SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Hybrid Energy-Efficient Reinforcement And Optimization with Carbon-Intensity-Based Container Scheduling

Ms. A. Janaki Devi[1*], Dr. V J Chakravarthy[2], Dr. R. Shobana[3], Dr. V. Dheepa[4]

[1]Research Scholar Department of Computer Science Dr. M. G. R. Educational and Research Institute Chennai, India.
E-mail: janakidevi08@gmail.com
[2]Professor, Faculty of Computer Applications Dr. M. G. R. Educational and Research Institute Chennai, India.
E-mail: chakravarthy.mca@drmgrdu.ac.in
[3]Associate Professor Department of CSE Aarupadai Veedu Institute of Technology Vinayaka Mission's Research Foundation (DU).
E-mail: shobana@avit.ac.in
[4]Assistant Professor Department of Computer Applications Women's Christian College Chennai, India. E-mail: dheepa@wcc.edu.in
*Corresponding Author: Ms. A. Janaki Devi, E-mail: janakidevi08@gmail.com

Abstract
The rapid growth of containerized cloud applications has significantly increased energy consumption and CO_2 emissions in large-scale data centers, creating major sustainability challenges. This study proposes HERO-CS 2.0 (Hybrid Energy-Efficient Reinforcement and Optimization with Carbon-Intensity-Based Container Scheduling), a scalable carbon-aware scheduling framework that integrates Power-Aware Best Fit Decreasing (PABFD), Dynamic Voltage and Frequency Scaling (DVFS), and Deep Q-Network (DQN)-based reinforcement learning. The framework combines heuristic optimization, hardware-aware power management, adaptive learning, and real-time carbon-intensity-aware scheduling using external energy-monitoring APIs to identify low-carbon computing nodes. Implemented using CloudSim Plus, Kubernetes, Java, and TensorFlow/PyTorch, the proposed model enhances energy efficiency and sustainability in cloud environments. Experimental results demonstrate significant improvements, including energy consumption reduction from 710 kWh to 470 kWh (approximately 33% savings), SLA violation reduction from 4.2% to 2.0%, and execution time improvement from 25.5 seconds to 15.3 seconds. Additionally, carbon intensity impact decreases from 0.81 kg to 0.45 kg. The framework achieves 92% resource allocation accuracy, 95% QoS satisfaction, and 91% host consolidation efficiency. HERO-CS 2.0 is scalable and suitable for deployment in distributed cloud infrastructures, supporting the development of sustainable and green cloud computing environments.

Keywords: Cloud Computing, Energy Efficient, Container, Scheduling, Response Time, Execution Time

1. Introduction

The rapid growth of cloud computing (CC) and containerized applications has significantly increased energy consumption in large-scale data centres, creating major challenges for operational efficiency and environmental sustainability [1]. Containers provide lightweight virtualization, scalability, and faster deployment compared to traditional virtual machines, making them highly suitable for dynamic cloud environments [2]. However, efficient container scheduling remains a critical issue because continuously shifting workloads can lead to excessive energy usage, thermal imbalance, and increased carbon emissions [3]. Most existing scheduling algorithms mainly focus on performance-related objectives such as resource utilization, throughput, response time, and SLA compliance, while environmental factors like energy efficiency and carbon footprint receive limited attention [4]. At the same time, carbon emissions have become a global concern affecting sustainable development across all sectors [5]. Many countries have established carbon neutrality targets and adopted low-carbon strategies within their national development frameworks [6]. Similar concerns are increasingly relevant in cloud data centres, where large-scale computing infrastructures consume substantial electrical power and indirectly contribute to greenhouse gas emissions. Therefore, modern cloud environments require intelligent scheduling approaches that can simultaneously optimize system performance and reduce environmental impact through energy-aware and carbon-aware resource management [7].

Green Cloud Computing (Green CC) has emerged as an important solution for reducing the environmental impact of cloud infrastructures [8]. This approach focuses on minimizing carbon emissions through energy-efficient hardware, renewable energy utilization, and intelligent workload distribution [9]. Task scheduling plays a vital role in Green CC because it directly influences server utilization, power consumption, and operational cost. Although several scheduling methods have been proposed, important limitations still exist. Traditional techniques such as PABFD mainly concentrate on virtual machine consolidation and

reducing server uptime, but do not consider real-time carbon intensity during workload placement [10]. DVFS-based scheduling methods reduce processor power consumption by adjusting CPU frequency and voltage; however, DVFS-based scheduling methods often ignore workload dynamics, thermal conditions, and carbon-aware decision-making [11]. Similarly, many Reinforcement Learning (RL)-based schedulers improve adaptive resource allocation but mainly prioritize performance metrics instead of sustainability objectives. These limitations highlight a significant research gap in container-level carbon-aware scheduling for sustainable cloud environments [12]. Figure 1 represents the architecture of the Green Broker framework for energy-efficient cloud service allocation by selecting cloud resources based on energy efficiency and carbon emission information.

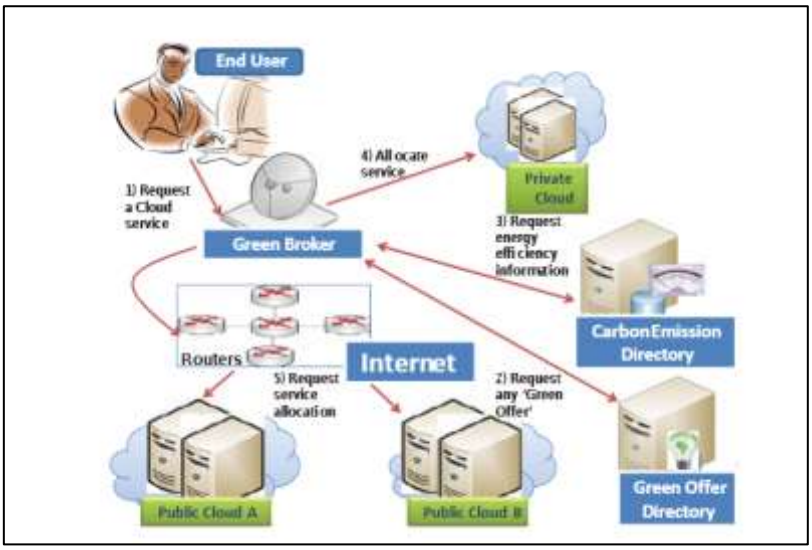

Figure 1: Carbon Aware Green Cloud Architecture [13]

To address this challenge, this study proposes HERO-CS 2.0 (Hybrid Energy-efficient Reinforcement and Optimization with Carbon-intensity-based Container Scheduling), a novel framework designed for adaptive, energy-efficient, and carbon-aware container allocation [14]. The proposed framework combines Deep Q-Network (DQN)-based Reinforcement Learning with optimization techniques to achieve intelligent scheduling decisions. The RL component continuously learns optimal scheduling policies from dynamic workload conditions and system feedback, enabling adaptive decision-making in changing cloud environments [15]. The optimization component works together with the RL engine by refining placement decisions according to multiple constraints, including CPU utilization, SLA compliance, thermal conditions, energy consumption, and carbon intensity [16].

The hybrid integration of RL and optimization is a key contribution of HERO-CS 2.0. While the RL model predicts suitable scheduling actions based on learned experience, the optimization mechanism evaluates and selects the best container placement by minimizing total energy usage and carbon emissions under operational constraints [17]. Carbon intensity is incorporated into the scheduling process by measuring the carbon emission factor associated with energy consumed by computing resources. This enables the scheduler to prioritize low-carbon nodes and reduce the environmental impact of cloud operations. The objectives are followed as:

- To develop an energy-efficient and carbon-aware container scheduling framework for cloud data centres.
- To integrate Deep Reinforcement Learning (DQN) with optimization techniques for adaptive container scheduling.
- To optimize container placement using CPU utilization, SLA, thermal conditions, and carbon intensity parameters.
- To evaluate HERO-CS 2.0 against existing scheduling methods in terms of energy efficiency, SLA compliance, execution cost, and carbon reduction.

