

# A Reinforcement Learning Based Framework for Holistic Energy Optimization of Sustainable Cloud Data Centers

Daming Zhao <sup>1</sup>, Jiantao Zhou <sup>2</sup>, Jidong Zhai <sup>3</sup>, *Senior Member, IEEE*, and Keqin Li <sup>4</sup>, *Fellow, IEEE*

**Abstract**—The widespread adoption of cloud data centers has led to a rise in energy consumption, with the associated carbon emissions posing a further threat to the environment. Cloud providers are increasingly moving towards sustainable data centers powered by renewable energy sources (RES). The existing approaches fail to efficiently coordinate IT and cooling resources in such data centers due to the intermittent nature of RES and the complexity of state and action spaces among different devices, resulting in poor holistic energy efficiency. In this paper, a reinforcement learning (RL) based framework is proposed to optimize the holistic energy consumption of sustainable cloud data centers. First, a joint prediction method MTL-LSTM is developed to accurately evaluate both energy consumption and thermal status of each physical machine (PM) under different optimization scenarios to improve the state space information of the RL algorithm. Then, this framework designs a novel energy-aware approach named BayesDDQN, which leverages Bayesian optimization to synchronize the adjustments of VM migration and cooling parameter within the hybrid action space of the Double Deep Q-Network (DDQN) for achieving the holistic energy optimization. Moreover, the pre-cooling technology is integrated to further alleviate hotspot by making full use of RES. Experimental results demonstrate that the proposed RL-based framework achieves an average reduction of 2.83% in holistic energy consumption and 4.74% in brown energy, which also reduces cooling energy consumption by 13.48% with minimal occurrences of hotspots. Furthermore, the proposed MTL-LSTM method reduces the root mean square error (RMSE) of energy consumption and inlet temperature predictions by nearly half compared to LSTM and XGBoost.

**Index Terms**—Cloud data center, energy optimization, reinforcement learning, renewable energy.

Received 8 May 2024; revised 27 October 2024; accepted 4 November 2024. Date of publication 11 November 2024; date of current version 6 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62162046, in part by the Major Project of Inner Mongolia Natural Science Foundation under Grant 2019ZD15, in part by the Research and Application of Key Technology of Big Data for Discipline Inspection and Supervision under Grant 2019GG372, and in part by the Science and Technology Plan Special Project of Hohhot. (*Corresponding author: Jiantao Zhou.*)

Daming Zhao is with the College of Computer Science, Inner Mongolia University, Hohhot 010021, China, and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: cszdm@mail.tsinghua.edu.cn).

Jiantao Zhou is with the College of Computer Science, Inner Mongolia University, Hohhot 010021, China (e-mail: cszhoujiantao@qq.com).

Jidong Zhai is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zhaijidong@tsinghua.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TSC.2024.3495495

## I. INTRODUCTION

THE rapid expansion of cloud data centers has brought attention to the issue of energy consumption. In recent times, the energy consumption of a single server rack has surged from less than 3 kW to 30 kW [1]. The main components of energy consumption have also changed. Besides IT devices, around 40% of the energy consumed by these data centers can be attributed to the Computer Room Air Conditioning (CRAC) systems [2]. This trend will also exacerbate the impact on the environment, and it is estimated that cloud data centers will contribute up to 8% of global carbon emissions by 2030 [3]. Therefore, major cloud providers (such as Google and Amazon) are prioritizing the use of renewable energy sources (RES) like solar and wind energy to build sustainable cloud data centers. Although promising, the energy requirements of these data centers cannot always be satisfied due to the intermittent nature of RES, leading to increased reliance on non-renewable (brown) energy sources. In this context, the energy consumption should be further optimized according to the RES availability.

Recent studies are focusing on predicting energy consumption and thermal status. By proactively understanding the impact of computing resource consolidation and cooling parameters adjustments on the holistic energy efficiency, cloud data center providers can make well-informed decisions. Current approaches primarily adopts mathematics models, computational fluid dynamic (CFD) technology [4] and thermodynamics methods [5], causing substantial computational overhead. Although data-driven solutions is promising [6], they often separately estimate energy consumption and inlet temperature without exploring shared characteristics, resulting in limited generalization capabilities and inaccuracies that contribute to additional energy consumption.

For holistic energy consumption optimization, energy-aware and thermal-aware approaches have indeed achieved significant reductions in both computing and cooling energy consumption. However, these strategies often encounter difficulties in reconciling the inherent conflict between these two systems, resulting in 4% to 30% holistic energy wastage [7]. Considering an example that a large-scale cloud data center consumes about 100 megawatts (MW) of power continuously, and the average U.S. household consumes about 10715 kWh per year [8], the energy consumption from just a 4% reduction could power approximately 3000 households. The primary issue arises from the aggressive consolidation of virtual machines (VMs) onto fewer active physical machines (PMs) to transition others into low-power states, inadvertently creating “hotspots” by localized temperature increases. Consequently, although IT energy

consumption is effectively optimized, the cooling systems have to set lower Computer Room Air Conditioning (CRAC) temperature, resulting in poor holistic energy efficiency.

This problem will become further challenging in RES-powered data centers [9]. Current solutions attempt to postpone batch workloads until RES is available [10], yet uncertainties such as resource requests, thermal status, and RES availability will introduce high-dimensional state and action spaces that need frequent reconfiguration and manual intervention. In contrast to traditional algorithms [11], reinforcement learning (RL) offers a more flexible and efficient solution by making real-time energy-aware scheduling decisions through adaptive interactions with the environment once trained [12]. Information about the cloud system's status is seamlessly integrated into the action value function estimates, enabling an RL agent to continuously enhance its decision-making capabilities. However, most existing RL methods are limited to handling either discrete action space (e.g., VM consolidation for reducing computing energy consumption) or continuous action space (e.g., CRAC adjustment for alleviating hotspots) separately. It is still intractable for the RL-based methods to synchronize control across both systems and hybrid action space to optimize holistic energy consumption.

In this paper, a novel reinforcement learning based framework is proposed, which considers RES availability to coordinate VM migrations and CRAC adjustments to improve the holistic energy efficiency of sustainable cloud data centers.

The main contributions are as follows:

- Propose an energy-efficient framework based on reinforcement learning, where a prediction approach is implemented to improve the accuracy of energy consumption and thermal status estimates within the state space, and a scheduling strategy is designed to avoid IT and cooling resources' asynchronous allocations in the action space, thereby optimizing the energy consumption.
- Propose a new joint prediction method named MTL-LSTM, where the LSTM component is utilized as the parameter-sharing layer within the MTL model, effectively leveraging shared representations across energy-related tasks. This combination enables the creation of a more detailed and informative state space by simultaneously modeling both energy consumption and temperature dynamics, while further improving the accuracy of predictions.
- Propose a novel energy-aware approach named Bayes-DDQN<sup>1</sup> to tackle conflicts of energy consumption optimization. In Bayes-DDQN, the discrete CRAC parameters within the hybrid action space are explored by the Bayesian optimization, which will be provided to DDQN to refine the decision-making policy for dynamic VM migration based on real-time resource requirements and RES availability.
- Propose an innovative hotspots mitigation strategy where the pre-cooling technology is employed to lower the supplied CRAC temperature when sufficient RES is available to satisfy the increased cooling energy demand. This proactive adjustment stabilizes temperatures without falling into coordinating extra VM migrations within the hybrid action space to satisfy thermal constraints, and the improved RES utilization will reduce significant brown energy consumption accordingly.

The rest of this paper is organized as follows. Section II discusses the related works and highlights their limitations. System models are defined and formulated in Section III. Section IV describes the proposed RL-based framework. Section V shows the experimental results and compares them with the state-of-art approaches. Section VI concludes the paper.

## II. RELATED WORK

In recent years, there has been extensive attention towards the holistic energy optimization of cloud data centers, especially highlighting the impact of hotspots caused by excessive VM migrations. This section introduces the research of predicting system status and optimizing energy-related management strategies.

Accurate energy consumption and thermal status evaluation are becoming indispensable in the holistic energy consumption optimization process. Compared with energy consumption which can be induced by popular linear mathematical model [13] or SPECpower benchmark,<sup>2</sup> temperature estimation is a non-trivial problem. Existing solutions are not efficient due to computational complexity and imprecision. The theoretical model makes predictions based on the thermodynamic characteristics of heat and the physical properties of the data center [28], which may struggle to capture the intricate relationships known as heat recirculation matrix (HRM) while lacking the scalability as the HRM has to be changed to adapt to different data centers. Computational fluid dynamics (CFD) [14] provides detailed and accurate airflow patterns, temperature distributions, and heat transfer mechanisms within a data center. However, CFD is highly specialized, which needs substantial computational resources and time, making real-time predictions challenging. Data-driven approaches utilize large-scale historical data, which is suitable for dynamic workload variations and ambient temperature behaviors [15]. Nevertheless, the aforementioned methods never consider the correlation between energy consumption and inlet temperature jointly, parameter changes in one model can affect the other and hence requiring more accuracy predictions to avoid worse energy efficiency or more hotspots.

