Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Green Orchestration of Digital Twin Analytics: Joint Optimization of AI, IoT, and 5G/6G Resources

[1]Mohamed Fahad A* [2]Gorkem Barutcu [3]Mohamed Elawadi Elsayed Eid [4]Amr Ezzat Abdou Shallal

IT-District Technology Management, KAFD Development & Management Company, Riyadh, Saudi Arabia

ARTICLEINFO

Received: 02 Nov 2024

Revised: 18 Dec 2024

Accepted: 29 Dec 2024

ABSTRACT

Digital Twin (DT) systems built on the Internet of Things (IoT), artificial intelligence (AI), and 5G/6G networks continuously mirror physical processes, but the energy and carbon cost of keeping them synchronized is rarely optimized end to end. This paper proposes a Green Orchestration framework, the Green Adaptive Multi-Objective Digital Twin Resource Optimization (GAMDTRO) architecture, which jointly optimizes AI model complexity, edge-cloud placement, IoT resource allocation, 5G/6G communication resources, DT synchronization interval, and carbon-aware scheduling in one multi-objective formulation. GAMDTRO couples a Non-dominated Sorting Genetic Algorithm II (NSGA-II) outer loop for structural decisions with a Multi-Agent Proximal Policy Optimization (MAPPO) inner loop for real-time power and bandwidth control, gated by grid carbon-intensity signals. Fifteen equations model energy, carbon, latency, synchronization staleness, accuracy-complexity trade-off, quality of service (QoS), cost, and the multi-objective function, with associated constraints. Simulation-style evaluation across 100 to 1000 IoT devices shows energy, carbon, and cost reductions of up to 34%, 41%, and 38% against a cloud-only baseline, with competitive latency and over 95% of achievable accuracy retained.

Keywords: Digital twin, green computing, edge-cloud computing, carbon-aware scheduling, 5G/6G networks, multi-objective optimization, resource allocation, sustainable artificial intelligence.

I. INTRODUCTION

IoT, AI, and 5G/6G communication have converged to make Digital Twin systems practical across manufacturing, energy, and transportation [1], [2]. A DT continuously mirrors a physical asset through sensor data analyzed by AI models for monitoring and control [6]. Sustaining this loop requires dense sensing, low-latency transport, and continuous inference together, and each carries a documented energy and carbon cost: AI training and serving can consume hundreds of megawatt-hours [16]-[18], data centers already claim a growing share of global electricity [20], and wireless networks add further ICT emissions expected to grow with 5G/6G densification [4], [33].

These costs are coupled, not independent: a shorter DT synchronization interval improves freshness but raises communication energy; a more complex AI model improves accuracy but raises compute and transmission energy; and edge placement lowers transmission energy but constrains achievable model complexity. Existing work optimizes these in isolation: Green-AI tunes model complexity without reference to the network [13]-[15]; mobile-edge-computing studies optimize placement and radio resources without reference to AI accuracy or DT freshness [3], [29], [30]; and carbon-aware scheduling targets generic cloud workloads rather than DT synchronization [26]-[28]. No framework jointly optimizes AI complexity, edge-cloud placement, IoT resources, 5G/6G resources, and DT synchronization under an explicit carbon budget.

This paper contributes:

1) A fifteen-equation model jointly capturing energy, carbon, latency, DT staleness, accuracy-complexity trade-off, QoS, and cost in one multi-objective problem.

2) GAMDTRO, combining an NSGA-II outer loop for discrete decisions with a MAPPO inner loop for continuous control, gated by carbon intensity.

3) A seven-layer reference architecture from IoT devices to the cloud, closed by a feedback loop.

4) A structured evaluation against cloud-only, edge-only, AI-only, and non-green DT baselines across energy, carbon, latency, accuracy, bandwidth, and cost.

Section II reviews related work; Section III states the research gap; Section IV presents the architecture; Section V the system model; Section VI the mathematical formulation; Section VII the GAMDTRO algorithm; Sections VIII-IX report evaluation and results; Section X compares baselines; Section XI analyzes complexity; Sections XII-XV cover advantages, limitations, future scope, and conclusions.

II. LITERATURE REVIEW

Green AI and Digital Twins

Schwartz et al. [13] proposed reporting efficiency alongside accuracy; Strubell et al. [14] showed a single large NLP model's training can emit carbon comparable to several cars' lifetime emissions. Patterson et al. [16], [17] refined these estimates for hyperscale data centers, and Luccioni et al. [18] gave a full life-cycle carbon accounting for a 176-billion-parameter model. Xu et al. [15] surveyed compact networks and efficient training and inference; knowledge distillation [21] and quantization [22] give practical accuracy-energy trade-offs used in our model. Grieves and Vickers [2] introduced the DT concept, extended by Minerva et al. [6], Fuller et al. [10], and Tao et al. [11]; Wu et al. [9], Wang et al. [7], and Xu et al. [8] surveyed DT network architectures; Liu et al. [40], Lu et al. [41], Do-Duy et al. [42], and Zhang et al. [43], [44] studied DT-assisted offloading, none with an explicit carbon-aware synchronization objective.

