ESBO-FDDPG: A Federated Reinforcement Learning-Based Continuous Resource Scheduler for Cross-Domain Multi-Cloud Systems

Wenwei Su[1*], Zhengxiong Mao[1], Fengbo Kong[1], Yan Shi[2], Pan Xiao[3]
[1]Yunnan Power Grid Co., Ltd. Information Center, Yunnan, 650228, China
[2]Qujing Power Supply Bureau of Yunnan Power Grid Co., Ltd., Yunnan, 650100, China
[3]Southern Power Grid Digital Platform Technology (Guangdong) Co., LTD, Guangdong, 518101, China
E-mail: WenweiSu@outlook.com

Keywords: cloud computing, resource scheduling, multi-data center management, energy efficiency, cross-domain efficient satin bowerbird optimizer-driven federated deep deterministic policy gradient (CG-ESB-FDDPG)

Received: June 6, 2025

With the rapid development of cloud computing, the demand for efficient resource management across multiple data centers has become increasingly critical. Cloud platforms typically span geographically distributed data centers, offering scalability, flexibility, and on-demand resource provisioning. This research proposes a unified cross-domain continuous resource scheduling method for various data centers within the cloud platforms, aiming to tackle challenges such as uneven workload distribution, dynamic service demands, and energy inefficiency across multiple data centers. This research introduces a unified cross-domain continuous resources scheduling framework based on an Efficient Stain Bowerbird Optimizer-driven Federated Deep Deterministic Policy Gradient (ESBO-FDDPG) model. Workload traces from the Cross-Domain Cloud Scheduling Dataset were preprocessed using Min-Max normalization, and Statistical Pattern Recognition (SPR) with Fast Fourier Transform (FFT) and Radial Inverse Distance (RID) kernels was applied for feature extraction. The proposed method enables privacy-preserving, real-time, and adaptive scheduling by integrating federated learning with deep reinforcement learning. The proposed method supports continuous monitoring and coordination of workloads across multiple geographically distributed data centers, dynamically adjusting resource allocation based on service request patterns and power usage profiles. The system embeds vertical federated learning for cross-domain model updates without data sharing and uses reinforcement learning to continuously adapt to fluctuating workloads and infrastructure constraints. Simulation experiments were conducted using Python. Simulations conducted with different load and network conditions show that the suggested ESBO-FDDPG approach lowers average latency (63ms) and enhances energy efficiency (4.6 × 〖10〗^5bits/joules), and task completion rate (98.8%). This research offers important insights for sustainable cloud computing practices and a scalable, energy-conscious solution to unified resource scheduling in contemporary cloud systems.

Povzetek: Predlagana je metoda za sprotno, zasebnostno varno usklajevanje virov med več podatkovnimi centri v oblaku, ki z združevanjem federativnega učenja in globokega okrepitevnega učenja izboljša porazdelitev bremen ter zmanjša zakasnitev in porabo energije.

1 Introduction

Cloud platforms are massive collections of computers in different locations. It enables people and organizations to upload files, run applications, and access services from computers that do not manage hardware [1]. Each computer host can provide processing power, memory, and storage space. Moreover, the data can be allocated to different users anywhere in the world. The popularity of cloud platforms continues to improve with all types of tasks, from video-streaming to data analytics to file storage [2]. The systems can efficiently manage these resources, especially during peak times when many users can be active.During periods of predictable demand, the cloud platform dynamically allocates computing and storage resources while continuously monitoring system utilization to ensure optimal performance[3].

These cloud platforms run across numerous data centers, which are huge buildings packed with computers. A single center can have varying workloads based on time of day, regional demand, or the kinds of tasks being executed [4]. Intelligent systems are needed to determine when and how data should be shared between centers to maintain

efficiency and reduce energy consumption [5]. Cloud systems use advanced learning techniques to prevent data redundancy, maintain privacy, and make better decisions, conserving time and energy, making them more environmentally friendly by reducing costs and pollution [6].

Overloaded cloud systems can cause delays and critical tasks, requiring platforms to balance workloads on all computers and centers, allowing others to assist if necessary [7]. A smart resource scheduling system efficiently manages workload, adjusts task assignments, and reacts to changes, ensuring smoother operations and anticipating needs to prevent issues [8]. Enhancements in cloud services lead to faster user services, better resource utilization, and reduced energy consumption, necessitating intelligent workload management for reliable, quick, and environmentally friendly platforms [9].

Research questions

- ***RQ1:*** How can federated reinforcement learning improve energy efficiency in multi-data center cloud environments?
- ***RQ2:*** Can ESBO-FDDPG maintain or reduce task latency while dynamically allocating resources?
- ***RQ3:*** How effectively does the proposed method preserve privacy while aggregating insights across distributed data centers?

Research objective and system overview

The research introduced new unified cross-domain continuous resource scheduling methods.The ESB-FDDPG framework enables multiple data centers on cloud platforms to satisfy a required set of conditions over time. The method employs federated learning that combines with deep reinforcement learning to optimally allocate resources while managing and minimizing energy usage and enhancing overall performance within a service-level agreement and privacy constraints.

The organization of the research includes five sections. Phase 1: discuss the introduction of the research. Phase 2 includes related works based on resource scheduling over cloud environments. Phase 3 contains the methodological framework that includes the overall view of the multi-data centers on cloud platforms for resource scheduling. Phase 4 includes the results and discussion. Phase 5 involves the conclusion of the research.

2 Relevant works

Mao et al. proposed a novel strategy for managing Task Management and Virtual Machine Control (TM-VMC), which improved total energy usage while preventing hot spots on a global scale. The experimental findings suggested that the strategy could prevent hot spots and minimize energy usage while maintaining minimal service level agreement breaches [10]. Lin et al. distributed a high-definition deep learning or hybrid deep deterministic learning (HDDL) algorithm for job planning and a deep Q-network (DQN) framework used for resource allocation. The suggested framework improved the benchmark method in terms of energy consumption and task delay [11].

Fan et al. used cloud computing, which was quickly evolving, and managing resources in data centers anddistributed computing networks (DCNs)was critical to its performance. This discussion covers advancements in resource management and optimization, including intelligent resource search algorithms and scheduling methods, and suggests integrating AI within (DCNs) [12].

A framework for providing medical services is provided by cloud computing. Subsequently, exchanging medical information with unidentified cloud neighbours could jeopardize private medical service data. A personalized cloud offers a secure method of safeguarding personal healthcare data. Gong et al. suggested a secure cloud-based effective resource planning solution for healthcare facilities in healthcare information systems [13]. Using testing using Google clusters trace information, Malik et al. suggested a multiple-resource forecasting strategy that combines Functional Link Neural Network (FLNN), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). It shows improved accuracy compared to traditional approaches [14].