2. Related Works

Recent literature has shown great progress made in energy efficient and carbon-literate cloud computing, edge computing, and AI scheduling systems. Beena et al., (2026)[18] introduced CarbonWise CX, which combines the RAG and LLM analytics for providing a green cloud framework. Using 8,469 real-world customer records, their system reduced their carbon emissions by 34.6%, was 41.8% faster to respond and improved SLA compliance by 29.5%. Cao et al., (2026)[19] proposed the architecture called GFEC-ICA for an architecture involving AI for IoT, consuming 540-770 kWh, 96% learning accuracy and 92% system

resilience, while scaling across 500 nodes with 680 operational units. Ozkan et al., (2025)[20] proposed the Federated Carbon Intelligence (FCI) framework which was able to decrease cumulative CO_2 emissions by up to 45% in a three-year simulation period by using reinforcement learning and telemetry-aware scheduling. Likewise, Paramanayakam et al., (2025)[21] introduced Ecomap in edge servers which cut back the carbon emissions by 30% and carbon delay product by 25%.

Peng et al., (2025)[22] introduced hierarchical ensemble learning to construct building energy prediction, which outperformed the traditional supervised building energy prediction models with a minimum Mean Squared Error (MSE) of 3.02. To enhance the transparency and cloud sustainability by making the deployment of cloud services more dynamic based on carbon intensity forecasting, Suryadevara et al., (2024)[23] presented AI based carbon-aware orchestration to streamline cloud optimization. Ntzoumanis et al., (2024)[24] examined how implementing IMO carbon regulations impacts maritime energy efficiency and discussed the operational advantages of Energy Efficiency Existing Ship Index (EEXI) and Carbon Intensity Indicator (CII) policies in reducing GHGs. In the field of smart grids, Li et al., (2023)[25] summarized the integration of Edge-Cloud computing, highlighting decentralized intelligent energy management and flexible edge computation in renewable energy power systems.

Sun et al., (2026) [26] proposed LACE-RL, a deep reinforcement learning approach for serverless cloud environments. Their solution cut the carbon emissions during idle use by 77.08% and cold-start latency by 51.69% from static scheduling approaches. Euchi et al., (2026)[27] investigated renewable energy powered logistic systems and found that AI-based drone routing and solar-powered delivery hubs can lead to energy savings of 54% to 84%. In addressing the challenge of scheduling tasks in serverless computing, Hadi et al., (2025)[28] developed a PSO-ACO hybrid scheduling method that decreased energy consumption by 30% while ensuring more than 91% SLA adherence. Li et al., (2024)[29] designed a hybrid supervised and reinforcement learning framework for WBAN systems, which minimizes energy consumption near optimally and has low computational complexity. Attia et al., (2024)[30] suggested a multi-layer reinforcement learning scheme for mobile networks to achieve the optimum performance levels for throughput, energy efficiency, and coverage; study showed that the proposed scheme performs substantially better than traditional optimization schemes. Last but not least, Zhou et al., (2022)[31] proposed a hierarchical multi-agent DRL system for mobile edge computing, which achieved almost 50% more system energy efficiency and learning performance than model-free approaches. To conclude, these studies collectively highlight the increasing significance of sustainable, optimization-based approaches with AI and a carbon awareness approach and with reinforcement learning for cloud and distributed computing systems.

Existing works optimize energy efficiency, reduce carbon emission and enhance SLA performance with AI and reinforcement learning; however, few works embed the carbon-aware scheduling at the container level in realtime adaptive optimization. Implementing current solutions also introduces scalability constraints, cloud deployment and configuration variations, flexible handling of dynamic workloads and reliable monitoring of carbon intensity in distributed cloud-edge infrastructures in real time.

Significance of the Research:

Excessive emission of carbon dioxide has become a concern for human society across the world in a recent global warming issue. The development of renewable and sustainable energy sources has become urgent to reduce carbon emissions (Fu Jiang et al. 2024). The increased dependence on cloud services and containerized applications, along with their higher usage has sharply increased energy consumption and carbon emissions of large-scale data centers. This increases the cost of operations and contributes to worldwide environmental problems. Many of the classical energy-aware scheduling techniques like PABFD and DVFS focus on reducing energy at hardware or virtualization level not considering the dynamic behaviour of workload and environmental parameters. The inflexibility of these methods in heterogeneous and carbon-intensive settings creates an urgent need for a smart mechanism for scheduling that simultaneously optimizes energy, quality of service, and sustainability.

The HERO-CS 2.0 framework suggested reducing this gap by deploying DQN with optimization-based decision-making to develop a hybrid, adaptive and carbon-aware scheduling method. HERO-CS 2.0 can utilise resources efficiently by taking into account the real-time CPU utilisation, SLA compliance, and carbon intensity. TOTAL energy consumption and SLA violation ratesreduced. This study is important for the development of green CC, encourages the expansion of AI-powered solutions for sustainability, and provides a scalable model that can help in the development of eco-efficient container orchestration on distributed cloud infrastructure in future.

3. Research Methodology

This section introduces proposed energy and carbon efficient container scheduling framework, HERO-CS 2.0, in cloud data centers. The methodology combines PABFD, DVFS and reinforcement learning mechanism

based on deep Q-learning to optimize container placement, decrease energy consumption, decrease SLA violation, and increase the environmental sustainability [32]. Figure 2 represents the Framework of methodology.

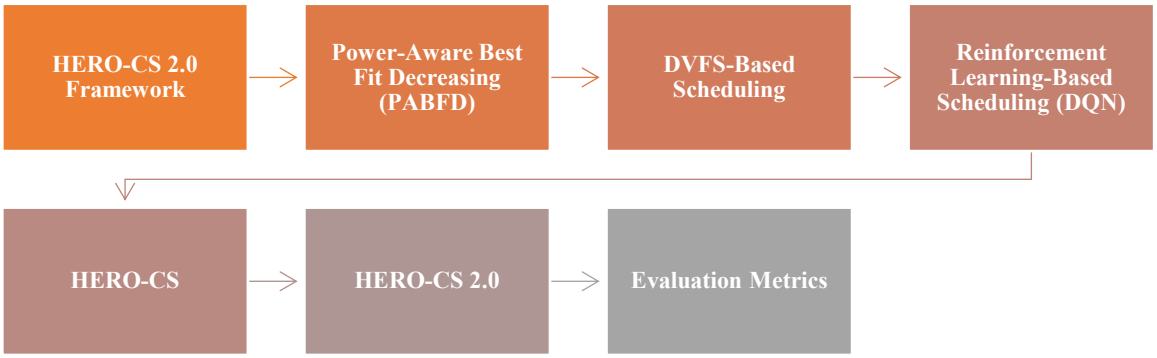

Figure 2: framework of proposed work

3.1 HERO-CS 2.0 Framework

The cloud data center architectures in the figure 3 are used for the architecture of the HERO-CS 2.0 which enables intelligent, adaptive and sustainable container scheduling in cloud data center environments [33]. The framework is built around several layers that are interdependent and designed for monitoring, optimization and decision making. The physical infrastructure layer continuously collects data of CPU utilization, power consumption, thermal status and carbon intensity from the cloud nodes [34]. These metrics are passed to the monitoring and profiling layer to manage them efficiently [35]. The framework integrates PABFD for initial energy-aware placement, DVFS for dynamic power optimization, and DQN-based reinforcement learning for adaptive scheduling and migration. Lastly, the carbon-aware optimization layer reduces carbon emissions, SLA violations and energy consumption [36].

