CFWS: DRL-Based Framework for Energy Cost and Carbon Footprint Optimization in Cloud Data Centers

Daming Zhao, Jian-tao Zhou, and Keqin Li, *Fellow, IEEE*

***Abstract*—The rapid growth and widespread adoption of cloud computing have led to significant electricity costs and environmental impacts. Traditional approaches that rely on static utilization thresholds are ineffective in dynamic cloud environments, and simply consolidating virtual machines (VMs) to minimize energy costs does not necessarily result in the lowest carbon footprints. In this paper, a deep reinforcement learning (DRL) based framework called CFWS is proposed to enhance the energy efficiency of renewable energy sources (RES) supplied data centers (DCs). CFWS incorporates an adaptive thresholds adjustment method TCN-MAD by evaluating the predicted probability of a physical machine (PM) being overloaded to prevent unnecessary VM migrations and mitigate service level agreement (SLA) violations due to imbalanced workload distribution. Additionally, CFWS introduces a novel action space in the DRL algorithm by representing VM migrations among geo-distributed cloud data centers as flattened indices to accelerate its execution efficiency. Simulation results demonstrate that CFWS can achieve a superior optimization of energy costs and carbon footprints, saving 5.67% to 13.22% brown energy with maximized RES utilization. Furthermore, CFWS reduces VM migrations by up to 86.53% and maintains the lowest SLA violations within suboptimal execution time in comparison to the state-of-art algorithms.**

***Index Terms*—Carbon emission, cloud data centers, energy cost, renewable energy, resource allocation, workload shifting.**

I. Introduction

THE widespread application of cloud computing technology promotes the scale and number of data centers (DCs), resulting in the energy consumption problem has become increasingly prominent. According to the Energy Information Administrator (EIA) report [1], global data centers are expected to consume 95 TWh energy by 2040, which is twice as high as in 2020. The impact of high energy consumption is two-fold. On one hand, data center operators have to pay extra millions of dollars a year due to the dramatic increment in energy consumption. On the other hand, huge energy consumption will cause negative environmental impacts. A report by McKinsey [2] highlights that cloud data centers contributed to 3.5% of the world's CO_2 emissions in 2018, and this figure is projected to increase to 14% by 2040. Therefore, carbon emission optimization is also worthy of attention.

In current studies, improving resource utilization through workload shifting is viewed as an efficient approach to relieve the high energy costs and carbon footprints of data centers. One method for achieving this is through adaptive overloaded detection, which utilizes multi-thresholds or regression-based threshold adjustment approaches to reflect the changing workload patterns and prevents service level agreement (SLA) violations [3] by proactive virtual machine (VM) consolidation from potentially overloaded physical machines (PMs). Toward this goal, the over-utilization resource can be migrated to a limited number of active PMs and the remaining PMs can be switched to the inactive mode for saving energy [4]. Although promising, unpredictable workloads and inaccurate thresholds setting will also lead to energy wastage or high SLA violations.

Another effective way to mitigate the energy crisis and adverse environmental impact is renewable energy sources (RES) based resource scheduling. IT giants such as Apple and Facebook have both achieved carbon neutrality through their solar-supplied data centers [5]. This can be fulfilled through shifting workload to cheaper or cleaner data centers, but electricity prices and carbon footprint rates vary temporally and spatially, leading to a complicated decision process. Even though existing heuristic cost and carbon-aware algorithms attempt to fulfill their objectives by maximizing RES utilization [6], they have to introduce so many computationally and dynamically intractable hyperparameters. Deep reinforcement learning (DRL) is increasingly recognized as a critical component for developing self-sufficient resource management algorithms in such changeable cloud environments [7], which can dynamically adjust agents' behaviors based on environmental conditions and optimize resource allocation. However, migrating VMs among geo-distributed typically needs to traverse all data centers and PMs for determining the consolidation scheme, which is difficult to learn and represent an accurate value function or policy in high-dimensional spaces, resulting in scalability and responsiveness challenges.

In this paper, a novel DRL-based framework, named CFWS, is proposed for achieving a trade-off between energy cost and carbon footprint through workload shifting. CFWS could

Manuscript received 12 February 2023; revised 15 April 2024; accepted 17 April 2024. Date of publication 22 April 2024; date of current version 5 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, in part by the Science and Technology Plan Special Project of Hohhot. Recommended for acceptance by A. Louri. *(Corresponding author: Jian-tao 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: 1477166037@qq.com).

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

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/TSUSC.2024.3391791

periodically adjust the upper threshold to detect overloaded PMs for minimizing performance degradation and then design a DRL algorithm to perform VM migration for improving energy efficiency. The followings are the paper's main contributions:

- Propose a multi-objective workload shifting framework CFWS where an intelligent DRL-based VM migration is implemented with the consideration of time-variability electricity prices and spatial-variability carbon footprint rates (CFRs) among geo-distributed cloud data centers to relieving energy costs and carbon footprints by maximizing RES utilization.
- Propose an adaptive PM overloaded detection algorithm TCN-MAD where the combined use of the temporal convolutional network (TCN) and median absolute deviation (MAD) enhances the threshold adjustment process by considering both the temporal characteristics and the distribution of the workload, avoiding unnecessary migrations and severe SLA violations.
- Propose a DRL-based VM migration method where a flattened index is introduced in DRL's action space to simplify the representation of possible migration actions by assigning a unique index to each potential destination for obtaining cost and carbon-aware VM migration schemes, reducing the complexity and computational overhead involved in evaluating migration possibilities in contrast to existing approaches.
- Evaluate the CFWS considering realistic data center configuration in comparison to four state-of-art algorithms. Performance results demonstrate that the proposed algorithm can reduce 5.67% to 13.22% brown energy by maximizing RES utilization. Furthermore, CFWS successfully balances the trade-off between energy cost and carbon footprint, while also minimizing the number of VM migrations by 46.49% to 86.53% and achieving a minimized probability of SLA violations within suboptimal execution time.

The rest of this paper is organized as follows. Section II reviews the related works and their limitations. Section III outlines the system model. Section IV details the proposed workload shifting framework CFWS. Section V summarizes the findings of the simulation and compares them to the state-of-art approaches. At last, Section VI concludes this paper and depicts its future research plans.

II. Related Work

Workload shifting through VM consolidation is viewed as a promising method to save energy costs and relieve carbon emissions. In this section, the previous research works are categorized into three subsections including adaptive overloaded detection, RES-based resource scheduling and DRL-based workload shifting.

A. Adaptive Overloaded Detection

Many existing researches have focused on multiple threshold-based overloaded detection methods to adapt to the varying workload patterns for energy optimization. Arshad et al. [6] presented a dual-threshold approach to classify hosts into three primary categories by interquartile range, which can effectively capture and analyze different levels of host utilization for improved energy-related management. Zhang et al. [8] proposed an enhanced adaptive threshold classification principle using the least median square regression technique, which enables resource migration among four distinct groups for optimal SLA compliance and energy utilization. However, these reactive methods fail to consider the recent workload trend. Consequently, PMs with unstable requests need to reserve a large number of resources for an extended period, which is not conducive to developing energy-efficient management strategies.

In this regard, regression-based approaches apply statistical analysis techniques to adjust the utilization thresholds accordingly. Yadav et al. [9] introduced the stochastic gradient descent method for effectively detecting overloaded hosts, while also designing an energy-aware VM selection policy based on predicted minimum utilization. Chen et al. [10] provided a proactive adjustment for the upper CPU utilization, employing a statistical measure of dispersion that assigns higher weights to values with larger deviations from the median. Rawas et al. [11] proposed a location-aware VM consolidation approach (LECC) for geo-distributed cloud DCs, which evaluates several overloaded detection methods in advance and then selected the minimum carbon and cost data center for migrating VMs. Nevertheless, the aforementioned methods may struggle to accurately predict requests with large variations that exhibit significant noises in the data, which leads to undesired VM migrations and SLA violations.

B. RES-Based Resource Scheduling

In response to the rising energy costs and carbon footprints for increased computation power, geographically distributed data centers supplied with renewable energy sources have become increasingly prevalent. Nadalizadeh et al. [12] presented a renewable-aware geographical load balancing algorithm GreenPacker that considers the RES availability and varied electricity prices to process cost-awareness resource scheduling. Xu et al. [13] designed an innovative workload management strategy that addresses the challenge of carbon emissions by prioritizing cloud DCs with sufficient RES or lower CFRs in multi-cloud environments. However, it is important to note that optimizing both objectives simultaneously often leads to a conflict, as data centers with cheaper electricity prices may suffer from higher carbon footprints, thereby invalidating cost-aware algorithms.