Dey et al. presented SmartNoshWaste, a multifaceted blockchain-powered architecture that uses cloud technology, QR codes, and reinforced learning to reduce wasted food across the supply chain. The effectiveness of the structure was assessed using actual data from the nosh app, showing a 9.46% decrease in food wasted over the initial data [15].Cloud computing constitutes an internet-based innovation where optimizing resources is essential. A resource administration strategy for the Bin-Packing challenge is presented by Nehraand Kesswani, to enable the allocation of VMs to Physical Machines (PMs) in a cost-effective way. The CloudSim Plus Simulator experiment findings show better resource utilization, less energy usage, and fewer engaged PMs [16].

The Bin-Packing challenge is addressed in Vijayaand Srinivasan, a resource administration approach for the effective distribution of VMs among Physical Machines (PMs). The results of simulations conducted with CloudSim show enhanced usage of resources, decreased energy usage, and a decrease in outstanding PMs [17].Cloud computing needs to consider the environment and energy consumption without sacrificing service excellence. As a means to reduce energy consumption and carbon emissions, Buyya et al. suggested a learning-focused management strategy for workloads and combining resources. It also presents a conceptual design that increases ecological and energy conservation [18].

Table 1 represents a summary of existing resource scheduling methods.

Table 1: Summary of existing resource scheduling methods

Reference	Methodology	Metrics	Dataset	Reported results
Mao et al. (2023) [10]	Thermal-aware scheduling using Task Management and Virtual Machine Control (TM-VMC)	Energy consumption, SLA breaches	Synthetic cloud data	Energy reduced by ~7.9%, SLA violations <0.005%
Lin et al. (2020) [11]	High-definition Deep Learning (HDDL) + Deep Q-Network (DQN)	Energy consumption, task delay	Google cluster traces	Energy reduced by 5.7–9.7%, delay improved
Fan et al. (2020) [12]	Locality-based scheduling in Data Center Networks (DCNs)	Latency, throughput	DCN simulations	Improved latency and throughput
Gong et al. (2022) [13]	Secure cloud-based resource management for hospital systems	Resource allocation efficiency	Healthcare data	Operational cost reduced by 23%
Malik et al. (2022) [14]	Functional Link Neural Network (FLNN) + GA + PSO for resource prediction	Prediction accuracy, resource utilization	Google cluster traces	Higher prediction accuracy vs benchmarks
Nehra & Kesswani, (2023) [16]	Load-balanced multi-dimensional Bin-Packing heuristic	Resource utilization, energy consumption	CloudSim Plus simulations	Better resource use and fewer active servers
Vijaya & Srinivasan, (2024) [17]	Multi-objective meta-heuristic for VM placement (ACO + SCA)	Energy consumption, CPU utilization	Cloud simulations	Improved energy efficiency
Buyya et al. (2024) [18]	Integrated energy management strategy	Energy use, carbon emissions	Data center workload simulations	Conceptual vision, no numerical results

2.1 Research gap

Despite several advancements in cloud resource scheduling, existing methods exhibit notable drawbacks. Mao et al.improved energy efficiency but rely on static scheduling that does not adapt well to dynamic workloads [10]. Lin et al.enabled task delay and energy consumption using DL but lack continuous scheduling and privacy-preserving mechanisms for multi-data center environments [11]. Fan et al.concentrated on locality-based scheduling; nonetheless, the method has issues with large-scale implementation and energy efficiency [12]. Although they don't directly optimize real-time scheduling, Malik et al.increased prediction accuracy for resource consumption [14]. Although Vijaya and Srinivasan, used meta-heuristic virtual machine placement to increase energy efficiency, they are unable to expand efficiently or dynamically manage workload fluctuations [17].

The ESBO-FDDPG architecture, which offers continuous, privacy-preserving, and scalable resource scheduling across many cloud data centers while enhancing latency, energy efficiency, and task completion rates, is motivated by these gaps.

3 Materials and methods

The methodology section clearly explains the process of data transmission between the vehicle users and the multi-edge computing server (MEC) server (roadside unit (RSU) with the help of resource scheduling using ESBO-FDDPG, also describes the data collection, preprocessing using Min-Max scaling, and the feature extraction approach, which includes statistical pattern recognition (SPR) to identify workload data patterns. The step-by-step process is shown in Figure 1.

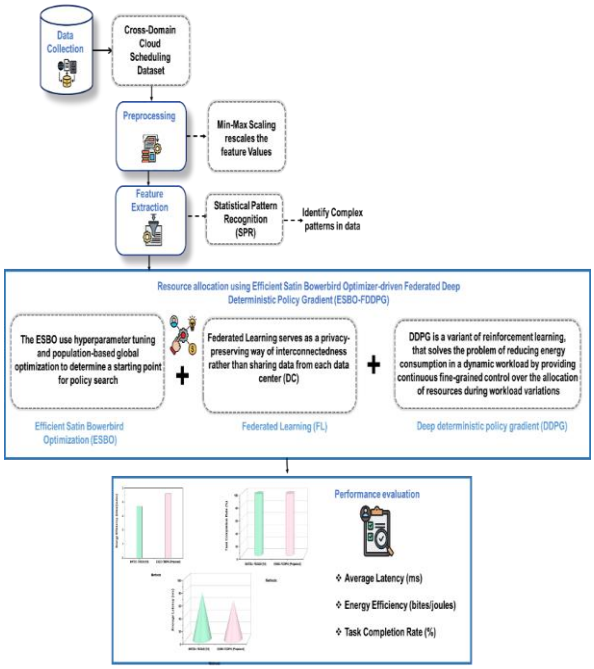

Figure 1: Methodological framework

3.1 Data collection

The experimental evaluation in this research is conducted using the Cross-Domain Cloud Scheduling Dataset (https://www.kaggle.com/datasets/programmer3/cross-domain-cloud-scheduling-dataset), which provides workload and energy-related metrics for simulating multi-data center scheduling scenarios. Each record corresponds to a cloudlet (task) scheduled on a virtual machine (VM) and contains both input parameters and execution outcomes, making it suitable for analyzing cross-domain scheduling policies. The dataset includes key attributes such as CloudletLength (task length in million instructions), CloudletFileSize and OutputSize (I/O data size), NumOfPes (number of processing units allocated), and utilization models for CPU, RAM, and bandwidth. It also tracks Vmld (assigned VM), ActualCPUTime (execution time), and Status (Success/Failure).

This dataset was selected because it provides diverse workload traces across computational, storage, and bandwidth dimensions, reflecting the heterogeneous demands in real cloud environments. Its inclusion of both utilization behaviours and outcome measures enables performance evaluation in terms of latency, task success rate, and energy efficiency, which align directly with the goals of the proposed ESBO-FDDPG model. To ensure replicability, preprocessing involved Min-Max normalization of continuous attributes, SPR-based feature extraction (FFT-RID kernel), and a 70:15:15 split into training, validation, and testing sets. This choice allows consistent benchmarking while preserving generalization across multiple data centers.

