learning can give the best decision. The idea of reinforcement learning comes from behavioral psychology. In 1911, Thorndike proposed the Law of Effect [3]. Behavior that makes an animal feel good in a given situation is associated with that situation more strongly, and when the situation is reproduced, the behavior is more likely to be reproduced. Like this, reinforcement learning is a learning method that depends on trial and error. It realizes motivation with a rewarding mechanism. As shown in Fig. 1, when the agent executes specific actions, the environment will turn into a new state. Observing the new environment will give training signals (positive or negative) based on the rewarding mechanism. Then the agent will execute new action according to the environment feedback. Through reinforcement learning, the agent will know how to get the most rewarding. However, if we use reinforcement learning to train from the start without any base,

Yitao Lei, Xingxiang Zhai, and Burra Venkata Durga Kumar are with School of Computing & Data Science, Xiamen University Malaysia (e-mail: SWE1909475@xmu.edu.my, DMT1909203@xmu.edu.my, venkata.burra@xmu.edu.my).

the training process will be inefficient and very slow. This is because of the incremental parameter adjustment and weak inductive bias, so we use a traditional neural network to predict the data flow for each point in the distributed system network. The reinforcement learning rewarding function will be based on the prediction of peak data flow. The DRN (deep reinforcement network) will make distribution for the network resource to get full use of every point.

TABLE I
CURRENT REPRESENTATIVE ALGORITHMS FOR DISTRIBUTED SYSTEM COMPUTING RESOURCE SCHEDULING [2]

Name	Method	Factor	Cost	Time	Dependent	Independent	Resource Utilization	Speed	Dynamic
QoS guided Min-min	Batch Mode	Bandwidth of Tasks	X	X	X	√	X	X	X
Min-Mean heuristics	Batch Mode	----	X	X	X	X	X	X	X
Qos guided weighted Mean Time-min	Batch Mode	QoS and Network BandWidth	X	X	X	√	X	X	X
QoS based predictive Max-min, Min-min switcher	Batch Mode	Heuristics	X	X	X	√	X	√	X
Load Balanced Min-min	Batch Mode	List of heavy load resources	X	X	X	√	√	X	X
Optimized Resource Scheduling	Cooperative	Allocation Request	X	X	X	X	√	√	√
Improved Cost Based	Batch Mode	Meta Tasks	√	X	X	X	X	X	X
A Compromised Time Cost	Batch Mode	Count of Workflow instances	√	√	X	X	X	X	X
HEFT Workflow	Dependent	Highest Rank	X	X	√	X	X	X	√

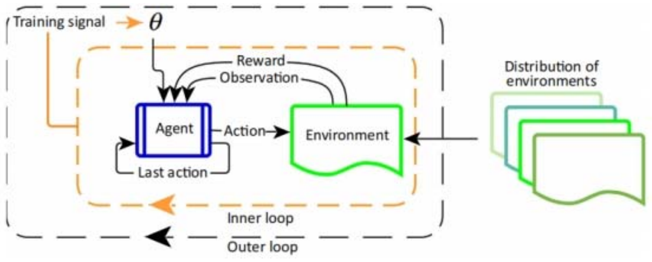

Fig. 1 Learning process of reinforcement learning [3]

Problems

Resource scheduling is an essential factor in improving distributed system performance. Traditional computing resource scheduling algorithms are first come, first serve, round-robin, minimum completion time, minimum execution time, min-min, max-min, and RASA (Real-Time Automated Self-Assessment). However, these classic resource scheduling algorithms cannot realize improvement in distributed system computing. They can just be considered as a resource scheduling rule. With the development of distributed systems, computer scientists have found more and more important factors that influence the performance of the distributed system. They introduce advanced algorithms like ant colony optimization, genetic algorithm, and other heuristic algorithms. Some practical computing resource scheduling algorithms are proposed: Qos-based predictive Max-min, Min-min switcher [4], Load Balanced Min-min [5], Optimized Resource Scheduling [6], Improved Cost Based [7], a Compromised Time Cost [8], and so on. However, ant colony optimization has serious inertia. The peak data traffic in the distributed system is relatively high. This makes the ant colony optimization cannot get the best results after the peak data traffic. Genetic algorithms cannot process high dimension data. The factors that influence the performance of distributed computing are complex and difficult. If the genetic algorithm considers many influential factors, the computation will cost many resources. Besides, the genetic algorithm also has a similar problem as ant colony optimization—insufficient stability because of the randomness of the optimization algorithm.

