Hindawi
Mobile Information Systems
Volume 2022, Article ID 8716132, 11 pages
https://doi.org/10.1155/2022/8716132

Research Article

Energy Efficiency Strategy for Big Data in Cloud Environment Using Deep Reinforcement Learning

Neeraj Kumar Pandey,[1] Manoj Diwakar,[2] Achyut Shankar,[3] Prabhishek Singh,[4] Mohammad R. Khosravi ,[5] and Vivek Kumar[6]

[1]*School of Computing, DIT University, Dehradun, Uttarakhand, India*
[2]*Graphic Era Deemed to be University, Dehradun, India*
[3]*Amity School of Engineering & Technology, Amity University Uttar Pradesh, Noida, India*
[4]*School of Computer Science Engineering and Technology, Bennett University, Noida, India*
[5]*Department of Computer Engineering, Persian Gulf University, Bushehr, Iran*
[6]*CSIT, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India*

Correspondence should be addressed to Mohammad R. Khosravi; mohammadr.khosravi@iran.ir

Received 29 December 2021; Revised 24 February 2022; Accepted 22 June 2022; Published 11 August 2022

Academic Editor: Robin Singh Bhadoria

Big data entails massive cloud resources for data processing and analysis, which consumes more energy to run. The resources and tasks are increasing exponentially in the cloud environment for the processing of big data, which results in an increment in power consumption to run the cloud data center. So, there is always a scope for optimizing the energy utilization in cloud data centers. This paper presents a visionary architecture in a cloud environment for big data with a proposed energy-efficient strategy based on LSTM-DQN (long-short-term memory-deep Q network) using reinforcement learning (RL). The traditional techniques are not so efficient when the tasks are allocated dynamically, and the generic RL strategies are not able to store the data iterated in the last cycles of processing, so the LSTM is considered for this purpose. In the proposed model, integration of DPSO and DQN is used for better estimation and rectification of the curse of dimensionality. The proposed strategy is compared with different variants of PSO (particle swarm optimization) such as DPSO and QoS-PSO. The improvement in results through proposed model is recoded over the algorithm such as load aware (8.01%), DQN (13.36%), EA-DQN (34.16%), L-No-DEAF (15.62%), DPSO (62.68%), QoS-PSO (72.69%), FFO-EVSM (75.42%), and MIMT (76.39%) on the parameter of energy efficiency, tasks completion time, and energy consumption over the timeline. So, the proposed model is encouraging in the energy-efficient cloud environment for big data with the challenges that the technological world is facing and the emergence of deep learning as one propitious field.

1. Introduction

Big data is the most advanced data analysis field used to process massive data in a high-speed dynamic environment. In this scenario, the most advanced real-time resources are required to cope-up with the industrial requirement. Cloud computing has proven the most reliable computing paradigm of sustainable computing in big data. Big data requires massive storage, high bandwidth for real-time data streams, high-performance data analysis applications, and high-end visualization of data, so these needs cannot be fulfilled with ordinary computing infrastructure available on-premises.

Cloud computing technology is known for aggregating the physical resources in data centers and provides a single system view to each user in a fully virtualized environment. The vicinity of the cloud has increased exponentially in the last five years due to ad-hoc mobile [1] cloud services where a user is accessing cloud utilities through mobile devices. This mechanism has increased the challenges of the cloud (like resource allocation and scheduling, load balancing, and power management) and it requires new dynamic strategies to handle traditional optimization problems. Assignment of every single task consumes some resource over the cloud and that too consumes some power to run in the cloud server. So power management is always a hot spot for the researchers of cloud computing.

In last few years, swam intelligence has gained popularity in solving complex problem in diversified research areas and has given very efficient solutions for dynamic problems. There are many strategies, which come in the meta-heuristic category, which helps as individual or in integration with other nature-inspired approaches to solving complex problems of scheduling, load balancing, and resource migration. The integrated solutions are more focused as it may address multiple parameters of a single problem.

1.1. LSTM. LSTM is a special strategy used in combination with RNN to deal with vanishing gradient issues. The RNN can improve the results in every recurrence, but it cannot store the previous results required for future predictions. RNN works on the large sequence and it cannot carry the data from starting of the sequence to the later stages so LSTM is used to resolve this issue, which gives efficient results in RL with RNN. It is used in different optimization problems in cloud environment, specifically in energy efficiency [2], resource allocation [3, 4], VM migration [5], and load balancing [6]. The CIES (Cloud-based intelligent evaluation service) framework has been used for video homework using LSTN-CNN [7]. It is the tool used for the assessment of video assignments and classifies the data using different features captured from the videos. This is based on the classification; the qualified/unqualified category has been allotted in a cloud environment using LSTM.

1.2. DPSO. PSO is the most commonly used nature-inspired resource allocation strategy in the cloud environment for resource allocation. There are various formats and modifications of PSO used by different researchers with integration and with different algorithms. Integer-PSO, MPSO (modified-PSO), GA-PSO (genetic algorithm-PSO), DPSO (discrete PSO), EE-PSO (energy efficient PSO), RND-PSO (rounding–off PSO), SPV-PSO (smallest position value-PSO), EEDPSO (energy efficient dynamic-PSO), BPSO (binary-PSO), and MOPSO (multiobjective-PSO) are the most common PSO used in the management of resources of the cloud data center. This paper focuses on the DPSO for improving energy efficiency in resource allocation and scheduling. The major strategies used are PSO and its different combinations, but it does not look sufficient in the dynamic environment of cloud, so the RL is applied along with scheduling strategies so that more optimization can be achieved.

DPSO is used with the discrete and binary form of PSO in cloud environment for workflow scheduling. In DPSO every particle parameter is used to calculate the fitness of the particles. The major steps involved in the particle are as follows [8]:

(1) Initialize the data center and dimension of all particles where the best previous position of each and all particles are Pbest and Gbest, respectively.

(2) The objective fitness of all particle are calculated between the node points considering the distance and cost factor.

(3) Updating the position of each particle is done depending on the number of generations.

(4) Updating position of all particles and new values are assigned to Gbest.

(5) Position and velocity of every particle are calculated and updated values that are assigned to corresponding variables.