Alternatively, other researches are concentrating on designing methods to coordinate these two objectives. Renugadevi et al. [14] proposed an optimization function that considers electricity and carbon costs under task deadline constraints, which incorporates the concept of application brownout and batch task delays to maximize RES utilization. Hu et al. [15] presented a two-stage method to address the energy variances caused by geo-distributed RES generators, which quantifies the greenness of each energy source using the average carbon emission rate and establishes a distribution power model to minimize total energy costs. Nevertheless, the aforementioned methods may

TABLE I
OPTIMIZATION OBJECTIVES OF WORKLOAD SHIFTING ALGORITHMS

Reference	Energy	Cost	Carbon	Adaptive THR	Proactive THR	RES	DRL Method
[6]	✓			✓			
[8]	✓			✓			
[9]	✓			✓	✓		
[10]	✓			✓	✓		
[11]	✓	✓	✓	✓	✓		
[12]	✓	✓				✓	
[13]	✓		✓			✓	
[14]	✓	✓	✓			✓	
[15]	✓	✓	✓			✓	
[16]	✓						✓
[17]	✓						✓
[18]	✓	✓				✓	✓
[19]	✓	✓	✓			✓	✓
This paper	✓	✓	✓	✓	✓	✓	✓

not be well-suited for handling changing workload patterns, resource availability and system dynamics, potentially leading to unnecessary migrations.

C. DRL-Based Workload Shifting

DRL-based workload shifting technology has gained significant attention in recent years for improving energy efficiency, as it allows an agent to learn and optimize its actions without prior knowledge in dynamic environments. Zeng et al. [16] developed a DRL-based VM consolidation method, which introduces an Influence Coefficient to evaluate the impact of each VM on overloaded hosts and combines a LSTM-based state prediction model to identity suitable hosts for energy-efficient VM migration. Shaw et al. [17] proposed a combined variable action space that considers both PM utilization and VM size for preventing exhaustive VM consolidation searches, which is guided by a reward shaping technique to accelerate the well-known SARSA and Q-Learning process for a greater energy gains. However, these methods are all aimed at single cloud DC scenarios and do not consider the influence of RES, resulting in the unpredictable costs and inevitable carbon footprints.

However, the application of DRL in RES-based data centers has been relatively limited. Xu et al. [18] proposed an RL-based job scheduling algorithm that incorporated two techniques into the neural network to improve learning performance. Their approach also took into account the characteristics of RES generation to significantly reduce electricity costs associated with brown energy. Wang et al. [19] developed an energy quota planning scheme for situations where there is a shortage of RES. They simplified the cost calculation process by using a multiple agents-based DRL reward function to represent the monetary cost and carbon emissions of each RES generator. As a result, this approach successfully minimized SLA violations and demonstrated superior performance. Nevertheless, the aforementioned methods will face challenges with exhaustive searching, resulting in limited scalability of action spaces.

Table I presents a summary of relevant studies. The proposed approach is unique in its proactive adjustment of the upper threshold (THR) for energy-aware VM consolidation, while utilizing DRL technology to optimize carbon emissions in multi-electricity RES-powered geo-distributed data centers.

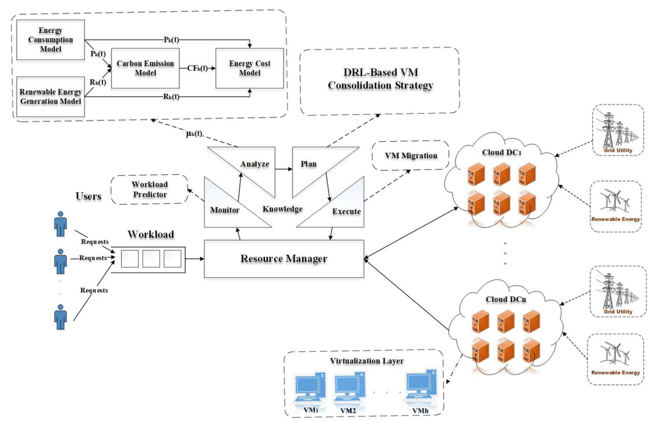

Fig. 1. Architecture of the data center powered by both RES and traditional energy.

This novel combination of adaptive threshold adjustment and DRL is a notable contribution to the field.

III. SYSTEM MODEL

In this section, a typical Infrastructure as a Service (IaaS) cloud system is considered where wind and traditional energy are used to supply n geo-distributed DCs, as shown in Fig. 1. The incoming workload is formed as VMs and delivered to servers among geographically distributed DCs. In the practical implementation, the proposed CFWS architecture follows the principle of MAPE-K, which is the abbreviation of monitor, analyze, plan, execute, and knowledge. The resource monitoring system of the cloud data center can be viewed as a monitor that collects users' requests and continuously evaluates the status of various servers according to the workload predictor in real-time. Once resource utilization of DCs is collected, the analyze module will identify patterns and trends to understand the current DCs' states through four mathematical models. The energy consumption model calculates the power consumption of each DC and conveys them to the carbon emission model. The renewable energy generation model calculates the wind power of each DC and conveys them to the carbon emission model. Then, the output of the carbon emission model, together with the output of the energy consumption model and renewable energy generation model, will be used to calculate the energy cost. Based on the analysis results, the plan module will generate VM migration strategies, which involves forecasting future resource demands and identifying potential overload by the proposed TCN-MAD workload predictor, and developing a DRL-based VM consolidator to address these challenges proactively. After that, the execution module migrates VMs according to the identified optimal strategies. At last, the pre-defined objectives (such as energy cost, carbon footprint) and the aforementioned models will be recorded in the knowledge module to improve the efficiency of VM migration across cloud data centers. In this section, details of the monitor module and analyze module will be introduced.

A. Workload Model

For cloud service providers, establishing cloud data centers in various regions is feasible to offer services to users.

Definition 1: Let D be the set of n geo-distributed cloud data centers, which can be expressed as

$$D = \{D_1, D_2, \ldots, D_k, \ldots D_n\}, \tag{1}$$

where each cloud data center is considered to be powered by traditional energy and renewable energy.

These data centers run multiple PMs, which are interconnected through high-speed network to collectively provide resources to cloud users.

Definition 2: Let S_k be the set of m heterogeneous physical servers running in the kth data center, which can be defined as

$$S_k = \{S_{1^\sim k}, S_{2^\sim k}, \ldots, S_{jk}, \ldots S_{mk}\}, \tag{2}$$

where S_{jk} is the jth PM in DC k, and its CPU utilization at time t can be depicted as $U_{jk}^{PM}(t)$.

In each time slot $t\epsilon\{1, 2, \ldots T\}$, the incoming user requests are viewed as instances and executed by h VMs.

Definition 3: Let VM_{jk} be the set of h VMs hosted on the jth PM of the kth data center, which can be formulated as

$$VM_{jk} = \{VM_{1jk}, VM_{2jk}, \ldots, VM_{ijk}, \ldots, VM_{hjk}\}, \tag{3}$$

where VM_{ijk} is the ith VM of the jth PM in DC k, and its CPU utilization at time t can be depicted as $U_{ijk}^{VM}(t)$.

Accordingly, for the jth PM in DC k, its CPU utilization $U_{jk}^{PM}(t)$ can be calculated as

$$U_{jk}^{PM}(t) = \sum_{i\epsilon VM_{jk}} U_{ijk}^{VM}(t) \times x_{ijk}(t), \tag{4}$$

where $x_{ijk}(t)$ is a binary integer that represents whether VM i is assigned to PM j of DC k (1) or not (0).

In this paper, the kth DC's CPU utilization at time t is expressed as the average CPU utilization of its hosted PMs as

$$\mu_k(t) = \frac{\sum_{j\in S_k} U_{jk}^{PM}\{t\}}{m}. \tag{5}$$

B. Energy Consumption Model

Since the energy spent on cooling needs a fine-grained model, both supplied cooling temperature and inlet temperature will decide cooling costs, which is regarded as a separate study. Therefore, the simplified energy consumption model introduces PUE to incorporate cooling energy consumption.