3.1.1 Preprocessing using Min-Max scaling

The preprocessing technique used to normalize data while rescaling feature values to a certain range is preferable within$[0,1]$. The scaling is performed using Eq. (1):

$$W_{scaled} = \frac{W - W_{min}}{W_{max} - W_{min}} \quad (1)$$

Where$W$ stands for the cross-domaindata point when the characteristic. $W_{min}$is the number that determines the minimum value of the feature. W_{max}represents the value of the maximal feature. The result, W_{scaled}, is the transformed value for the given data that is in the range of$W$. The variables in a dataset are measured on different scales or units to prevent them from being submerged because of their higher value. It has large or small values in the data that affect the scaling of other values.

`SPR plays a crucial role in handling heterogeneous and noisy workload traces that arise in cloud data centers. By analyzing global frequency trends (via FFT) and localized temporal variations (via RID-based TDA), the method captures sudden surges or drops in demand more accurately than conventional time-domain analysis. This dual-domain approach ensures that the scheduler can detect recurring periodic workloads (e.g., daily or weekly peaks) as well as unexpected spikes, leading to more robust resource allocation. Furthermore, SPR reduces dimensionality while retaining the most relevant workload features, which directly benefits the ESBO-FDDPG model's learning efficiency. SPR is a feature extraction technique used in fields like diagnostics, speech recognition, and data analysis to identify complex patterns in sensor data. It helps in cross-domain continuous resource scheduling problems in cloud platforms by identifying temporal patterns in physiological and behavioral measures. SPR is particularly useful in investigating sensor data captured under various academic and environmental conditions. SPR used two methods, Fast Fourier Transform (FFT) and Time-Frequency Analysis (TFA), for better feature extraction from time-series data, enabling efficient management with noisy inputs and adaptation to sudden changes in monitored electronics assembly.

FFT for Feature Extraction in the Frequency Domain: FFT is a mathematical technique that converts a time-domain signal into its frequency components. It decomposes complex workload signals into sinusoidal components, revealing periodic demand cycles that help identify recurring task patterns in cloud environments. By mapping the signal into the frequency spectrum, FFT enables the system to anticipate regular fluctuations in resource demand.FFT is used in the field to extract frequency-domain features from time-series data to capture any periodic characteristics that can be related to fluctuating resource demand. It was used in Eq. (2).

$$X(f) = \sum_{n=0}^{N=1} x(n) e^{j2\pi fn/N} \quad (2)$$

The RID kernel is a specialized windowing function used in TFA to analyse workload signals simultaneously in the time and frequency domains. Unlike FFT, which loses time resolution, the RID kernel captures localized variations, making it effective for identifying transient spikes in demand. This ensures the scheduling model remains responsive to short-term irregular workloads as well as long-term cycles.

Time-Frequency Analysis using resource identifier (RID)kernel:Using an RID kernel for TFA enables the extraction of features that provide information about the localized responses from the data over both time and frequency that could help detect low-frequency variations that signify the resource demand by Eq. (3).

$$S(t,f) = \left| \int_{-\infty}^{\infty} x(\tau) e^{j2\pi f\tau}\, w(t-\tau) d\tau \right|^2 \quad (3)$$

In Eq. (2), $x(n)$ represents the discrete-time signal at index n, f denotes the frequency, and N is the total number of samples in the signal. In Eq. (3), $S(t,f)$ is the time-frequency representation of the signal, $w(t-\tau)$ is the window function for time localization, and τ is the integration variable in the continuous-time analysis.Using these SPR methods, the system monitors and predicts low resource demand and thus provides a real-time alert for resource scheduling.

3.1.2 Power Spectral Density (PSD) using FFT: PSD is used to quantify how the power of a signal is distributed across different frequencies. In cloud workload analysis, PSD highlights dominant periodic patterns in task arrivals or resource demand.

$$O(e) = \frac{1}{M} |W(e)|^2 \quad (4)$$

In Eq. (4), where $O(e)$is the power spectral density at frequency $W(e)$ is the FFT of the signal, and M is the total number of samples. This quantifies the strength of workload demand variations at different frequencies.

3.1.3 Short-Time Fourier Transform (STFT) for Localized Analysis: STFT provides a time-frequency representation of a signal. It is particularly useful in cloud scheduling to capture transient bursts in demand while preserving local time information.

$$STFT(s,e) = \sum_{m=-\infty}^{\infty} w(m)\omega(m-s) f^{-i2\pi em} \quad (5)$$

In Eq. (5), $STFT(s,e)$ represents the time-frequency representation of the signal at time s and frequency f, $\omega(m-s)$is the sliding window function for localizing the signal in time, i is the imaginary unit, πem represents the phase rotation corresponding to the frequency component e at time m., and $w(m)$ is the discrete-time workload signal. This helps detect temporary demand bursts in specific time intervals.

3.2 Resource allocation using efficient satin bowerbird optimizer-driven federated deep deterministic policy gradient (ESBO-FDDPG)

The ESB-FDDPG hybrid system uses three methodologies tobetter manage cloud resources. The efficient satin bowerbird optimizer (ESBO) uses hyperparameter tuning and population-based global optimization to determine a starting point for policy search. Federated Learning (FL) serves as a privacy-preserving way of interconnectedness rather than sharing data from each data center (DC). Deep deterministic policy gradient (DDPG) is a variant of reinforcement learningthat solves the problem of reducing energy consumption in a dynamic workload by providing continuous, fine-grained control over the allocation of resources during workload variations. This hybrid system combines ESB's global optimization methodology, federated learning's data privacy, and DDPG's adaptability to provide a more energy-efficient scheduling system while managing resources across multiple heterogeneous distributed cloud systems.

Efficient Satin Bowerbird Optimization (ESBO):The ESBO algorithm is a metaheuristic that finds a globally optimal solution to multi-datacenter resource scheduling problems. It is a simple, robust, and efficient population-based method that generates candidate solutions within defined limits, guiding the best chances of optimal resource allocation across data centers. Eqs. (6) and (7) define how candidate solutions are evaluated and probabilistically selected, forming the foundation for the iterative optimization in ESBO.

$$Prob_j = \frac{fit_j}{\sum_{m=1}^{NB} fit_m} \quad (6)$$

$$fit_j = \begin{cases} \frac{1}{1+e(w_j)}, & e(w_j) \geq 0 \\ 1 + |e(w_j)|, & e(w_j) < 0 \end{cases} \quad (7)$$

$Prob_j$is the probability of selecting the j^{th}solution based on fitness, fit_j is the fitness of the j^{th}solution,m is the index of each candidate solution in the population, and NB is the total number of solutions. NB is the error value for the j^{th}solution.$w_j$represents the decision variable of the j^{th}candidate solution. $e(w_j)$ is the error or deviation from the optimal allocation for that candidate.

The ESBO approach uses exclusivity to find optimal solutions, based on satin bowerbirds' behavior. The attractiveness of the bower is determined by decorations, allowing for optimization and effective resource scheduling across cloud systems as follows in Eq. (8).

$$w_{jl}^{new} = w_{jl}^{old} + \lambda_l \left(\left(\frac{w_{jl} + w_{elite,l}}{2} \right) - w_{jl}^{old} \right) \quad (8)$$

In the given Eq. (15), w_{jl}^{new} represents the updated weight of the j^{th} solution in the l^{th} dimension, w_{jl}^{old} is the old weight of the j^{th} solution in the l^{th} dimension, $w_{jl}$is the positive weight component, and $w_{elite,l}$ is the elite solution weight in the l^{th} dimension. λ is the step size factor in the l^{th} dimension.