For saving computing energy consumption, existing approaches target consolidating VMs to deactivate more PMs. Wu et al. [16] proposed a cyclic usage prediction for one-day VM usage based on the ordinary least squares (OLS). The VMs with obviously fluctuation patterns are then consolidated to reduce unnecessary migrations, and hence minimizing associated energy consumption. Yao et al. [17] designed an energy-efficient load balance strategy, which considers PM fragmentation caused by imbalanced resource utilization and predict load fluctuations during VM consolidation with exponential smoothing, hence reducing energy consumption within the acceptable service level agreement (SLA) violations. Such methods might struggle to operate with fluctuating scenarios where resource requests change rapidly, whereas RL-based methods excel in adapting to dynamic conditions by continuously learning and updating VM consolidation policies based on real-time feedback. Shaw et al. [18] presented a RL-based autonomous model capable of optimizing the distribution of VMs to achieve greater energy efficiency while guaranteeing the delivered quality of service, which improves energy efficiency by 25% over the power-aware heuristic algorithm. Wei et al. [19] proposed an asynchronous

<sup>1</sup>Code is open-source at <https://github.com/cszdm/BayesDDQN>

<sup>2</sup>[https://www.spec.org/power\\_ss/2008/results](https://www.spec.org/power_ss/2008/results)

TABLE I  
COMPARISON OF RELATED WORK

Reference	Energy Optimization	Prediction Model		Algorithm	RES-powered	Objective
		Energy	Thermal			
Radi et al. [13]	IT	Linear		Genetic algorithm		Energy, VM migrations
Li et al. [14]	Holistic	SPECpower benchmark	CFD	Greedy-based algorithm		Energy, CPU temperature
Gill et al. [15]	Holistic	Linear	RNN+GRU	Thermal-aware heuristic algorithm		Energy, VM migrations, CPU Temperature
Wu et al. [16]	IT	OLS		Load balance-aware heuristic algorithm		Energy, resource utilization
Yao et al. [17]	IT	SPECpower benchmark		Resource and load-aware heuristic algorithm		Energy, VM migrations
Shaw et al. [18]	IT	SPECpower benchmark		Q-learning and SARSA algorithm		Energy, VM migrations
Wei et al. [19]	IT	Linear		Actor-Critic algorithm		Energy, VM Migrations, Inactive PMs
Li et al. [20]	Holistic	Linear	HRM	GA and SA algorithm		Energy, Inlet temperature, Execution time
Aghasi et al. [5]	Holistic	Linear	HRM	Fuzzy controlled gravitational search algorithm		Energy, CPU temperature, Hotspots
Tuli et al. [21]	Holistic	GGCN	GGCN	Performance to power heuristic algorithm		Energy, VM migrations, CPU temperature
Ran et al. [22]	Holistic	Linear	Temperature sensors	DQN algorithm		Energy, VM migrations, Outlet temperature
Li et al. [23]	Holistic	Linear	HRM	Thermal-aware heuristic algorithm	✓	Energy, RES utilization, CRAC temperature
Renugadevi et al. [24]	IT	Linear		Renewable energy-aware heuristic algorithm	✓	Energy, Carbon emission, Operating cost
Wang et al. [25]	IT	SARIMA		MARL algorithm	✓	Energy, Carbon emission, Monetary cost
Jayanetti et al. [26]	IT	Linear		MARL algorithm	✓	Energy, Makespan
Zhao et al. [27]	IT	Linear		PPO algorithm	✓	Energy, Operating cost
Proposed method	Holistic	MTL+LSTM	MTL+LSTM	DDQN algorithm	✓	Energy, VM migrations, Inlet temperature, Hotspots, RES utilization

advantage actor-critic strategy, the workload distribution and active PMs are considered in the reward function for updating the optimal VM migration mapping. However, these methods may suffer a higher possibility of hotspots due to aggressive VM consolidation, resulting in increased cooling energy consumption.

Towards mitigating hotspots, thermal-aware methods have emerged, which considers the impact of cooling devices on holistic energy consumption. In this context, Li et al. [20] proposed a thermal-friendly workload distribution scheme based on genetic algorithm (GA) and simulated annealing algorithm (SA), which seeks to improve the holistic energy efficiency by minimizing the maximum inlet temperatures of computing nodes. Aghasi et al. [5] designed a self-adaptive mechanism based on fuzzy logic to guide the gravitational search algorithm for IT and cooling energy optimization. Tuli et al. [21] presented a comprehensive resource management approach, employing the gated graph convolution network to capture the thermal dependencies of PMs. This method establishes a performance-to-power ratio to intelligently balance the workload across cloud PMs. Ran et al. [22] devised a framework that parametrizes the action space of job scheduling and airflow rate adjustment based on a Deep Q-Network (DQN) to minimize energy consumption, which saves up to 10% energy compared to heuristic-based joint control optimization algorithm. Nevertheless, these mentioned approaches unavoidably encounter challenges related to carbon emissions and asynchronous issues when concurrently scheduling IT and cooling resources.

To make data centers greener, transitioning from coal-based fuels to renewable energy is essential. Li et al. [23] presented a pre-cooling strategy which utilizes redundant renewable energy to provide lower CRAC temperature to transfer the heat generated by hotspots without introducing extra VM migrations. Renugadevi et al. [24] proposed a renewable-aware algorithm to schedule the workload to the cloud data centers with sufficient renewable energy, which also performs on-demand Dynamic Voltage Frequency Scaling (DVFS) to eliminate hotspots caused by high processor utilization. Meanwhile, RL algorithms are well-suited for optimizing policies over extended periods and managing large state and action spaces, making them ideal for handling the uncertain relationship between energy consumption and RES generation in large-scale deployments. Wang et al. [25] designed a multi-agent reinforcement learning (MARL) based method to postpone urgent requests upon insufficient renewable energy supply so that carbon emission can be reduced. Jayanetti et al. [26] proposed a hierarchical MARL-based framework where the global and local RL agents cooperate to determine the optimal destination data centers

and VM mappings strategy through workflow scheduling. Zhao et al. [27] devised a model-free proximal policy optimization (PPO) algorithm, which automatically shifts workload between public and private clouds according to the RES availability and reduces energy consumption by 34.61% compared to heuristic-based maximum renewable least utilization first approach. Although promising, none of them simultaneously proceed VM consolidation and CRAC parameters adjustment for improving holistic energy efficiency while making full use of RES.

Table I presents a comparative overview between the proposed method and relevant researches. The proposed RL-based framework is holistic energy-effective in sustainable cloud data centers since it can explore VM migration and CRAC adjustment through jointly evaluate energy consumption and thermal to proactively avoid energy consumption optimization conflicts. In such dynamic scenarios, it use DDQN and pre-cooling technology to lower cooling temperature with enlarged RES utilization without introducing extra VM migrations, which is not considered in other literature.

### III. SYSTEM MODEL

#### A. Overview of Holistic Energy Optimization Framework

The holistic energy-aware framework of sustainable cloud data centers powered by a combination of the traditional electricity grid and renewable energy sources is shown in Fig. 1. The system models of this framework follow the MAPE principle where the Monitor module collects user requests and system information. The Analyze module then employs the MTL-LSTM method to jointly predicts energy and thermal status. In this context, the BayesDDQN algorithm in the Plan module will output the optimal IT and cooling resources scheduling strategy. At last, the corresponding VM migrations and CRAC adjustments are carried out in the Execute module.

#### B. The Holistic Energy Consumption Model

In recent researches, the holistic energy consumption of a cloud data center is deemed to be contributed by PMs and CRAC devices [29].

*Definition 1:* Let  $E^{DC}(t)$  be the holistic energy consumption of the cloud data center at time  $t$ , which is contributed by each PM's computing energy  $E_i^{IT}(t)$  and the cooling energy consumed by CRAC  $E_i^{CRAC}(t)$ .  $E^{DC}(t)$  can be defined as:

$$E^{DC}(t) = \sum_{i=1}^N E_i^{IT}(t) + E^{CRAC}(t) \quad (1)$$

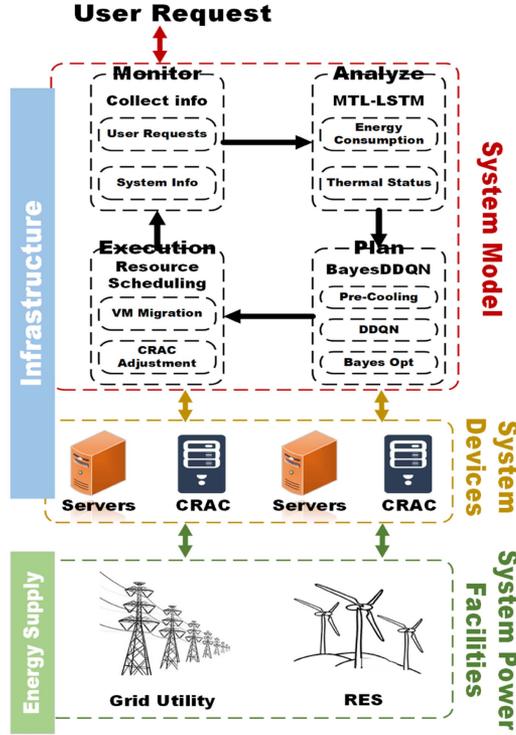


Fig. 1. The framework for holistic energy optimization of sustainable cloud data centers.

where  $N$  is the number of PMs in a cloud data center. The computation of  $E_i^{IT}(t)$  will be explained in the following section.  $E^{CRAC}(t)$  is derived as the ratio of computing energy consumption to the Coefficient of Performance (CoP) of the cloud data center, which is formulated as:

$$E^{CRAC}(t) = \frac{\sum_{i=1}^N E_i^{IT}(t)}{CoP(T^{sup}(t))} \quad (2)$$

where the value of CoP varies with the supplied CRAC temperature  $T^{sup}(t)$ . Given that accurately modeling the CoP model requires specialized and time-intensive CFD tools, which are currently only applicable to specific rack deployments. This paper references established research findings from the HP Utility Laboratory [30], which have also been utilized in current studies [5], [20] as:

$$CoP(T^{sup}(t)) = 0.0068(T^{sup}(t))^2 + 0.0008T^{sup}(t) + 0.458 \quad (3)$$

where the decrement of CRAC temperature  $T^{sup}(t)$  will result in a lower CoP value, leading to more cooling and DC energy consumption as depicted in (1) and (2), and vice versa.

For sustainable cloud data centers, RES is employed to mitigate the carbon emissions associated with increased computational and cooling energy demands. This paper intentionally opted wind energy for ensuring that the findings can effectively guide the practical deployment of local wind-powered cloud data centers situated in approximately 40°N latitude and 111°E longitude. This area is characterized by its strong and favorable wind conditions, making it suitable for wind energy generation. Moreover, wind energy has minimal carbon footprints among

renewable sources [31], and the energy generated by wind turbines can be modeled as a function of wind speed [32].

*Definition 2:* Let  $RES(t)$  be the generated renewable energy by the wind turbines at time  $t$ , which is decided by the actual wind speed  $v(t)$  and the number of wind turbines  $W$ .  $RES(t)$  can be defined as:

$$RES(t) = Wind(v(t)) \times W \quad (4)$$

$$Wind(v(t)) = \begin{cases} 0 & v(t) < v_{in}, v(t) > v_{out} \\ P_r \times \frac{v(t) - v_{in}}{v_r - v_{in}} & v_{in} < v(t) < v_r \\ P_r & v_r < v(t) < v_{out} \end{cases} \quad (5)$$

where  $Wind(v(t))$  is the generated wind energy of a wind turbine.  $P_r$  is the rated power of a wind turbine.