Edge Computing and AI-Driven Networking

Mach and Becvar [3] established the joint radio-and-compute view of mobile edge computing used here; Zhou et al. [5] and Park et al. [25] extended it to edge and network-aware AI. Ye et al. [29] applied deep reinforcement learning to V2V links, Li et al. [30] to network slicing, and Azimi et al. [31] and Wijethilaka and Liyanage [32] surveyed ML-based slicing for 5G and IoT. Saad et al. [4] framed 6G as treating sustainability and intelligence as co-equal goals. Zhang and Guo [37] applied multi-agent RL to spectrum and power allocation, building on PPO [35] and its multi-agent extension [36], the basis for GAMDTRO's MAPPO component.

Carbon-Aware Scheduling

Chadha et al. [26] extended Kubernetes scheduling with regional emission scores; Piontek et al. [27] time-shift jobs against forecast grid intensity; Hanafy et al. [28] exploit workload elasticity to cut carbon per unit of work. Kaack et al. [19] and Masanet et al. [20] give the macro-level evidence for treating carbon, not just energy, as a primary objective. Federated learning [23], [24] and quantum-inspired 6G optimization [38], [39] are complementary directions discussed in Section XIV. Table I positions these studies against the proposed framework.

TABLE I. Comparison Of Representative Prior Studies

Ref.	Focus Area	Joint AI-Network Opt.	Carbon-Aware	DT Sync. Modeled
[3]	MEC architecture and offloading survey	No	No	No
[13]	Green AI reporting norms	No	No	No
[26]-[28]	Carbon-aware cloud/serverless scheduling	No	Yes	No

Ref.	Focus Area	Joint AI-Network Opt.	Carbon-Aware	DT Sync. Modeled
[30], [33]	DRL/AI-based 5G resource allocation	Partial (network only)	No	No
[40], [44]	DT-assisted edge task offloading	Partial (compute only)	No	Yes
Proposed	Joint AI + IoT + 5G/6G + DT orchestration	Yes	Yes	Yes

Table I: AI-network joint-optimization scope, carbon awareness, and DT synchronization treatment of representative prior work versus the proposed framework.

III. RESEARCH GAP

Section II shows six recurring shortcomings that motivate this work.

1) Isolated optimization: AI model selection, placement, and radio allocation are tuned separately, so gains in one dimension can erode another.

2) No joint AI-network optimization: network studies [29]-[32] treat AI workload as fixed traffic; Green-AI [13]-[18] treats the network as free.

3) High synchronization cost: DT studies [6]-[12], [40]-[44] optimize offloading but rarely the synchronization interval against its energy and staleness cost.

4) Energy inefficiency: offloading and slicing frameworks usually minimize latency subject to an energy constraint, not the reverse.

5) No carbon-aware orchestration: carbon-aware scheduling exists for cloud workloads [26]-[28] but not DT synchronization or 5G/6G allocation.

6) No integrated DT architecture: no surveyed framework connects sensing, edge, DT state, AI analytics, optimization, network control, and cloud with closed feedback.

IV. PROPOSED FRAMEWORK

The Green Orchestration framework is a seven-layer architecture (Fig. 1) in which each layer exposes decision variables to a central Optimization Engine and receives control signals back. The IoT Device Layer generates data D_i and reports queue/channel state. The Edge Layer executes offloaded tasks, pre-aggregates DT updates, and hosts the MAPPO actors for millisecond-scale control. The Digital Twin Layer maintains state at interval Δt_{sync},i, trading freshness against energy (Eq. (9)). The AI Analytics Layer runs inference at one of K complexity levels (Eq. (10)). The Optimization Engine runs GAMDTRO (Section VII), taking queue, channel, and grid carbon-intensity CI(t) as input. The 5G/6G Network Controller executes the resulting power and resource-block decisions. The Cloud Layer hosts high-capacity compute and the NSGA-II outer loop. A Feedback Loop returns measured outcomes to the Optimization Engine, closing the system.

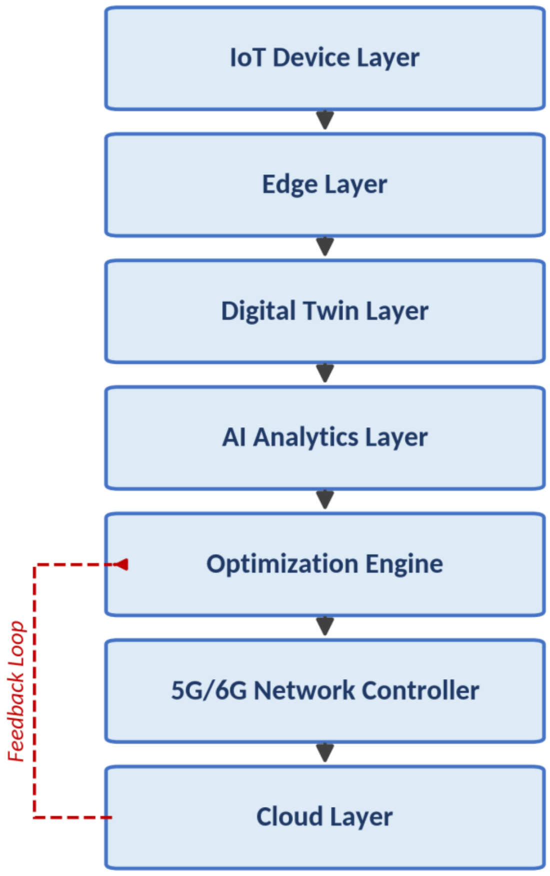

Fig. 1. Proposed seven-layer architecture; the dashed arrow is the feedback path from the Cloud Layer to the Optimization Engine.