To consider so many factors which influence the performance of the distributed system computing, we introduce the neural network, which is a kind of heuristic algorithm. It can process large amounts of factors. The neural network has strong information processability. It can analyze the non-linear relations between the results and variables. However, the data process in the neural network does not consider the results. As a result, the decision-making process may lead to negative optimization outcomes at times. This is why we apply reinforcement learning; the rewarding mechanism gives the reinforcement learning good decision ability, which ensures the optimization of the computing resource scheduling algorithms in the distributed system.

Objectives

The distributed system computing multiple resources to collaborate with each other so that they can carry out large-scale computing tasks. However, unreasonable resource scheduling may cause low efficiency and resource waste. As a result, we should design a resource scheduling algorithm to help distributed system scheduling utilize the resource effectively.

1) To study the influence of different parameters (response time, unit capacity, and so on) on the distributed system computing efficiency.
2) To analyze the advantages and disadvantages of different distributed system resources scheduling algorithms.
3) To provide a neural network-based model that can learn the effect of different distributed system parameters.
4) To provide reinforcement learning to make decisions on the resource scheduling algorithm.
5) To combine the deep learning method and reinforcement learning and develop a deep reinforcement learning method to develop a high-efficient resource scheduling algorithm.

Research Questions

1) What are the issues with the traditional genetic algorithm and ant colony optimization?
2) What are the advantages of deep reinforcement learning resource scheduling over other heuristic algorithms?
3) What are the best rewarding functions that enable the reinforcement learning part always make the best decision?
4) How to process the distributed system computing factors data so that the computing can be effective?

II. Framework & Method

A. Frameworks of Existing DRL (Deep Reinforcement Learning) Algorithms for Resource Scheduling

1. DeepRM_Plus
2. DeepRM_Plus [9] utilizes imitation learning in the reinforcement process to reduce the dimension of the parameter, which helps reduce the training time of the optimal policy. It also uses a convolutional neural network to capture the resource management model.

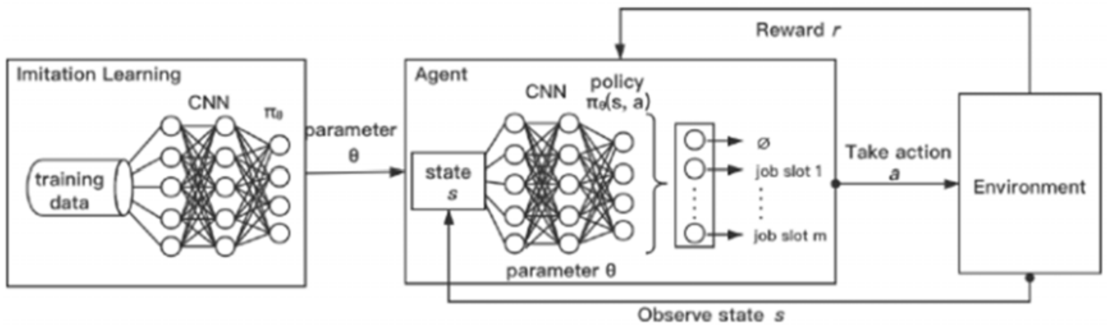

Fig. 2 Overall framework of DeepRM_Plus [9]

3. DERP

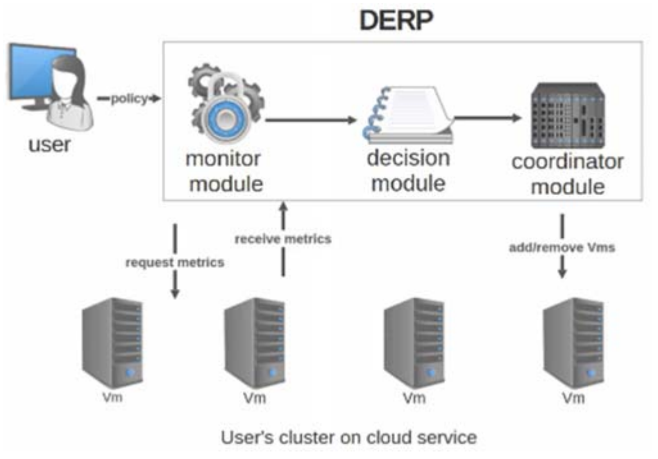

Fig. 3 Framework of DERP [10]

DERP [10] takes some policy parameters from users to define the reward function. Users can select the most crucial parameters for him, and DERP will respond to change its module. If an increase or decrease in the cluster's size is decided, then DERP uses its coordinator module to add or subtract VMs accordingly.