(6) The velocity and position values of particles are measured, which exceed the corresponding range.

(7) If the termination criteria are not met, the process will be repeated from step 2. It depends on the number of generations is taken for consideration or if not achieved.

1.3. DQN. In deep Q network (DQN), RL plays a big role in capturing the features of dynamic resource allocation strategies in cloud environments. RL is used to integrate the state with the corresponding action so that immediate decisions can be triggered in a dynamic environment. This mapping function is framed to evaluate action value based on the reward expected. So the primary focus is on the process of resource scheduling with different parameters like throughput, makespan, energy efficiency, and VM migration. This process creates an abstracted layer between the CSP and the client. The primary interface of cloud does not disclose the strategy used for VM management and resource orchestration. In the proposed methodology, RL is combined with LSTM along with DQN. DQN is used to store transition data generated in training so that it can make the method more efficient due to the reusability of data. So, DQN makes the sample data selected in more wide way, which provides speed to the convergence and stability. In the paper [9], the job scheduling model is used using double DQN so that the overestimation can be reduced produced by DQN.

The above-mentioned strategies are used by different researchers in various optimization techniques, which motivates them to work in this area, where there is a lot of scope of optimizing the energy efficiency in the cloud environment for big data, specifically the paper [10–12].

(i) The amalgamation of different static and nature inspired technique is used for reducing energy consumption in cloud with conservative energy reduction models.

(ii) The RL based DQN and LSTM with DPSO is used in this paper for addressing the energy efficiency issue in cloud where LSTM helps to process the time stamp data with larger size and gap. It helps DPSO in handling such issues and making it more efficient with DQN.

The paper is organized in the following way as Section 2 describes the architecture of big data services in cloud environment. In Section 3, related work of different researchers is compared in table by considering technique used, parameter covered, goal, and simulating environment. In Section 4, proposed model for resource scheduling of cloud data center has been discussed, which includes LSTM, RL-based model, DQN structure, and DQN-DPSO algorithm. In Section 5, implementation and results have been discussed along with the comparison with different strategies. The Section 6 discusses the conclusion and future scope.

2. Architecture of Big Data Services in Cloud

The integration of two major computing paradigm (cloud and big data) can be a game-changer for all big players in industrial computing. The architecture of big data services in the cloud is shown in Figure 1. Big data provides a user ability platform for the distributed query processing for large datasets, while in cloud, an engine for the data processing in a distributed platform like Hadoop [13]. The architecture has three major parts i.e., data sources, cloud integration with big data platforms like Hadoop, data visualization, and decision-making. Data sources are the combination of data generation sources through cloud like real-time data streaming sources, web sources, etc. These sources work like raw material for the big data cloud platform. The big data platform started from the HDFS, which is a distributed file system and storage technology for Hadoop and used for storing unstructured data received from random data sources. Then, there is creation of databases for the unstructured data, which is an input for the distributed programming model along with algorithms used for the processing of large datasets. These data are then available for the query engine for filtering, analysis, and report generation. Finally, data analysis is used for data visualization for better understanding and decision-making. Different tools, API, and libraries are used to visualize extensive data generated on the big data platform over the cloud. Cloud computing provides not only the platform and resources for massive big data processing but also a dedicated service model for big data services and applications. There are specific big data services provided by all major cloud computing players like IBM, Google, Microsoft, Cloudera, and Salesforce. The major big data service on cloud platforms includes availability, data imports, NoSQL, machine learning, relational DBMS, MapReduce, and Hadoop.

In the paper [14], the author has state-of-the-art review of the energy efficiency strategy of big data in cloud environment. The contribution of high-performance computing (HPC) for big data analysis has been underlined with the help of supercomputer, cluster, and grid computing. In the paper [15], an energy efficiency technique in the cloud data center for big data has been proposed. Hadoop is the most famous platform for big data in cloud environment. It is highly available and known for services over a highly distributed environment for the high-speed data stream.

2.1. Challenge in Handling Big Data in Cloud Environment. Handling such a big volume of data in a real-time environment is itself a big challenge, and it gets multifold when

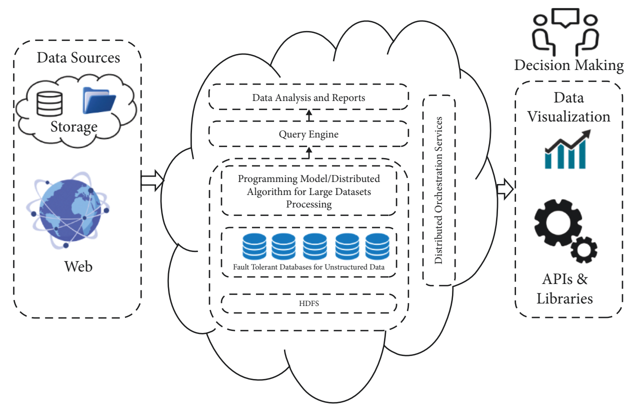

Figure 1: Architecture of big data services on cloud.

handled in a cloud environment. The cloud challenges are also added with big data in this scenario. So the major challenges increase, like scalability, which is generally handled by cloud in every situation, but in this case, the actual test of this basic property of cloud has started. The volume of data in big data is too high that CSP can easily reach on the threshold, so big data services provided by cloud have to address this issue in particular. Different researchers [16–19] have projected many problems and challenges for big data in cloud environment, which respect to the energy efficiency using traditional and nature-inspired strategies. The major concerns include availability, security, storage, heterogeneity, privacy, data quality, energy efficiency, and data staging and scaling.

Storing and processing high volume data streams require high availability, robustness, and scalability in the system. So the cloud computing plays a significant role to provide all aforementioned services through virtualization for big data. Cloud is capable to provide all network based services in all its generic service models to big data in highly scalable environment. So, due to high demand of big data services in cloud environment, many cloud service providers are providing big data as a dedicated service in their service models. The major big data issues handled by cloud are security, storage, data handling policy, heterogeneity, network services, and disaster recovery policy.