TABLE II
WATTS DECIDED BY THE CPU UTILIZATION OF SERVERS

Servers	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

Definition 4: Let $P_k(t)$ be kth DC's power consumption at time t, which is calculated by the product of its IT devices power consumption $P_k^{IT}(t)$ and PUE value PUE_k. $P_k(t)$ can be defined as

$$P_k(t) = PUE_k\left(\mu_k(t), H_k(t)\right) \times P_k^{IT}(t), \tag{6}$$

where the value of the kth DC's PUE changes with utilization and the ambient temperature $H_k(t)$. The representative research [20] calculates the PUE as

$$PUE_k\left(\mu_k(t), H_k(t)\right) = 1 + \frac{0.2 + 0.1\mu_k(t) + 0.01\mu_k(t)H_k(t)}{\mu_k(t)}. \tag{7}$$

Furthermore, the $P_k^{IT}(t)$ in (6) can be calculated by summing up all servers' power consumption in the kth DC, which can be formalized as

$$P_k^{IT}(t) = \sum_{j=1}^{m} P_{jk}^{IT}\left(U_{jk}^{PM}(t)\right). \tag{8}$$

Considering that constructing an accurate PM energy model is quite complicated, the SPECpower benchmark [21] is adopted to evaluate P_{jk}^{IT}, which is decided by the jth server's CPU utilization, as shown in Table II.

C. Renewable Energy Generation Model

For RES-based DCs, the availability of RES is critical. Considering a data center is powered by wind energy, the feasibility of which depends on two general aspects. One is whether the location of the DC has sufficient wind speed to drive the wind turbine to generate clean energy, and the other is whether the DC has built enough on-site wind turbines upfront to meet the energy demand.

Definition 5: Let $RES_k(t)$ be the generated renewable energy at time t, which is decided by the actual wind speed $v_k(t)$ of the kth data center and the number of installed wind turbines M_k. The wind power can be defined as

$$RES_k(t) = Wind\left(v_k(t)\right) \times M_k \tag{9}$$

$$Wind\left(v_k(t)\right) = \begin{cases} 0 & v_k(t) < v_{in}, v_k(t) > v_{out} \\ P_r \times \frac{v_k(t) - v_{in}}{v_r - v_{in}} & v_{in} < v_k(t) < v_r \\ P_r & v_r < v_k(t) < v_{out} \end{cases}, \tag{10}$$

where $Wind(v_k(t))$ is the generated energy of a wind turbine. It can be also found that when $v_k(t)$ is lower than the cut-in speed v_{in} or higher than the cut-out speed v_{out}, the output power is set to 0. The wind power will increase linearly when wind speed stays within the cut-in and rated thresholds, otherwise resulting in rated output.

D. Carbon Emission Model

The coal-based energy, known as brown energy, will emit carbon footprint into the environment. Although RES is assumed to generate 0 carbon emission [22], this paper follows the settings in [23], [24], considering a more realistic scenario where the CFR value of wind is treated as a constant.

Definition 6: Let $P_k^b(t)$ and CFR_k be the brown energy consumption and CFR of the kth data center respectively, both of which jointly determine the carbon footprint $CF_k(t)$ at time t, and can be defined as

$$CF_k(t) = \sum_{k=1}^{n} P_k^b(t) \times CFR_k + RES_k(t) \times CFR^{wind}, \tag{11}$$

where CFR^{wind} is the CFR of wind energy. $P_k^b(t)$ is affected by RES and can be calculated as

$$P_k^b(t) = \max\left(0, P_k(t) - RES_k(t)\right). \tag{12}$$

E. Energy Cost Model

For cloud service providers, they have to afford costs associated with the negative environmental impact of carbon emissions and electricity expenses of the traditional grid due to insufficient renewable energy to meet their energy consumption demands.

Definition 7: Let $Cost_k(t)$ be the energy cost of the kth data center at time t, which mainly comes from the purchased traditional grid due to insufficient renewable energy $Cost_k^{grid}(t)$ and the carbon emission cost $Cost_k^{Carbon}(t)$. $Cost_k(t)$ can be defined as

$$Cost_k(t) = \sum_{k=1}^{n} \left(Cost_k^{grid}(t) + Cost_k^{Carbon}(t)\right), \tag{13}$$

where $Cost_k^{grid}(t)$ and $Cost_k^{Carbon}(t)$ are determined by the electricity price $Price_k(t)$, carbon emission price $Price^{carbon}$, and CFR.

For the $Cost_k^{grid}(t)$, a pricing method in real-time is adopted, offering temporal-varied electricity prices for geographically distributed DCs. At time t, the kth DC' energy cost then can be calculated as

$$Cost_k^{grid}(t) = P_k^b(t) \times Price_k(t). \tag{14}$$

For the $Cost_k^{Carbon}(t)$, the carbon emission price is assumed to be a constant, and thus the carbon cost can be calculated as

$$Cost_k^{Carbon}(t) = CF_k(t) \times Price^{carbon}. \tag{15}$$

F. Objective Function

Given the status of PMs, the intuitions of VM migrations among geographically distributed DCs are lowering electricity bills by consolidating more VMs into DCs with cheaper electricity prices and reducing the consumed brown energy by maximizing RES utilization, which will eventually fulfill the minimized energy cost and carbon footprint while satisfying PMs' upper threshold constraints μ_k^H. Moreover, each VM needs to be assigned to a PM as shown in (19). Therefore, the optimization objectives can be defined as follows:

Minimize

$$CF = \int_1^T \sum_{k=1}^{n} CF_k(t)dt \tag{16}$$

$$Cost = \int_1^T \sum_{k=1}^{n} Cost_k(t)dt, \tag{17}$$

subjected to

$$U_{jk}^{CPU}(t) \leq \mu_k^H \tag{18}$$

$$\sum_{j \epsilon S} \sum_{k \epsilon D} x_{ijk}(t) = 1. \tag{19}$$

Algorithm 1: VMC Procedure of the CFWS Framework.

```
   Input: The placement of VMs situated in PMs among geographically
          distributed DCs
   Output: VM consolidation strategy
1  Obtain realistic electricity prices
2  Obtain realistic CFRs
3  for t = 1, T do
4      PM_Status ← Collect PMs' resource utilization information from
         Monitor module
5      Overloaded_PM_List ← TCN-MAD based Workload Predictor
         (PM_Status)
6      Underloaded_PM_List ← Default_Threshold
7      PM_List ← Overloaded_PM_List ∪ Underloaded_PM_List
8      VM_List ← VMs hosted on PM_List
9      for each VM in VM_List do
10         Migration_Map ← DRL based VMC Algorithm (VM_List)
11         Allocate VM to destination PMs based on Migration_Map
12         Store the Migration_Map and system status to knowledge
             base
13     end
14     Update PMs and VMs information
15 end
16 return VM consolidation strategy
```

IV. The Proposed Framework for Workload Shifting

In this section, the proposed CFWS framework is introduced to devise an adaptive overloaded host detection strategy and a DRL-based VM consolidation algorithm to improve the energy efficiency of RES-supplied cloud DCs.

A. CFWS Framework

The proposed CFWS framework aims to achieve an optimization between energy cost and carbon emissions by using the proposed TCN-MAD method to detect overloaded PMs and a DRL-based VM consolidator to perform the optimal VM-PM mapping accordingly. Algorithm 1 outlines the procedure for VM consolidation within the proposed CFWS framework. First, the workload predictor utilizes the designed TCN-MAD method to detect overloaded PMs, and underloaded PMs are identified by a predefined static threshold, forming the source PM_List (Lines 4–7). Subsequently, the VM consolidator establishes a sequential decision model for finding the most suitable PM and achieving the best mapping for each VM in VM_List (Lines 8–13). The model is solved by the DRL-based VM consolidator (Line 10), ensuring that data centers consume the minimum cost and carbon emissions. The detailed process of which will be introduced in Algorithm 2. Afterward, the execute module (Line 11) will migrate VMs (Line 8) associated with the source PMs (Line 7) according to the VM consolidation strategy. Finally, the entire VM consolidation procedure will be stored in the knowledge base of the MAPE-K loop for future scheduling (Line 12), and PMs' status and VMs' allocation on each PM will be updated (Line 14). For the rest time, the above process will be repeated until there are no overloaded or underloaded PMs. In general, the complexity of Algorithm 1 is $O(R \times n \times m)$, where R represents the number of VMs running on the identified source PM. In fact, $n \times m$ is a two-dimensional array that indicates the distribution of PMs in geo-distributed data centers, consuming significant computation resources. To this regard, this paper proposes a novel flattened index to transform the array into a one-dimensional array, which will be discussed in Section IV-C2.