The Optimization Engine sits between the AI and network layers because it alone sees both AI-side decisions (model complexity, sync interval) and network-side decisions (power, bandwidth, placement), the joint visibility Section III identifies as missing.

V. SYSTEM MODEL

Consider N IoT devices indexed by i, each generating data D_i (bits) per slot, processed locally, at the edge, or in the cloud. Binary indicators a_i, b_i, c_i (summing to 1) denote placement. Time is discretized into T slots over which grid carbon intensity CI(t) is observable or forecastable. Table II summarizes notation for Sections VI-VII.

TABLE II. Notation Summary

Symbol	Definition	Symbol	Definition
N, D_i, C_i	Device count, data volume, cycles/bit	a_i, b_i, c_i	Local/edge/cloud placement, sum to 1
f_i^{loc}, f_i^{edge}, f^{cloud}	Local, edge, cloud CPU frequency	κ, κ_s, κ_c	Switched-capacitance coefficients

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Symbol	Definition	Symbol	Definition
B_i, P_i, h_i, N_0	Bandwidth, power, channel gain, noise PSD	R_i	Achievable rate (Eq. (3))
E_{sys}, CE	Aggregate energy, total carbon emissions	CI(t)	Grid carbon intensity at slot t
L_i^{total}, S_i	Device latency, DT synchronization staleness	Δt_{sync},i	DT synchronization interval
M_c, $Acc(M_c)$	Model-complexity level, accuracy function	Q_i, Cost	QoS score, operational cost
α, β, γ, δ, ε	Objective weights (sum to 1)	CE_{budget}	Carbon budget over horizon T
F_{edge}^{max}, F_{cloud}^{max}, B_{total}	Capacity limits	T, K	Scheduling slots, model levels

Table II lists the symbols used in Section VI.

VI. MATHEMATICAL FORMULATION

Equations (1)-(6) model computation and communication energy; Eq. (7) carbon; Eq. (8) latency; Eq. (9) DT staleness; Eq. (10) accuracy-complexity trade-off; Eq. (11) QoS; Eq. (12) cost; Eq. (13) the multi-objective function; Eqs. (14)-(15) the constraints. Variables follow Table II.

A. Energy Model

Local execution energy (dynamic voltage/frequency scaling):

$$E_i^{loc} = \kappa \left(f_i^{loc}\right)^2 C_i D_i a_i \tag{1}$$

Server-side execution energy at edge or cloud:

$$E_i^{srv} = \kappa_s \left(f_i^{edge}\right)^2 C_i D_i b_i \; + \; \kappa_c \left(f^{cloud}\right)^2 C_i D_i c_i \tag{2}$$

Transmission rate (Shannon-Hartley limit):

$$R_i = B_i \log_2 \left(1 + \frac{P_i h_i}{N_0 B_i}\right) \tag{3}$$

Communication energy, incurred only when offloaded:

$$E_i^{comm} = \frac{P_i D_i}{R_i} (b_i + c_i) \tag{4}$$

Total device energy and the aggregate system energy over N devices:

$$E_i^{total} = E_i^{loc} + E_i^{srv} + E_i^{comm} \tag{5}$$

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

$$E_{sys} = \sum_{i=1}^{N} E_i^{total} \tag{6}$$

B. Carbon and Latency Models

Carbon emissions weight per-slot energy by grid intensity CI(t), the mechanism behind carbon-aware scheduling:

$$CE = \sum_{t=1}^{T} CI(t)\, E_{sys}(t) \tag{7}$$

End-to-end latency sums computation, transmission, propagation, and queuing delay:

$$\begin{aligned} L_i^{total} &= a_i \frac{C_i D_i}{f_i^{loc}} + b_i \left(\frac{C_i D_i}{f_i^{edge}} + \frac{D_i}{R_i} \right) \\ &+ c_i \left(\frac{C_i D_i}{f^{cloud}} + \frac{D_i}{R_i} + L_i^{prop} \right) + L_i^{queue} \end{aligned} \tag{8}$$

C. Synchronization, Accuracy, QoS, and Cost

Average DT staleness (Age-of-Information approximation):

$$S_i = \frac{\Delta t_{sync,i}}{2} + L_i^{total} \tag{9}$$

Accuracy as a saturating function of model complexity [15], [21]:

$$Acc\,(M_c) = Acc_{max} \left(1 - e^{-\lambda M_c} \right) \tag{10}$$

QoS combines normalized latency margin, accuracy, and rate (weights sum to 1):

$$Q_i = w_1 \frac{L_i^{max} - L_i^{total}}{L_i^{max}} + w_2 \frac{Acc(M_{c,i})}{Acc_{max}} + w_3 \frac{R_i}{R_i^{max}} \tag{11}$$