It can be found from (5) that when the wind speed is lower than the cut-in speed  $v_{in}$  or higher than the cut-out speed  $v_{out}$ , the wind turbine will not generate energy. If the actual wind speed is between cut-in and rated, the wind power will increase as the wind speed increases. If the wind speed is higher than the rated wind speed and lower than the cut-out speed, the wind turbine will output the rated power.

Based on the aforementioned definitions, brown energy consumption is characterized as the actual electricity acquired from the grid.

*Definition 3:* Let  $E^{Brown}(t)$  be the brown energy of the cloud data center, which is resulted by the consumed energy  $E^{DC}(t)$  and generated renewable energy  $RES(t)$ .  $E^{Brown}(t)$  is defined as:

$$E^{Brown}(t) = \max(0, E^{DC}(t) - RES(t)) \quad (6)$$

Therefore, if the generated renewable energy is insufficient to meet the energy consumption of the cloud data center, the cloud providers have to purchase the required electricity from the traditional grid, resulting in brown energy consumption. Conversely, the cloud data center will entirely depend on renewable energy.

### C. Inlet Temperature Model

During the holistic energy optimization, it is also necessary to address the hotspots caused by the air heat recirculation within the cloud data center. The inlet temperature is selected to estimate the cloud data center's thermal status, which considers the entire cooling process and workload variations. Cloud providers can then adjust CRAC parameters to adapt to the current thermal status of the data center [33].

*Definition 4:* Let  $T_i^{in}(t)$  be the inlet temperature of PM  $i$  at time  $t$ . Its value should be limited under the peak inlet temperature  $T_{red}$  to avoid hotspots and decrease the device failure possibility, which can be defined as:

$$\max\{T_i^{in}(t)\} \leq T_{red} \quad (7)$$

In this context,  $T^{sup}(t)$  can be adjusted with the help of pre-cooling technology if the generated RES is sufficient to satisfy the energy consumption. This method can be applied in the Plan module, and the supplied temperature after adjustment can be calculated as:

$$T^{sup'}(t) = T^{sup}(t) + T_{red} - \max\{T_i^{in}(t)\} \quad (8)$$

This paper also considers PMs' CPU temperature to further ensure the reliable operation of the cloud data center. According to the research conclusions from the widely used RC model [34],

the CPU temperature of a PM is primarily influenced by factors such as its thermal resistance, heat capacity and inlet temperature.

*Definition 5:* Let  $T_i(t)$  be the CPU temperature of the  $i$ th PM at time  $t$ ,  $R$  and  $C$  are the thermal resistance and heat capacity of this PM.  $T_i(t)$  can be defined as:

$$T_i(t) = E_i^{IT}(t) \times R + T_i^{in}(t) + T' \times e^{-\frac{t}{R \times C}} \quad (9)$$

$$T' = T_{initial} - E_i^{IT}(t) \times R - T_i^{in}(t) \quad (10)$$

Likewise, the CPU temperature should also not violate its maximum temperature  $T_{max}$ , which can be expressed as:

$$T_i(t) \leq T_{max} \quad (11)$$

#### D. Problem Formulation

Suppose that there is a cloud data center supplied with the electricity grid and renewable energy. The resource requests arrive at PMs in the form of VMs. The proposed BayesDDQN method aims to synchronously adjust computing and cooling resources based on available RES within  $[t_1, t_2]$ , so as to minimize the brown energy consumption. In this case, the objective function can be expressed as:

*Minimize:*

$$E_{t_1 \rightarrow t_2}^{Brown} = \int_{t_1}^{t_2} E^{Brown}(t) dt \quad (12)$$

*S.t. :*

$$\sum_{j=1}^{M_i} \mu_{i,j}^{vm} \times C_{i,j}^{vm} \leq C_i \quad (13)$$

$$(7), (11) \quad (14)$$

where  $M_i$  is the VMs hosted on PM  $i$ .  $\mu_{i,j}^{vm}$  is the utilization of VM  $j$  on the  $i$ th PM and  $C_{i,j}^{vm}$  is the capacity of this VM, which are summed to restrict the actual resource usage not exceed the capacity of the  $i$ th PM. Besides, the constraints in (14) prevent PMs from being hotspots.

#### IV. THE PROPOSED RL-BASED FRAMEWORK FOR HOLISTIC ENERGY OPTIMIZATION

In this section, a RL-based framework for holistic energy optimization of sustainable cloud data centers is proposed. The framework first designs a joint prediction method MTL-LSTM to evaluate energy consumption and thermal status. Then, a novel energy-aware approach BayesDDQN is presented. It explores potential combinations of VM migration and CRAC adjustment by integrating Bayesian optimization with DDQN to coordinate IT and cooling energy consumption within such hybrid action spaces. Moreover, this framework utilizes redundant RES to alleviate hotspots without migrating extra VMs. The framework's details are presented in the following.

##### A. MTL-LSTM Prediction Approach

The proposed MTL-LSTM energy-temperature prediction algorithm, as illustrated in Fig. 2, treats energy consumption and inlet temperature predictions as two subtasks with a shared LSTM layer. The underlying rationale for this approach stems from the fact that while improving CPU utilization through VM

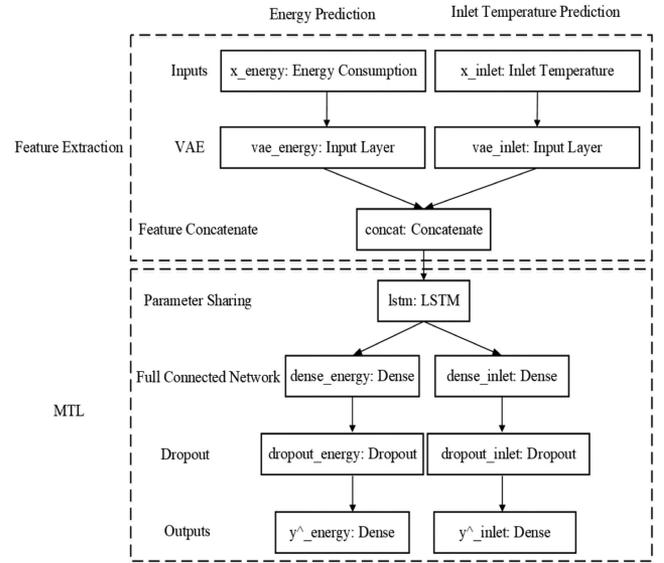


Fig. 2. Structure of MTL-LSTM.

consolidation effectively reduces IT energy consumption, it can also lead to localized hotspots. This, in turn, accelerates fan speeds to maintain lower temperatures, resulting in a decline in cooling energy efficiency. In this regard, MTL facilitates the learning of shared representations across these tasks, optimizing training time and exploring energy-related mutual relationships compared to training separate models for each task. LSTM complements MTL by capturing temporal and long-range dependencies, thereby incorporating contextual information into joint predictions. This integration enhances the generalization and accuracy of the predictive model.

In MTL-LSTM, PM's energy consumption and inlet temperature from the previous  $L$  time points at time  $t$  are considered as the original inputs, which can be depicted as:

$$x_{energy}(t) = [x_{energy}(t-L), \dots, x_{energy}(t-1)]$$

$$x_{inlet}(t) = [x_{inlet}(t-L), \dots, x_{inlet}(t-1)] \quad (15)$$

Then, VAE is proposed to perform feature extraction for reducing dimensionality while retaining relevant information, which is a valuable choice compared to Random Forest and XGBoost, especially in scenarios where complex relationships need to be captured [35]. The VAE-encoded features  $vae_{energy}(t)$  and  $vae_{inlet}(t)$  are transmitted as the input of a hard-sharing mechanism, which divides the model into a parameter-sharing layer and subtask learning layers. Specifically, the inputs of these two subtasks are first concatenated and then passed to the LSTM-based task-specific layers. The predicted values are denoted as:

$$\hat{y}_{energy}(t) = LSTM(vae_{energy}(t)) \quad (16)$$

$$\hat{y}_{inlet}(t) = LSTM(vae_{inlet}(t)) \quad (17)$$

LSTM models are well-suited for time series data prediction and have advantages in training efficiency with fewer parameters [36]. Given the widespread adoption of LSTM in existing literature, this paper will not provide further detailed definitions. In this context, each prediction task is treated as a subtask within MTL, trained by minimizing the mean squared error loss

between the predicted and actual values of energy consumption and inlet temperature over a training dataset. The loss function  $Loss$  is given by:

$$Loss = \frac{1}{S} \sum_{t=1}^S [(\hat{y}_{energy}(t) - y_{energy}(t))^2 + (\hat{y}_{inlet}(t) - y_{inlet}(t))^2] \quad (18)$$

where  $S$  is the total number of predicted data points.

Here, the training and inferencing energy consumption of the MTL-LSTM is not considered. The reason is that the energy consumption of such machine learning-based methods can vary significantly depending on the complexity of the model, the hardware used, and the scale of the dataset [37]. Consistent with most related research [21], [25], this paper emphasizes optimizing the improved energy efficiency of the scheduling strategy based on resource predictions rather than addressing the energy consumption of training these prediction models.

### B. Overview of BayesDDQN

The producer of the BayesDDQN algorithm is depicted in Algorithm 1. The MTL-LSTM prediction algorithm proposed in the analyze module is initially employed to evaluate the PM status with the collected information from the Monitor module (Line 3). If overloaded or overheated PMs are identified based on predefined static thresholds of CPU utilization or inlet temperature (Line 4), the *migrateServer* function will add them into the migrate PM list (Line 5). Accordingly, a list of VMs hosted on these PMs is generated, which sorts VMs in descending order of CPU utilization (Line 6). Line 7 calculates the brown energy using (6). After that, this algorithm will consider two scenarios according to the RES availability as illustrated in Lines 8-15. In cases where RES is sufficient to accommodate the increased cooling energy demand due to lowering CRAC temperature, the pre-cooling technology is activated to adjust  $T^{sup}(t)$  through (8), which is the most direct and effective way to mitigate hotspots without migrating any VMs (Line 9). Otherwise, the Plan module will create a decision model for resource scheduling based on the proposed BayesDDQN algorithm (Line 12). The output action set will prompt the Execute module to consolidate VMs and adjust CRAC temperature simultaneously (Lines 13-14). At last, the optimal holistic energy-aware strategy will be returned (Line 18).