3. Related Work

Energy efficiency is a rich area of exploration and full of the possibility of optimization in cloud environment for big data services. Cloud datacenter is always in need of huge power because of high consumption of resources, scalability, and availability in a dynamic business environment. Every other computing paradigm is an option for researchers in integration with cloud computing for optimization of energy consumption in cloud datacenter. In the paper [20], an efficient energy-aware strategy for mobile edge computing (MEC) is used for the prediction of offloads. LSTM is used for the task prediction and connecting the MEC server. A three-layer architecture is proposed for the data transmission, energy efficiency of resource allocation, and communication. In the paper [21], an integration model of CNN-

LSTM is proposed for resource utilization. The input data of energy efficiency are analyzed by regression and residual data are passed with CNN for analysis. The feature extraction of each VM is achieved after passing on the LSTM model.

This paper claims to achieve the accuracy from 3.8% to 10.9% and the error in the implementation is reduced by 7% to 8.5%. In Table 1, various energy-efficient strategies have been discussed using different simulating environment. They are satisfying different goals and covering many parameters. If the major contributions are analyzed, it can be concluded from Table 1 that most strategies are heuristic, optimization-based, and multiobjective. The focus of covering parameters includes SLA violation, energy efficiency, makespan, and task scheduling. Clousim, Python, and different public cloud environment are used for simulation and performance analysis for big data in cloud. In the paper [29], dynamic PSO is used for resource allocation with higher efficiency. The energy-efficient DPSO (EEDPSO) is used to optimize the energy utilization used in the resource allocation in dynamic environment for big data applications. The strategy is implemented through Google workload trace to minimize resource wastage and achieve better resource utilization. In the paper [30], the two-stage multitype PSO (TMPSO) is used for container consolidation in an energy-aware environment. The major focus of the paper is on energy-efficient, VM selection, and placement. The strategy was implemented on WS-dreams for validating a set of applications with different resources requirement. In the paper [31], DPSO and its variants (MOPSO, SBDMPSO, and APDPSO) are used for the load balancing, and time taken in communication is based on number of task executed. The CloudSim and MATLab were used for this. In the paper [32], a swarm optimization-based workload optimization (SOWO) strategy is proposed to improve the efficiency and performance of resources in cloud data centers. In the implementation part, the performance of the strategy is compared with the OpenStack scheduler and 50% less consumption of resources is recorded. In terms of handling big data, including cloud security, there are other researchers that show its major impact on other applications related to medical, satellite, and optical imaging field, IoT based cloud security, decision support system, etc.

The above methods and strategies have used the heuristic methods for energy efficiency. PSO work in iterative manner and it can process time series data, but if the time intervals are long so it may not store the previous data. This motivates us to propose LSTM-DQN model for energy efficiency, including reinforcement learning (DQN) with LSTM. In addition to this, it addresses the more energy consumption, no memory to store previous values, and not performing well for EE limitations mentioned in the Table 1. This models plays vital role to increase the efficiency using DPSO in cloud environment.

4. DQN-DPSO Model

The PSO is most widely used nature inspired strategy for different optimization technique. It follows very different way of sharing information, unlike ACO, GA, PIO, EHO, RCO, etc. These strategies do not have crossover and mutation operator. PSO and its variation algorithms update the operator values with the help of memory elements and internal velocity, which makes it more advantageous. The inbuilt guidance strategy of PSO helps to find more accurate and useful solution of its nearby solution and update itself accordingly in every iteration.

So, in this section, energy-efficient solutions have been provided to minimize the energy consumption by cloud data centers in resource scheduling. The proposed model will cover the LSTM (long short-term memory) based model for prediction and RL (reinforcement learning) based model. The major reason for using RL is because energy efficiency is a sequential decision problem due to time varying energy states. It entails precise model with temporal characteristic between sequential decisions where supervised/unsupervised learning strategies do not give predictable results because of dynamic work state and high computational cost. The supervised/unsupervised learning methodology is used in the problems where class type and underlying patterns are predictable, while EE does not come under this category, so the RL is used, in which learning agent works as a reward and action system. This plays a major role in the proposed model to optimize the existing methods mentioned in the result section (Algorithms 1 and 2).

4.1. LSTM-Based Model for Load Prediction. Resource scheduling in cloud data center requires calculating load of the node in the upcoming time slot. The summation of the load of every node can help to calculate the load of the data center. LSTM is a combination of LSTM units and RNN (recurring neural network). In RNN, hidden layers of the network is connected to the hidden layers of RNN. The most important properties lie in RNN that it can learn time series data efficiently. It has the ability to process the prior data during learning in the existing hidden layer. So, it can easily handle the dynamic nature of the time sequence. This property helps RNN to predict future directions using the past data values. If the time intervals between data are large, RNN cannot store the past data in an efficient way so to overcome this issue, LSTM is combined with RNN.

The proposed model is used to predict the load of the resource node for which a fixed time slot is along with using RNN. LSTM is used for load prediction when abnormal characteristics of data are found. At the time t_1 LSTM prediction model is given as

$$\begin{aligned} \mathrm{s_t} &= \Delta\left(\mathrm{X_h\, i_t + Y_h s_{t-1} + d_h}\right), \\ \mathrm{o_t} &= \Delta_{\mathrm{y}}\left(\mathrm{X_y s_t + d_y}\right). \end{aligned} \tag{1}$$

In the above equation, i_t is input at time t, o_t is an output unit at time t, and the status of RNN is determined by s_t (current status), while end time status is represented by h_{t-1} at input i_t. X_h is a weight from input to hidden layer, Y_h is weight of self-circulation, d_h is used for representing deviation, and Δ_h and Δ_y are the activation functions. The time sequence represents data of RNN and network output layer data shows the prediction of time sequence in the next time interval.

Table 1: Energy efficiency-related work in cloud environment.