In the mutation process after each round of the ESBO method, the solution w_{jl}^{old} is randomly adjusted according to defined probabilities. This mutation process uses a standard normal M distribution, and the new solution $w_{jl}^{new}$is is created based on the mean of the previous solution w_{jl}^{old} and a defined variance σ^2. The mutation is defined mathematically as in Eqs. (9-11).

$$\lambda_l = \frac{\alpha}{1+o_i} \tag{9}$$

$$w_{jl}^{new} \sim M\left(w_{jl}^{old}, \sigma^2\right) \tag{10}$$

$$M\left(w_{jl}^{old}, \sigma^2\right) = w_{jl}^{old} + \left(\sigma^* M(0,1)\right) \tag{11}$$

Where $M\left(w_{jl}^{old}, \sigma^2\right)$ is the normal distribution, in which the values resemble some random variation drawn from a mean of the normal distribution and the variance σ of the normal distribution. It adds diversity to the population and ultimately helps the algorithm effectively explore the solution space so it can schedule resources across the data centers optimally.

Federated Learning (FL): In the proposed framework, each data centre independently trains a local DDPG model using its own workload data, ensuring privacy is preserved. During training, the adaptive loss function evaluates resource scheduling errors, such as underutilization or overutilization, guiding the optimization process locally. Once local updates are completed, the gradients are aggregated using a federated averaging strategy to form a unified global model. This global model is then redistributed back to each data center, allowing consistent scheduling decisions across all domains. Both the actor and critic networks in DDPG are updated through this process, ensuring that the policy and value estimation remain aligned for efficient cross-domain resource allocation.This sub-section discusses resource scheduling using FL in a multi-data center, focusing on local FL models trained independently and updated globally. The adaptive loss function measures scheduling penalties of utilization errors, ensuring proper scheduling over the distributed functional block. The FL system consists of 5 geographically distributed data centers, each acting as a client. Each client trains its local model for 10 epochs before sending updates. Global model aggregation is performed every 5 local training rounds. In total, 50 communication rounds are executed to update the global model across all clients.FL optimizes resources for cloud consumption, providing energy efficiency, response times, and performance in Eqs. (12 and 14).

$$E_l(x_m) = \frac{1}{|\mathcal{D}_l|}\sum_{j\in\mathcal{D}_l} \mathrm{K}\left(\mathrm{f} \ominus (\mathrm{x}_l), \mathrm{y}_l\right), \forall l \in \mathcal{K} \tag{12}$$

$$E(x_m) \triangleq \sum_{l\in\mathcal{K}_m} \frac{|\mathcal{D}_l| E_l(x_m)}{|\mathcal{D}_m|} = \frac{1}{|\mathcal{D}_m|}\sum_{l\in\mathcal{K}_m}\sum_{j\in\mathcal{D}_l} e\left(x_m; t_{l,j}, y_{l,j}\right) \tag{13}$$

$$x_m^* = \arg\min E(x_m), \quad \forall m \in \mathcal{N} \tag{14}$$

$\mathcal{D}_l$represents *the* total number of data points in the local dataset $\mathcal{D}_l$of the l^{th} data center. $j \in \mathcal{D}_l$ iterates over each data point j in the local dataset. $\mathrm{K}\left(\mathrm{f} \ominus (\mathrm{x}_l), \mathrm{y}_l\right)$ is the loss function comparing model prediction $f(\mathrm{x}_l)$ and the true label y_l. $l \in \mathcal{K}$ iterates over all local model centers in the set K. $\mathcal{K}_m$is the set of data centers contributing to the global modal x_m. $\mathcal{D}_m$is the total number of samples across all contributing data centers. $t_{l,j}$represent input features of the j^{th} data point in the l^{th} data center. $x_m^*$is the optimal global model parameters after federated aggregation. $\mathcal{N}$ is the total number of candidate models. $\arg\min E(x_m)$selects the model x_m that minimizes the global loss $E(x_m)$.

In scheduling resources across geographically distributed data centers, it is not possible to compute the global model parameters $E(x_m)$ using the raw datasets corresponding to the local resource units for privacy and security, by Eq. (12).The solution here proposes the concept of model averaging in the resource scheduling process by utilizing FL to maintain the data privacy of each unit. The distributed approach can be applied using a gradient-averaging approach, similarly to optimize resource allocation across geographically distributed data centers securely and cost-effectively, utilizing the sensitivity of their local data while also maintaining scalability and performance.

Deep deterministic policy gradient (DDPG): The method can solve distributed resource scheduling problems in multiple data centers. Using the Q-learning method, it generates a value function right arrow$q$by stating the value that can be obtained by policy πat states t^l in. The value function provides a measurement of the effectiveness of scheduling resources to multiple data centers, taking into consideration infinite horizons and discounting in the DDPG in Eq. (15).

$$U^\pi(t) = \mathbb{E}_\pi[\sum_{l=0}^{\infty} \gamma . q^l\left(t^l, b^l\right) | t^0 = t] = \mathbb{E}_\pi[q^l(t^l, b^l) + \gamma . U^\pi(t^{l+1}) | t^0 = t] \tag{15}$$

To find the optimal policyπ, the optimal action in each state can be obtained from the optimal value function. This can be depicted as Eq. (16).

$$U^*(t) = \max_{b^l}\{\mathbb{E}_\pi[q^l(t^l, b^l) + \gamma . U^\pi(t^{l+1})]\} \tag{16}$$

Where$U^*(t)$is the optimal action-value function for l^{th}state t^l and l^{th} action b^l, to provide insight into which action is the best choice for an optimal allocation of resources in each data center.

However, the classical Q-learning algorithm could not effectively find the optimal policy when the state-action space is large or continuous, and it computes Q-values on the Q-table. Long computational time with multi-data center resource scheduling, high-dimensional data was used to optimize the representation of DDPGand implement a neural network model with Q-value $R^*(t^l, b^{l+1}, \theta)$, with θ being the weights of the model DDPG. After comparing the Q-value for each possible action, the optimal scheduling policy $\pi$for effective resource scheduling can now be represented in Eqs.(17-20).

$$R^{l+1}(t^l, b^l) = R^l(t^l, b^l) + \alpha[q^l(t^l, b^l) + \gamma max R^l(t^l, b^{l+1}) - R^l(t^l, b^l)] \quad (17)$$

$$\pi^*(t) = \arg\max_{b^l} R^*(t^l, b^{l+1}, \theta) \quad (18)$$

$$\tilde{R}(t^l, b^l, \omega) = q(t^l, b^l, \omega) + \gamma max_{b^{l+1}}[R(t^{l+1}, b^{l+1}, \theta)] \quad (19)$$

$$K = F\left[\left(\tilde{R}(t^l, b^l, \omega) - R(t^{l+1}, b^{l+1}, \omega)\right)^2\right] \quad (20)$$

The ESB-FDDPG hybrid system employs three approaches: ESBO, FL, and DDPG. It optimizes cloud computing resources by utilizing hyperparameter tuning, population-based optimization globally, and data security. The ESBO-FDDPG method pseudocode is provided in Algorithm 1.