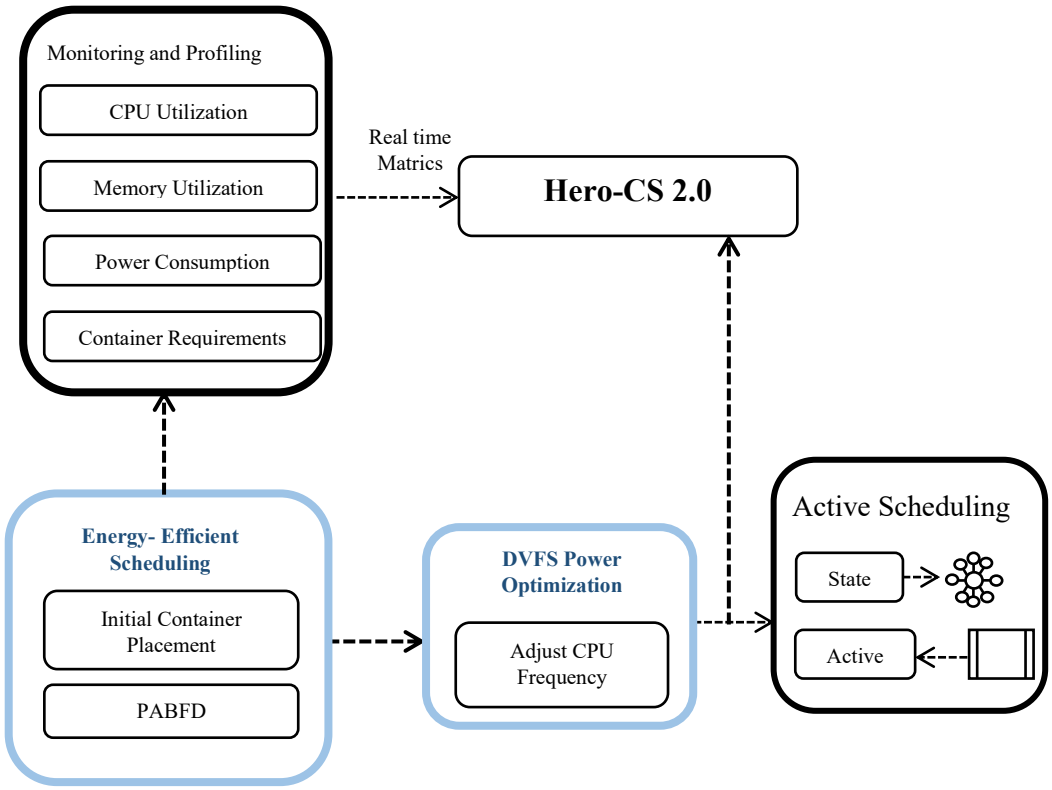

Figure 3: Proposed Contained Scheduling Architecture (HERO-CS 2.0)

3.2 Power-Aware Best Fit Decreasing (PABFD)

The suggested container scheduling method sorts all incoming containers in descending order based on the CPU utilization or resource demand. By placing high-demand containers first, it reduces the chance of fragmented resources while improving the resource allocation process effectively [37]. When a more

massive workload given for the first time, the scheduler is able to perform better host selection in terms of energy consumption in the further placement.Once a virtual machine selected, the VMplaced in the host with the lowest energy consumption by the "best fit" selection. Using this approach reduces energy consumption without compromising the performance and reliability of cloud infrastructure (AbdelhadiAmahrouch et al., 2025).

Every physical host in the data center records the current CPU utilization and the power model used by the host [38]. When a container is selected for placement, the scheduler reviews all the active hosts having sufficient available capacity and computes the increased power (ΔP) predicted due to the container deployment on each active host. Calculate the increase in power using a simple linear model.

$$\Delta P = P_{max} \times U_{new} - P_{max} \times U_{current} - - - (1)$$

P_{max}, the maximum power of the host, $U_{current}$, its current utilization, and U_{new}, the utilization after the container is assigned. The host that produces the least $\Delta P$is chosen to avoid wasting extra energy. If there is no available host to run the container, a host is activated from the pool. The process continues until every container is scheduled; thus, resources are used efficiently and energy use in the data center is reduced [39].

3.3 DVFS-Based Scheduling

The method based on DVFS help minimize the energy consumption of cloud data centres. This is done via dynamic adjustment of the CPU voltage and frequency levels based on the workload. When less processing power is required, the workload would reduce the frequency and voltage of the processor to decrease the power while meeting SLA requirements.DVFS based scheduling method executed to the cloud servers without any application-oriented implementation or configuration as it is their beneficial point (Hirofumi Noguchi et al., 2024). A Processor's overall power consumption can be expressed as.

$$P_{total} = P_{dynamic} + P_{static} - - - (2)$$

$P_{dynamic}$is the switching power and P_{static} is the leakage power.

The main affecting parameters for dynamic power are voltage and frequency and it is given as.

$$P_{dynamic} = C \times V^2 \times f - - - (3)$$

In this case, C represents the effective switching capacitance, V is the supply voltage, and f is the frequency. When the utilization is low, the voltage and frequency are reduced to lower energy consumption over time, which can be expressed as.

$$E = P_{total} \times t - - - (4)$$

E represents the energy consumed and t indicates the ET.

Schedulers that use DVFS techniques dynamically check the CPU utilization and the characteristics of the workload and determine a suitable frequency level that reduces energy without degrading performance. As the workload gets more intense, the frequency is increased to ensure throughput at all levels that are difficult to implement at circuits. When the workload gets lighter, the frequency scaled down to save power. Thus, it ensures good energy efficiency while not compromising on performance levels [40].

3.4 Reinforcement Learning-Based Scheduling (DQN)

This technique uses DRL to control container deployment and resource allocation in the cloud. This process, which is scheduling modelled as a decision-making problem that takes place in a sequence. Moreover, in this statement, the agent that is the scheduler interacts with the environment. Their further aim is to minimize the consumption of energy and SLA violations while enhancing the efficiency of performance.A recent work of Jie et al. 2021 proposed a Deep Q-Network (DQN)-based task processing to minimize the total latency in edge computing.

The agent in this approach observes the state (S) of the system, which may include the usage of CPU, the usage of memory, the state of hosts that are active, etc. Further, an action to be taken (A) is selected. For instance, allocate a container to a host. After each action, based on the performance of the system it receives a reward (R). The agent learns over time a policy (π) that specifies the best action for every system state. Learning behaves according to the Q-learning behaviour, according to the following equation.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'}(s', a') - Q(s,a)\right] - - - (5)$$

Where,

- The value of choosing an action in a state is defined as Q(s,a).
- The value of α is the learning rate
- γ represents the future rewards discounting factor.
- r represents the immediate reward
- Next state after taking action a is s′

In order to improve the repeatability and transparency of experiments, the DQN scheduling model can be further enriched with the detailed setting information of the training, including learning rate, the size of

the replay memory, batch size, exploration–exploitation strategy (ε-greedy), discount factor, and frequency of network updates on the target network. In DQN variant, the Q-function approximated with a NN(Neural Network) to manage High dimensional state spaces efficiently. The model updates itself continuously through experience replay and target network stabilization. The DQN scheduler learns on its own to get the best energy-efficient schedule strategy for different workload situations. This satisfies energy consumption, execution time SLA in a well-balanced approach.

3.5 HERO-CS

To ensure effective energy utilization and workload balance between energy and service time in cloud data centers, the HERO-CS methodology combines PABFD for heuristic initial placement, DVFS for hardware-level energy control, and DQN for adaptive scheduling.