4. DPM

In the DPM [11] method, the work is divided into job scheduling and local power management. Each server queues assigned jobs and allocates resources for them according to the first-come-first-serve policy. If a server does not have enough resources, it waits until completed jobs release sufficient resources. On the other hand, the local tier turns each server on/off in a distributed manner to manage the power.

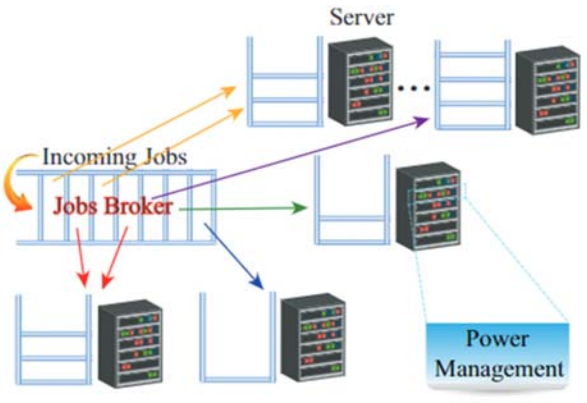

Fig. 4 Framework of DPM [11]

B. Our Framework

To solve the issue that the current DRL method cannot capture the relationship of different distributed system computing parameters. Our framework applies Recurrent Neural networks as the deep learning part. A recurrent neural network is recursive in the direction of data evolution, and all nodes are chained together. For the reinforcement part, we use Q-learning. Q-learning is an off-policy reinforcement learning algorithm. It re-learns the old data to realize sample-efficient, which is suitable for our situation that lacks the data.

III. Findings & Discussion

A. Findings in the Articles

To better tackle the challenging multi-resource scheduling problem, the DRL algorithm is implemented from different aspects.

- DeepRM_Plus: Employing CNN and imitation learning to reduce the complexity of parameters, DeepRM_Plus [9] enables an agent to learn an optimal policy more efficiently.

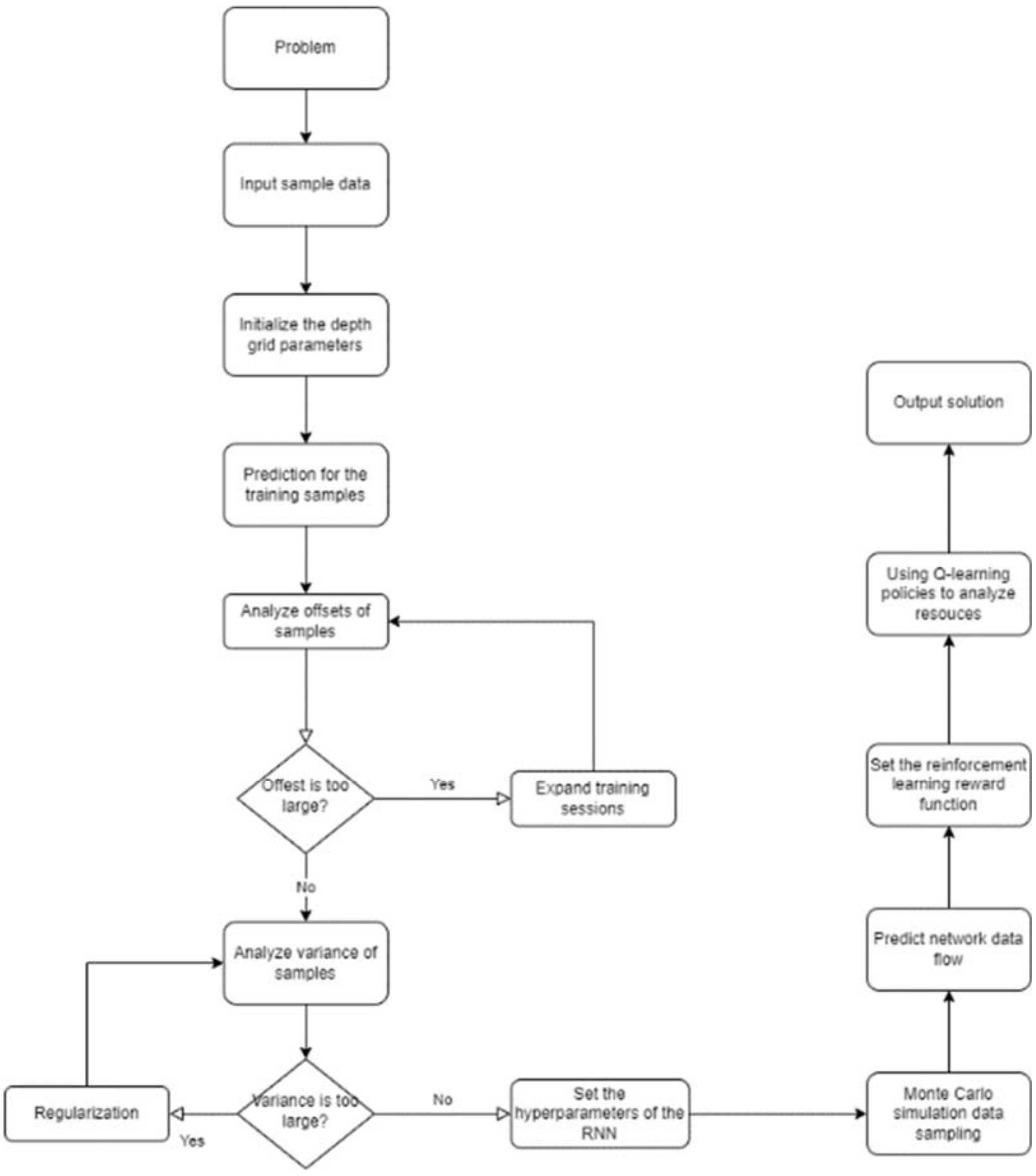
Problem
Input sample data
Initialize the depth grid parameters
Prediction for the training samples
Analyze offsets of samples
Offest is too large?
Yes
Expand training sessions
No
Analyze variance of samples
Variance is too large?
Yes
Regularization
No
Set the hyperparameters of the RNN
Monte Carlo simulation data sampling
Predict network data flow
Set the reinforcement learning reward function
Using Q-learning policies to analyze resouces
Output solution