Algorithm 1: ESBO-FDDPG

Input:
- $N = 5$ *agents/nodes*
- *Local environments* $\{Env\_1, \ldots, Env\_5\}$
- *Actor network* π_θ *(2 hidden layers: 64 neurons each, ReLU)*
- *Critic network* Q_φ *(2 hidden layers: 64 neurons each, ReLU)*
- *Replay buffer size* $B\_i = 10000$
- *Satin Bowerbird Optimizer (SBO):*
  - *Population = 10 candidates*
  - *Iterations = 5 per federated round*
- *Federated aggregation rounds R = 50*
- *Hyperparameters:*

$\gamma = 0.99$ *# Discount factor*
$\tau = 0.005$ *# Soft target update*
$lr\_actor = 0.001$
$lr\_critic = 0.002$
Mini-batch size = 64
Exploration noise σ = 0.1

Output:
- *Federated global actor network* π_θ_global
- *Federated global critic network* Q_φ_global

1. Initialize actor π_θ_i *and critic* Q_φ_i *for each agent i*
2. Initialize target networks: $\pi\_\theta\_target\_i \leftarrow \pi\_\theta\_i, Q\_\varphi\_target\_i \leftarrow Q\_\varphi\_i$
3. Initialize local replay buffers $Bi \leftarrow \emptyset$
4. For each Federated Round $r = 1$ *to 50 do:*
4.1 For each Agent i in parallel:
a) Interact with local environment Env_i *for* $T = 1000$ *steps:*
- Select action: $a\_t = \pi\_\theta\_i(s\_t) + N(0, 0.1)$ *# Gaussian noise*
- Observe reward r_t *and next state* $s\_\{t+1\}$
- Store $(s\_t, a\_t, r\_t, s\_\{t+1\})$ *in Bi*
b) Sample mini-batch of 64 transitions from Bi
c) Update Critic Q_φ_i *using:*

$$yt = rt + 0.99 * Q\_\varphi\_target\_i(s\_\{t+1\}, \pi\_\theta\_target\_i(s\_\{t+1\}))$$

Loss = $MSE(Q\_\varphi\_i(s\_t, a\_t), y\_t)$
d) Update Actor π_θ_i *using policy gradient:*

$$\nabla\theta\_i J \approx 1/64 * \Sigma \nabla\_a Q\_\varphi\_i(s, a)|\_\{a = \pi\_\theta\_i(s)\} \nabla\theta\_i \pi\_\theta\_i(s)$$

e) Soft update target networks:

$$\theta\_target\_i \leftarrow 0.005 * \theta\_i + 0.995 * \theta\_target\_i$$
$$\varphi\_target\_i \leftarrow 0.005 * \varphi\_i + 0.995 * \varphi\_target\_i$$

4.2 Federated Aggregation:
- Collect π_θ_i *and* Q_φ_i *from all agents*
- Aggregate globally:

$$\theta\_global = (\Sigma\ \theta\_i) / 5$$
$$\varphi\_global = (\Sigma\ \varphi\_i) / 5$$

- Broadcast θ_global *and* φ_global *to all agents*
4.3 ESBO Hyperparameter Optimization (Optional):
- Initialize SBO population with 10 hyperparameter sets $(lr\_actor, lr\_critic, \tau)$
- For $t = 1$ *to 5 iterations:*
a) Evaluate candidate hyperparameters on local agent performance
b) Update candidates using SBO mating/selection rules
- Update best hyperparameters for next federated round
5. Return θ_global, φ_global

The ESBO-FDDPG framework described in Algorithm 1 combines an FDDPG approach to cross-domain resource scheduling with the SBO. Each of the five agents uses environment-related data to autonomously train local actor-critic networks while maintaining anonymity. Using federated averaging, the local upgrades are combined to create a global framework that improves scheduling effectiveness. Every round, SBO dynamically adjusts hyperparameters like developing and updating rates to guarantee optimal performance and stable convergence. The hyperparameter tuning (Table 2) strategy follows the adaptive optimization mechanism of the SBO integrated within the federated reinforcement learning framework.

Table 2: Hyperparameter configuration and SBO search ranges used in ESBO-FDDPG training

Hyperparameter	Symbol	Default Value	SBO Search Range	Description
Discount Factor	γ	0.99	[0.90, 0.999]	Weight of future rewards in the critic update.
Soft Target Update Rate	τ	0.005	[0.001, 0.01]	Interpolation factor for soft update of target networks.
Actor Learning Rate	lr_actor	0.001	[0.0001, 0.01]	Learning rate for the actor network optimizer.
Critic Learning Rate	lr_critic	0.002	[0.0001, 0.01]	Learning rate for the critic network optimizer.
Replay Buffer Size	Bi	10,000	Fixed	Size of experience replay memory per agent.
Mini-Batch Size	$batch\_size$	64	[32, 128]	Number of samples used for each gradient update.
Exploration Noise Std Dev	σ	0.1	[0.05, 0.3]	Standard deviation of Gaussian noise for action exploration.
Actor Network Hidden Units	$hidden\_a$	64	[32, 128]	Number of neurons per hidden layer in the actor network.
Critic Network Hidden Units	$hidden\_c$	64	[32, 128]	Number of neurons per hidden layer in the critic network.
Number of Agents	N	5	Fixed	Number of federated nodes training locally.
Federated Rounds	R	50	Fixed	Number of communication rounds for federated aggregation.
SBO Population Size	pop_size	10	[5, 20]	Number of candidate hyperparameter sets in Satin Bowerbird Optimizer.
SBO Iterations per Round	$iter\_sbo$	5	[3, 10]	Number of SBO iterations per federated round for hyperparameter optimization.
Mutation Rate	μ	0.05	[0.01, 0.1]	Probability of random variation in candidate solutions during optimization.

4 Result and discussion

In this section, the unified cross-domain continuous resource scheduling approach for multi-data center environments is presented. The proposed method is evaluated based on some key metrics. Also, some comparisons to standard models highlight the improvements provided by the proposed model to combine resource scheduling on a cross-domain basis in distributed cloud environments.

4.1 System configuration

The experimental setup for the unified resource scheduling system includes Python 3.8.10 with libraries including Tensorflow, PyTorch, and NumPy. The hardware configuration (Intel Core i7 9th Gen, 16GB RAM (3200 MHz))is for enhanced resource scheduling.

4.2 Research output

Figure 2 demonstrates the energy consumed by method ESBO-FDDPG compared to frequency and shows how ESBO-FDDPG reduces energy consumed. It provides evidence of ESBO-FDDPG's slightly higher energy efficiency compared to baseline methodsin resource scheduling across multiple data centers, also making an energy-efficient cloud system.