B. Workload Predictor

The workloads of PMs change dynamically over time necessitating the use of time series prediction models. In this context, a workload predictor named TCN-MAD is proposed. TCN is a deep learning model that specializes in capturing temporal dependencies in sequential data. It utilizes convolutional layers with dilated convolutions to capture patterns and dependencies across different time steps. By applying TCN to historical CPU information of hosts, it can effectively learn the patterns and trends that indicate an overloaded state. MAD, on the other hand, is a statistical technique used to identify deviations from the average performance, allowing for the real-time detection of whether a host is operating within normal bounds or experiencing overload. This innovative combination of adaptive overloaded host detection considers the workload fluctuations and changes in resource demands, which avoids unnecessary VM migrations and reduces SLA violations.

In particular, at time $t+1$, the predicted CPU utilization of PMs allocated on the kth DC can be obtained by TCN method as: $\beta_k^{t+1} = \{\beta_{1\tilde{}k}^{t+1}, \beta_{2\tilde{}k}^{t+1}, \ldots, \beta_{jk}^{t+1}, \ldots, \beta_{mk}^{t+1}\}$, which is used as the input of MAD. The upper threshold of the PMs allocated in the kth DC can be defined by

$$\mu_k^H = 1 - s \times MAD, \tag{20}$$

where the value of s is 2.5 through experimental results mentioned in [25]. If the host CPU usage exceeds the threshold, it is deemed as overloaded and some of the VMs will be migrated out. On the contrary, VMs belonging to underloaded hosts should be migrated out altogether. The study by Beloglazov et al. [26] shows that DCs can obtain excellent performance when the lower threshold is 40%, regarded as $\mu_k^L = 40\%$. Therefore, hosts with CPU utilization greater than μ_k^H or less than μ_k^L are regarded as source PMs. The combination of them can effectively avoid unnecessary migrations while guaranteeing the SLA satisfaction, as confirmed by the experiments in the next section.

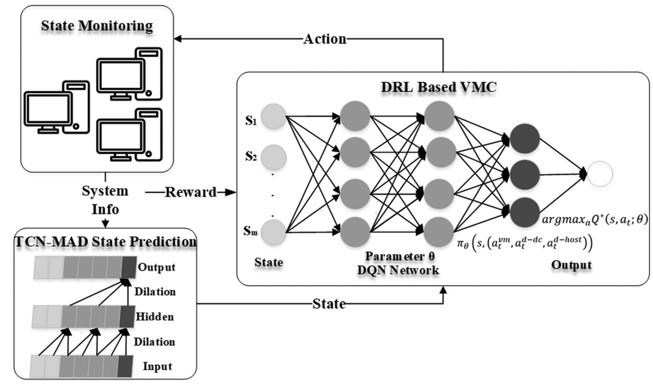

Fig. 2. The process of the DRL-based VM consolidaton.

C. DRL-Based VM Consolidator

DRL-based VM consolidator, as an indispensable part of CFWS, is responsible for interacting with the cloud environment and making energy-aware decisions on when and where to migrate VMs. This can be formulated as a Markov decision problem and the application of DRL methods has been proven to provide an optimal solution. Compared with traditional methods, the DRL agent can adjust its policy accordingly for better responsiveness to varying conditions, which is particularly advantageous in dynamic cloud environments. However, the VM consolidation problem poses challenges due to the high dimensionality of the action space, which involves selecting a target data center and a destination host for VM migration in a geo-distributed environment. The curse of dimensionality makes it computationally demanding and time-consuming for the DRL agent to explore and evaluate all possible actions. In view of this, a novel DRL-based VM consolidation method is proposed in this paper, which utilizes the flattened index to represent the action space for addressing these problems. The subsequent subsections will provide further details on this approach.

1) DRL for Energy-Efficient VM Consolidation: In this paper, the basic idea of using DRL to solve the VM consolidation problem is to employ a neural network-based agent to learn an optimal policy. The agent interacts with the environment, which represents the RES-powered cloud DCs, and takes actions (such as migrating VMs) based on the observed states (e.g., CPU utilization, workload distribution) to maximize a long-term cumulative reward (e.g., energy cost, carbon emission). The agent learns the optimal actions for different states, leading to energy-efficient VM consolidation decisions.

The proposed DRL-based VM consolidator encompasses three key steps, as depicted in Fig. 2: (1) Collecting the current system state and VM migration histories from data centers; (2) Training a computational model for predicting future system states using the provided TCN-MAD model (Section IV-B); (3) Integrating DRL with hash mapping, where the main and target neural networks utilize the flatten index technology to map input states to a set of actions, including the migrated VM, target data center, and destination host. Each output node eventually outputs the maximum Q-value, which is then commissioned to actuate in the data centers upon training completion. Subsequently, the environment receives reward feedback and turns to a new state. The detailed definitions of them are as followings:

1) State space: The VM migration problem is represented as a Markov process for sequential decision-making issues while facing unfavorable uncertainty. s_k^t represents the state space of the kth data center at time t, which is formulated as a CPU utilization tuple

$$s_k^t = <\beta_{1k}^t, \beta_{2k}^t, \ldots, \beta_{jk}^t, \ldots, \beta_{mk}^t>, \tag{21}$$

where β_{jk}^t is the CPU utilization of the jth host of DC k. Here, CPU utilization is used to represent state space for two main reasons. First, as mentioned in Section III-B, the CPU is the primary contributor to energy consumption. Second, when the VM consolidation happens, the CPU capacity constraint is used to evaluate the feasibility of the migration strategy to meet the requirements of the destination host, as demonstrated by (15).

2) Reward: The reward $r(t)$ aims to capture the long-term rewards rather than immediate rewards, providing insights into the operational state and efficiency of the VM migration policy at time t. As lower electricity costs do not necessarily correspond to lower carbon footprints, a linear weighting function is designed to represent the reward function for each state. The objective is to achieve lower energy costs and carbon emissions as follows:

$$\begin{aligned} r(t) = \beta_1 \times \left(\sum_{k=1}^{n} Cost_k(t-1) - \sum_{k=1}^{n} Cost_k(t) \right) \\ + \beta_2 \times \left(\sum_{k=1}^{n} CF_k(t-1) - \sum_{k=1}^{n} CF_k(t) \right), \end{aligned} \tag{22}$$

where $Cost_k(t-1)$ and $CF_k(t-1)$ are the energy cost and carbon footprint at time $t-1$. In order to eliminate the differences in data, both of them will be scaled to the same order of magnitude. Moreover, the coefficients are the positively weighted values that sum up to 1. Maximizing $r(t)$ will be viewed as the reference for subsequent VM migrations, hence ensuring that the energy cost and carbon footprint are reduced compared to the previous time. The objective of training the DQN network is to maximize the expected reward to determine the best VM migration. However, if a PM's utilization $U_{jk}^{CPU}(t)$ exceeds the upper threshold μ_k^H, which is not able to accommodate any VMs, the expected reward $r(t)$ is set to negative infinity.

2) Flatten-Based Action Space for DRL: As the most important component of DRL, the action space of this paper is defined as the VM migration schemes to change overloaded or underloaded PMs' status. Considering that migrating VMs to suitable PMs among geo-distributed cloud DCs introduces a $n \times m$ two-dimensional action space and it is advised to explore as many VM migration schemes as possible by interacting with the DRL environment. This paper provides an innovative hash-map based action space, which reduces the complexity of the action space by transforming it into a one-dimensional representation, known as a flattened index. This approach facilitates the identification of migrated VMs belonging to the source PM and enables the selection of appropriate destination PM and DC.

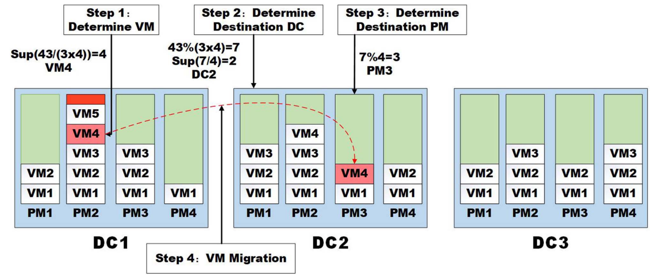

Fig. 3. Example of VM migration.