The operation begins with a layer of monitoring and profiling that constantly collects real-time metrics such as CPU utilization, memory utilization, and power consumption of all nodes. Each node power model is expressed as a linear equation.

$$P = P_{idle} + (P_{max} - P_{idle}) \times U - - - (6)$$

The parameters P, P_{idle}, P_{max} and U refer to current power, idle power, maximum power of a host; and current CPU utilization, respectively. Moreover, we track the deadlines and QoS levels of container SLA parameters to meet our service need.

The first step requires the incoming containers sorted in the order of CPU demand, i.e. in a descending order. After which placement done as per PABFD principle. The host with the least energy increment places each container within the host.

$$\Delta P = P_{new} - P_{current} - - - (7)$$

This results in a near-perfect initial allocation with minimal computation and no energy waste. After placement, DVFS changes the voltage and frequency of the CPU with respect to workload. To minimize dynamic power consumption, the value of V and f are decreased for underutilized nodes. Dynamic power consumption is calculated as:

$$P_{dynamic} \propto V^2 f - - - (8)$$

It avoids wasted energy but still has to maintain SLA and performance stability. Next, applying DQN for decision improvement over time. The RL (Reinforcement Learning) agent can see system states (utilization, power, demand) and take actions (container placement, migration). The Q-learning update rule guides this adaptive process.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t + \gamma \max_{A'} Q(S_{t+1}, A') - Q(S_t, A_t)] - - - (9)$$

The agent learns continuously to improve its scheduling policy so that it minimizes the total energy used and SLA violations. Lastly, underused hosts detected in time by the Container Migration and Consolidation phase and the DQN learned policy is used to migrate their containers to active ones. After we finish moving containers, shut down the spare nodes to save power. It continuously ensures a sound allocation of data center resources without compromising performance, energy footprint and latency. The HERO-CS framework utilizes heuristic, hardware level and intelligent learning. This unifies all approaches in a single framework that is sustainable, adaptable and efficient.

3.6 HERO-CS 2.0

The HERO-CS 2.0 framework improves upon existing energy-aware scheduling by incorporating carbon-intensity into the process. The main objective is to reduce the energy and the carbon footprint of containerized workloads running in cloud data centres while ensuring high performance and low SLA violations. The combination of heuristic placement (PABFD), hardware power control (DVFS) and intelligent learning (DQN) along with the addition of carbon-intensity factor constitutes the environmentally friendly scheduling. Currently the framework includes values of carbon intensity from external APIs and energy monitoring sources, but the methodology of producing carbon data in real time, verifying it, and gauging how reliable it is needs to be further defined to be consistent across geographically dispersed cloud environments.

The system monitors the CPU utilization (Ui), power consumption (P_i), and carbon intensity (C_i) of each node's electricity source. The carbon-aware energy cost for a node is computed as.

$$E_{carbon,i} = P_i \times C_i - - - (10)$$

This means P_i is power in kilowatts, and C_i is carbon intensity (gCO2/kWh) or carbon emitted in grams over kilowatt-hour.

The aim of algorithm is to minimize an overall multi objective cost function.

$$Minimize : F = w_1 E_{total} + w_2 SLA_{viol} + w_3 E_{carbon} - - - (11)$$

The notation$E_{total} = \sum iP_i$

Where SLA_{viol} is SLA violation rate and E_{carbon} is the total carbon-based energy cost. The weights w_1,w_2,w_3 can tune the importance of each factor so that can be changed flexibly to energy or sustainable priorities.

The choice of the weight factors (w_1, w_2, w_3) in the multi-objective cost function has a considerable influence on the scheduling behavior. These parameters can be justified and tuned for optimal operation under different load conditions through future work of sensitivity analysis or adaptive weight tuning techniques.

HERO-CS 2.0 uses PABFD heuristic to successfully place the containers on the nodes which generate less power increment. Afterwards, power is saved using the principle on underloaded nodes by reducing voltage and frequency. The DQN agent selects placement or migration actions to adaptively modify scheduling decisions. The DQN agent observes states such as CPU utilization, power, and carbon intensity. Its learning is guided by a carbon-aware reward function.

$$R_t = w_1\left(1 - \frac{E_t}{E_{max}}\right) + w_2\left(1 - \frac{C_t}{C_{max}}\right) - w_3 SLA_{viol} - - - (12)$$

The proposed HERO-CS 2.0 framework is illustrated in Algorithm 1. The algorithm combines heuristic container placement, power optimisation based on the dynamic voltage/frequency scaling (DVFS), reinforcement learning-based adaptive scheduling, and carbon-aware decision making to reduce energy usage, SLAs violation, and carbon emissions in the cloud data centres.

Algorithm 1: HERO-CS 2.0 Carbon-Aware Container Scheduling

```
Input:
  C = {c1, c2, ..., cn}  // Incoming containers
  H = {h1, h2, ..., hm}  // Available hosts
  CPUi = CPU utilization of host i
  Pi = Power consumption of host i
  CIi = Carbon intensity of host i
  SLA = Service Level Agreement constraints
  DQN = Deep Q-Network scheduling agent
Output:
  Optimal container placement with minimum
  energy consumption and carbon emission
Begin
1. Initialize monitoring and profiling module
2. Collect real-time system metrics:
     CPU utilization
     Memory utilization
     Power consumption
     Thermal conditions
     Carbon intensity
3. Sort all incoming containers in descending
   order according to CPU demand
4. For each container c ∈ C do
5.    For each host h ∈ H do
6.       Check available resource capacity
7.       Estimate power increment:
           ΔP = Pnew − Pcurrent
8.       Compute carbon-aware cost:
           Ecarbon = Pi × CIi
9.    End For
10.   Apply PABFD heuristic
11.   Select host hbest with:
        Minimum(ΔP + Ecarbon)
12.   Deploy container c on hbest
13.   Apply DVFS optimization
14.   If CPU utilization of hbest < Threshold then
15.      Reduce CPU voltage and frequency
16.   Else
17.      Maintain normal operating frequency
18.   End If
19.   Construct system state S:
        S = {CPU usage, Power,
           SLA status,
           Thermal condition,
```

```
        Carbon intensity}
20.  DQN agent selects action A:
       A ∈ {Placement,
            Migration,
            Consolidation}
21.  Execute selected action A
22.  Calculate reward Rt:
:contentReference[oaicite:0]{index=0}
23.  Store transition experience:
       (S, A, Rt, S′)
24.  Update DQN parameters using:
       Experience replay
       Target network optimization
25. End For
26. Detect under-utilized or high-carbon hosts
27. Migrate containers from inefficient hosts
    to energy-efficient hosts
28. Consolidate workloads and shut down
    idle hosts to reduce energy consumption
29. Return optimized container allocation
End
```

The key points of the proposed algorithm are that it can continuously collect resource utilization, energy consumption, and carbon intensity data and make intelligent scheduling and migration decisions. HERO-CS 2.0 uses heuristic optimization coupled with DQN-based learning to optimise their execution dynamically, thereby helping to maintain energy efficiency and environmental sustainability, even in the face of workload variability.

3.7 Evaluation Metrics

• Total Energy Consumption (E_t)

Exploring research questions considering uncertainty has a high potential for energy management and operations planning (Jasper Stoter et al., 2025). This metric measures the total energy consumed by all physical hosts while the container is running. It includes both dynamic and static power usage.

$$E_t = \sum_{i=1}^{N} P_i(t) \times \Delta t - - - (13)$$

Where:

- The power consumed by host i at time t is represented by $P_i(t)$.
- Δt denotes Measurement interval.

Lower values of E_t indicate better energy efficiency.