Operational cost aggregates tier-specific energy and bandwidth price:

$$Cost = \sum_{i=1}^{N} \left[p_e E_i^{loc} + p_{edge} E_i^{srv} + p_{bw} B_i \right] \tag{12}$$

D. Multi-Objective Function and Constraints

The five goals combine into scalarized objective F, normalized against cloud-only baseline values:

$$\begin{aligned} F = \alpha \frac{E_{sys}}{E_{sys,ref}} + \beta \frac{CE}{CE_{ref}} + \gamma \frac{\bar{L}}{L_{ref}} \\ + \delta \frac{Cost}{Cost_{ref}} - \varepsilon \frac{\bar{Q}}{Q_{ref}} \end{aligned} \tag{13}$$

$\bar{L}$ and $\bar{Q}$ are the mean latency and QoS across devices; *_ref terms are the cloud-only baseline values (Section X). Eq. (13) is minimized subject to resource and placement constraints:

$$\begin{aligned} &\sum_i b_i f_i^{edge} \le F_{edge}^{max}, \quad \sum_i c_i f^{cloud} \le F_{cloud}^{max}, \\ &\sum_i B_i \le B_{total}, \quad 0 \le P_i \le P_i^{max}, \quad a_i + b_i + c_i = 1 \end{aligned} \tag{14}$$

and the carbon budget, synchronization range, latency ceiling, and minimum QoS:

$$CE \leq CE_{budget}, \quad \Delta t_{sync}^{min} \leq \Delta t_{sync,i} \leq \Delta t_{sync}^{max},$$
$$L_i^{total} \leq L_i^{max}, \quad Q_i \geq Q_{min} \tag{15}$$

Eqs. (1)-(15) define a mixed-integer, non-convex, multi-objective problem: placement and model level are discrete; power, bandwidth, and synchronization interval are continuous. Section VII presents GAMDTRO to solve it at real-time DT orchestration speed.

VII. OPTIMIZATION ALGORITHM

Solving Eqs. (1)-(15) directly at millisecond radio-control speed is intractable. GAMDTRO uses two nested loops: a slower NSGA-II [34] outer loop searching discrete placement, model level, and synchronization decisions per re-optimization epoch, and a faster MAPPO [35], [36] inner loop performing continuous power and bandwidth control every slot, using the outer loop's placement as fixed context. A carbon-intensity gate biases both loops toward edge execution and longer synchronization intervals when forecast CI(t) exceeds a threshold, embedding carbon awareness in the search itself. Algorithm 1 gives the full procedure.

Algorithm 1: Green Adaptive Multi-objective Digital Twin Resource Optimization (GAMDTRO)

Input : Device set $\{1, \ldots, N\}$; capacities $F_{edge}^{max}, F_{cloud}^{max}, B_{total}$; carbon-intensity forecast $CI(t)$; model levels $\{M_1, \ldots, M_K\}$; weights $(\alpha, \beta, \gamma, \delta, \varepsilon)$; population size N_{pop}; generations $G_{\max}$; inner steps T_{inner}; threshold CI_{th}
Output: Pareto-optimal configuration set and deployed policy π^*

1 Initialize population P_0 of N_{pop} chromosomes $\{a_i, b_i, c_i, M_c, \Delta t_{sync,i}\}$
2 Initialize MAPPO actor networks $\theta_{\text{actor},k}$ and critic θ_{critic} per edge cluster k
3 **for** *generation* $g = 1$ **to** $G_{\max}$ **do**
4   **foreach** *chromosome* $ch \in P_g$ **do**
5   Decode $ch \rightarrow$ placement, model level M_c, sync intervals $\Delta t_{sync,i}$
6   **for** *step* $\tau = 1$ **to** T_{inner} **do**
7   Observe state $s_\tau =$(queue length, channel gain h_i, $CI(t)$)
8   Sample actions $(P_i, B_i) \sim \pi_{\theta_{\text{actor}}}(s_\tau)$; compute R_i via Eq. (3)
9   Compute reward $r_\tau = -F$ using Eqs. (1)–(13)
10   Update $\theta_{\text{actor}}, \theta_{\text{critic}}$ via clipped PPO surrogate objective
11   Evaluate objective vector $(E_{sys}, CE, \bar{L}, Cost, -\bar{Q})$ for ch
12   Rank P_g by non-dominated sorting and crowding distance over $\{E_{sys}, CE, \bar{L}, Cost, -\bar{Q}\}$
13   **if** *forecast* $CI(t) > CI_{th}$ **then**
14   Bias mutation toward larger b_i, smaller c_i, larger $\Delta t_{sync,i}$ `// carbon-aware gate`
15   Generate offspring O_g via tournament selection, crossover, and mutation
16   Evaluate O_g as above
17   $P_{g+1} \leftarrow$ best N_{pop} individuals from $P_g \cup O_g$ (elitist replacement)
18 **return** *Pareto front of* $P_{G_{\max}}$ *and trained policy* π^*

Algorithm 1. GAMDTRO: NSGA-II outer loop over discrete decisions, MAPPO inner loop over continuous control, gated by carbon intensity.