Consequently, the action space A at time t can be expressed as

$$a_t = [1, R \times (n \times m)] \xrightarrow[map]{hash} a_t = \left\{ a_t^{vm}, a_t^{d-DC}, a_t^{d-host} \right\}, \tag{23}$$

where the multiple of n and m has been introduced at the beginning of the previous section. By multiplying these values together, a flattened index is generated, which accommodates all possible actions and corresponds to the range of randomly selected VMs. To effectively represent the actions, a hash map is employed to convert the flattened action index into an action set denoted as $\{a_t^{vm}, a_t^{d-DC}, a_t^{d-host}\}$. Fig. 3 depicts an example to illustrate such process. It is assumed that there are 3 data centers, each containing 4 PMs, and the second PM of DC1 is overloaded, which includes 5 VMs and some of which need to be migrated out. As a result, the scale of the action space will be $5 \times 3 \times 4$. If the flattened index randomly outputs the value 43, it will be further mapped to determine the specific migrated VM and the destination PM. The following steps outline this process in detail:

Step 1: Determine the migrated VM. In this case, the value 43 corresponds to a three-dimensional array that is the flattened index of the migrated VMs. To be specific, 43 is divided by the total number of PMs in all data centers (3×4). Since all indexes in this example start with 1, the quotient indicates that VM4 will be selected, a_t^{vm} is 4.

$$\begin{array}{c c} & \begin{array}{cccc} PM1 & PM2 & PM3 & PM4 \end{array} \\ \begin{array}{c} DC1 \\ DC2 \\ DC3 \end{array} & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix} \end{array}. \tag{24}$$

Step 2: Determine the destination DC and PM. The value 7, which is the remainder of Step 1, represents a two-dimensional array indicating the target data center and destination PM. To better illustrate this decision process, an array with 3 rows and 4 columns is introduced as (24).

It can be noticed that 7 is located at array [2], [3], which corresponds to the PM3 of DC2. However, this array can be transformed into a one-dimensional array using the flattened index with a length of 12. In other words, array [7] is the hash map of array [2], [3]. Therefore, the ordering and relationship between different actions are preserved with the flattened index, allowing for easier comparison and selection of actions based on their index values. This process can be mathematically explained in the following two substeps:

Step 2(a): Divide the remainder (7) of Step 1 by the number of PMs of each DC (4), and round up the quotient as the index of the target data center. Thus, DC2 is selected, a_t^{d-DC} is 2.

Step 2(b): Determine the destination PM. Particularly, regard the remainder (3) of Step 2(a) as the index of the PM. As a result, PM3 is selected a_t^{d-host} is 3.

Step 3: If the destination PM is not the source PM, then VM4 in PM2 of DC1 will be migrated to PM3 of DC2. Otherwise, return to Step 1. Furthermore, if the destination PM has no spare capacity to accommodate the migrated VM, the reward function will be punished.

It should be noticed that since the initial value of the action index is randomly selected, the above steps will be repeated in $TimeStep$ until a maximum reward value is achieved.

3) DQN-Based VM Consolidation: Algorithm 2 gives the whole VM migration procedure with the DQN method, which tries to obtain the best consolidation strategy as the destination of the migrated VMs on overloaded or underloaded PMs. In particular, Lines 1–3 initialize the network with the initial PM status and return a Q function about the action value. Lines 4–33 are the training procedure of the DQN network. Among them, Lines 7–12 employ ϵ-greedy policy to explore the best VM migration strategy based on a predefined reward function (20), so that high reward can be returned more frequently and the exploration probability ϵ will be decreased accordingly to reduce exploration. During the process, Line 13 executes a flattened index action a_j to determine the migrated VM, destination DC and PM according to the hash map as mentioned in Section IV-C2. Then, observe the next observation s_{j+1} and reward r_j, where r_j is calculated through (20), which is not only affected by the consumed energy consumption and generated wind energy but also decided by the PM's capacity (Lines 14–19). Besides, experience replay in Lines 20–22 is used to evaluate for improving learning process stability by breaking the correlation of learning data and reducing update variance. After that, Lines 29–31 build a target Q network to eliminate divergence use experience replay to learn about their environment and update networks. Finally, the set of actions will be returned as the VM consolidation strategy.

In general, in order to optimize the trade-off between energy costs and carbon emissions in a dynamic cloud environment, the proposed DRL-based VM consolidator first detects the overloaded PM by the proposed TCN-MAD method. Also, the introduction of the flattened index in the algorithm reduces complexity and enhances the effectiveness of VM consolidation during the exploration phase of DRL.

Algorithm 2: DRL-Based VMC Algorithm.

Input: VM_List
Output: Migration actions on VMs
1 Initialize replay memory Δ to capacity ω
2 Initialize action-value function Q with random weights θ
3 Initialize target action-value Q' with weights $\theta' \leftarrow \theta$
4 **for** $episode = 1, Episode$ **do**
5   Reset VM status of source PM to initial state
6   **for** $j = 1, TimeStep$ **do**
7   **if** $\epsilon \geq probability$ **then**
8   choose a random action a_j
9   **end**
10   **else**
11   choose $a_j \leftarrow argmax_a Q(s_j, a; \theta)$
12   **end**
13         $a_j^{vm}, a_j^{d-DC}, a_j^{d-host} \leftarrow$ The flattened index action a_j executes hash map to determine these values as mentioned in Section 4.3.2.
14   **if** *the destination PM's capacity is available* **then**
15             $r_j \leftarrow$ Obtain the reward value calculated by Eq. (20).
16   **end**
17   **else**
18             $r_j \leftarrow$ -∞
19   **end**
20   Update state space $s_{j+1} \leftarrow$s$_j, a_j$
21   Store transition (s_{j+1}, a_j, r_j, s_j) in Δ
22   Sample random minibatch of transitions (s_{j+1}, a_j, r_j, s_j) from Δ
23   **if** *episode terminates at step* $j + 1$ **then**
24             $target_j \leftarrow r_j$
25   **end**
26   **else**
27             $target_j \leftarrow r_j + \gamma max_{a'} Q'(s_{j+1}, a'; \theta')$
28   **end**
29   Perform a gradient descent step on $(target_t - Q(s_j, a_j; \theta))^2$
30   Train evaluation network and decrease ϵ every τ steps
31   Copy Q to $Q^{'}$ every ζ steps
32   **end**
33 **end**
34 **return** All actions a_j

V. Simulation Evaluation

This section considers CFWS in detail, creating the model and fulfilling the algorithm using PyCharm 3.3 running on a 16 GB RAM PC using an Intel(R) Core i7-8750H processor with 2.2 GHz. The scenario under analysis is consistent with the real environment, with four geographically distributed DCs in the United States. Moreover, workload traces from Google and renewable energy traces from National Renewable Energy Laboratory (NREL) [27] are used to build an IaaS cloud environment.

A. Simulation Settings

1) Data Centers Configuration: The effect of environmental fluctuations on renewable energy will be simulated using four US data centers in different locations across Arizona, California, Oregon, and Louisiana. These data centers are strategically chosen to represent diverse climate influences due to their distinct geographical locations and time zones. Additionally, they exhibit variations in electricity prices and CFRs. To be specific, each data center has a varying number of heterogeneous PMs, with half of them being HP ProLiant ML G4s and the other part HP ProLiant ML110 G5s [28]. Table II provides detailed information on the energy consumption of these server types at different workload levels. Since the CPU is the primary energy contributor, the server capacity is expressed in terms of CPU frequency in MIPS. Table III outlines the CPU capacities and the number of cores for PMs and VMs.

2) Workload: Google Cluster Dataset (GCD) [29] is used to simulate the resource requests by estimating the daily active users from various regions. The reason for utilizing GCD is its comprehensive coverage of diverse applications and users worldwide. It includes workload data from a 12,500-machine cluster collected over 29 days in May 2011. In the context of this paper, VM will be migrated to address situations where the

TABLE III
SERVER/VM TYPES AND CAPACITY

Name	CPU	Core
HP ProLiant G4	1860 MIPS	2
HP ProLiant G5	2660 MIPS	2
VM 1	2500 MIPS	1
VM 2	2000 MIPS	1
VM 3	1000 MIPS	1
VM 4	500 MIPS	1

TABLE IV
PARAMETERS OF WIND TURBINE

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

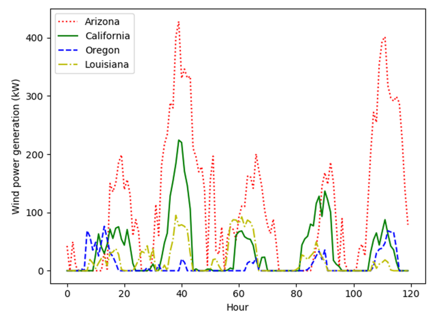

Fig. 4. Wind power generation.