Fig. 5 Framework of proposed method

- DERP: DERP combines cutting-edge algorithmic techniques in both cloud resource management and deep learning areas. DERP [10] can handle complex and large space-state issues, adapt quickly to its environment, generalize inputs in an adequate manner, and get greater rewards over its lifetime.
- DPM: The proposed framework [11] comprises a global tier and a local tier. The global tier can allocate VM resources to the servers, and the local tier can manage the power. Hence, the best trade-off is achieved in a server cluster.

B. Gaps in the Previous Research

The environment of distributed systems is complex. It needs to deal with large-scale user requests and complex physical resources. In addition, the actual running processes of electronic components and software programs are hard to express using simulation [12]. Crucially, it is challenging to model the mappers of decision and feedback due to the high dimensionality and the continuity of state-space. Generally speaking, the gaps in the research can be concluded as follows:

1) DRL consumes large computing power when training. For a large-scale system, the computational complexity requires to be optimized.
2) Deep learning is like a black box, which means unpredictable results can be generated, as a result, the performance of the worst case is hard to evaluate.
3) When practically allocating resources, scheduling still relies on predicting dynamic tasks without preemption and

prior knowledge.

4) The gradient descent algorithm used in DRL or the Bellman equation used in QL has inherent limitations that lead to local optimization rather than global optimization.

IV. Recommendations

To solve the problem that many factors can influence the performance of the distributed system computing, our framework employs DRL, and we improve the workflow of the data. Our framework is aimed to achieve the resource allocation problem while considering the constraints of task processing time, network bandwidth, and network delay in distributed environments. It first takes a large amount of prior data from network nodes to train the RNN (Recurrent Neural Network). Then, Monte Carlo simulation is used to reduce the data dimension, and Q-learning is used to allocate network resources.

1) Obtain data from network nodes: Firstly, in a distributed environment, traffic data of each branch node are collected as training samples. Then, according to sample size and real-time requirements, we determine the number of hidden layers and nodes of the deep learning network, and initialize network parameters. RNN is used to fit data flow and analyze deviation of fitting results which can help capture high dimension data.
2) Reduce the dimension of data: In a distributed environment, the peak value of data traffic is high, and the traffic is dynamic. Therefore, Monte Carlo simulation is used to predict the maximum flow direction of sampling data, and the prediction results are given. Reducing the dimension of data helps save computing power.
3) Employ the advanced reinforcement learning method: According to Monte Carlo's prediction results, the reward function of reinforcement learning is set. Applying advanced DRL algorithms like DQN (Deep Q-Network) and DDPG (Deep Deterministic Policy Gradient) can effectively improve the learning ability of the DRL method for resource scheduling.
4) Employ the advanced deep learning method: Due to the correlation of distributed resource parameter data, some deep learning models that better capture the relationship of variables would be better suited to perform resource scheduling work. In this paper, we use RNN to capture the contextual information of different variables further. In addition, some advanced deep learning algorithms such as LSTM (Long Short Term Memory) and transformer can also better capture the correlation between variables, thus enabling the model to gain better learning capability.