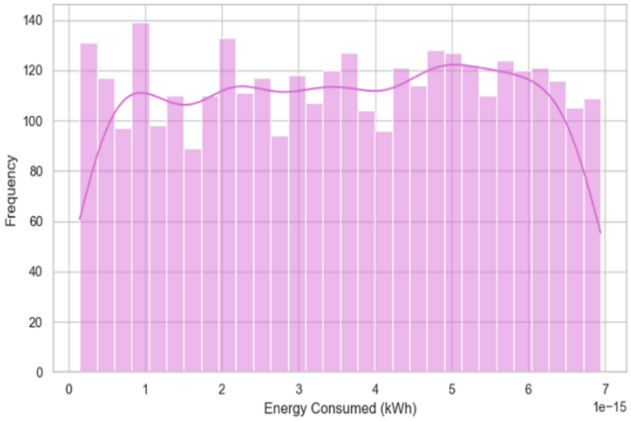

Figure 2: Energy consumption distribution across 5 geographically distributed data centers under varying workload intensities, measured during ESBO-FDDPG resource scheduling simulations.

Figure 3 exhibits the distribution of task data size processed by the proposed ESBO-FDDPG method and the distribution of task data sizes from different data centers. It indicates the differing types of

workloads processed by the proposed method in a cloud computing environment.

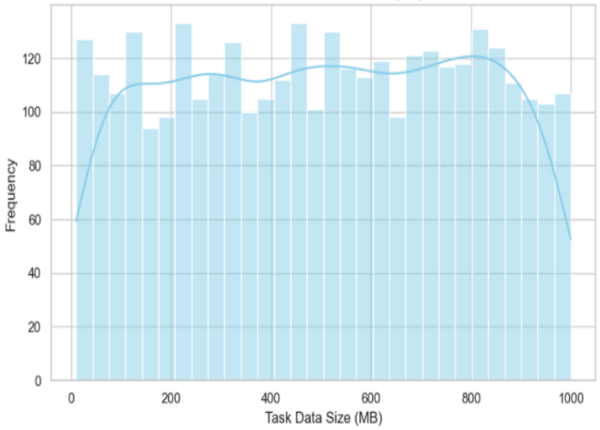

Figure 3: Distribution of task data sizes processed by multiple cloud data centers in the ESBO-FDDPG simulation environment, highlighting variations under dynamic workloads.

Figure 4 shows the distribution of CPU cycles used by the tasks in the ESBO-FDDPG system. This shows variation in CPU cycles across the tasks, indicating that the system can allocate and manage the CPUs efficiently, as the workload varies across multiple connected data centers in a cloud environment.

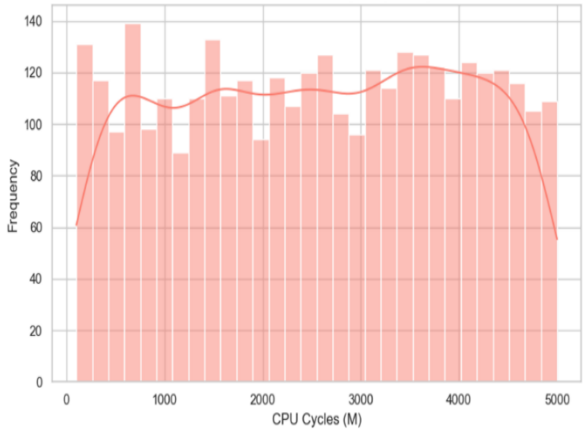

Figure 4: Distribution of CPU cycles required per task across all data centers, showing workload heterogeneity and resource allocation efficiency using ESBO-FDDPG.

Figure 5 shows bandwidth consumption distribution under the ESBO-FDDPG framework, illustrating network resource allocation relative to scheduling. This demonstrates system capacity, optimal performance, and efficient resource distribution across cloud systems.

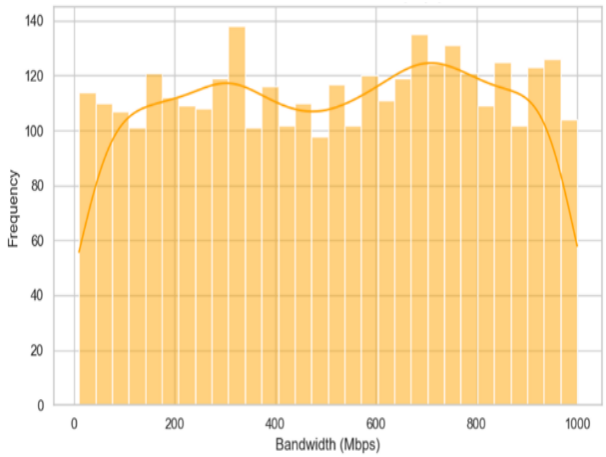

Figure 5: Bandwidth usage distribution across data centers during task scheduling, demonstrating network resource utilization under different workload scenarios.

4.3 Comparison analysis

The proposed ESBO-FDDPG model was evaluated against three benchmark methods such MATD3-FD3QN [19], Deep Deterministic Policy Gradient (DDPG) and Federated-Deep Deterministic Policy Gradient (F-DDPG) to assess performance in multi-data center cloud environments. The comparison was conducted using three key indicators: Average Latency (ms), Energy Efficiency (bits/joule), and Task Completion Rate (%). As shown in Table 3.The following metrics of the findings are given below.

Average Latency (ms): Identifies latencies as task processing takes place across the system, continuing until some completion response using Eq. (19).

$$\text{Average Latency (ms)} = \frac{\sum_{i=1}^{n} T_i}{n} \tag{19}$$

Where T_i is the processing time (in milliseconds) of the i^{th} task, and n is the total number of tasks.

Energy Efficiency (bits/joules): Indicates the number of bits processed per joule of electrical energy by Eq. (20).

$$\text{Energy Efficiency } \left(\frac{\text{bit}}{\text{s joules}}\right) = \frac{Total\ bits\ processed}{Total\ Energy\ Consumed\ (J)} \tag{20}$$

Here, "$Total\ bits\ processed$" represents the cumulative data handles, and "$Total\ Energy\ Consumed$" is in joules (J).

Task Completion Rate (%): The percentage of tasks completed within a reasonable time using Eq. (21).

$$\text{Task Completion Rate (\%)} = \frac{Completed\ Tasks}{Total\ Tasks} \times 100 \tag{21}$$

Where "$Completed\ Tasks$" are those finished within the desired SLA time, and "$Total\ Tasks$" is the total number of submitted tasks.

Table 3: Comparison of key performance indicators in cloud-based scheduling methods

Method	Average Latency (ms)	Energy Efficiency (bits/joule)	Task Completion Rate (%)
MATD3–FD3QN [19]	75	3.7×10^5	98.5
DDPG	71	3.9×10^5	98.6
F-DDPG	67	4.2×10^5	98.7
ESBO-FDDPG (Proposed)	63	4.6×10^5	98.8

Table 3 and Figure 6 illustrate the average latency comparisons between two resource scheduling techniques. The MATD3-FD3QN method had an average latency of 75 ms, while DDPG achieved 71 ms, F-DDPG further improved it to 67 msand the proposed ESBO-FDDPG method had an average latency of 63 ms. This shows important improvements in latency and thus resource scheduling with the use of the ESBO-FDDPG method, and overall improvement in the system's responsiveness in the cloud system.