TABLE V
CARBON FOOTPRINT RATE (TONS/MWH)

	Arizona	California	Oregon	Louisiana
CFR	0.658	0.350	0.147	0.690
Servers	3300	2800	3200	2500

available resources of a PM are insufficient to meet the resource requirements of the incoming workload.

3) Wind Energy: The RES traces utilized are sourced from the NREL, offering records of sampled points across the world at 10-minute intervals. This research utilizes such information to perform an analysis over a period lasting from April 1st to 30th 2020; an overall time consisting of 720 hours. In particular, records implying the first 20 days forming 480 hours have been taken as the training set which includes environmental temperature, humidity, wind speed, and time. The next five days make up 120 hours and are used as the test sample, and the last five days are used as the verification samples, allowing for the prediction of wind power generation. In contrast to previous studies that consider both solar and wind energy [28], this paper assumes that NE-3000 wind turbines [30] are installed at each data center to fulfill the electric necessities of cloud DCs. Table IV provides an overview of the wind turbine parameters, and the generated wind power is calculated using (6), as depicted in Fig. 4.

4) Carbon Footprint Rate and PUE: The US Department of Energy Electricity Emission Factors [31] provides the carbon footprint rates of the four data centers, which are presented in Table V. These values are used to calculate the carbon emissions for each data center. Also, the number of servers contained

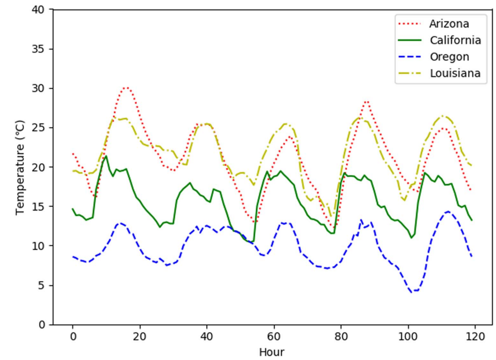

Fig. 5. Temperature variation.

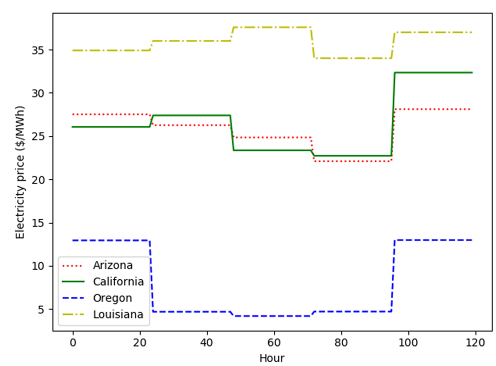

Fig. 6. Electricity price.

within these data centers is presented. Among the four data centers, Oregon emits the least carbon footprint when consuming the same amount of energy. For the CFR of wind, it is assumed to be 0.0225 (Ton/MWh) as mentioned in [23]. For the PUEs of geo-distributed cloud data centers, this paper considers their dynamics as modeled in (7), which integrates the impact of environmental temperature variations as shown in Fig. 5. Such settings are beneficial to provide a more accurate evaluation of data centers' comprehensive energy efficiency across varying CPU utilization levels and temperature conditions.

5) Electricity Price: Electricity prices are provided by the US EIA. Fig. 6 illustrates energy prices over a 5-day period with a 24-hour interval. Moreover, this paper considers cloud DCs are powered by on-site wind energy, which is assumed to have no additional costs since it incurs one-time capital and maintenance expenses regardless of its utilization. The carbon cost of wind is considered to be 20 ($/Ton) [32].

6) Baseline Method: Since there are no existing studies specifically addressing the research problem presented in this paper, the authors have developed four baseline methods by drawing insights from relevant works. These comparison methods serve as baseline methods for evaluating the performance and effectiveness of the proposed approach.

- *Greenpacker:* This method is derived from [12], which focuses on RES-powered data centers. In the proposed baseline method, it traverses geo-distributed data centers and adds hosts with sufficient resources for accommodating the migrated VM to the PM list. However, the baseline

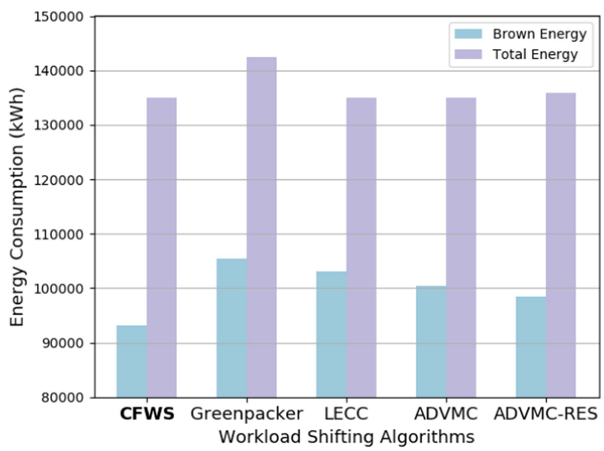

Fig. 7. Energy consumption.

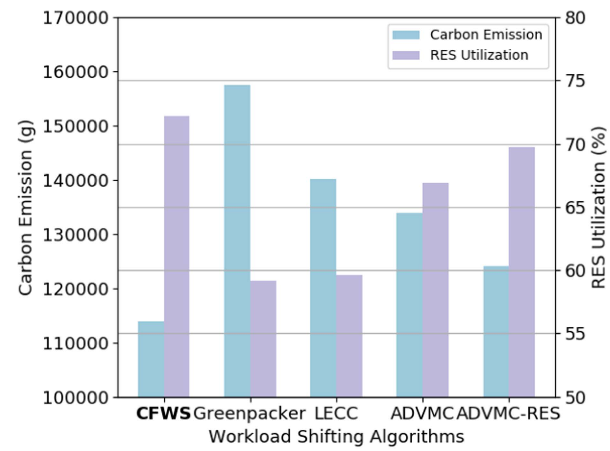

Fig. 8. Carbon emission and RES utilization.

algorithm disregards the fragmentation cost and instead aims to select the PM that minimizes brown energy and energy cost in a greedy manner.

- *LECC:* This method is derived from [11], which heuristically explores the VM migration scheme for minimizing the carbon emission cost. Based on the findings of that study, the proposed baseline method is designed to identify overloaded and underloaded PMs based on MAD and minimum CPU utilization, respectively. Moreover, the VM that has a higher correlation of the CPU utilization is selected to migrate, which is known as maximum correlation. The PM with the least availability is viewed as the destination host.
- *ADVMC:* This method is derived from [16], which develops a DRL-based VM placement strategy and adopts LSTM to predict the system status in a traditionally powered cloud data center. Referring to it, the proposed baseline method sets upper and lower CPU thresholds with 0.9 and 0.2, respectively. Furthermore, the VM selection is performed using the influence coefficient policy, and DRL is employed to migrate the selected VMs in order to minimize the total energy consumption while avoiding SLA violations.
- *ADVMC-RES:* This method is a variant of ADVMC, which expands the experimental setup to include multiple RES-powered data centers. In comparison to ADVMC, this baseline method is designed to prioritize cloud DCs that are abundant in RES instead of energy consumption to optimize carbon emissions.

B. Simulation Results

To further evaluate the proposed CFWS framework, simulations were done over 5 days to investigate the energy consumption, energy cost, carbon emission, RES utilization and the number of migrations of four data centers. Each simulation was executed 30 times using different initial virtual machine placements.

1) Energy Consumption: The energy consumption comparison is illustrated in Fig. 7. Meanwhile, brown energy is introduced because it is a key contributor to carbon emissions. Notably, the proposed algorithm CFWS can significantly reduce 5.67%–13.22% brown energy compared with baseline algorithms while consuming similar total energy. This is because the CFWS optimizes brown energy consumption over extended periods by considering future rewards and long-term workload variations, which also incorporates the TCN method to relieve the gap between RES generation and energy consumption. Among other DRL-based algorithms, ADVMC-RES focuses on migrating VMs to the data center with sufficient RES, and hence achieving less brown energy to ADVMC (98431.77 kWh versus 100363.63 kWh). Among heuristic algorithms, LECC performs better for the reason that it adopts the MAD to dynamically adjust thresholds to improve resource utilization as CFWS does. On the contrary, Greenpacker does not design elaborate PM overloaded identification schemes, which exhibits the highest energy consumption in both metrics.