### C. BayesDDQN-Based Holistic Energy Optimization Method

BayesDDQN-based energy optimization algorithm, as the core component of the proposed framework, is a model-free approach implemented by DDQN. This method employs two parallel neural network models with identical architectures to alleviate the overestimation challenge. The computationally intensive training and learning process are carried out prior to deployment by using extensive historical data, enabling real-time handling of diverse scheduling scenarios. Additionally, to effectively address the asynchrony issue while jointly executing VM consolidation and cooling control, DDQN is combined with Bayesian optimization to make optimal decisions. The following are detailed definitions for the state space, action space, and reward function of DDQN:

- *State Space*: The state space of DDQN includes the relevant information about the IT and cooling devices, which can

### Algorithm 1: DDQN-Based Algorithm.

---

**Input:** PM\_List,  $T^{sup}(t)$   
**Output:** VM migration and CRAC temperature adjustment strategy

```

1 Obtain RES generation
2 for  $t = 1, T$  do
3   PM_Status  $\leftarrow$  Apply the proposed MTL-LSTM prediction
   algorithm to evaluate the PM status
4   while There is an overloaded or overheated PM do
5     Migrate_PM_List  $\leftarrow$  migrateServer(PM_Status)
6     Migrate_VM_List  $\leftarrow$  VMs hosted on Migrate_PM_List
7     Brown_Energy  $\leftarrow$  Calculate the brown energy by Eq.(6)
8     if Renewable energy can satisfy the increased cooling energy then
9        $T^{sup}(t) \leftarrow$  Apply Pre-cooling technology
10    end
11  else
12    VM Migration, CRAC Adjustment  $\leftarrow$ 
    BayesDDQN(Migrate_PM_List, Migrate_VM_List)
13    Consolidate VMs based on Migration Strategy
14    Adjust  $T^{sup}(t)$  according to CRAC Adjustment
15  end
16 end
17 end
18 return Holistic Energy Optimization Strategy
```

---

be formulated as a tuple:

$$s_t = (s^{IT}(t), s^{Temp}(t)) \quad (19)$$

where  $s^{IT}(t)$  represents PMs' CPU utilization that contributes the most temperature and directly relates to energy consumption as described in Section IV-A. Furthermore, the possibility of PM overload can be predicted according to changes in CPU utilization, thus enhancing the efficiency of VM consolidation.  $s^{Temp}(t)$  describes system's thermal status, which evaluates both inlet and CPU temperatures to assess the efficiency of CRAC adjustment.

- *Action Space*: The agent should perform a series of actions to change the IT and cooling resource distribution in the cloud data center. In this paper, each discrete VM migration action is matched with continuous CRAC temperature settings. This combination forms a hybrid action space that covers a wide range of potential action strategies, which can be expressed as:

$$a = (k, x_k) \quad (20)$$

where  $k$  corresponds to a unique hash key that is mapped to a specific target PM and its corresponding rack, which is defined as  $k \in \{0, 1, \dots, N-1\}$ . Considering a simplified scenario with 12 PMs evenly distributed in 3 racks (Rack A, Rack B, Rack C) as shown in Fig. 3, the order of PMs is from PM0 to PM11. Assuming that PM1 in Rack A is overloaded and 10 is returned as the hash key of the maximal DDQN action value (Case 1). In this context, PM10 is the returned target PM, and parts of VMs on the overloaded PM1 are further migrated to the third PM (PM10) of Rack C. After the VM migration process, the workload on PM1 is reduced, while PM10's utilization and temperature may increase due to the migrated VMs. Accordingly, the cooling temperature will be adjusted to adapt to the new workload distribution, where  $x_k$  represents the supplied CRAC temperature that usually ranges from 15 °C to 25 °C. To address the challenge posed by the hybrid action space, this paper discretizes CRAC temperature into 101 levels at 0.1 intervals as mentioned in [38]. Each level corresponds to a potential adjustment step for CRAC temperature, which can be described as  $x_k \in \{15.0, 15.1, \dots, 24.9, 25.0\}$ . Existing DDQN-based

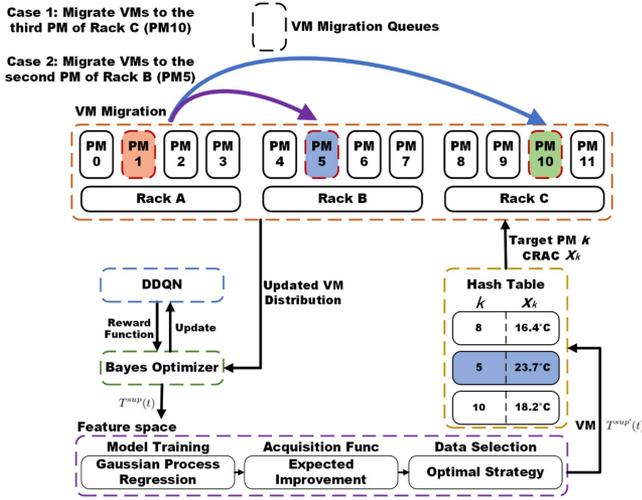


Fig. 3. Process of BayesDDQN.

methods often struggle to output both actions simultaneously, causing energy optimization conflicts in most cases. To address this issue, Bayesian optimization is integrated, exploring the influence of VM migration and CRAC adjustment combinations within the hybrid action space of DDQN for holistic energy optimization. The specifics of this combination will be elaborated in the following subsection.

- **Reward Function:** Reward  $r(t)$  is crucial for DDQN's learning process. It tries to replace the immediate rewards with the maximized long-term rewards, which can reflect the correct system state and the resource scheduling efficiency at time  $t$ . As stated in (12), the main optimization objective is to minimize brown energy by maximizing RES utilization. If the agent consumes more energy than the available renewable energy, the environment gives a penalty. Besides, both inlet temperature and CPU temperature are considered to avoid hotspots. Those values are normalized to the same scale with min-max technology. Therefore, the constraints defined in (14) are added to the reward function which can be defined as:

$$r(t) = \beta - \beta_1 \times E_{t_1 \rightarrow t_2}^{Brown} - \beta_2 \times T_i^{in}(t) - \beta_3 \times T_i(t) \quad (21)$$

where the coefficients are the negatively weighted values and the sum of them is 1. If the scheduling is feasible,  $\beta$  is set large enough to obtain positive rewards. The maximized reward value  $r(t)$  will be viewed as the candidate for the next scheduling. Otherwise, if the allocated resources or the thermal properties on a PM violate the defined constraints, the reward  $r(t)$  will be presumed to be negative infinity.

#### D. Jointly Optimization by BayesDDQN

To solve most RL methods' poor performance in the aforementioned hybrid action space while jointly scheduling IT and cooling resources, this paper proposes to combine Bayesian optimization with DDQN and the main idea of BayesDDQN is two-fold. On the one hand, BayesDDQN takes the reward function of DDQN as BO's objective function which could

explore the optimal VM migration action under the supplied CRAC temperature range. On the other hand, Bayesian optimization facilitates the sharing of parameters between different types of actions. These features encompass aspects of the data centers such as workload distribution and system temperatures. This allows DDQN to adapt its policies based on real-time feedback and to explore optimal energy optimization strategies in dynamic environments with multiple variables and dependencies.

The detailed process of BayesDDQN is shown in Fig. 3. During training, the DDQN algorithm learns the Q-values associated with hybrid actions, and the reward function of DDQN serves as the objective function for the Bayes optimizer that is fulfilled by the *bayes\_opt* function in the Python environment. The integration of Bayesian optimization enhances the exploration of the action space by providing estimates of the potential outcomes for different action combinations, which takes the supplied CRAC temperature  $T^{sup}(t) \in x_k$  as its hyperparameter. Subsequently, the default Gaussian process regression and expected improvement function are applied to obtain the optimal scheme. In particular, a hash table is designed to record the optimal VM migration strategy with the supplied cooling temperature as depicted in Fig. 3. Considering the important influence of CRAC settings on holistic energy consumption as discussed in (2), BayesDDQN will prioritize the highest CRAC temperature (23.7 °C) and regard the associated PM5 as the target PM. In this context, take the same example from the previous subsection, VMs on an overloaded server (PM1) are recommended to be migrated to the second PM (PM5) of Rack B (Case 2). The updated VM distribution will be then transmitted to the Bayes optimizer for achieving further optimization strategy. In general, the feature space of the Bayes optimizer outputs the optimal cooling temperature  $T^{sup'}(t)$ , and its corresponding VM migration strategy can be retrieved from the hash table.

#### E. Example of the Proposed Framework

To illustrate the working of the proposed framework, a complete (small-sized) instance of the problem is presented, including all necessary inputs and definitions. The scenario involves a RES-powered data center with four PMs dedicated to serving user resource requests, which are collected by the **Monitor** and the CRAC ( $T^{sup}(t)$ ) is set to 20 °C.

In **Analyze**, the energy consumption and inlet temperature of these PMs are predicted using the MTL-LSTM method, which are assumed as (60W, 24°C), (70W, 25°C), (90W, 29°C), (100W, 30°C). The holistic energy consumption is evaluated by the following steps:

- With *Definition1*: The sum of each PM's IT energy consumption  $\sum_{i=1}^N E_i^{IT}(t)$  is 320 Watts.
- According to (3) (*Definition1*): The CoP value  $CoP(20^\circ\text{C})$  is 3.2.
- In (2) (*Definition1*): The cooling energy consumption  $E^{CRAC}(t)$  is 100 Watts.
- Based on (1) (*Definition1*): The holistic energy consumption  $E^{DC}(t)$  is 420 Watts.

Then, the thermal status including inlet temperature and CPU temperature are evaluated by the following steps:

- With *Definition4*: The maximized inlet temperature  $\max\{T_i^{in}(t)\}$  of all PMs is 30°C. Assuming that the peak

inlet temperature  $T_{red}$  is 25 °C, the last two PMs in this example are viewed as hotspots.

- With *Definition5*: The CPU temperature is simplified to be satisfied with the peak constraint.