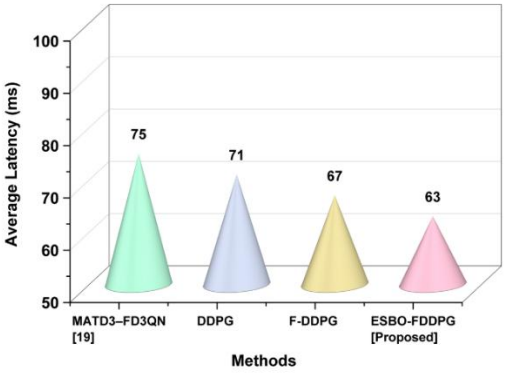

Figure 6: Comparison of average latency between different scheduling methods

Figure7 and Table 3 define the energy efficiency of MATD3-FD3QN [19] and the proposed ESBO-FDDPG method. Efficient energy utilization for resources allocated to the scheduling of multi-data centers with ESBO-FDDPG, as it has an energy efficiency (4.6×10^5) bits joule compared to MATD3-FD3QN, which has an energy efficiency (3.7×10^5) bits joule, while DDPG improved it to 3.9×10^5 bits/joule, and F-DDPG further enhanced performance with 4.2×10^5 bits/joule.

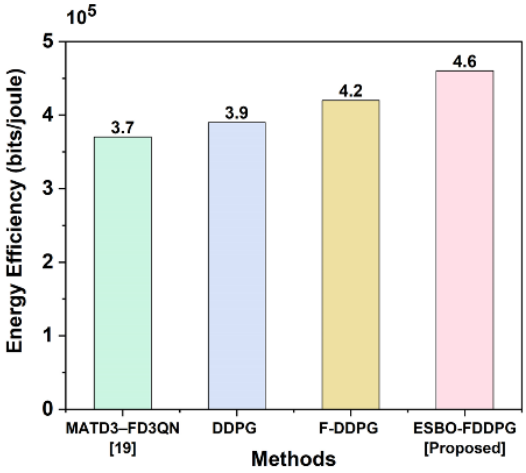

Figure 7: Scheduling algorithm performance of energy efficiency

Figure 8 and Table 3 show that the proposed ESBO-FDDPG method achieved a task completion rate of 98.8% compared to 98.5% for the baseline MATD3-FD3QN,followed by DDPG with 98.6%, and F-DDPG with 98.7%.indicating a modest improvement.The proposed method is marginally better at completing tasks due to improved efficiency in resource scheduling and improved performance when workloads fluctuate in multi-data center cloud platforms.

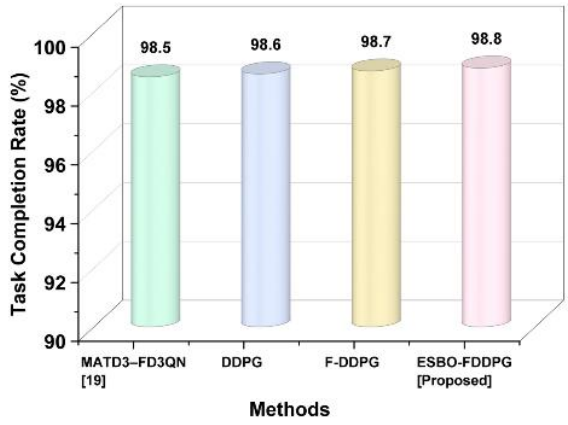

Figure 8: Task completion rate improvement observed in ESBO-FDDPG vs MATD3-FD3QN

The MATD3-FD3QN [19] algorithm has limitations, such as reliance on specific infrastructure, fixed RSUs, and mobile UAVs, which may not be realistic in all vehicular networks. It's also not suitable for deployment in dense urban environments. The algorithm's scalability could be demonstrated through testing across diverse networks. It is sensitive to network variability and task arrival patterns, potentially affecting its applicability in broader contexts; the proposed ESBO-FDDPG method helps improve some of these limitations as it provides more flexible designs in multiple contexts, such as highly populated urban

areas, and scalability compared to classical methods, and it also adjusts more appropriately with variability in network and during shifts in task arrival time.

A quantitative comparison between the suggested ESBO-FDDPG method and current cloud scheduling approaches is shown in Table 4. Every prior research focuses on particular optimization parameters, such as CPU utilization accuracy, energy consumption, or execution time. The usefulness of the suggested ESBO-FDDPG in preserving balanced performance in cross-domain situations is validated by its superior task completion rate of 98.0%.

Table 4: Comparative Summary of State-of-the-Art (SOTA) Cloud Scheduling Methods and the Proposed ESBO-FDDPG Model

Model / Study	**Approach / Architecture**	**Metrics**	**Reported Findings / Values**
Mao et al. (2023) [10]	Thermal-aware scheduling using TM-VMC	Execution Time (ms)	25–44 ms
Lin et al. (2020) [11]	Two-stage job scheduling with DQN, FERPTS, MLF	Energy Consumption (%)	5.7–9.7%
Malik et al., 2022	FLNN + Evolutionary Algorithms for resource utilization prediction	MAE (CPU)	0.25–0.36
Nehra & Kesswani, (2023) [14]	Load-balanced multi-dimensional Bin-Packing heuristic	Energy Reduction (%)	5.46–7.78%
Vijaya & Srinivasan, (2024) [17]	Multi-objective meta-heuristic (ACO + SCA) for VM placement	Power (W)	17294–33654 W
ESBO-FDDPG [Proposed]	ESBO + Federated DDPG for cross-domain scheduling	Task Completion Rate (%)	98.8%

Discussion

Blockchain and cloud computing were highlighted by author of [15] for smart city technologies; however, their emphasis was on applications related to waste minimization rather than scalability, scheduled effectiveness. An energy-efficient multi-objective meta-heuristic for virtual machine deployment was proposed by work of [17], although it came with greater latency compromises in unpredictable workloads. Energy conservation and effective resource administration are vital, as research of [18] pointed out, although their work was mainly theoretical and lacked specific optimization outcomes. The suggested ESBO-FDDPG outperforms current methods with a latency of 63 ms, an energy efficiency of 4.6 × 10^5 bits/joule, and a task completion rate of 98.8%. These findings demonstrate that an approach guarantees real-time flexibility across several data facilities in addition to achieving the long-term objectives anticipated in earlier work. Therefore, when contrasted with the state-of-the-art techniques now in use, ESBO-FDDPG provides a more useful and effective approach.