2) Carbon Emission: Fig. 8 shows the experimental results of carbon emissions, which further introduces the comparative results of RES utilization. The RES utilization indicates the proportion of wind energy utilized in the data center relative to the total generated wind energy. It is evident that CFWS performs best in both metrics, the reason of that can be attributed to two main factors. On the one hand, CFWS achieves the highest RES (72.19%) to trade off environmental impacts of carbon emission (113966.14 g), whereas Greenpacker tops the carbon emission (157566.57 g) with the lowest RES utilization (59.16%). Similarly, ADVMC-RES considers the real-time availability and variability of RES, which also prioritizes the utilization of RES and decreases 9716.04 g carbon emissions compared to ADVMC. On the other hand, CFWS considers the geographic heterogeneity of CFRs during VM migration. This is also reflected in the fact that although the carbon-aware LECC only improves 0.48% RES than Greenpacker, it reduces 17473.75 g carbon emission.

3) Energy Cost: Fig. 9 compares energy costs and carbon costs with baseline algorithms. As expected, the CFWS will pay less costs due to its outstanding performances in brown energy reduction and carbon emission optimization as discussed in prior subsections. In comparison to LECC, which also considers price

TABLE VI
COMPARISON RESULTS ON EVALUATING OVERLOADED PM DETECTION METHODS

Policies	Energy Cost	Carbon Cost	Total Energy	Brown Energy	Carbon Emission	RES	Migrations	SLAV
Greenpacker	9227.49	3151.35	142513.66	105465.11	157566.57	59.16	735	0.0368
LECC	8911.81	2801.86	**135011.69**	103036.95	140092.82	59.64	543	0.0335
ADVMC	8710.77	2676.98	135021.57	100363.63	133849.26	66.89	185	0.0222
ADVMC-RES	8632.23	2482.66	135838.14	98431.77	124133.22	69.73	188	0.0228
TCN-MAD-2.5 (CFWS)	**8100.46**	**2279.32**	135017.99	**93154.23**	**113966.14**	**72.19**	**99**	**0.0174**
LSTM-MAD-2.5	8151.32	2323.58	135021.57	93808.15	116179.11	72.15	101	0.0188
MAD-2.5	8200.19	2364.80	135124.53	94007.66	118240.08	72.16	103	0.0193
TCN-IQR-1.5	8134.94	2290.02	134777.56	95202.80	114500.87	72.16	101	0.0181
LSTM-IQR-1.5	8152.04	2325.88	135024.60	93958.31	116293.79	71.48	115	0.0218
IQR-1.5	8216.52	2430.63	135064.38	94400.87	121531.50	70.88	123	0.0240
TCN-THR-0.8	8440.08	2611.73	135162.40	94523.70	130586.51	70.31	135	0.0256
LSTM-THR-0.8	8437.75	2634.82	135300.22	94804.92	131740.66	70.27	133	0.0256
THR-0.8	8636.49	2775.39	136241.13	96069.58	138769.53	69.83	152	0.0340

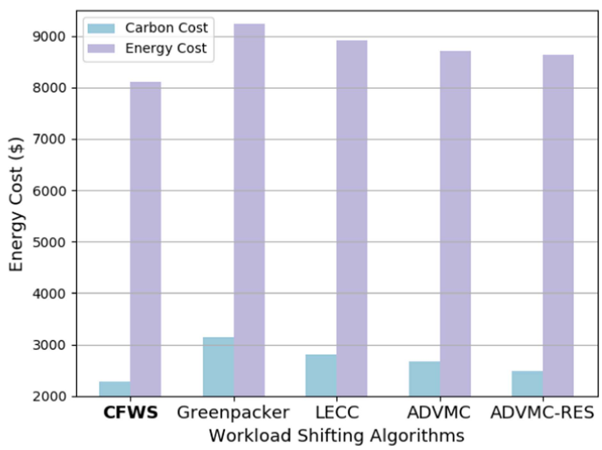

Fig. 9. Energy cost.

variations among geo-distributed data centers, CFWS achieves even greater cost savings by reducing carbon costs by \$522.54 and energy costs by \$811.35. This highlights CFWS's ability to adapt and optimize migration strategies based on the real-time electricity market, leading to significant cost reductions. The results also suggest that the introduction of RES is effective to eliminate brown energy as demonstrated by ADVMC-RES, which leads to the reduction of carbon cost by 7.26% than ADVMC. Furthermore, the Greenpacker causes the most costs in this scenario. It treats electricity prices at all data centers as a constant value and fails to make decisions according to their price differences.

4) Migrations: The last two columns in Table VI illustrate SLA violations and the necessary VM migrations associated with them. SLA violations are defined as the ratio of overloaded PMs that exceed the CPU utilization threshold to the total number of active PMs. Compared to baseline algorithms, CFWS demonstrates a remarkable reduction in VM migrations, ranging from 46.49% to 86.53%, with an average decrement of 36.52% in SLA violations. This achievement can be attributed to the proposed TCN-MAD in CFWS, which proactively estimates unseen overloaded situations in advance to mitigate the need for frequent migrations. Experimental results further highlight the superiority of DRL-based methods (CFWS, ADVMC, ADVMC-RES) over heuristic-based algorithms (Greenpacker, LECC) in terms of reducing VM migrations and minimizing SLA violations. This is because DRL-based methods can continuously update their migration policies based on real-time feedback and adjust their decision-making processes accordingly, whereas heuristic algorithms require extra migrations to adapt to changing conditions.

In addition to the above, comparisons about overloaded PM detection are also recorded in the last 8 rows of Table VI to evaluate the effectiveness of the proposed TCN-MAD on migrations and SLAs. The table presents nine combinations using different overloaded detection algorithms, including TCN-MAD, LSTM-MAD, MAD, IQR, and THR (a static threshold set to 0.8 [33]). The aggressive parameters, denoted as s, are set to 2.5 for MAD and 1.5 for IQR [25]. For the threshold adjustment performance, it can be found that TCN-MAD-2.5 could achieve optimal results in most cases with the least migrations and SLAs. This is because the threshold adjustment method based on MAD will lead to fewer VM migrations (e.g., MAD-2.5 performs better through reducing VM migrations by 16.26% and SLA violations by 19.58% than IQR-1.5). Compared with the static threshold setting (THR-0.8), the proposed TCN-MAD-2.5 avoids 34.87% VM migrations and 48.82% SLA violations. On the other hand, this paper introduces the well-known LSTM method and designs LSTM-MAD-2.5, LSTM-IQR-1.5 and LSTM-THR-0.8 adaptive threshold adjustment method to evaluate the validity of the TCN-based workload prediction. Since TCN has been shown to have better accuracy while predicting the workload variation than LSTM [34], [35], TCN-based methods reduce subsequent migrations to rebalance the workload (eg. TCN-MAD-2.5 reduces 2 VM migrations and 0.14% SLA violations than LSTM-MAD-2.5) and the default static threshold methods perform worst.

5) Execution Time: Fig. 10 depicts the execution time of the proposed algorithm compared to the state-of-the-art approaches, providing insights into the computational overhead of each method. The results demonstrate that CFWS exhibits a slightly higher execution time compared to LECC. This can be attributed to the fact that LECC pre-determines the target data center. Therefore, the destination PM determined by sorted available resources will result in a computational complexity of $O(R \times m \log m)$, whereas CFWS has a complexity of $O(R \times n \times m)$ as discussed in Section IV-A. Despite the higher complexity, CFWS may still be preferred in scenarios where carbon

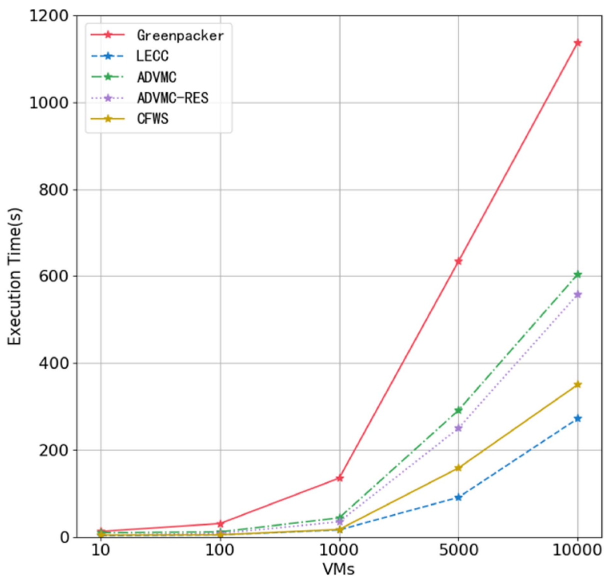

Fig. 10. Execution time.

emissions and energy costs are of primary concern. The advantage of the proposed flatten-based action space is evident when comparing it with variations of traditional DRL-based algorithms such as ADVMC and ADVMC-RES. where the action spaces are designed based on sorted data centers and PMs, leading to execution time with $O(R \times n \log n \times m \log m)$. Among all the scenarios, Greenpacker exhibits the slowest execution time. This is due to its need for two inner for loops and an outer $while$ loop to iterate all available PMs for migrating. Consequently, its complexity is the largest at $O(R \times n \times m + R^2 \times n)$.