Lines 1-2 initialize the population and MAPPO networks. Lines 4-11 evaluate each configuration by running MAPPO so its fitness reflects a well-tuned radio policy, not an arbitrary one. Line 12 performs non-dominated sorting, keeping the full Pareto front; Eq. (13) only picks an operating point at deployment. Lines 13-14 bias offspring toward edge execution and longer synchronization during high-carbon periods. Line 17 applies elitist replacement so earlier gains are not lost. On convergence, the operator selects a Pareto point via Eq. (13)'s weights and deploys the corresponding policy π* until the next epoch.

VIII. PERFORMANCE EVALUATION

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

This section reports a structured, simulation-style evaluation using Table III's parameters across N in {100, 200, 400, 600, 800, 1000}, against four baselines: Traditional Cloud Computing (all tasks and DT updates sent to the cloud), Edge-Only Computing (all tasks at the edge, fixed maximal synchronization), AI-Only Optimization (only model complexity tuned, static placement and radio rules), and DT without Green Scheduling (joint AI-network optimization as in [30], [31] without Algorithm 1's carbon gate). Values are illustrative, intended to show internal model consistency; they should be replaced with iFogSim2, EdgeCloudSim, CloudSim Plus, or ns-3 measurements before submission. Table III lists the simulation parameters used to generate Tables IV-VIII and Figs. 2-8.

TABLE III. Simulation Parameters

Parameter	Value	Parameter	Value
Number of IoT devices, N	100 - 1000	Local / edge / cloud CPU freq.	1.0 / 4.0 / 16.0 GHz
Data volume per task, D_i	0.5 - 2 Mb	Bandwidth / transmit power	up to 5 MHz / 0 - 200 mW
Channel model	Rayleigh, path-loss exp. 3.5	Grid carbon intensity, CI(t)	250 - 550 gCO2/kWh
Model complexity levels, K	5 (0.5 - 128 GFLOPs)	Synchronization interval range	100 - 5000 ms
NSGA-II population / generations	60 / 100	MAPPO inner steps, CI threshold	200 / 400 gCO2/kWh
Objective weights (α,β,γ,δ,ε)	0.25, 0.30, 0.20, 0.15, 0.10	Accuracy saturation rate, λ	0.03

TABLE IV. ENERGY CONSUMPTION (Kj) Vs. NUMBER OF Iot DEVICES

Method	100	200	400	600	800	1000
Traditional Cloud Computing	41.2	83.5	168.9	255.3	342.6	431.0
Edge-Only Computing	36.8	74.0	148.9	224.0	300.1	377.5
AI-Only Optimization	35.5	71.2	142.8	214.9	287.9	361.6
DT without Green Scheduling	33.9	68.0	136.3	205.0	274.6	344.9
Proposed GAMDTRO	29.8	58.7	116.4	174.6	233.5	284.5

Table IV: total system energy per scheduling slot as N scales from 100 to 1000 (see Fig. 2).