In **Plan**, the proposed BayesDDQN approach firstly considers the gap between energy consumption and RES generation to decide the optimal resource scheduling strategy. The optimization process are designed by the following steps:

- With *Definition2*: Suppose there are five wind turbines, each with a capacity of 100 Watts. The generated renewable energy  $RES(t)$  is 500 Watts.
- With *Definition3*: The available RES (500 Watts) is sufficient to power the data center (420 Watts) without consuming brown energy ( $E^{Brown}(t) = 0$ ).
- Based on (8) (*Definition4*): The pre-cooling technology is applicable, the supplied temperature ( $T^{sup}(t)$ ) can be lowered to 15 °C. Ideally, this adjustment would eliminate all hotspots. In such a scenario, the CoP value  $CoP(15^\circ C)$  is 2. The cooling and holistic energy consumption will be increased to 160 Watts and 480 Watts respectively, which can still be fully powered by the RES.
- Otherwise, according to Algorithm 1, BayesDDQN is employed to achieve the optimal energy optimization strategy.

In **Execute**, the VM deployment and CRAC setting will be adjusted accordingly as depicted in Sections IV-C and IV-D.

#### F. Overhead Analysis of BayesDDQN

The BayesDDQN algorithm is designed to optimize holistic energy consumption by integrating Bayesian optimization with the DDQN. For DDQN, its primary computational overhead arises from the forward pass through the neural network for Q-value estimation, which is typically  $O(P)$ , where  $P$  represents the number of neurons in the network. Additionally, the training phase involves updating the network weights based on the gradient descent algorithm, with a time complexity of  $O(P^2)$  due to matrix operations. For Bayesian optimization, it refines its search for the optimal action iteratively. The time complexity of such process depends on the number of iterations  $R$  and the complexity of evaluating the reward function of DDQN at a set of candidate points, which has a time complexity of  $O(Q)$ , where  $Q$  is the number of candidate points. The time complexity of Bayesian optimization can be then approximated as  $O(R \times Q)$ . Therefore, the overall time complexity of BayesDDQN is a combination of the time complexities of its components, resulting in a time complexity of  $O(R \times Q + P^2)$  for the training phase and  $O(P)$  for the inference phase. In general, while the training overhead is significant, it is typically performed offline that allows for real-time holistic energy optimization decisions without further training.

## V. EXPERIMENTAL EVALUATION

This section first details the cloud data center configuration, and renewable energy traces. Then, four state-of-art approaches are introduced as baseline algorithms to evaluate the feasibility and effectiveness of the proposed RL-based framework, which are fulfilled by PyCharm 3.3 and simulated on a PC with Intel(R) Core i7-8750H processor with 2.2 GHz CPU and 16 GB RAM. Finally, the superior experimental performance of the proposed framework is further analyzed.

TABLE II  
VM TYPES AND CAPACITY

VM Type	CPU (MIPS)	Memory (GB)	Storage (GB)	Number of Cores
c1.medium	2500	0.85	350	2
m1.large	2000	3.75	840	2
m1.small	1000	1.7	160	1
ti.micro	500	0.613	1	1

TABLE III  
PARAMETERS OF WIND TURBINE

Parameter	$v_{in}$ (m/s)	$v_{out}$ (m/s)	$v_r$ (m/s)	$P_r$ (kW)
	2.5	35	11	3

#### A. Experimental Setup

The simulation environment in this study is an expanded version of previously published works [39], [40], which is designed to emulate a data center environment with 10 racks, each containing 5 PMs [6]. The setup includes four types of VMs with different computational capabilities inspired by Amazon EC2 instance types,<sup>3</sup> with the Million Instructions Per Second (MIPS) value indicating the CPU performance as shown in Table II, and CPU thresholds for PMs are set to 0.9 to identify overload status. The simulator operates by generating resource requests based on estimated daily active users from various regions, as modeled on the Google cluster trace dataset [41]. This trace includes diverse applications and users around the world and collects workload data from a 12500-machine cluster over 29 days. Moreover, the simulator follows the proposed holistic energy optimization framework in Sections III and IV to design and execute the optimal IT and cooling resource scheduling according to the analysis of varied requests' effects on energy and thermal conditions. As this paper focuses on scheduling workloads viewed as VM requests with required resources, it does not consider application types running within VMs.

To evaluate the RES availability, the RES data traces come from the National Renewable Energy Laboratory (NREL) [42] are adopted. This database provides time-series data from 30000 measurement points worldwide at 10-minute intervals, and the Arizona wind speed traces on September 3rd in 2019 is adopted. In experiments, the total experiment period is set to 24 hours and the scheduling interval to 10 minutes. The cloud data center is assumed to be equipped with the 5 NE-3000 wind turbines, the parameters of the wind turbine are shown in Table III.

From the hotspots perspective, exceeding the device temperature threshold significantly raises the risk of device failure [33]. Therefore, in this paper and this example, the maximum allowable CPU temperature  $T_{max}$  (11) and the peak inlet temperature  $T_{red}$  (7) of a server are set to 105 °C and 25 °C according to the American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE) [43] advice. Furthermore, the parameters of thermal resistance and the heat capacity in (9) are set to 0.34K/W and 340J/K, respectively [34].

In addition, the data traces for MTL-LSTM training and testing are sourced from the Melbourne Research Cloud datasets [34], which collect resource utilization and cooling

<sup>3</sup><https://aws-pricing.com/instances.html>

parameters every 10 minutes, including CPU load, network utilization, fan speeds, RAM utilization, running VMs, and CPU cores. In experiments, the parameter  $L$  in (15) is set to 30, indicating that the last 5-hour data are used for predicting energy consumption and thermal status at time  $t$ . To align with the 24-hour scheduling period mentioned earlier, the parameter  $S$  in (18) is set to 144, corresponding to 6 data points per hour with 10-minute intervals.

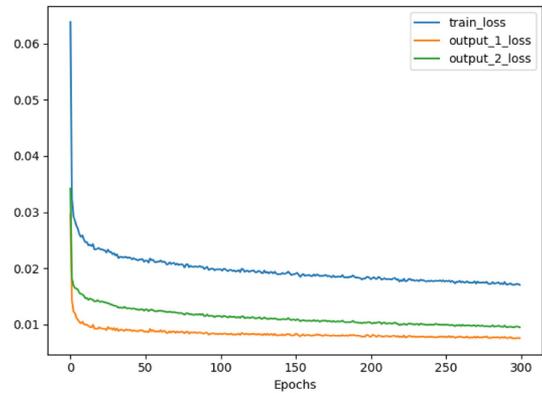
### B. Comparison Algorithms

Since recent studies have not proposed a RL-based approach to coordinate IT and cooling resources for achieving the holistic energy optimization of sustainable cloud data center, four most related algorithms are introduced to evaluate the performance of the proposed BayesDDQN algorithm.

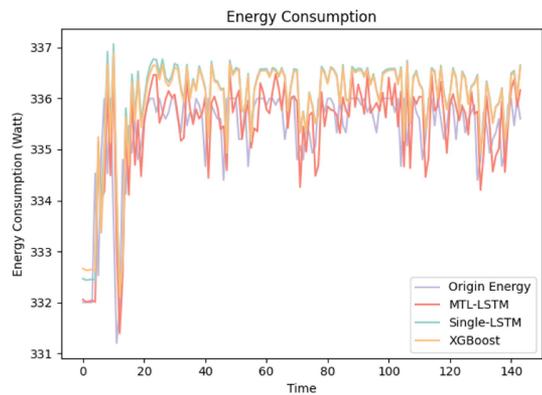
- **Thermal Prediction for Energy Management (TAS)** [34]: TAS considers hotspots detection in traditional data centers, which proposes a gradient-boosting machine learning model for temperature prediction. Once a hotspot occurs, VMs are migrated to the PM with the lowest temperature for reducing the peak temperature. The reduced cooling costs contribute to significant energy optimization.
- **Thermal-Aware Scheduling (TA)** [11]: TA considers thermal imbalance in traditional data centers, which designs a deep neural network to calculate the thermal imbalance. TA executes thermal-aware migrations for minimizing the temperature difference between the hottest and coldest PMs, and hence reducing the holistic energy consumption.
- **DeepEE** [22]: DeepEE considers conflicts between IT and cooling energy consumption in traditional data centers, which parameterizes the Deep Q-Network action space to jointly optimize the resource scheduling of VM consolidation and CRAC adjustments, which selects the destination server according to the predefined theoretical optimum temperature.
- **Thermal-Aware Workload Management (TAWM)** [23]: TAWM method considers the availability of RES, which applies the pre-cooling technology to dynamically adjust the CRAC temperature or power on idle servers to mitigate hotspots when RES is sufficient.
- **BD-LSTM**: BD-LSTM is a variation of the proposed framework, which employs LSTM as the resource prediction method instead of the MTL-LSTM strategy for guiding BayesDDQN to optimize resource allocation.
- **BD-XGBoost**: BD-XGBoost is another alternative version of the proposed framework, which utilizes XGBoost to predict resource variations and BayesDDQN to schedule resources.

### C. Experimental Results

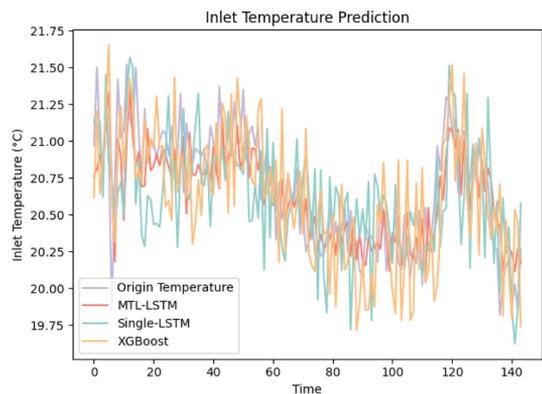
In experiments, a series of sensitivity analysis are conducted to determine how fluctuating parameters and various approaches (e.g., VM distributions, prediction methods, and scheduling strategies) affect the simulator's outcomes. To further quantify the potential impact of noise and variability in real-world systems, all possible PM anomalies scenarios under different VM initial deployments are simulated, including overload, violation of  $T_{red}$  inlet temperature, violation of  $T_{max}$  (CPU temperature), and all combinations of the three anomalies. Simulation experiments run 10 times, the average results from these executions



(a) Training loss



(b) Energy consumption prediction



(c) Inlet temperature prediction

Fig. 4. Performance and accuracy evaluation for MTL.

are used to evaluate the framework's performance in terms of prediction accuracy, energy consumption, temperature, RES utilization, and VM migrations.