• SLA Violation Rate (SLAV)

The SLA typically outlines the requirements and defines the Quality of Service (QoS) parameters that govern the relationship between cloud users and the service provider (Umer Arshad et al., 2022). This checks how often the system does notfulfil SLAs related to performance deadlines or response time thresholds.

$$SLAV = \frac{N_{violations}}{N_{total\_tasks}} - - - (14)$$

Where:

• $N_{violations}$ is the number of SLA violations.

• The variable N_{total_tasks} denotes the total number of tasks performed.

A lower SLAV indicates more reliable scheduling.

• Average Response Time (ART)

ART represents the average duration a system or application takes to reply to a user's request or execute a task. It serves as an essential performance indicator that measures the efficiency and responsiveness of computing or cloud-based services. In CC, a lower ART signifies improved service performance and a better user experience.This metric assesses how quickly the system responds to a user or container request.

$$ART = \frac{\sum_{i=1}^{N} \left(T_{completion_i} - T_{arrival_i} \right)}{N} - - - (15)$$

Where.
• $T_{completion_i}$ refers to the total time required for task 1 completion.
• The starting time for the execution of task I represented $T_{arrival_i}$
Minimizing ART ensures better QoS and user experience.

• Execution Time (ET)

Task scheduling is a crucial aspect of CC that helps optimize resource usage, minimize ET, and improve overall system performance (Fatih Kaplan et al., 2025).This keeps track of everything done with the containers.

$$ET = T_{end} - T_{start} - - - (16)$$

Less time to execute means more efficient schedule and less overhead.

• Carbon Intensity Impact (CII)

The placement of data centers is also a key factor in influencing carbon footprints, as the CI of energy generation differs significantly from one region to another (Haoyang Liu et al., 2025)
This is an indication of how the scheduling is environmentally efficient by incorporating carbon footprint of the energy used.

$$CII = \sum_{i=1}^{N} (E_i \times CI_i) - - - (17)$$

Where.
• Ei is the energy consumed by host i.
• The carbon intensity of energy consumed ($kgCO_2/kWh$) denoted as CII
A smaller CII value indicates a more sustainable system.

4. Results And Discussion

The results of the experiments showed that HERO-CS 2.0 beats all the baseline methods on all metrics. The framework proved efficient in terms of energy by minimizing overall energy consumption using effective container placement and power scaling. With this advanced reinforcement learning, the system was capable of dynamic decision-making to learn the optimal scheduling strategies. It reduces wastage of resources while enhancing utilization. By implementing CI-based scheduling, the model also ensured that workload assignment favoured nodes with lower carbon intensity, which is also environmentally friendly. Table 1 represents Scheduling Algorithms Comparative Performance Evaluation Metrics.

Table 1 Scheduling Algorithms Comparative Performance Evaluation Metrics

Metric/Algorithm	E_t(kWh)	SLAV (%)	ART(s)	CII(kg)	ET (s)
DVFS	710	3.7	1.70	0.81	25.5
PABFD	670	4.2	1.62	0.79	21.3
DQN	580	3.8	1.32	0.71	19.1
HERO-CS	530	2.7	1.21	0.59	17.2
HERO-CS 2.0	470	2.0	1.10	0.45	15.3

Many of the algorithms have similar characteristics. Five algorithms, DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0, have similar characteristics, as shown in figure 4. The highest energy consumption is recorded by DVFS with around 700 kWh, which is very close to PABFD at around 670 kWh. The DQN saves about 580 kWh and the HERO-CS saves almost 530 kWh. HERO-CS 2.0 consumes the least energy: approximately 470 kWh. On the energy side, HERO-CS 2.0 saves approximately 230 kWh of energy over DVFS, a saving of almost 33%. The chart shows overall that the use of advanced optimization techniques, particularly HERO-CS 2.0, results in a considerable improvement in energy efficiency compared to traditional optimization techniques.

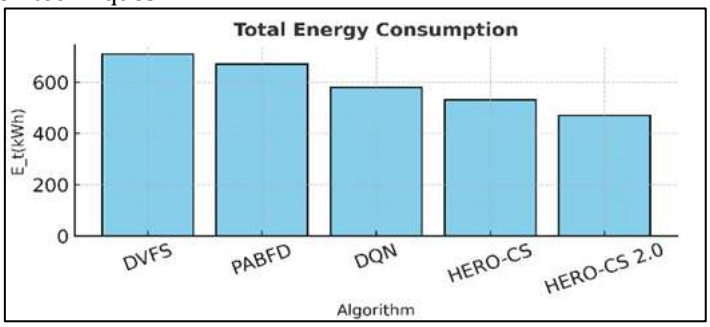

Figure 4: Total Energy Consumption

To compare the performance of all the algorithms, we plotted the percentage of SLA violations for five algorithms, as shown in figure 5: DVFS, PABFD, DQN, HERO-CS, and the proposed HERO-CS 2.0. The SLA violation rate with the highest is with PABFD, around 4.1% which shows lower service reliability. The next two, with approximately 3.7% and 3.8% respectively, are DVFS and DQN. HERO-CS decreases violations to almost 2.7%, thereby improving the performance. HERO-CS 2.0 performs optimally at a low SLA violation rate of ~2.0%. Compared to PABFD, HERO-CS 2.0 reduces SLA violations by nearly 2.1 percentage points - a significant improvement of almost 51% - indicating its superior quality-of-service performance and system stability.

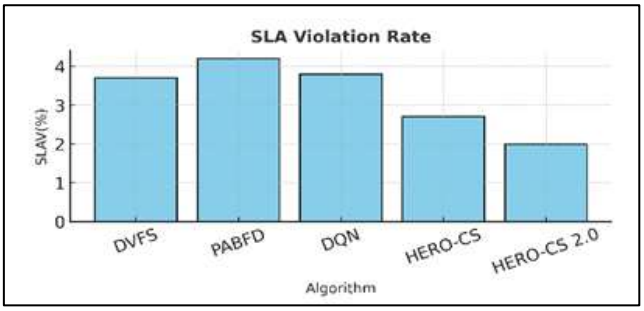

Figure 5: SLA Violation Rate

The response time of five algorithms, i.e., DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0, are compared in figure 6. The highest average response time is recorded by DVFS and is approximately 1.7 seconds followed by PABFD with an average response time of about 1.6 seconds. DQN performance is improved by almost 1.3 seconds and HERO-CS further decreases the response time to around 1.2 seconds. With a response time of approximately 1.1 seconds, HERO-CS 2.0 is the most efficient in terms of the fastest response time. The reduction in response time for HERO-CS 2.0 above DVFS is approximately 0.6 seconds, or about 35%, numerically being the more efficient of the two processes and the system becoming the more responsive.

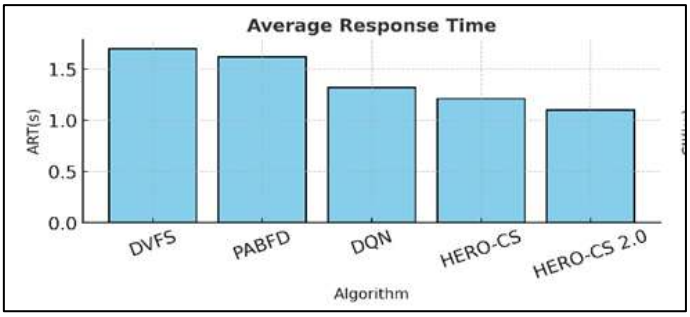

Figure 6: Average Response Time

The carbon emission effect of five algorithms, namely DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0 are compared in figure 7. HERO-CS 2.0 has the lowest carbon intensity with about 0.45 kg of carbon emission while DVFS has (0.80 kg) of carbon emission, a difference of approximately (0.35 kg) is a nearly (44%) reduction. In general, the chart confirms the success of HERO-CS 2.0 in enhancing environmental sustainability and minimizing energy-related carbon footprint.