Ref no.	Technique used	Parameter covered	Goal	Simulating environment	Limitations
[22]	ISCOLA-LSTM	Energy consumption	Energy efficiency	Python 3.6	More energy consumption in iterative manner
[1]	DRL and DQN	Energy consumption, utility, delay, and rewards	Offloading scheme	Cloudsim 3.0.3	No memory to store previous values
[23]	Cuckoo optimization	SLAV, SLATAH, and PDM	Load balancing and energy management	Cloudsim 3.0.3	Cannot handle discrete/ multiobjective problems
[24]	Multiobjective optimization and VM migration	Power consumption, SLAV, watts per core, NPA, and DVFS	Energy efficiency and resource allocation	Cloudsim 3.0.3	Less parameters are considered
[4]	Energy efficiency and temporal load-aware resource allocation	Power consumption models (linear, nonlinear, multistate, DVFS) and VM migration	Energy efficiency	OpenStack	Not performing well for EE in high load
[25]	LSTM, RNN, WORN-DEAR, SVM, KNN, and L-NO-DEAF	Accuracy, residual energy, energy consumption, packet delivery ratio, and throughput	Energy efficiency and scheduling in body-fog-cloud	Contiki-OS-cooja simulator and IBM watson/thingSpeak	Outperformed in real-time requirement
[26]	KMGA strategy, meta-heuristic, adaptive link rate, virtual network embedding, sleep mode, and green routing	Energy-QoS, make-span, energy consumption, VM migration, and overall SLA	Energy efficiency and resource management	CloudSim 3.0.3	More VM migration with more make span
[27]	EA-VM consolidation, server under load detection, VM selection/placement, maximum fit	QoS, energy consumption, live migration, SLA violation, and server load	VM consolidation and energy consumption	CloudSim 3.0.3	Load balancing with high VM migration
[28]	Deadline partition strategy, energy and cost model, and task and resource selection	DVFS, execution time, computation cost, SLA violation, execution time, and energy consumption	Energy-aware scheduling, and workflow scheduling	CloudSim 3.0.3	Performance decreases/ more energy consumption in more VM migration

4.2. RL-Based Model for Decision. Discrete particle swarm optimization (DPSO) generic algorithm is used for optimizing resource scheduling. There are P resource nodes with R tasks to be assigned. The distribution matrix and velocity matrix is given below to minimize the energy consumption while allocating the resource to the task, so the green computing can be achieved in the cloud data center.

$$\begin{aligned} \mathrm{A_i} &= \left(\mathrm{a_{11\ldots}a_{1p}}\right)\left(\mathrm{a_{1p,\ldots\ldots\ldots\ldots}a_{rq}}\right), \\ \mathrm{B_i} &= \left(\mathrm{b_{11,\ldots\ldots\ldots}b_{1\mathit{q}}}\right)\left(\mathrm{b_{1p,\ldots\ldots\ldots\ldots}b_{rq}}\right), \end{aligned} \tag{2}$$

$$P_{\text{cost}} = (1-\alpha)\sum_{i=1}^{p}\cdot\sum_{j=1}^{q}\text{pow}_{ij}a_{ij}, \tag{3}$$

$$b_{ij}^{k+1} = \omega b_{ij}^{k} + c_1\left(p_{ij}^{k} - a_{ij}^{k}\right) + c_2\left(p_{ij}^{k} - a_{ij}^{k}\right). \tag{4}$$

In equation (2), A_{ij} represents that i^{th} task is assigned to j^{th} server, and $A_{ij} \in \{0, 1\}$ and B_{ij} have the probability lies between 0 and 1 of the A_{ij} position. The particle speed b_{ij} the $\text{sig}(b_{ij})$ is mapped with b_{ij} between 0 and 1. Equation (3) is used to calculate the power consumed, which is in the focus of optimization. pow_{ij} shows that power require to i^{th} task executed on j^{th} server. The following fitness function (equation (4)) is used for the positions of the particles. The $\mathrm{p_{ij}}$ is the optimal matrix of the position of the swarm where k decides the number of iterations to be taken for positioning with α (weight factor). Here, $c1$ and $c2$ are the factors used in learning with ω.

DPSO has great ability for optimization of scheduling the resources in cloud environment, but when the resource and tasks are dynamic in nature, it will be very hard by DPSO to handle because it takes a long time to execute. It reduces the efficiency of the model of scheduling. So, RL is used to learn from every iteration of scheduling and accumulate experience. Q-learning is used to learn from some reward after some action on specific position. In each iteration, the reward value is updated and approached towards maximization, which results the output state. A table is used to record the rewards and actions on a specific state known as state-action (in combination). So, this Qtable is updated in each iteration using the following equation:

$$Q(s,a) = r + \Upsilon(\max(Q(\acute{s},\acute{a}))), \tag{5}$$

where s is known as current state and a is the action, r is reward (noted in Qtable), Υ is loss factor, $\acute{s}$ is next state of action, and $\acute{a}$ is action taken in next position. In new state, LSTM is used for prediction and the task is used as input to

RL-based decision model for nonlinear and distributed relationship. So, Q value from Qtable is used to select initial state of particles. Many other particles correspond to optimized solution in many other positions. This position can be formed by searching other particles. The following figure represents the stable structure of DQN (deep Q network). It is divided into convolutional layer and fully connected later to reach out to Qtable with updated Q value. This Q value is received on every iteration and maintained in Qtable so that the optimized (max.) values can be achieved. The Q value is achieved by equation (6), where Q function is measured in terms of status at time t and the action over it. So, it is $Q(s, a)$, where a can be specific policy at status s i.e., $a = \pi(s)$ while if there any change in the action, it will lead to the next state $\acute{s}$, so the updated action function will be like $\acute{a} = \pi(\acute{s})$ and so on. And finally, Q value can be represented mathematically like

$$Q^{\pi}(s_t, a_t) = \overline{E}_{\pi}\left[\sum_{i=t}^{T} \mu^{i-t} G(t) | s_t a_t\right]. \tag{6}$$

Here, $G(t)$ shows the total rewards achieved on the given time slot t and μ is the discount factor that lies between 0 and 1. $\mu = 0$ shows the agents consider current rewards only, while $\mu \leq 1$ shows that the rewards are considered on later time slots also. As per Bellman's equation, the Q learning is defined as

$$Q(s_t, a_t) = Q(s_t, a_t) + \beta[G(t) + \mu \max Q(s_{t+1}, \acute{a}_{t+1}) - Q(s_t, a_t)], \tag{7}$$

where β is represented as learning rate. The Qtable contains all possible values generated by equation (7) but when the state-action increases, the convergence of Qtable is like a challenge. It generates new problem known as the curse of dimensionality [33]. So here, DQN structure (Figure 2) is used as a solution to this problem, where DQN is used as an alternative for better estimation, in which Conv 1 and Conv 2 are the convolutions, and FC denotes the fully connected.