V. Conclusion

DRL has greatly contributed to natural language processing, drug design, gaming, and other fields in recent years. Research on DRL still needs plenty of attention for its ability to combine the perception ability of deep learning and the decision-making ability of reinforcement learning [13]. Regarding resource allocation, the problem can be divided into many aspects according to the optimization objectives. In distributed environments, there are energy consumption optimization problems, time optimization problems, load balancing optimization problems, and other problems.

In this paper, we first research and explain the architecture of RL (Reinforcement Learning) and DRL, then we discuss and review three deep RL algorithms for resource scheduling in distributed system computing: the DeepRM_Plus, DERP, and DPM. Finally, we proposed our framework, which comprises a flow prediction model, data sampling model, and RL model, and we found that the deep learning algorithm ensures the fast and accurate analysis of data, prevents the occurrence of locally optimal solutions, and provides better decisions for resource scheduling by comparison. However, the proposed approach has some limitations. Although DRL-based scheduling algorithms show advantages in the reviewed literature, they have only been tested in laboratory simulations, where the magnitude of tasks is much smaller than in real distributed environments. Furthermore, the importance of different resources cannot be the same for different organizations. The proposed model may need adjustment to be applied to a specific field.

All in all, DRL contributes a lot to managing the resource in the distributed system, and the essential meaning is that it demonstrates the great potential of multiple technologies and deep RL for solving complex resource scheduling problems [14]. Along this line, we, in the plan, will explore other RL methods in the domain of resource management.

References

[1] M. van Steen and A. Tanenbaum, "A brief introduction to distributed systems", *Computing*, vol. 98, no. 10, pp. 967-1009, 2016. Available: 10.1007/s00607-016-0508-7.

[2] O. M. Elzeki, M. Z. Rashad, M. A. Elsoud "Overview of Scheduling Tasks in Distributed Computing Systems", International Journal of Soft Computing and Engineering, Volume-2, Issue-3, July 2012.

[3] H. Zhang, X. Wu and J. Wang, "Study on the Evaluation Theoretical Structure Building of Deep Learning", China Education Technology, pp. 51-55, 2014.

[4] M. Botvinick, S. Ritter, J. Wang, Z. Kurth-Nelson, C. Blundell and D. Hassabis, "Reinforcement Learning, Fast and Slow", Trends in Cognitive Sciences, vol. 23, no. 5, pp. 408-422, 2019. Available: 10.1016/j.tics.2019.02.006.

[5] Singh. M and Suri. P.K, "QPS A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid", "Information Technology Journal", vol. 7, Issue. 8, 2008, pp. 1176- 1181.

[6] T. Kokilavani, Dr. D.I. George Amalarethinam, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing", "International Journal of Computer Applications", vol. 20, no. 2, April 2012, pp. 43-49.

[7] Hai Zhong, Kun Tao, Xuejie Zhang, "An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems ", 5th Annu. Conf. China Grid Conference, China, 2010.

[8] Y. Yang, K. Liu, J. Chen, X. Liu, D. Yuan and H. Jin, "An Algorithm in SwinDeW-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows", 4th IEEE International Conference on e-Science, 374-375, Indianapolis, USA, December 2008.

[9] W. Guo, W. Tian, Y. Ye, L. Xu, and K. Wu, "Cloud resource scheduling with deep reinforcement learning and imitation learning," IEEE Internet Things J., vol. 8, no. 5, pp. 3576–3586, 202

[10] C. Bitsakos, I. Konstantinou, and N. Koziris, "DERP: A deep reinforcement learning cloud system for elastic resource provisioning," in 2018 IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2018, Nicosia, Cyprus, December 10-13, 2018. IEEE Computer Society, 2018, pp. 21–29

[11] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," Future Gener. Comput. Syst., vol. 102, pp. 847–861, 2020

[12] Liu, N., Li, Z., Xu, J., Xu, Z., Lin, S., Qiu, Q., ... & Wang, Y. (2017, June). A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. In 2017 IEEE 37th international conference on distributed computing systems (ICDCS) (pp. 372-382). IEEE.

[13] Zhou, G., Tian, W., & Buyya, R. (2021). Deep Reinforcement Learning-based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions. arXiv preprint arXiv:2105.04086.

[14] W. Xu, L. Chen, and H. Yang, "A comprehensive discussion on deep reinforcement learning," 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), 2021.