VI. Conclusion

In this paper, a DRL-based framework CFWS is proposed to optimize energy costs and reduce carbon footprints via workload shifting for RES-supplied cloud DCs. To be specific, it first provides an adaptive overloaded PM detection method TCN-MAD that helps reduce VM migrations by proactively identifying periods of anticipated resource overload, thus reducing unnecessary migrations and the occurrence of SLA violations. Based on that, a flattened index is introduced to determine the destination of migrated VMs among geo-distributed data centers, which promotes better energy-efficient exploration with the consideration of the temporal and spatial-variability of electricity prices and CFRs to increase the likelihood of obtaining optimal migration strategies. The simulation results demonstrate the superiority of CFWS as compared to the state-of-art algorithms, which achieves the optimal energy cost and carbon emission while requiring fewer migrations and exhibiting lower SLA violations within satisfactory execution time. Additionally, CFWS achieves the highest RES utilization among the compared algorithms, reaching 72.19%.

In the future, the proposed algorithm is expected to be tested in a real cloud infrastructure such as OpenStack or extended in a workload management platform such as Aneka. Additionally, like the existing studies, the proposed CFWS only provides guidelines for optimizing the RES-based cloud data center and demonstrates its feasibility through simulation experiments. Hence, there is also a necessity that the proposed CFWS be practically implemented or validated in modern built sustainable data centers powered by renewable energy. Furthermore, the rest of future work will construct a more realistic carbon emission estimation model that considers the spatial-temporal varied carbon footprint rates of RES. It is also expected to consider the impact of cooling and network transmission on energy consumption to prevent service quality degradation due to insufficient RES supply.

References

[1] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)–Centric resource management in cloud computing: A review and future directions," *J. Netw. Comput. Appl.*, vol. 204, 2022, Art. no. 103405.

[2] S. Rawas, "Energy, network, and application-aware virtual machine placement model in SDN-enabled large scale cloud data centers," *Multimedia Tools Appl.*, vol. 80, no. 10, pp. 15541–15562, 2021.

[3] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Netw.*, vol. 26, pp. 1905–1919, 2020.

[4] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y. C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing," *IEEE Access*, vol. 6, pp. 55923–55936, 2018.

[5] J. Patchell and R. Hayter, "The cloud's fearsome five renewable energy strategies: Coupling sustainable development goals with firm specific advantages," *J. Cleaner Prod.*, vol. 288, 2021, Art. no. 125501.

[6] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, "Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers," *Renewable Sustain. Energy Rev.*, vol. 167, 2022, Art. no. 112782.

[7] 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.

[8] W. Zhang, R. Yadav, Y.-C. Tian, S. K. S. Tyagi, I. A. Elgendy, and O. Kaiwartya, "Two-phase industrial manufacturing service management for energy efficiency of data centers," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 7525–7536, Nov. 2022.

[9] R. Yadav, W. Zhang, K. Li, C. Liu, and A. A. Laghari, "Managing overloaded hosts for energy-efficiency in cloud data centers," *Cluster Comput.*, vol. 24, no. 3, pp. 2001–2015, 2021.

[10] R. Chen, B. Liu, W. Lin, J. Lin, H. Cheng, and K. Li, "Power and thermal-aware virtual machine scheduling optimization in cloud data center," *Future Gener. Comput. Syst.*, vol. 145, pp. 578–589, 2023.

[11] S. Rawas, A. Zekri, and A. El-Zaart, "LECC: Location, energy, carbon and cost-aware VM placement model in geo-distributed DCs," *Sustain. Comput. Informat. Syst.*, vol. 33, 2022, Art. no. 100649.

[12] Z. Nadalizadeh and M. Momtazpour, "GreenPacker: Renewable-and fragmentation-aware VM placement for geographically distributed green data centers," *J. Supercomputing*, vol. 78, no. 1, pp. 1434–1457, 2022.

[13] M. Xu and R. Buyya, "Managing renewable energy and carbon footprint in multi-cloud computing environments," *J. Parallel Distrib. Comput.*, vol. 135, pp. 191–202, 2020.

[14] T. Renugadevi and K. Geetha, "Task aware optimized energy cost and carbon emission-based virtual machine placement in sustainable data centers," *J. Intell. Fuzzy Syst.*, vol. 41, no. 5, pp. 5677–5689, 2021.

[15] X. Hu, P. Li, and Y. Sun, "Minimizing energy cost for green data center by exploring heterogeneous energy resource," *J. Modern Power Syst. Clean Energy*, vol. 9, no. 1, pp. 148–159, 2021.

[16] J. Zeng, D. Ding, X. K. Kang, H. Xie, and Q. Yin, "Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2991–3002, Nov. 2022.

[17] 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, 2022, Art. no. 101722.

[18] C. Xu, K. Wang, P. Li, R. Xia, S. Guo, and M. Guo, "Renewable energy-aware Big Data analytics in geo-distributed data centers with reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 205–215, First Quarter 2020.
[19] 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.
[20] A. Khosravi, L. L. Andrew, and R. Buyya, "Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 183–196, Second Quarter 2017.
[21] Z. Zhou et al., "Fine-grained energy consumption model of servers based on task characteristics in cloud data center," *IEEE Access*, vol. 6, pp. 27080–27090, 2017.
[22] X. Wang, G. Zhang, M. Yang, and L. Zhang, "Green-aware virtual machine migration strategy in sustainable cloud computing environments," *Cloud Comput.-Architecture Appl.*, 2017.
[23] S. Aslam, S. Aslam, H. Herodotou, S. M. Mohsin, and K. Aurangzeb, "Towards energy efficiency and power trading exploiting renewable energy in cloud data centers," in *Proc. Int. Conf. Adv. Emerg. Comput. Technol.*, 2020, pp. 1–6.
[24] J. Gao, H. Wang, and H. Shen, "Smartly handling renewable energy instability in supporting a cloud datacenter," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2020, pp. 769–778.
[25] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012.
[26] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.
[27] X. Lu, D. Jiang, G. He, and H. Yu, "GreenBDT: Renewable-aware scheduling of bulk data transfers for geo-distributed sustainable datacenters," *Sustain. Comput.*, vol. 20, pp. 120–129, 2018.
[28] D. Alsadie, Z. Tari, and E. J. Alzahrani, "Online VM consolidation in cloud environments," in *Proc. IEEE 12th Int. Conf. Cloud Comput.*, 2019, pp. 137–145.
[29] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: Format schema," Google Inc., White Paper, vol. 1, pp. 1–14, 2011.
[30] C. Gu, Z. Li, C. Liu, and H. Huang, "Planning for green cloud data centers using sustainable energy," in *Proc. IEEE Symp. Comput. Commun.*, 2016, pp. 804–809.
[31] S. K. Garg, C. S. Yeo, and R. Buyya, "Green cloud framework for improving carbon efficiency of clouds," in *Proc. Eur. Conf. Parall. Process.*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 491–502.
[32] K. Than, "Estimated social cost of climate change not accurate," *Stanford Scientists Say, Stanford News January*, vol. 12, no. 2015, pp. 1391–1398, 2015.
[33] J. P. B. Mapetu, L. Kong, and Z. Chen, "A dynamic VM consolidation approach based on load balancing using pearson correlation in cloud computing," *J. Supercomputing*, vol. 77, no. 6, pp. 5840–5881, 2021.
[34] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv: 1803.01271*.
[35] E. Golshani and M. Ashtiani, "Proactive auto-scaling for cloud environments using temporal convolutional neural networks," *J. Parallel Distrib. Comput.*, vol. 154, pp. 119–141, 2021.

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.

Jian-tao Zhou received the PhD degree from 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.

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 990 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 CPSCom-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, 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).