Accurately predicting energy consumption and inlet temperature is a key measure of the effectiveness of the proposed MTL-LSTM method. Merely obtaining the root mean square error (RMSE) of the prediction method is insufficient as the number of episodes required for the training loss to converge should remain current with the highly dynamic conditions in multi-resource prediction environments. As evidenced through Fig. 4(a), the

TABLE IV  
PREDICTION RMSES OF ENERGY CONSUMPTION AND INLET TEMPERATURE

Prediction	Energy Consumption	Input Temperature
MTL-LSTM	1.63	0.23
LSTM	3.03	0.62
XGBoost	3.45	0.38

training loss curves of inlet temperature (*output\_1\_loss*) and energy consumption (*output\_2\_loss*) in MTL-LSTM begin to converge after 30 epochs. During the experiments, each training epoch lasts approximately 2 seconds. Consequently, for a MTL-LSTM model with 300 iterations, retraining occurs every 10 minutes, with convergence happening after 1 minute. Moreover, MTL-LSTM leverages the shared parameters to capture complex non-linear relationships in multi-features. As a result, the prediction curve of MTL-LSTM closely aligns with actual traces as shown in Fig. 4(b) and (c), averaging a nearly 50% reduction in RMSE errors for energy consumption and inlet temperature compared to LSTM and XGBoost as presented in Table IV. Regarding other techniques, XGBoost is particularly effective when dealing with structured data with categorical variables, such as thermal-related attributes, reducing 38.7% inlet temperature prediction RMSE error than LSTM. On the other hand, LSTM can capture long-term dependencies in sequential data, which reduces 12.2% RMSE error of energy prediction than XGBoost.

Powering DCs with RES provides new opportunities to reduce reliance on brown energy, but the uncertain feature of RES also leads to massive energy wastage. In this context, besides the total energy consumption, the actual consumed brown energy that is directly related to the purchased fossil fuel is analyzed. Furthermore, the feasibility of the pre-cooling method in such environment is evaluated according to saved cooling energy consumption. As shown in Fig. 5, the detailed consumed energy at each time slot aims to assess the effectiveness of the scheduling decisions taken by the proposed framework, including total energy consumption, brown energy consumption, and cooling energy consumption. The total time slot can be categorized into two Phases based on the availability of RES, where no sufficient RES in the first 15 hours and the last 3 hours are depicted as Phase 1 in Table V, and the rest hours are depicted as Phase 2 in Table V. During Phase 1, due to the lack of sufficient RES, all algorithms prioritize thermal awareness, focusing on hotspot mitigation and cooling cost reduction. The RL-based BayesDDQN and DeepEE can proactively learn and excel at adapting to limited renewable energy scenarios. As shown in Table V, they achieve average reductions in brown energy consumption of 7.53% and 5.2% than heuristic-based methods, respectively. In Phase 2, Fig. 5 illustrates abundant RES generation, which allows BayesDDQN and TAWM to adjust their CRAC temperatures by applying the pre-cooling technology without incurring additional brown energy, and hence consuming less cooling energy than others. In contrast, TAS, TA and DeepEE only process VM consolidation with default CRAC temperatures, and DeepEE incurs extra brown energy usage at times 18, 20, and 21 to satisfy thermal constraints. Nevertheless, with the help of the neural network, DeepEE can better optimize energy performance holistically, reducing average 3.98% brown energy consumption and 4.38% cooling energy consumption compared to TAS, TA

and TAWM across all phases. Furthermore, when BayesDDQN is combined with LSTM or XGBoost prediction methods, as shown at the bottom of Table V, the enhanced accuracy of the proposed MTL-LSTM method leads to an average reduction of 2657.22 kWh in total energy consumption and 1099.84 kWh in cooling energy consumption. In general, compared to baseline methods, the energy-saving distributions of BayesDDQN over ten simulation runs, along with their confidence intervals, are visually represented by the Kernel Density Estimation (KDE) plot in Fig. 5(f). The blue curve depicts the estimated probability density function of energy savings. The red dashed line denotes the mean savings, while the green dashed lines indicate a 95% confidence interval, suggesting that the energy savings are likely to fall between 2.68% and 2.98%. This demonstrates that the observed energy improvements from BayesDDQN are not due to random fluctuations but rather a direct outcome of the optimization framework proposed, and thus achieving an average reduction of 2.83% in holistic energy consumption, a 4.74% decrease in brown energy usage, and a 13.48% reduction in cooling energy consumption, which are crucial for relieving carbon footprint and preventing over-cooling energy wastage.

To ensure that data centers operate in a safety environment without localized overheating caused by excessive workload concentration, this paper conducts a comprehensive temperature comparison, including the average CPU temperature, average inlet temperature, and hotspots variation over a 24-hour period as shown in Fig. 6. Although these metrics can be met with lower CRAC settings, this results in significant cooling energy consumption as depicted in (2). Therefore, the average cooling temperature is also compared to evaluate the effectiveness of temperature adjustment strategies employed by these methods. It can be found from Fig. 6(a) and (b) that both CPU temperature and inlet temperature align with ASHRAE recommendations as depicted in Section V-A, ensuring the reliability of all algorithms. In such scenario, BayesDDQN effectively prevents over-cooling by maintaining the highest average CRAC temperature (21.27 °C) as shown in Fig. 6(c), resulting in the lowest cooling energy consumption of 67817.69 kWh (Table V). Meanwhile, BayesDDQN does not incur excessive hotspots during this optimization process as shown in Fig. 6(d), and its overheated situation is comparable to the TAWM algorithm which generates the fewest hotspots. Therefore, BayesDDQN proves feasible for saving cooling energy with a higher CRAC temperature setting, also contributing to reduced brown energy consumption and promoting sustainable cloud data centers as detailed in Table V. In specific, compared to dynamic CRAC setting methods TAWM and DeepEE, BayesDDQN reduces 7.16% (154011.49 kWh vs 143714.50 kWh) and 3.81% (149194.32 kWh vs 143714.50 kWh) brown energy consumption, respectively. On the contrary, static CRAC setting methods TAS and TA tend to provide lower cooling temperature, leading to increases in cooling energy consumption by 26.85% and 27.25%. Furthermore, although BD-LSTM and BD-XGBoost perform similarly to BayesDDQN in temperature-related metrics, they have to consume more energy to handle additional hotspots due to significant RMSE gaps as discussed in Table IV, wasting an additional 2177.43 kWh and 2494.56 kWh of brown energy, respectively.

Fig. 7 illustrate the comparative results of RES utilization that indicates the ratio of the used wind energy to the actual energy consumed by the data center. This metric provides a clear

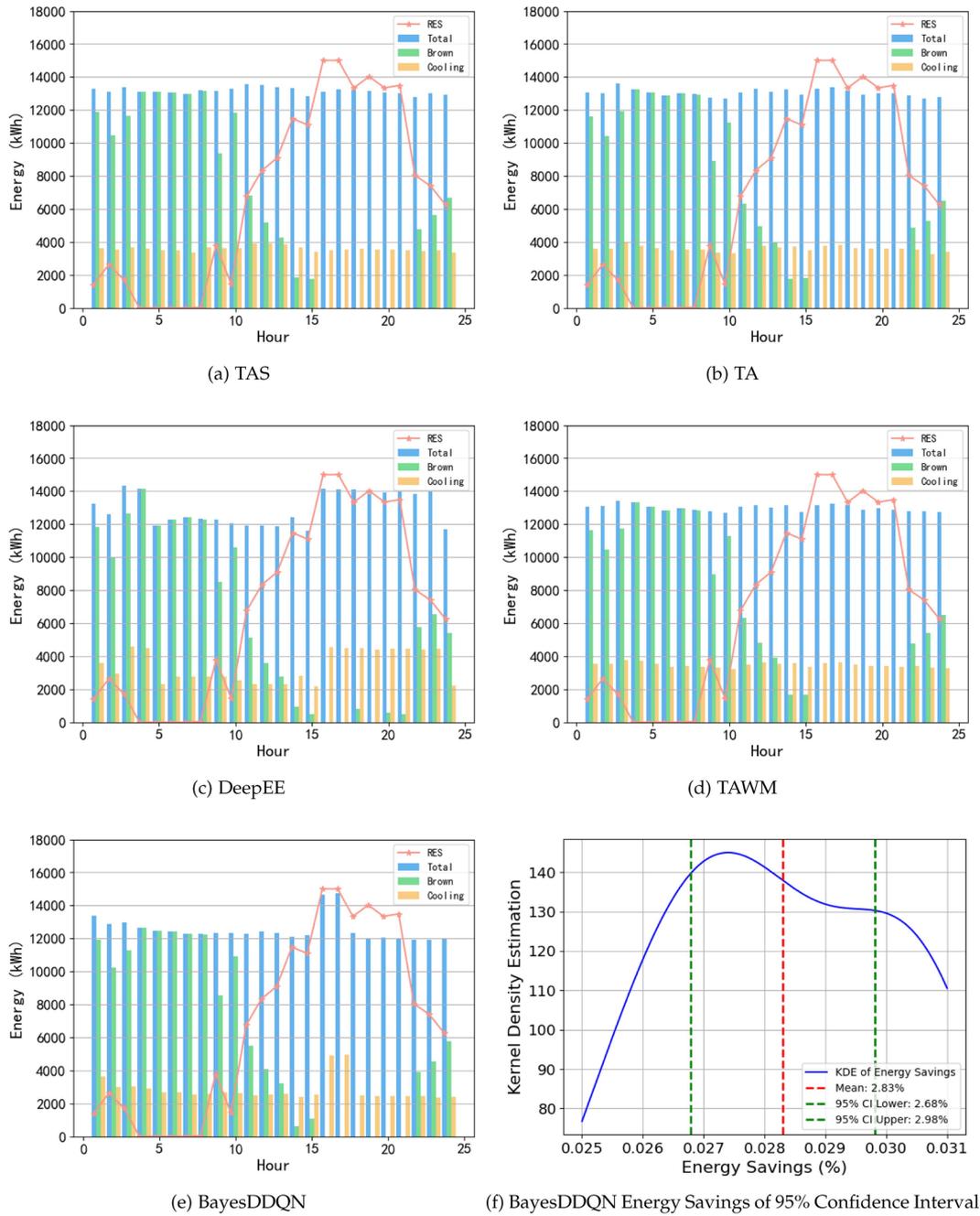


Fig. 5. Energy consumption comparison.