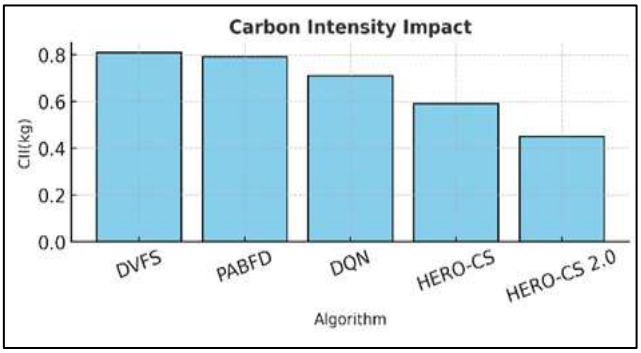

Figure 7: Carbon Intensity Impact

The execution time of five algorithms (DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0) are compared in figure 8. The maximum execution time recorded by DVFS is around 25 seconds which shows low processing efficiency. The time PABFD can execute is cut down to nearly 21 seconds by DQN's further performance enhancement, which is about 19 seconds. HERO-CS runs in about 17 seconds, which is better than the computational efficiency of HERO. HERO-CS 2.0 achieves the best performance and the shortest execution time (~15 seconds). But when it comes to actual numbers, the HERO-CS 2.0 would save almost 10 seconds of execution time versus DVFS, or approximately a 40% boost. In conclusion, the graph illustrates a superior optimization and speed advantage of HERO-CS 2.0.

Figure 8: Execution Time

Figure 9 compares the efficiency of the 5 scheduling algorithms, namely DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0. The lowest CPU utilization efficiency is around 62% as observed through DVFS whereas PABFD slightly improves the performance around 68%. Further, DQN significantly improves resource utilization efficiency to almost 75%, showing better resource management. HERO-CS achieves a close-to-82% efficiency, showing better workload balancing and scheduling optimisation. The highest CPU utilization efficiency can be obtained with HERO-CS 2.0, about 88%. HERO-CS 2.0 is nearly 26 percentage points more efficient than DVFS numerically, and this is around a 42% gain in resources consumed and overall performance of the system.

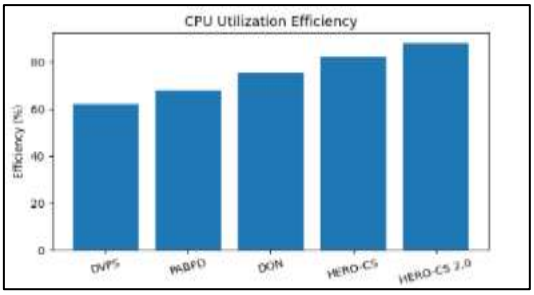

Figure 9: CPU Utilization Efficiency

The performance of five scheduling algorithms for correct cloud resources allocation is compared in Figure 10. DVFS is the least accurate of all the allocations with about 70% accuracy, and PABFD slightly improves the accuracy to around 74%. The second improvement is the nearly 80% accuracy in allocation, which translates to a more efficient workload distribution and decision-making. HERO-CS shows considerable improvement rate of approximately 87% which shows that the scheduling and allocating the resources are done efficiently. The highest allocation accuracy is achieved by HERO-CS 2.0 at around 92%. HERO-CS 2.0 makes it possible to improve the allocation accuracy by almost 22 percentage points (around 31%) compared to DVFS, which results in better utilization of resources, less wastage of resources and makes the cloud systems more reliable.

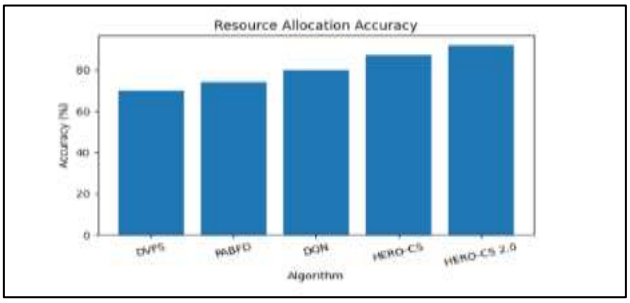

Figure 10: Resource Allocation Accuracy

Next, the results of five algorithms are compared in terms of thermal management capabilities, as shown in Figure 11: DVFS, PABFD, DQN, HERO-CS, and HERO-CS 2.0. Note that the thermal stability is lowest at about 55%, suggesting that DVFS is less effective in controlling temperature in cloud data center operations. By using intelligent workload handling, DQN further increases the stability up to nearly 68%, compared to 60% by PABFD. HERO-CS is able to significantly boost thermal stability to around 76%, which is an improvement in the energy-aware scheduling. At a thermal stability of around 84% HERO-CS 2.0 is the most thermally stable. In terms of numbers, HERO-CS 2.0 has increased the thermal stability by almost 29 percentage points compared to DVFS, which is roughly a 53% increase in thermal stability control.

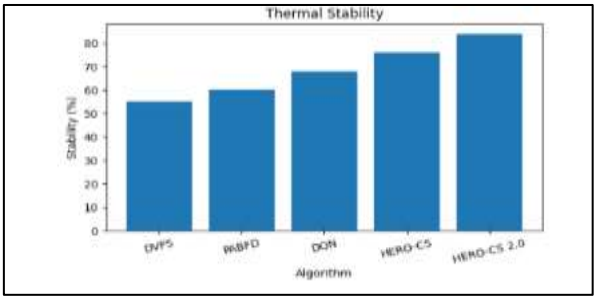

Figure 11: Thermal Stability

In Figure 12, we compare the Quality of Service satisfaction of five scheduling algorithms: DVFS, PABFD, DQN, HERO-CS and HERO-CS 2.0. DVFS gets the lowest QoS satisfaction rate with ~72% while PABFD marginally improves performance to ~76%. DQN also improves QoS delivery to almost 83%, which is due to the improvements in task scheduling and resource management. The satisfaction achieved by HERO-CS is approximately 90% which shows more responsiveness and reliability in cloud services. HERO-CS 2.0 offers the best quality of service satisfaction rate of around 95%. Additionally, at a numeric level, HERO-CS 2.0 shows a near 23 percentage points increase in QoS satisfaction, which is approximately a 32% increase in overall service quality and user experience when compared to DVFS.

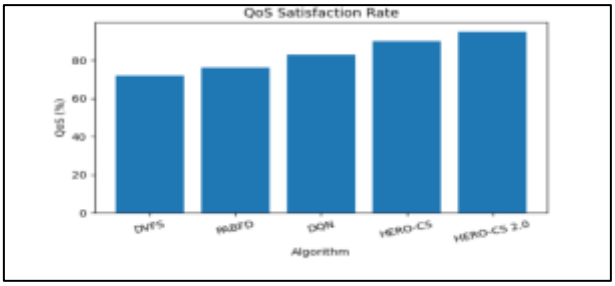

Figure 12: QoS Satisfaction Rate

Figure 13 illustrates the performance of various scheduling algorithms for reduction of the number of host machines required to run applications and reduce energy consumption. For the minimum host consolidation rate, the values are ~50% for DVFS and ~ 57% for PABFD. DQN further smartly schedules workloads, thereby boosting the workload consolidation to almost 66%. The performance of HERO-CS shows considerable improvement with the rate of consolidation around 79%, which reflects improved resource utilization and lower load on the hosts. The best host consolidation ratio is achieved by HERO-CS 2.0 with nearly 91%. The numerical improvement of HERO-CS 2.0 over DVFS – nearly an 82% improvement – is about 41 percentage points when computing consolidation, which reduces power usage and boosts efficiency of cloud infrastructure.