The power consumption and CPU utilization can be represented using a linear relationship equations (8) and (9), in which P is used to calculate the total power consumed, P_i is the total power used by server in idle position, P_b is the used for the total power used in fully utilized position, and x is the CPU utilization. As per the study, total power consumption by idle servers are 60% more than the total power consumption by the server on threshold utilization.

$$P(x) = P_i + (P_b - P_i)x, \tag{8}$$

$$P(x) = P_i \times P_b + (1 - P_i) \times P_b \times x, \tag{9}$$

$$\bar{\mathrm{E}} = \int_{t}^{\propto} P(x(t))\mathrm{d}t. \tag{10}$$

The total energy consumption can be calculated using equation (10), which defines that the total energy is the integration of power consumption with time (from starting to till the CPU is in awake state).

In order to calculate the complexity of proposed model, n is number of particles, d is the number of dimension, dn is total number of complex multiplication for update the velocity, and Nn is total number of complex additions, which are required in every iteration in DPSO. So, the complexity of the model is represented as $dNn + 4dn + 4 + (N-1)n$, where N is a size of block. The DQN behavior is like ensemble model for the evaluation of complexity, so the DQN-DPSO is complexity, which is totally based on the number of complex addition and multiplication per iteration and the size of blocks into the consideration. With this complexity RL agents deals with environment in discrete way and DPSO is used to support RL, whereas supervised/unsupervised learning cannot work in such scenario. RL have a capability to learn from the feedback in previous iteration, which makes it more efficient for energy efficiency problems.

5. Results and Discussion

In this paper, energy-efficient LSTM-RL based cloud computing model is proposed. The proposed model is implemented and verified using ClodSim. So the RL model of prediction is implemented with DPSO using CloudSim stimulating environment. CloudSim provides features of resource scheduling and management. The large scale cloud cluster can display and test on CloudSim using single machine and processing elements. The simulating environment parameter is given in Table 2 contains 50 VM's. Every VM's has same configuration and MIPS speed (i.e., 2500 MIPS).

The weight factor value (i.e., 0.5) is based on the capacity of local and global search. In the entire process, the value of inertia is decreased gradually with some constant difference (0.05), which results slow down the particle velocity and make it converge in an easier and linear way. There are various impractical studies [34] available for optimal particle size for getting most useful results, which is in the range of 10 to 30. So the particle size is 25 and $\alpha = 0.5$ in the proposed model of DPSO is used.

Based on the configuration stated in Table 2 the simulation is performed with all parameters. A similar dataset is used to train the prediction model using RL. Table 2 has all parameter list used for simulating DPSO in cloud environment using CloudSim. The load aware resource allocation algorithm is used to compare the proposed model of RL-based resource allocation algorithm in cloud computing for big data. The following couple of algorithms are used for the proposed model, in which the DQN-DPSO is used to improve energy utilization during the execution of big data tasks in cloud environment. The DQN improves the utilization in each iteration, which can store the data for the improvement in every cycle. Figures 3(a) and 3(b) represent the total energy consumed using CloudSim for three different resource scheduling algorithm. In Figure 3, the x-axis shows the size of requested tasks and the y axis shows the power consumption for the task in execution. For observation, the size of the task is arranged in increasing order. The proposed algorithms are compared with traditional DQN and load aware algorithms. Through the above figure, the performance of DQN-DPSO strategy can be easy analyzed. The two set of task is implemented for better understanding, comparison, and visualization. In Figure 3(b),

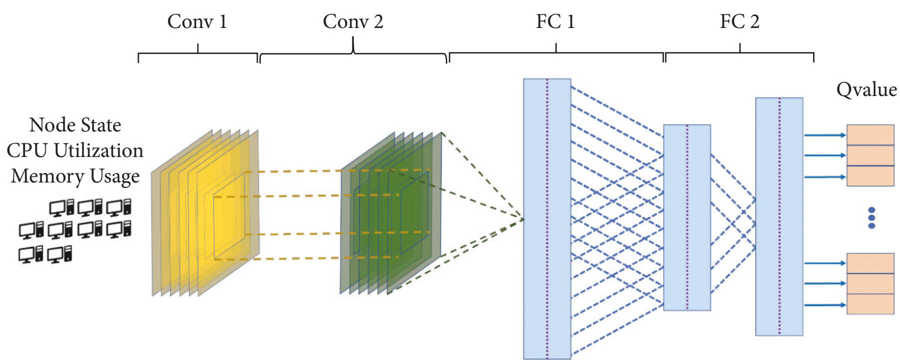

Figure 2: DQN structure with hidden layers.

DPSO algorithm

$t = 0$
for ($i = 1$ to N)
    $G(P_k^t)$
    $\bar{E}(P_k^t)$
    $P_{k(b)} = P_k^t$
    $G_{(b)}$ = best particle found in P_k^t
do{
    for ($i = 1$ to N)
    $V_k^{t+1} = c_1 V_k^t + c_2 (P_b - P_c) + c_3 (G_b - P_c)$
    $P_k^{t-1} = P_k^t + V_k^{t+1}$
    If ($P_k^{t+1} \sim P_k^t$)
    $P_{k(b)} = P_k^{t+1}$
    If ($P_k^{t+1} \sim G_b$)
    $G_b = P_k^{t+1}$
    $t = t + 1$
} while ($t < t_{\max}$)

Algorithm 1: DPSO algorithm for energy efficiency.

DQN algorithm

for ($i = 1$ to N)
    Memory initialization
    for ($i = 1$ to t)
        for ($i = 1 : N$)
            Prob. T and random action a_t
            Set $a_t = Q(s_t, a, \theta)$
            Observe reward and state S_{t+1}
            Add $(s_t^1, a_t^1, r_t^1, s_{t+1}^1) \rightarrow E_i$
            $y_i = \begin{cases} r_j \text{ if } s_t = n, \\ r_j + \min Q(s, a, \theta) \text{ else.} \end{cases}$
            Set $\theta^n = \theta$ in every iteration
        End for
    End for
    $\forall$ Agent replay update
End for
End procedure

Algorithm 2: DQN algorithm for energy efficiency.