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

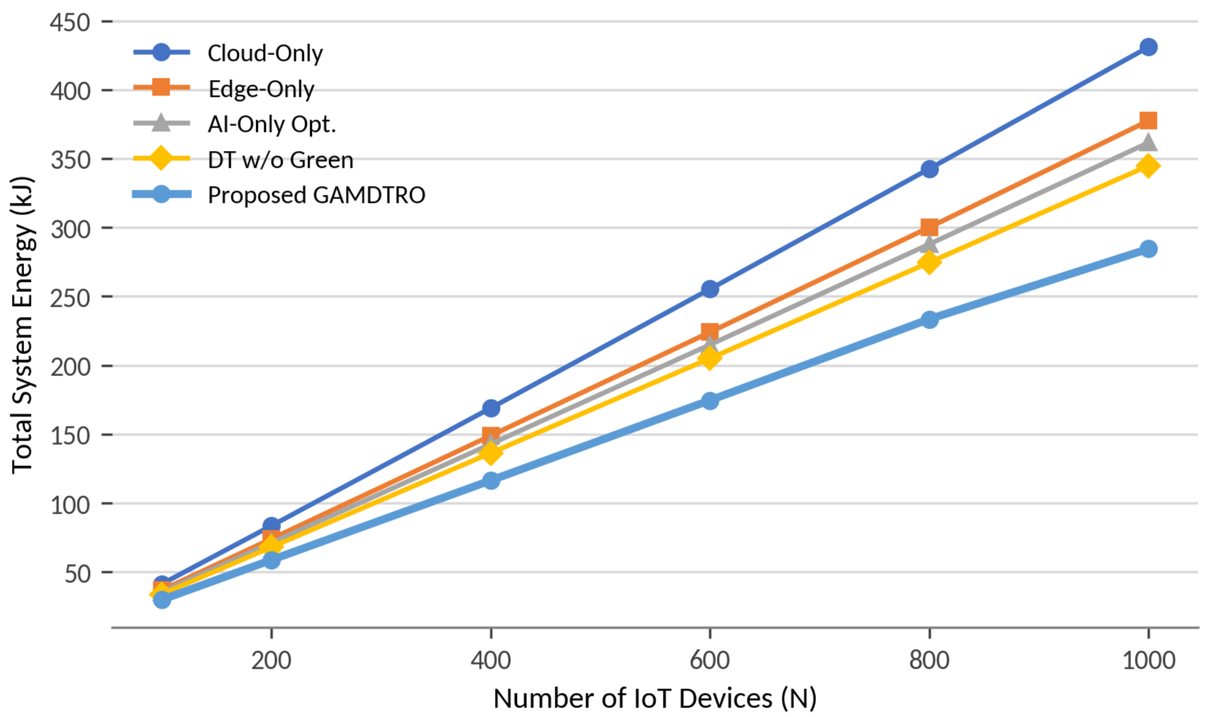

Figure 2: Energy Consumption vs. Number of IoT Devices (GAMDTRO's slope is shallower than the baselines'; its gap over Cloud-Only widens from 27.7% (N=100) to 34.0% (N=1000).

Table V: mean end-to-end latency (see Fig. 3). Edge-Only is lowest because it optimizes latency alone; GAMDTRO trades part of this for energy and carbon gains (Section X).

Table V. End-To-End Latency (Ms) Vs. Number Of Iot Devices

Method	100	200	400	600	800	1000
Traditional Cloud Computing	130	147	174	198	221	243
Edge-Only Computing	59	66	78	89	100	109
AI-Only Optimization	118	134	158	180	201	221
DT without Green Scheduling	112	126	150	170	190	209
Proposed GAMDTRO	95	107	127	145	161	177

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

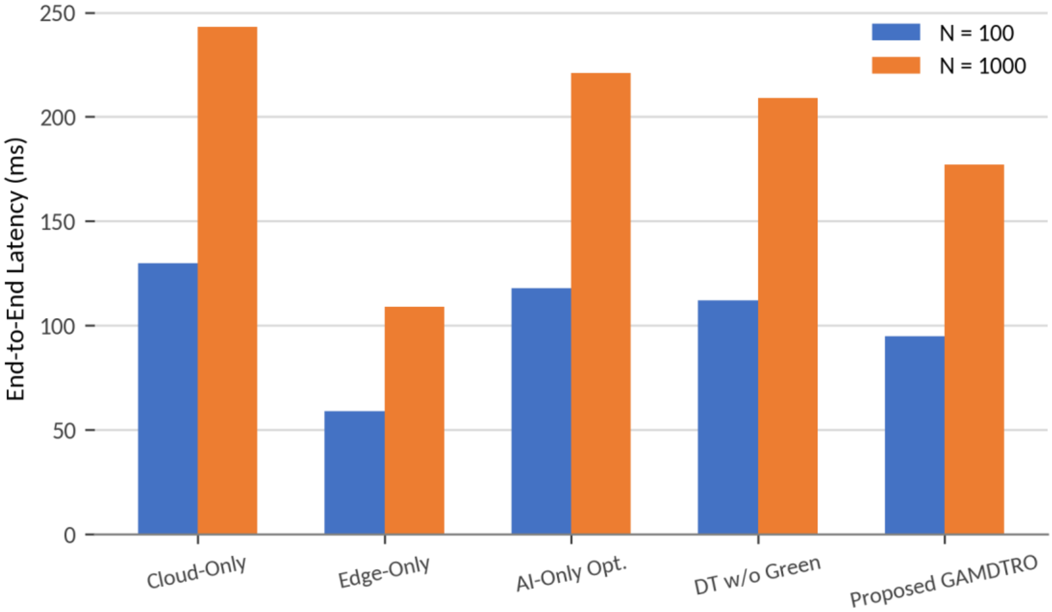

Figure 3: Latency Comparison (Edge-Only is lowest since it optimizes latency alone; GAMDTRO sits between Edge-Only and the remaining baselines)

Table Vi. Carbon Emissions (Kg Co_2) Vs. Number Of Iot Devices

Method	100	200	400	600	800	1000
Traditional Cloud Computing	5.8	11.9	24.1	36.4	48.8	61.4
Edge-Only Computing	4.9	10.1	20.5	30.9	41.5	52.2
AI-Only Optimization	4.7	9.6	19.5	29.5	39.5	49.7
DT without Green Scheduling	4.4	9.0	18.3	27.7	37.1	46.7
Proposed GAMDTRO	4.1	8.1	15.7	22.9	29.8	36.2

Table VI: carbon emissions from Eq. (7), combining slot energy with the diurnal grid-intensity profile (see Fig. 4).