While ESBO-FDDPG demonstrates notable improvements in energy efficiency and latency, the increase in task completion rate could be less significant under highly unpredictable or extreme workloads. The simulations used could not fully capture the complexity of real-world heterogeneous cloud infrastructures. Additionally, the federated learning process introduces slightly higher computational overhead due to model aggregation rounds. These factors suggest that performance gain could vary depending on deployment scenarios and system scales.

ESBO-FDDPG effectively addresses federated DRL challenges by enabling privacy-preserving model updates, scaling across heterogeneous data centers, and dynamically adapting to fluctuating workloads, which are limitations in most existing methods.

Conclusion

This research utilized cross-domain data from data centers of a cloud platform to manage data resource scheduling. The data goes through a preprocessing phase through a min-max scaling so that the data can be standardized and better facilitate efficient scheduling. TheSPR method is a feature extraction that was carried out to improve the ability of the model to learn the essential patterns necessary for resource scheduling. The ESB-FDDPG model consisted of coupling deep reinforcement learning with FL to manage computing, storage, and transmission resources, including data centers that aredistributed. The data analysis showed modest improvements in performance metrics, with an average latency of63ms, energy efficiency of (4.6 × 10^5 bites/joule), and task completion rate of (98.8%).

The research has several limitations as well because simulated datasets were used for the evaluation instead of actual multi-data center (multi-DC) logs. This limits the findings' applicability to real heterogeneous infrastructures with varying workloads and hardware diversity (such as CPU, GPU, containers, and virtual machines). Furthermore, different service-level restrictions, including latency, throughput, and energy trade-offs—all of which are crucial in real-world deployments—are not specifically taken into consideration in the current research. By combining adaptive mechanisms for various SLA requirements, incorporating heterogeneous resource settings, and verifying the model on real-world cloud traces, these shortcomings can be addressed in future work. Additionally, expanding the framework to include edge-cloud orchestration would improve responsiveness and scalability in distributed computing situations.

Funding：Science and Technology Projects of Yunnan Power Grid Co., Ltd.，Research and Application of Key Technologies for Cross-domain Continuous Operation of Cloud Platforms of China Southern Power Grid，YNKJXM20240066.

References

[1] Ali, S., Wadho, S. A., Yichiet, A., Gan, M. L., & Lee, C. K. (2024). Advancing cloud security: Unveiling the protective potential of homomorphic secret sharing in secure cloud computing. *Egyptian Informatics Journal*, *27*, 100519. doi: https://doi.org/10.1016/j.eij.2024.100519

[2] Tuli, S., Ilager, S., Ramamohanarao, K., &Buyya, R. (2020). Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE transactions on mobile computing*, *21*(3), 940-954. doi: https://doi.org/10.1109/TMC.2020.3017079

[3] Binyamin, S. S., & Ben Slama, S. (2022). Multi-agent systems for resource allocation and scheduling in a smart grid. *Sensors*, *22*(21), 8099. doi: https://doi.org/10.3390/s22218099

[4] Buyya, R., Ilager, S., & Arroba, P. (2024). Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads. *Software: Practice and Experience*, *54*(1), 24-38. doi: https://doi.org/10.1002/spe.2837

[5] Perko, I. (2023). Data sharing concepts: a viable system model diagnosis. *Kybernetes*, *52*(9), 2976-2991.doi: https://doi.org/10.1108/K-04-2022-0575

[6] Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J. (2023). Big data analytics using cloud computing-based frameworks for power management systems: Status, constraints, and future recommendations. *Sensors*, *23*(6), 2952. doi: https://doi.org/10.3390/s23062952

[7] Li, L., & Zhang, H. (2020). Delay optimization strategy for service cache and task offloading in three-tier architecture mobile edge computing system. *IEEE Access*, *8*, 170211-170224.doi: https://doi.org/10.1109/ACCESS.2020.3023771

[8] Anand, J., & Karthikeyan, B. (2025). Dynamic priority-based task scheduling and adaptive resource allocation algorithms for efficient edge computing in healthcare systems. *Results in Engineering*, *25*, 104342.doi: https://doi.org/10.1016/j.rineng.2025.104342

[9] Prasad, V. K., Dansana, D., Bhavsar, M. D., Acharya, B., Gerogiannis, V. C., & Kanavos, A. (2023). Efficient resource utilization in IoT and cloud computing. *Information*, *14*(11), 619.doi: https://doi.org/10.3390/info14110619

[10] Mao, L., Chen, R., Cheng, H., Lin, W., Liu, B., & Wang, J. Z. (2023). A resource scheduling method for cloud data centers based on thermal management. *Journal of Cloud Computing*, *12*(1), 84.doi: https://doi.org/10.1186/s13677-023-00462-2

[11] Lin, J., Cui, D., Peng, Z., Li, Q., & He, J. (2020). A two-stage framework for the multi-user multi-data center job scheduling and resource allocation. *IEEE Access*, *8*, 197863-197874.doi: https://doi.org/10.1109/ACCESS.2020.3033557

[12] Fan, W., He, J., Han, Z., Li, P., & Wang, R. (2020). Intelligent resource scheduling based on locality principle in data center networks. *IEEE communications magazine*, *58*(10), 94-100.doi: https://doi.org/10.1109/MCOM.001.1900324

[13] Gong, S., Zhu, X., Zhang, R., Zhao, H., & Guo, C. (2022). An intelligent resource management solution for hospital information system based on cloud computing platform. *IEEE Transactions on Reliability*, *72*(1), 329-342.doi: https://doi.org/10.1109/TR.2022.3161359

[14] Malik, S., Tahir, M., Sardaraz, M., &Alourani, A. (2022). A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques. *Applied Sciences*, *12*(4), 2160.doi:https://doi.org/10.3390/app12042160

[15] Dey, S., Saha, S., Singh, A. K., & McDonald-Maier, K. (2022). SmartNoshWaste: Using blockchain, machine learning, cloud computing and QR code to reduce food waste in decentralized web 3.0 enabled smart cities. *Smart Cities*, *5*(1),

162-176.doi:https://doi.org/10.3390/smartcities5010011

[16] Nehra, P., &Kesswani, N. (2023). Efficient resource allocation and management by using load balanced multi-dimensional bin packing heuristic in cloud data centers. *Journal of Supercomputing*, *79*(2).doi:https://doi.org/10.1007/s11227-022-04707-w

[17] Vijaya, C., & Srinivasan, P. (2024). Multi-objective meta-heuristic technique for energy efficient virtual machine placement in cloud data centers. *Informatica*, *48*(6).doi:https://doi.org/10.31449/inf.v48i6.5263

[18] Buyya, R., Ilager, S., & Arroba, P. (2024). Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads. *Software: Practice and Experience*, *54*(1), 24-38.doi:https://doi.org/10.1002/spe.3248

[19] Xiao, L., Shan, H., Zhu, J., Mao, R., & Pan, S. (2025). FD3QN: A Federated Deep Reinforcement Learning Approach for Cross-Domain Resource Cooperative Scheduling in Hybrid Cloud Architecture. *Informatica*, *49*(10).doi: https://doi.org/10.31449/inf.v49i10.7114