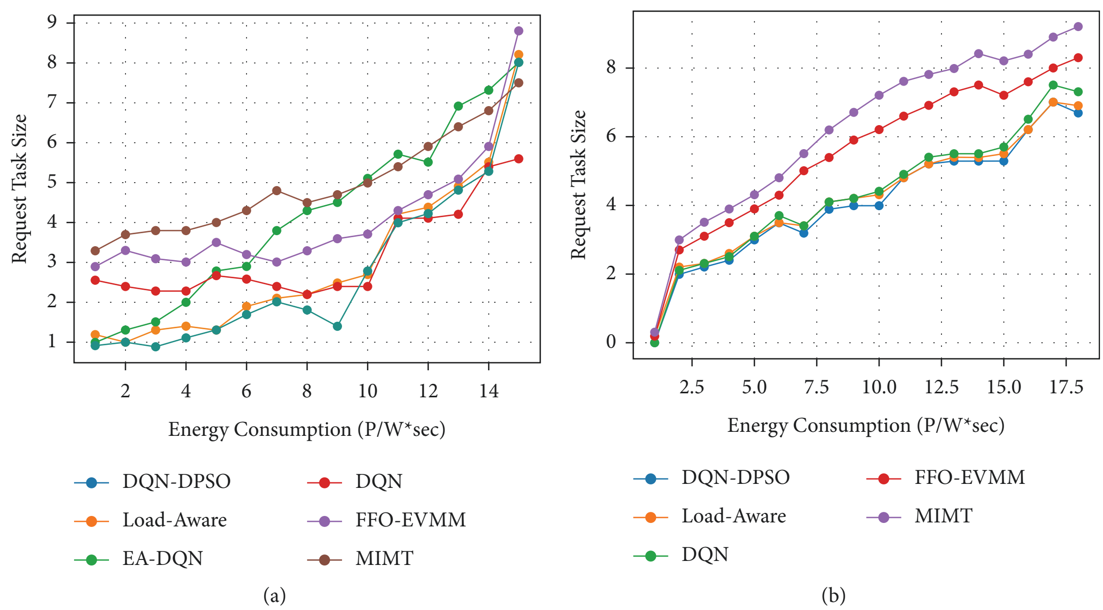

Figure 3: (a) Energy consumption of DQN-DPSO in cloud environment. (b) Energy consumption of DQN-DPSO in cloud environment.

the task set and the number of algorithms have been increased for better view of comparison and understanding. It shows that as the number of tasks increases the DQN-DPSO performs better in comparison to the DQN and load aware algorithm in big data.

In Figure 3(a), energy consumption of DQN-DPSO is shown (round-1) with the comparison with load aware, EA-DQN, and DQN algorithms [35]. As the task size increases, the power consumption also get increases (per task), but DQN-DPSO has minimum average energy consumption per unit task i.e., 0.3746 while load aware, EA-DQN, DQN, FFO-EVMM (firefly optimization-energy aware virtual machine migration), and MIMT (minimization of migration based on Tesa) have 0.4072, 0.5690, 0.4325, 1.489, and 1.7238, respectively. In the next cycle (round-2), number of tasks has been increased to check the performance of the proposed model (as shown in Figure 3(b). These two rounds of evaluation are considered for Case-1. The comparison of the proposed model is also being made with the load aware strategy and DQN. It has optimized performance as the average energy consumption per unit time of DQN-DPSO is 0.4596 and Load- Aware strategy, DQN, FFO-EVMM, and MIMT have 0.4739, 0.4850, 1.489, and 1.7238, respectively.

Big data applications have a great impact on the resources due to high volume and unstructured content, so it is necessary to store the previous data for the cleaning process. So, DQN supports in this way and DPSO has more impact on the energy-aware resource selection. DQN-DPSO makes a unique integration to meet the big data requirements so that the cloud resources may be used in such a way. This method optimizes the energy efficiency of strategies and is able to allocate resources to efficient processes. Every public cloud environment is providing dedicated big data services so this methodology may help them to optimize the energy efficiency so that it can deal with the hindrance of process allocation and save energy in the entire mechanism.

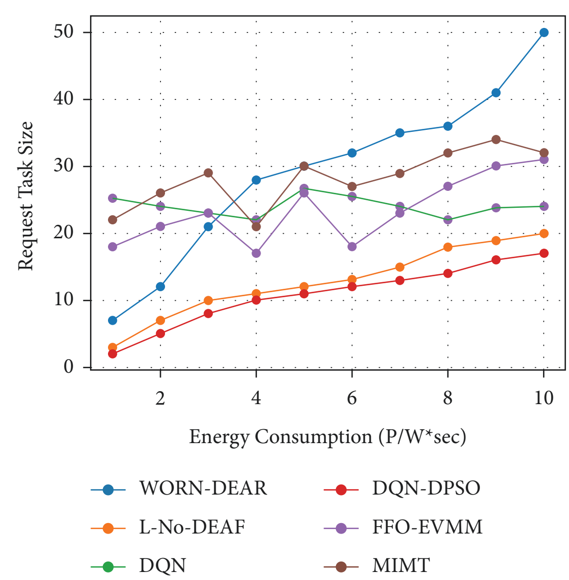

Figure 4: Energy consumption analysis with time.

Another comparative study has been done among the proposed strategy and DQN, Worn-Dear [36], and L-No-Deaf on the parameter of energy efficiency with DQN-DPSO proposed model. In Figure 4, it is shown that DQN-DPSO has less energy consumption as compared to DQN, Worn-Dear, L-No-Deaf, FFO-EVSM, and MIMT. These strategies and considered in literature survey for the comparison and motivation for proposed model (Case-2). Similarly, in Case-3, the completion of execution time as a parameter is also taken for the proposed model, which is compared with different variations of PSO, DBC, and EDF (Figure 5) [37]. The DQN-DPSO performs better than other compared strategies as the time of completion increases when the number of tasks increases because of big data applications specifically like EA-PSO, MIMT, and FFO-EVMM. So the performance of the