TABLE V  
ENERGY CONSUMPTION COMPARISON (kWh)

	Total Energy	Phase 1	Phase 2	Brown Energy	Phase 1	Phase 2	Cooling Energy	Phase 1	Phase 2
TAS	315830.13	236998.91	78831.22	157541.31	157541.31	0	86030.13	64819.91	21210.22
TA	312934.61	234169.62	78764.99	154712.02	154712.02	0	86300.61	64332.62	21967.99
DeepEE	310948.71	226785.51	84163.20	149194.32	147327.91	1866.41	81327.71	54438.51	26889.2
TAWM	311701.95	233469.09	78232.86	154011.49	154011.49	0	83173.95	62283.09	20890.86
<b>BayesDDQN</b>	<b>300944.49</b>	<b>223172.10</b>	<b>77772.39</b>	<b>143714.50</b>	<b>143714.50</b>	<b>0</b>	<b>67817.69</b>	<b>48118.70</b>	<b>19698.99</b>
BD-LSTM	304071.55	225908.17	78163.38	146209.06	146209.06	0	69334.64	49316.10	20018.54
BD-XGBoost	303131.87	225173.26	77958.61	145891.93	145891.93	0	68500.41	48693.73	19806.68

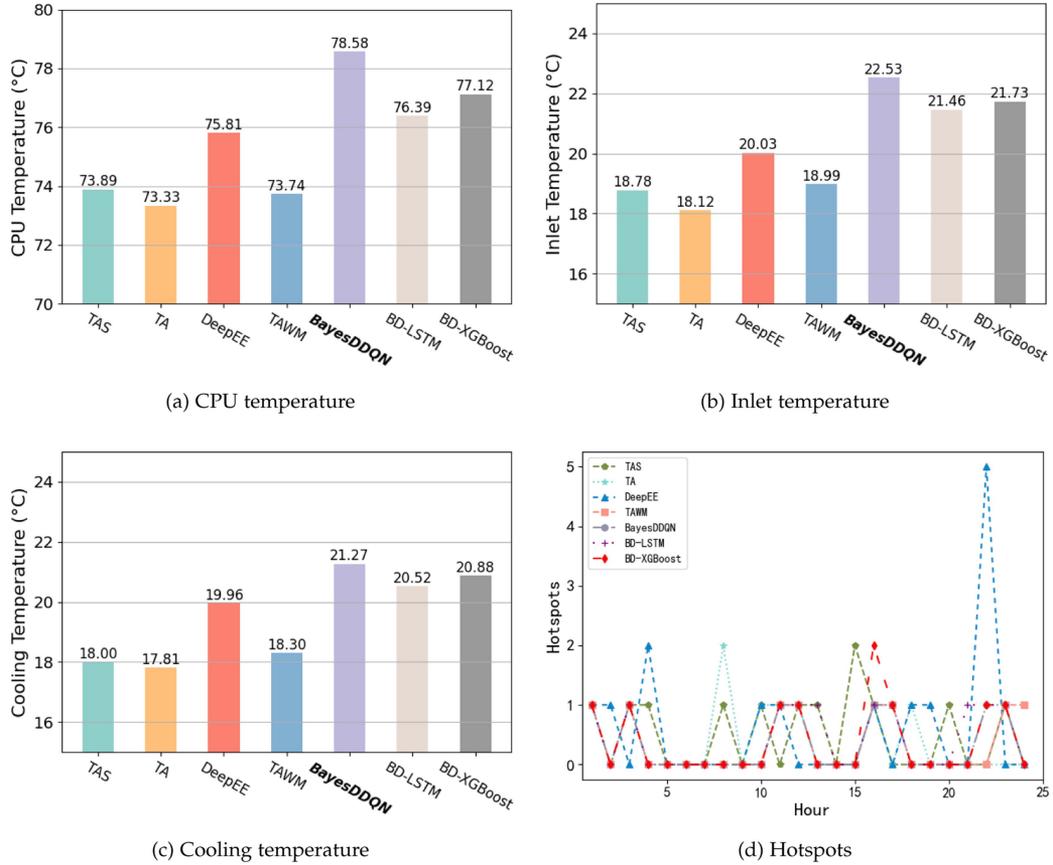


Fig. 6. Temperature comparison.

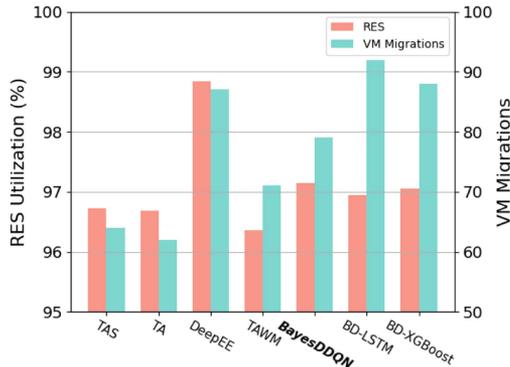


Fig. 7. RES utilization and VM migrations Comparison.

measure of how well different resource scheduling algorithms integrate and optimize the use of renewable energy sources. It can be found that the RES utilization of RL-based methods performs better, where BayesDDQN is lower than that of DeepEE, but BayesDDQN consumes less brown energy by saving 19.92% cooling energy (67817.69 kWh vs 81327.71 kWh) as detailed in Table V. This difference primarily stems from the energy consumption of BayesDDQN in Phase 2, which correlates with RES fluctuations as shown in Fig. 5(e). BayesDDQN executes finer-grained pre-cooling actions for the cooling equipment to maximize RES utilization and coordinates IT and cooling

energy consumption with a RL-based VM consolidator to reduce brown energy consumption. Similar advantages are seen in BD-LSTM and BD-XGBoost, where their average cooling energy consumption in Phase 2 is 35.04% less than that in DeepEE. On the contrary, the asynchronous control of such resource scheduling disables DeepEE to achieve the optimal solution under the predefined CRAC temperature. The benefits of the pre-cooling technology in enhancing RES utilization are suitable for heuristic algorithms as well. Specifically, TA and TAWM exhibit similar RES performance and brown energy consumption (154712.02 kWh vs 154011.49 kWh) as shown in Table V, their main energy differences are originated from cooling energy. With the help of pre-cooling, TAWM saves 5.15% cooling energy while improving only 0.63% RES utilization (93.56% vs 92.93%) in Phase 2. It is anticipated that cooling energy will be further reduced when considering longer-term RES utilization.

Comparing the number of VM migrations provides insight into the operational efficiency of different resource scheduling algorithms as shown in Fig. 7. Effective VM migrations can avoid increased overhead, including higher energy consumption and potential performance degradation due to the resource-intensive nature of the migration process. In this paper, VMs will be migrated if a server experiences a level of 90% CPU utilization or expresses the temperature constraints  $T_{red}$  and  $T_{max}$ . As expected, thermal-aware algorithms such as TAS and TA ignore the holistic energy optimization of both systems, which focuses on optimizing IT energy consumption with a

lower CRAR temperature as shown in Fig. 6(c). Such limited search space makes their migrations are mainly caused by overloaded servers, leading to fewer VM migrations. In contrast, TAWM utilizes pre-cooling technology to alleviate hotspots as shown in Fig. 6(d) while avoiding excessive VM migrations. For RL-based methods, DeepEE will execute 87 VM migrations to optimize various metrics, whereas BayesDDQN requires 79 migrations. The main reason can be inferred from Fig. 5 and Fig. 6(d). As shown, DeepEE has two more hotspots when RES is sufficient (Phase 2) and 7 more hotspots at time 4 and 22, while BayesDDQN applies pre-cooling in Phase 2 and dynamically determines the optimal suggested CRAC temperature with Bayes optimizer to avoid unnecessary VM migrations in Phase 1. For BD-LSTM and BDXGBoost, their increased migrations compared to BayesDDQN can be attributed to unpredicted hotspots due to their worse accuracy.

## VI. CONCLUSION

In this paper, a reinforcement learning based framework is proposed to optimize the holistic energy of sustainable cloud data centers. This framework begins with the development of a MTL-LSTM prediction method, where LSTM is employed as the parameter-sharing layer to improve the accuracy of jointly estimating energy consumption and thermal status. Following this, a novel energy-aware approach BayesDDQN is presented to tackle asynchrony challenges by integrating Bayesian optimization with DDQN. This combination excels in coordinating conflicts of IT and cooling energy consumption while effectively utilizing renewable energy, resulting in superior energy efficiency and environmental sustainability. Additionally, the framework incorporates pre-cooling technology, which minimizes unnecessary VM migrations and reduces reliance on brown energy by maximizing the use of RES. The experimental results show that the framework reduces holistic energy consumption by an average of 2.83%, with a 4.74% decrease in brown energy. Additionally, cooling energy consumption is reduced by an average of 13.48% while generating the fewest hotspots. Meanwhile, the proposed MTL-LSTM prediction method within this framework outperforms XGBoost and LSTM methods, resulting in half RMSE errors for energy consumption and inlet temperature.

In future work, the proposed algorithm is expected to be extended in a geographically distributed cloud environment, where hundreds and thousands of rack servers will contribute huge state space and computationally intractable action space. Moreover, this work focuses on a simplified differentiation between brown and wind energy, which can be refined by incorporating more granular distinctions among different types of renewable and non-renewable energy sources (e.g. solar, wind, hydro, coal). These energy mixes are also expected to be utilized to power both the training and execution of prediction and RL models, thereby improving the overall energy footprint. Finally, there is also a necessity that the proposed BayesDDQN be evaluated in a real cloud infrastructure such as OpenStack, where thermal sensors can be installed to collect thermal information.