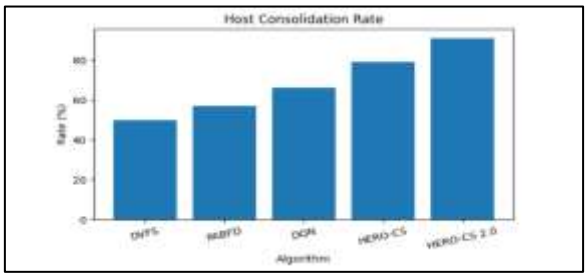

Figure 13: Host Consolidation Rate

Figure 14 shows that various scheduling policies have different levels of efficiency in distributing workloads evenly among cloud resources. DVFS achieves the lowest load balancing performance at about 58%, which means that the workload distribution is poorer. By intelligently allocating resources, DQN reaches nearly 71% accuracy while PABFD reaches just about 63%. HERO-CS considerably improves load balancing performance around 84%, indicating more efficient management of workloads and lower resource congestion. With adaptive scheduling capabilities, HERO-CS 2.0 delivers the highest performance at around 90%. In absolute values, HERO-CS 2.0 enhances load balancing by almost 32 percentage points when compared to DVFS, which is equivalent to an efficiency and stability of the overall cloud system of about 55%.]

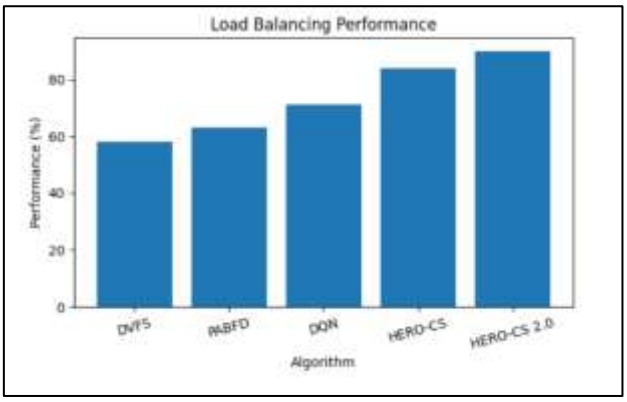

Figure 14: Load Balancing Performance

5. conclusion and future work

This section show that the proposed HERO-CS 2.0 framework is able to achieve much better energy efficiency, service reliability, and environmental sustainability gains within cloud data centers compared with DVFS, PABFD, DQN and HERO-CS methods. The achieved total energy consumption of the framework was 470 kWh, which was 710 kWh in the case of DVFS, resulting in almost 33% energy savings. The SLA violation rate in PABFD was decreased from 4.2% to 2.0% and the response time was reduced from 1.70 s to 1.10 s. The execution time was reduced from 25.5 s to 15.3 s, and the impact on carbon intensity was minimized from 0.81kg to 0.45kg. Furthermore, the resource allocation accuracy and QoS satisfaction of HERO-CS 2.0 were 92% and 95%, respectively, and the CPU utilization efficiency and host consolidation efficiency were 88% and 91%, respectively. Despite these advantages, the model has some drawbacks: it relies heavily on the accuracy of real-time carbon intensity data, it struggles to scale with geographic dispersion of clouds and it adds computational cost due to DQN training. The practical challenges for deployment, including interoperability with different cloud platforms, lag between deployment and monitoring in real time, and the variability of renewable energy also need to be examined. Areas that need further research are adaptive weight tuning, multi-cloud and edge-cloud deployment, integration of live renewable energy forecasting and lightweight reinforcement learning models for large-scale real-time scheduling. In general, HERO-CS 2.0 has a solid base for intelligence within carbon-aware cloud scheduling and it shows very good potential for a sustainable next-generation cloud infrastructure.

Funding:
The authors received no financial support, grant, or funding from any funding agency in the public, commercial, or not-for-profit sectors for the research, authorship, and/or publication of this article.

Ethics Declaration:
This study did not involve human participants, animals, human tissue, or identifiable personal data. Therefore, ethical approval and informed consent were not required.

References

1. Hirofumi Noguchi& Masashi Kaneko (2024), “Application-Independent DVFS Method Based on OS Metrics Monitoring for Cloud Servers," *2024 IEEE International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, Beijing, China, pp. 1-6, doi: 10.1109/CCCI61916.2024.10736461.
2. B. M. Beena, PrashanthCheluvasaiRanga, Thotapalli Sri Surya Manideep, Sneha Saragadam & Gari kipati Karthik(2025), “A Green Cloud-Based Framework for Energy-Efficient Task Scheduling Using Carbon Intensity Data for Heterogeneous Cloud Servers”, *IEEE Access*, Vol. 11, pp. 73916 – 73938.

3. AbdelhadiAmahrouch, Youssef Saadi& Said El Kafhali (2025), "Optimizing Energy Efficiency in Cloud Data Centers: A Reinforcement Learning-Based Virtual Machine Placement Strategy", *MDPI, Network*, Volume 5, Issue 2, 10.3390/network5020017.
4. Asfandyar Khan , FaizanUllah , Dilawar Shah , Muhammad Haris Khan , Shujaat Ali & Muhammad Tahir (2025), "EcoTaskSched: a hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments", *Sci Rep*, 10;15:12296. doi: 10.1038/s41598-025-96974-9.
5. Meixian Jiang, ShuyingLv, Yuqiu Zhang, Fan Wu , Zhi Pei &Guanghua Wu (2025), "A Low-Carbon Scheduling Method for Container Intermodal Transport Using an Improved Grey Wolf–Harris Hawks Hybrid Algorithm", *MDPI, Applied Sciences,* Vol. 15, Issue 9, 10.3390/app15094698.
6. Fu Jiang, Jie Chen , Jieqi Rong, Weirong Liu , Heng Li& Hui Peng (2024), "Safe reinforcement learning based optimal low-carbon scheduling strategy for multi-energy system", *Science Direct, Sustainable Energy, Grids and Networks,* Volume 39, September 2024, 101454, https://doi.org/10.1016/j.segan.2024.101454.
7. Darius Drungilas, Mindaugas Kurmis, Audrius Senulis , Zydrunas Lukosius , Arunas Andziulis, Jolan ta Januteniene , Marijonas Bogdevicius , Valdas Jankunas & Miroslav Voznak (2023), "Deep reinforcement learning based optimization of automated guided vehicle time and energy consumption in a container terminal", *Science Direct, Alexandria Engineering Journal*, Vol. 67, pp. 397-407, https://doi.org/10.1016/j.aej.2022.12.057.
8. Huaqiang Si&Jiyun Qin (2025), "Energy-Efficient Scheduling for Resilient Container-Supply Hybrid Flow Shops Under Transportation Constraints and Stochastic Arrivals", *MDPI, Journal of Marine Science and Engineering*, Volume 13, Issue 6, 10.3390/jmse13061153.
9. Ameni Kallel , Molka Rekik & Mahdi Khemakhem (2025), "A deep reinforcement learning-based optimization approach for containerized microservice scheduling in Hybrid Fog/Cloud environments", *Science Direct,* https://doi.org/10.1016/j.engappai.2024.109745.
10. Zaixing Sun, Hejiao Huang, Zhikai Li& Chonglin Gu (2025), "Energy-efficient real-time multi-workflow scheduling in container-based cloud", J*ournal of Combinatorial Optimization. Vol. 49, No. 2,* https://doi.org/10.1007/s10878-025-01265-8
11. Honghua Chen, Cong Shen, XinyuanQiu& Chuanqi Cheng (2024), "Container Scheduling Algorithms for Distributed Cloud Environments", *Processes*, Vol. 12, Issue 9, 10.3390/pr12091804.
12. Garg, Saurabh & Yeo, Chee Shin & Buyya, Rajkumar. (2011). Green Cloud Framework For Improving Carbon Efficiency of Clouds. 491-502. 10.1007/978-3-642-23400-2_45.
13. Ugwoke, F. N. "SmartGreen: A Novel Green Computing Architecture for Environmental Data Acquisition in IT/Automation DataCenter." African Journal of Computing & ICT 6, no. 5 (2013).
14. Jie X, Liu T, Gao H, Cao C, Wang P & Tong W (2021), "A DQN-based approach for online service placement in mobile edge computing, Proceedings of the 16th EAI International Conference on Collaborative Computing: Networking, Applications and Worksharing, *Springer* , pp. 169-183
15. Jasper Stoter , Xinyu Tang , Milos Cvetkovic , Peter Palensky , Henk Polinder , Çağatay Iris & Freder ik Schulte (2025), "Energy management and stochastic operations planning for electrified container terminals with uncertain energy supply and demand", *Journal of Cleaner Production*, Volume 527, 10 October 2025, 146383.
16. Umer Arshad , Muhammad Aleem, Gautam Srivastava &Jerry Chun-Wei Lin (2022), "Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers", *Science Direct, Renewable and Sustainable Energy Reviews*, Vol. 167, October 2022, 112782.
17. Fatih Kaplan & Ahmet Babalik (2025), "Performance Analysis of Cloud Computing Task Scheduling Using Metaheuristic Algorithms in DDoS and Normal Environments", *MDPI, Journals, Electronics,* Vol. 14, No. 10, 10.3390/electronics14101988.
18. Beena, B. M., Thotapalli Sri Surya Manideep, Sneha Saragadam, Sailesh Rathi, Ramaswamy Ramesh, and Vijay Holimath. "CarbonWise CX: An Agentic AI Framework for Carbon-Aware Customer Support Analytics Using RAG and LLM-Based Code Generation." IEEE Access 14 (2026): 53442-53460.
19. Cao, Jiaqi. "Intelligent computing architectures for sustainable large-scale IoT ecosystems enabling AI-driven industrial transformation." Discover Internet of Things (2026).
20. Ozkan, Mihrimah, and Cengiz S. Ozkan. "Federated carbon intelligence for sustainable AI: Real-time optimization across heterogeneous hardware fleets." MRS Energy & Sustainability (2025): 1-23.
21. Paramanayakam, Varatheepan, Andreas Karatzas, Dimitrios Stamoulis, and Iraklis Anagnostopoulos. "Ecomap: Sustainability-driven optimization of multi-tenant dnn execution on edge servers." IEEE Transactions on Computers (2025).
22. Peng, Kun, and Bin Xu. "Improving Building Energy Prediction Accuracy Using Hierarchical Supervised Learning." Available at SSRN 5201686.