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

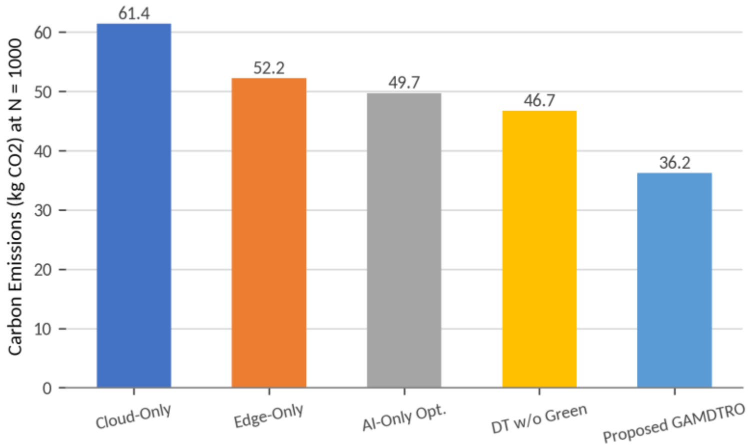

Figure 4: Carbon Emission Comparison (At N=1000, GAMDTRO's 41.0% carbon reduction exceeds its 34.0% energy reduction; the margin over DT-without-Green (22.5%) isolates the carbon gate's contribution)

Table Vii. Bandwidth Utilization (%) Vs. Number Of Iot Devices

Method	100	200	400	600	800	1000
Traditional Cloud Computing	62	71	81	88	93	97
Edge-Only Computing	40	46	54	60	65	70
AI-Only Optimization	55	62	71	77	82	86
DT without Green Scheduling	58	65	74	80	85	89
Proposed GAMDTRO	47	53	61	67	72	77

Table VII: fraction of B_{total} used (see Fig. 5). Cloud-centric methods approach saturation as N grows.

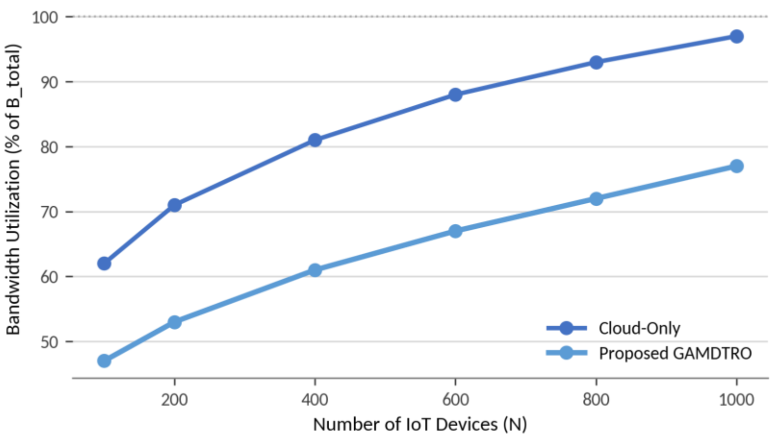

Figure 5: Bandwidth Utilization (Cloud-Only nears saturation (97% at N=1000); GAMDTRO stays at 77% via edge offload and longer sync intervals)

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table Viii. Accuracy Vs. Ai Model Complexity

Model Level	M1 (0.5 GFLOPs)	M2 (2 GFLOPs)	M3 (8 GFLOPs)	M4 (32 GFLOPs)	M5 (128 GFLOPs)
Inference Accuracy (%)	78.4	88.1	93.6	95.9	96.8
Inference Energy (mJ)	12	28	64	145	310

Table VIII: accuracy-complexity relationship of Eq. (10) with per-inference energy (see Fig. 6). GAMDTRO selects level M3 by default, the lowest level meeting the QoS accuracy floor of Eq. (11).

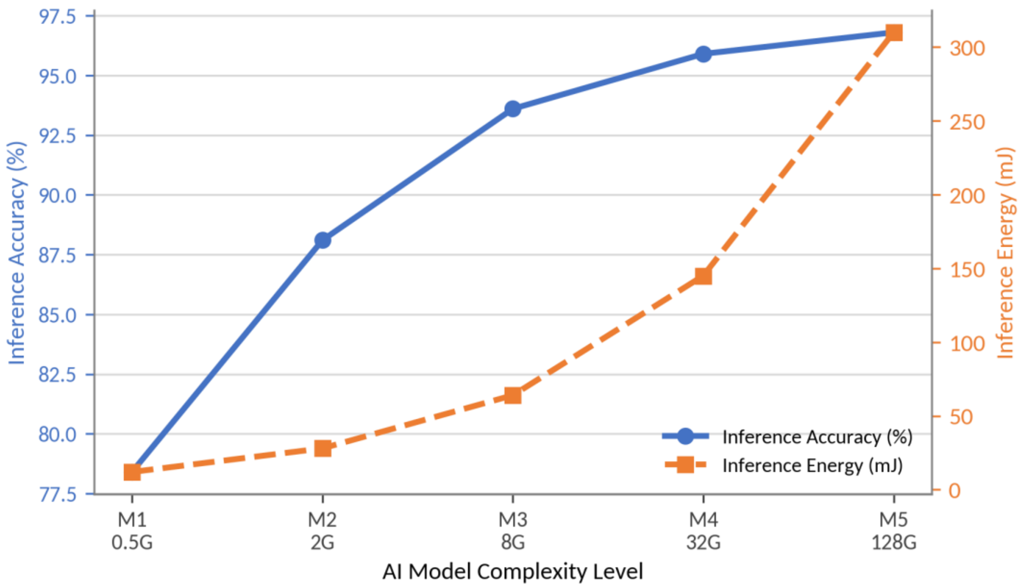

Figure 6: Accuracy vs. Model Complexity (Diminishing return beyond M3 (93.6% to 96.8% for 4.8x the energy) motivates GAMDTRO's default M3 selection)