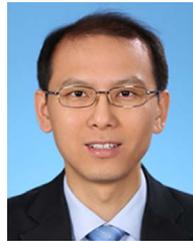
## REFERENCES

- [1] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, 2019.
- [2] K. Ajmera and T. K. Tewari, "Energy-efficient virtual machine scheduling in iaas cloud environment using energy-aware green-particle swarm optimization," *Int. J. Inf. Technol.*, vol. 15, no. 4, pp. 1927–1935, 2023.
- [3] Z. Cao, X. Zhou, H. Hu, Z. Wang, and Y. Wen, "Toward a systematic survey for carbon neutral data centers," *IEEE Commun. Surv. Tut.*, vol. 24, no. 2, pp. 895–936, Second Quarter, 2022.
- [4] A. Almoli, A. Thompson, N. Kapur, J. Summers, H. Thompson, and G. Hannah, "Computational fluid dynamic investigation of liquid rack cooling in data centres," *Appl. Energy*, vol. 89, no. 1, pp. 150–155, 2012.
- [5] A. Aghasi, K. Jamshidi, and A. Bohlooli, "A thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm (FC-BGSA)," *Cluster Comput.*, vol. 25, pp. 1015–1033, 2022.
- [6] S. Ilager and R. Buyya, "Machine learning in energy and thermal-aware resource management of cloud data centers: A taxonomy and future directions," in *Proc. 19th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, 2024, pp. 21–34.
- [7] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Gener. Comput. Syst.*, vol. 104, pp. 174–186, 2020.
- [8] J. DeCarolis and A. LaRose, "Annual energy Outlook 2023 with projections to 2050," US Energy Information Administration, 2023.
- [9] A. N. Toosi, C. Qu, M. D. de Assunção, and R. Buyya, "Renewable-aware geographical load balancing of web applications for sustainable data centers," *J. Netw. Comput. Appl.*, vol. 83, pp. 155–168, 2017.
- [10] M. Xu, A. N. Toosi, and R. Buyya, "A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 4, pp. 544–558, Oct.–Dec. 2021.
- [11] S. Akbar, R. Li, M. Waqas, and A. Jan, "Server temperature prediction using deep neural networks to assist thermal-aware scheduling," *Sustain. Comput.: Inform. Syst.*, vol. 36, 2022, Art. no. 100809.
- [12] G. Rjoub, J. Bentahar, O. Abdel Wahab, and A. Saleh Bataineh, "Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems," *Concurrency Comput.: Pract. Experience*, vol. 33, no. 23, 2021, Art. no. e5919.
- [13] M. Radi, A. A. Alwan, and Y. Gulzar, "Genetic-based virtual machines consolidation strategy with efficient energy consumption in cloud environment," *IEEE Access*, vol. 11, pp. 48022–48032, 2023.
- [14] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1317–1331, Jun. 2018.
- [15] S. S. Gill et al., "Thermosim: Deep learning based framework for modeling and simulation of thermal-aware resource management for cloud computing environments," *J. Syst. Softw.*, vol. 166, 2020, Art. no. 110596.
- [16] H. Wu, Y. Chen, C. Zhang, J. Dong, and Y. Wang, "Loads prediction and consolidation of virtual machines in cloud," *Concurrency Comput.: Pract. Experience*, vol. 35, no. 23, 2023, Art. no. e7760.
- [17] W. Yao, Z. Wang, Y. Hou, X. Zhu, X. Li, and Y. Xia, "An energy-efficient load balance strategy based on virtual machine consolidation in cloud environment," *Future Gener. Comput. Syst.*, vol. 146, pp. 222–233, 2023.
- [18] R. Shaw, E. Howley, and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," *Inf. Syst.*, vol. 107, 2021, Art. no. 101722.
- [19] P. Wei, Y. Zeng, B. Yan, J. Zhou, and E. Nikougoftar, "VMP-A3C: Virtual machines placement in cloud computing based on asynchronous advantage actor-critic algorithm," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 5, 2023, Art. no. 101549.
- [20] J. Li, Y. Deng, Y. Zhou, Z. Zhang, G. Min, and X. Qin, "Towards thermal-aware workload distribution in cloud data centers based on failure models," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 586–599, Feb. 2023.
- [21] S. Tuli et al., "HUNTER: AI based holistic resource management for sustainable cloud computing," *J. Syst. Softw.*, vol. 184, 2022, Art. no. 111124.
- [22] Y. Ran, H. Hu, X. Zhou, and Y. Wen, "DeepEE: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 645–655.
- [23] Y. Li, X. Wang, P. Luo, and Q. Pan, "Thermal-aware hybrid workload management in a green datacenter towards renewable energy utilization," *Energies*, vol. 12, no. 8, 2019, Art. no. 1494.
- [24] T. Renugadevi, K. Geetha, K. Muthukumar, and Z. W. Geem, "Optimized energy cost and carbon emission-aware virtual machine allocation in sustainable data centers," *Sustainability*, vol. 12, no. 16, 2020, Art. no. 6383.

- [25] H. Wang, H. Shen, J. Gao, K. Zheng, and X. Li, "Multi-agent reinforcement learning based distributed renewable energy matching for datacenters," in *Proc. 50th Int. Conf. Parallel Process.*, 2021, pp. 1–10.
- [26] A. Jayanetti, S. Halgamuge, and R. Buyya, "Multi-agent deep reinforcement learning framework for renewable energy-aware workflow scheduling on distributed cloud data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 4, pp. 604–615, Apr. 2024.
- [27] J. Zhao, M. A. Rodríguez, and R. Buyya, "A deep reinforcement learning approach to resource management in hybrid clouds harnessing renewable energy and task scheduling," in *Proc. IEEE 14th Int. Conf. Cloud Comput.*, 2021, pp. 240–249.
- [28] A. Aghasi, K. Jamshidi, A. Bohlooli, and B. Javadi, "A decentralized adaptation of model-free Q-learning for thermal-aware energy-efficient virtual machine placement in cloud data centers," *Comput. Netw.*, vol. 224, 2023, Art. no. 109624.
- [29] H. Cheng, B. Liu, W. Lin, Z. Ma, K. Li, and C.-H. Hsu, "A survey of energy-saving technologies in cloud data centers," *J. Supercomput.*, vol. 77, no. 11, pp. 13385–13420, 2021.
- [30] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling 'cool': Temperature-aware workload placement in data centers," in *Proc. USENIX Annu. Tech. Conf.*, 2005, pp. 61–75.
- [31] N. Scarlat, M. Prussi, and M. Padella, "Quantification of the carbon intensity of electricity produced and used in Europe," *Appl. Energy*, vol. 305, 2022, Art. no. 117901.
- [32] Y. Zhang, H. Sun, and Y. Guo, "Wind power prediction based on PSO-SVR and grey combination model," *IEEE Access*, vol. 7, pp. 136254–136267, 2019.
- [33] Q. Tang, T. Mukherjee, S. K. Gupta, and P. Cayton, "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," in *Proc. 4th Int. Conf. Intell. Sens. Inf. Process.*, 2006, pp. 203–208.
- [34] S. Ilager, K. Ramamohanarao, and R. Buyya, "Thermal prediction for efficient energy management of clouds using machine learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1044–1056, May 2021.
- [35] Y. Huang, D.-R. Liu, S.-J. Lee, C.-H. Hsu, and Y.-G. Liu, "A boosting resampling method for regression based on a conditional variational auto-encoder," *Inf. Sci.*, vol. 590, pp. 90–105, 2022.
- [36] Z. Shen, X. Zhang, B. Xia, Z. Liu, and Y. Li, "Multi-granularity power prediction for data center operations via long short-term memory network," in *Proc. 2019 IEEE Int. Conf. Parallel Distrib. Process. Appl. Big Data Cloud Comput. Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, 2019, pp. 194–201.
- [37] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, "Energy consumption of on-device machine learning models for IoT intrusion detection," *Internet Things*, vol. 21, 2023, Art. no. 100670.
- [38] P. Xiao, Z. Ni, D. Liu, and Z. Hu, "A power and thermal-aware virtual machine management framework based on machine learning," *Cluster Comput.*, vol. 24, pp. 2231–2248, 2021.
- [39] D. Zhao and J. Zhou, "An energy and carbon-aware algorithm for renewable energy usage maximization in distributed cloud data centers," *J. Parallel Distrib. Comput.*, vol. 165, pp. 156–166, 2022.
- [40] D. Zhao, J. Zhou, and S. Yu, "Elvmc: A predictive energy-aware algorithm for virtual machine consolidation in cloud computing," in *Proc. Int. Conf. Algorithms Architectures Parallel Process.*, Springer Int. Publishing, Cham, 2020, pp. 62–81.
- [41] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: Format schema," *Google Inc White Paper*, vol. 1, pp. 1–4, 2011.
- [42] X. Lu, D. Jiang, G. He, and H. Yu, "Greenbdt: Renewable-aware scheduling of bulk data transfers for GEO-distributed sustainable datacenters," *Sustain. Comput.: Inform. Syst.*, vol. 20, pp. 120–129, 2018.
- [43] D. V. Le, R. Wang, Y. Liu, R. Tan, Y.-W. Wong, and Y. Wen, "Deep reinforcement learning for tropical air free-cooled data center control," *ACM Trans. Sensor Netw.*, vol. 17, no. 3, pp. 1–28, 2021.



**Jiantao Zhou** received the PhD degree from the Tsinghua University, in 2005. Since 1999, she has been on the faculty with Inner Mongolia University, China, where she is a professor now. Her research interests include formal methods, cloud computing and software engineering.



**Jidong Zhai** (Senior Member, IEEE) received the BS degree in computer science from the University of Electronic Science and Technology of China, Chengdu, in 2003, and the PhD degree in computer science from Tsinghua University, Beijing, in 2010. He is a tenured professor with the Department of Computer Science and Technology, Tsinghua University, Beijing. His research interests include performance evaluation for high-performance computers, performance analysis, and modeling of parallel applications.



**Keqin Li** (Fellow, IEEE) received the BS degree in computer science from Tsinghua University, in 1985 and the PhD degree in computer science from the University of Houston, in 1990. He is a SUNY Distinguished professor with the State University of New York and a National Distinguished Professor with Hunan University (China). He has authored or co-authored more than 1060 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCCom-2022. He holds nearly 75 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for over twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department at the University of Houston, in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing, in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing, in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic), in 2023. He is a Member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, and an ACIS Founding Fellow. He is an academician member of the International Artificial Intelligence Industry Alliance. He is a Member of Academia Europaea (Academician of the Academy of Europe).



**Daming Zhao** received the PhD degree from Inner Mongolia University, Hohhot, China, in 2023. He is a postdoctoral researcher with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include energy-aware for cloud computing, resource scheduling, and deep reinforcement learning.