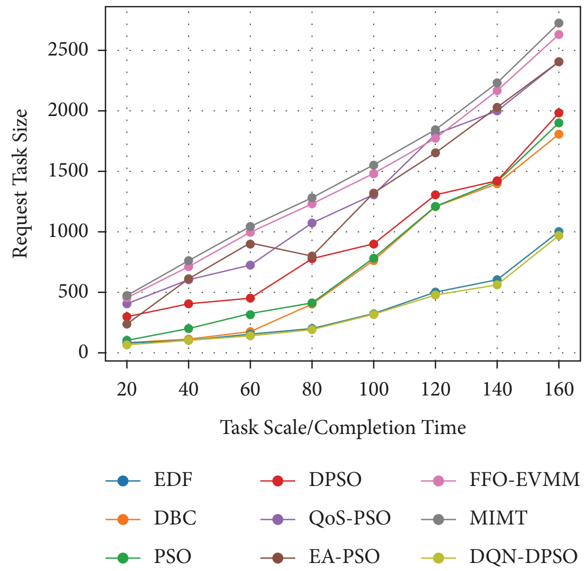

Figure 5: Comparison of proposed method with existing technique.

Table 2: Simulation parameter of DQN-DPSO in cloudSim.

DPSO algorithms	Parameter values
Particle size	25
Weight factor α	$\alpha = 0.5$
Learning factor c_1, c_2	$c_1 = c_2 = 2$
Inertia factor ω	$\omega = 1$
Iteration capacity	$I = 1000$

Table 3: Improvement recorded in EE (in %) proposed model with existing models in two rounds.

Algorithms	Improvement (round 1)	Improvement (round 2)
EA-DQN	34.16	28.27
Load aware	8.01	2.63
DQN	13.36	5.24
FFO-EVMM	49	34.72
MIMT	79.32	51.21

Table 4: Improvement recorded in EE (in %) proposed model with existing models using three cases.

Case-1		Case-2		Case-3	
Algorithms	Improvement	Algorithms	Improvement	Algorithms	Improvement
Load aware	2.63	Worn-DEAR	63	EA-PSO	71.78
DQN	5.24	L-No-DEAF	15.62	QoS-PSO	72.69
FFO-EVMM	41.61	DQN	55	DPSO	62.68
MIMT	65.26	FFO-EVMM	53	PSO	55.53
		MIMT	61	DBC	52.53
				EDF	4.42
				FFO-EVMM	75.42
				MIMT	76.39

proposed strategy is performing better than other integrated variants of PSO for big data cloud applications.

The improvement is shown in Figures 3–5 can be analyzed with the help of following Tables 3 and 4. The improvements are calculated with respect to the DQN-DPSO strategy over other algorithms. The proposed model has value 41.21 and 74 for round-1 and round-2, respectively, calculated in Case-1. In the next case (Case-2), a significant improvement has been recorded as compared to other proposed model i.e., 108 and in Case-3 the completion time is 2810 per second. The detailed improvement against all compared algorithms is shown in Tables 3 and 4.

The proposed model with PSO is extended with FIS for not getting premature convergence and local minimums, so that it may not lead to unexpected level of optimization but in proposed system, PSO is getting more optimal value in efficiency, which is not significantly increased in extended strategy with called FISDPSO. The improvement recorded is not so significant so it is not discussed in the results section but shown in the Tables 3 and 4. While two more strategy FFO-EVSM and MIMT is also compared with proposed model and it shows the significant improvement, i.e., 75.42% and 76.39%.

6. Conclusion and Future Scope

As the green cloud is today's computing requirement, there is a need for an intelligent self-learning strategy used with traditional optimization problems for big data in cloud environment. This paper focuses on the usage of energy while executing the task in cloud perspective for big data. The efficient integration of intelligent strategies (DQN and DPSO) gives the solution where less power is required for resource allocation issues. CloudSim provides such features for simulation so that cluster systems can run in a simulated environment and DQN strategy can be integrated with DPSO. The results can be compared with traditional load aware algorithms for big data. The performance of DQN-DPSO is found more efficient for two different set of task in execution. It shows that the proposed model provides an energy-efficient solution to the resource allocation problem in the green cloud environment. As the future strategy, more reinforcement learning can be explored more for the development of integrated strategies for energy efficiency for big data in the cloud environment. The traditional and streamlined algorithms may not perform well as the challenges in big data is increasing every day in terms of velocity and volume, so there is a need of the integration of multiple optimized algorithms to meet the challenges of present big data applications in cloud data centers. The energy efficiency can be used in the industrial strategy of energy optimization in cloud based industrial control system (especially during scheduling and load balancing of resources). This helps the industrial cloud and private cloud deployments to save energy in VM migration during handling of big data (e.g., AWS, Eucalyptus, and OpenNebula).

Data Availability

The conclusion and comparison data of this article are included within the article. For inquiries regarding raw data and codes, please contact the first author (email: neeraj.pandey@dituniversity.edu.in).

Conflicts of Interest

The authors declared that they have no conflicts of interest.

References

[1] D. L. Van and C. K. Tham, "A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds," in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 760–765, IEEE, Honolulu, HI, USA, April 2018.

[2] R. Yazdani, O. Ruwase, M. Zhang, Y. He, J. M. Arnau, and A. Gonzalez, "LSTM-sharp: An Adaptable, Energy-Efficient Hardware Accelerator for Long Short-Term Memory," 2019, https://arxiv.org/abs/1911.01258.

[3] Y. Liu, L. L. Njilla, J. Wang, and H. Song, "An LSTM enabled dynamic stackelberg game theoretic method for resource allocation in the cloud," in *Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 797–801, IEEE, Honolulu, HI, USA, February 2019.

[4] S. Vakilinia, "Energy efficient temporal load aware resource allocation in cloud computing datacenters," *Journal of Cloud Computing*, vol. 7, no. 1, p. 2, 2018.

[5] H. Ren, Y. Wang, C. Xu, and X. Chen, "SMig-RL," *ACM Transactions on Internet Technology*, vol. 20, no. 4, pp. 1–18, 2020.

[6] R. K. Ramesh, H. Wang, H. Shen, and Z. Fan, "Machine learning for load balancing in cloud datacenters," in *Proceedings of the 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pp. 186–195, IEEE, Melbourne, Australia, May 2021.