23. Suryadevara, Siva Sai Krishna. "AI-Driven Carbon-Aware Orchestration for Sustainable Cloud Deployments." International Journal of Emerging Research in Engineering and Technology 5, no. 2 (2024): 165-175.
24. Ntzoumanis, Vasileios. "The impact in ship operation as imposed by the energy efficiency existing shipping index." Master's thesis, Πανεπιστήμιο Πειραιώς, 2024.
25. Li, Junlong. "Internet-of-Things-enabled smart local energy management system." PhD diss., University of Bath, 2023.
26. Sun, Bowen, Christos D. Antonopoulos, Evgenia Smirni, Bin Ren, Nikolaos Bellas, and Spyros Lalis. "Green or Fast? Learning to Balance Cold Starts and Idle Carbon in Serverless Computing." arXiv preprint arXiv:2602.23935 (2026).
27. Euchi, Jalel, Abdeljawed Sadok, and Majdi Argoubi. "Energy-efficient drone technologies and carbon-aware systems in sustainable supply chains: innovations, challenges, and policy pathways." International Journal of Logistics Research and Applications (2026): 1-28.
28. Hadi, Andik Prakasa, Budi Hartono, Khoirur Rozikin, and Iman Saufik Suasana. "Energy Efficient Task Scheduling in Serverless Computing Using PSO and ACO Algorithms." In 2025 4th International Conference on Creative Communication and Innovative Technology (ICCIT), pp. 1-7. IEEE, 2025.
29. Li, Shuang, Hong-Chuan Yang, Fang Xu, Huimin Hu, and Fengye Hu. "Energy-efficient relay transmission for WBAN: Energy consumption minimizing design with hybrid supervised/reinforcement learning." IEEE Internet of Things Journal 11, no. 10 (2024): 17770-17779.
30. Attia, Bishoy Salama, Aamen Elgharably, Mariam Nabil Aboelwafa, Ghada Alsuhli, Karim Banawan, and Karim G. Seddik. "Self-optimized agent for load balancing and energy efficiency: A reinforcement learning framework with hybrid action space." IEEE Open Journal of the Communications Society 5 (2024): 4902-4919.
31. Zhou, Hang, Yusi Long, Shimin Gong, Kun Zhu, Dinh Thai Hoang, and Dusit Niyato. "Hierarchical multi-agent deep reinforcement learning for energy-efficient hybrid computation offloading." IEEE Transactions on Vehicular Technology 72, no. 1 (2022): 986-1001.
32. Haoyang Liu &JiangtaoZhaim (2025), "Carbon Emission Modeling for High-Performance Computing-Based AI in New Power Systems with Large-Scale Renewable Energy Integration", Processes 2025, *13*(2), 595;https://doi.org/10.3390/pr13020595
33. Seerangan, Koteeswaran, Malarvizhi Nandagopal, Tamilmani Govindaraju, Nalini Manogaran, Balamurugan Balusamy, and Shitharth Selvarajan. "A novel energy-efficiency framework for UAV-assisted networks using adaptive deep reinforcement learning." Scientific Reports 14, no. 1 (2024): 22188.
34. Hussain, Quadeer, Ahmad Shukri Mohd Noor, Muhammad Mukhtar Qureshi, Jianqiang Li, Atta-ur Rahman, Aghiad Bakry, Tariq Mahmood, and Amjad Rehman. "Reinforcement learning based route optimization model to enhance energy efficiency in internet of vehicles." Scientific Reports 15, no. 1 (2025): 3113.
35. Liu, Yuchen, Ting Shu, Xuesong Yin, and Jinsong Xia. "Synergistic Evolutionary Optimization with Reinforcement Learning for Multi-Objective Energy-Efficient Hybrid Flow Shop Scheduling." Axioms 15, no. 3 (2026): 193.
36. Sanjalawe, Yousef, Salam Al-E'mari, Budoor Allehyani, and Sharif Naser Makhadmeh. "Reinforcement Learning-Guided Hybrid Metaheuristic for Energy-Aware Load Balancing in Cloud Environments." Algorithms 18, no. 11 (2025): 715.
37. Gupta, Deeksha, Abhishek Bajpai, Naveen Kumar Tiwari, and Smriti Yadav. "Energy-efficient routing optimization for underwater internet of things using hybrid Q-learning and predictive learning approach." Procedia Computer Science 235 (2024): 45-55.
38. Salim, Mahmoud M., Khaled M. Rabie, and Ali H. Muqaibel. "Energy-efficient irregular RIS-aided UAV-assisted optimization: A deep reinforcement learning approach." IEEE Internet of Things Journal (2025).
39. Saeed, Rashid A., Elmustafa Sayed Ali, Maha Abdelhaq, Raed Alsaqour, Fatima Rayan Awad Ahmed, and Asma Mohammed Elbashir Saad. "Energy efficient path planning scheme for unmanned aerial vehicle using hybrid generic algorithm-based Q-learning optimization." IEEE access 12 (2023): 13400-13417.