[7] R. Song, Z. Xiao, J. Lin, and M. Liu, "CIES: cloud-based Intelligent Evaluation Service for video homework using CNN-LSTM network," *Journal of Cloud Computing*, vol. 9, no. 1, p. 7, 2020.

[8] J. Liu, T. Shi, and P. Li, "Optimal cloud storage problem in the distributed cloud data centers by the discrete PSO algorithm," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 156–163, IEEE, Sendai, Japan, May 2015.

[9] B. Han and J. Yang, "Research on adaptive job shop scheduling problems based on dueling double DQN," *IEEE Journals & Magazine | IEEE Xplore*, IEEE, vol. 8, 2020.

[10] A. Iqbal, M.-L. Tham, and Y. C. Chang, "Double deep Q-network-based energy-efficient resource allocation in cloud radio access network," *IEEE Access*, vol. 9, Article ID 20449, 2021.

[11] M. Siddaraju and Dr. K. Siddaraju, "Energy efficient dynamic particle swarm optimization (EEDPSO) resource allocation in cloud computing," *Webology*, vol. 17, no. 2, pp. 128–149, 2020.

[12] K. Oukfif, L. Bouali, S. Bouzefrane, and F. Boumghar, "Energy-aware DPSO algorithm for workflow scheduling on computational grids," in *Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud*, pp. 651–656, IEEE, Rome, Italy, August 2015.

[13] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of "big data" on cloud computing: review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[14] EC. Inacio and M. Dantas, "A survey into performance and energy efficiency in HPC, cloud and big data environments," *International Journal of Networking and Virtual Organisations*, vol. 14, no. 4, pp. 299–318, 2014.

[15] F. Mehdipour, H. Noori, and B. Javadi, "Chapter two—Energy-efficient big data analytics in datacenters," *Advances in Computers*, vol. 100, pp. 59–101, 2016.

[16] A. Kobusińska, C. Leung, C.-H. Hsu, S. Raghavendra, and V. Chang, *Emerging Trends, Issues and Challenges in Internet of Things, Big Data and Cloud Computing*, Elsevier, Amsterdam, Netherland, 2018.

[17] A. Dhabhai and Y. K. Gupta, "A study of big data in cloud environment with their related challenges," *International Journal of Engineering Sciences*, vol. 25, p. 11, 2016.

[18] S. Subbalakshmi and D. K. Madhavi, "Security challenges of big data storage in cloud environment," *Surveyor*, vol. 13, no. 17, p. 8, 2018.

[19] F. Mehdipour, H. Noori, and B. Javadi, "Energy-efficient big data analytics in datacenters," *Advances in Computers*, vol. 100, pp. 59–101, 2016.

[20] G. Wu, Y. Miao, Y. Zhang, and A. Barnawi, "Energy efficient for UAV-enabled mobile edge computing networks: intelligent task prediction and offloading," *Computer Communications*, vol. 150, pp. 556–562, 2020.

[21] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," *Neural Computing & Applications*, vol. 33, no. 16, Article ID 10055, 2021.

[22] N. Somu, G. R. M R, and K. Ramamritham, "A hybrid model for building energy consumption forecasting using long short term memory networks," *Applied Energy*, vol. 261, Article ID 114131, 2020.

[23] M. Yakhchi, S. M. Ghafari, S. Yakhchi, M. Fazeli, and A. Patooghi, "Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures," in *Proceedings of the 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1–5, IEEE, Istanbul, Turkey, May 2015.

[24] B. Patel and H. Patel, "Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 7, pp. 860–869, 2020.

[25] S. Murali and M. Murali, "RETRACTED ARTICLE: Deep learning based energy efficient novel scheduling algorithms for body-fog-cloud in smart hospital," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7441–7460, 2021.

[26] M. H. Askarizade, M. Maeen, and M. Haghparast, "An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms," *Wireless Personal Communications*, vol. 104, no. 4, pp. 1367–1391, 2019.

[27] S. Y. Zahedi Fard, M. R. Ahmadi, and S. Adabi, "A dynamic VM consolidation technique for QoS and energy consumption in cloud environment," *The Journal of Supercomputing*, vol. 73, no. 10, pp. 4347–4368, 2017.

[28] M. Khorsand and R. Khorsand, "Energy-aware scheduling algorithm for time-constrained workflow tasks in DVFS-enabled cloud environment," *Simulation Modelling Practice and Theory*, vol. 87, pp. 311–326, 2018.

[29] M. Bala Krishna, "Hybrid evolutionary algorithm based task scheduling mechanism for resource allocation in cloud environment," *Revista Gestão Inovação e Tecnologias*, vol. 11, no. 4, pp. 194–209, 2021.

[30] T. Shi, H. Ma, and G. Chen, "Energy-aware container consolidation based on PSO in cloud data centers," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 2018.

[31] Z. Miao, P. Yong, Y. Mei, Y. Quanjun, and X. Xu, "A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment," *Future Generation Computer Systems*, vol. 115, pp. 497–516, 2021.

[32] J. Yan, H. Zhang, H. Xu, and Z. Zhang, "Discrete PSO-based workload optimization in virtual machine placement," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 589–596, 2018.

[33] A. Koppel, G. Warnell, E. Stump, P. Stone, and A. Ribeiro, "Policy evaluation in continuous MDPs with efficient kernelized gradient temporal difference," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1856–1863, 2021.

[34] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, NJ, USA, 2007.

[35] A. Oroojlooyjadid, M. Nazari, L. Snyder, and M. Takáč, "A Deep Q-Network for the Beer Game: A Deep Reinforcement Learning Algorithm to Solve Inventory Optimization Problems," 2020, https://arxiv.org/abs/1708.05924.

[36] P. Prabukumar and M. Prabukumar, "Design and implementation of energy efficient reconfigurable networks (WORN-DEAR) for BAN in IOT environment (BIOT)," *International Journal of Reasoning-Based Intelligent Systems*, vol. 10, no. 3/4, p. 258, 2018.

[37] W. Jing, C. Zhao, Q. Miao, H. Song, and G. Chen, "QoS-DPSO: QoS-aware task scheduling for cloud computing system," *Journal of Network and Systems Management*, vol. 29, no. 1, p. 5, 2020.