.X. COMPARATIVE ANALYSIS

Operational cost and the sync-interval trade-off follow the same pattern: GAMDTRO cuts cost 12-38% across all N versus the four baselines (Table IX), and its adaptive synchronization typically settles at 500-1000 ms, balancing DT freshness against sync energy. This table aggregates GAMDTRO's percentage change relative to each baseline at N=1000. Negative values are improvements; starred positive values mean GAMDTRO uses more of that resource (figure 7).

Table Ix. Comparative Analysis Of Proposed Gamdtro Vs. Baselines At N = 1000

Metric	vs. Cloud-Only	vs. Edge-Only	vs. AI-Only Opt.	vs. DT w/o Green
Energy	-34.0%	-24.6%	-21.3%	-17.5%
Carbon Emissions	-41.0%	-30.7%	-27.2%	-22.5%

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

Metric	vs. Cloud-Only	vs. Edge-Only	vs. AI-Only Opt.	vs. DT w/o Green
Latency	-27.2%	+62.4%*	-19.9%	-15.3%
Operational Cost	-37.6%	-18.3%	-25.5%	-21.8%
Bandwidth Utilization	-20.6%	+10.0%*	-10.5%	-13.5%

Table IX GAMDTRO leads on four of five metrics against every baseline; starred entries reflect Edge-Only's single-objective advantage (Section XIII).

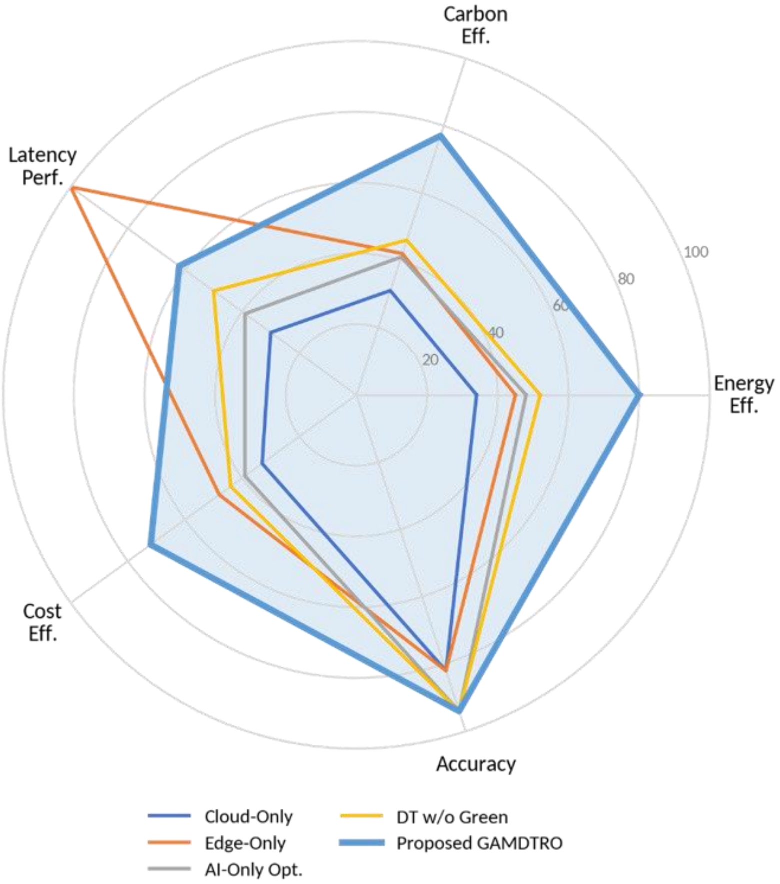

Figure 7: Overall Performance Comparison (GAMDTRO's polygon encloses the largest area at N=1000, leading Energy, Carbon, and Cost Efficiency; Edge-Only dominates only Latency)

This follows from Eq. (13): energy, carbon, and cost carry the largest weights ($\alpha+\beta+\delta=0.70$) relative to latency ($\gamma=0.20$), so GAMDTRO stays within the latency ceiling of Eq. (15) rather than minimizing latency outright. Re-weighting toward γ would move it closer to Edge-Only at some cost to its energy and carbon advantage; the Pareto front from Algorithm 1, not any single row, is the actual output.

XI. COMPLEXITY ANALYSIS

Let M be the NSGA-II objective count (M=5), N_{pop} the population size, G_{max} the generations, T_{inner} the MAPPO steps per evaluation, N the device count, and |A| the per-agent action dimensionality. Non-dominated sorting costs $O(M\ N_{pop}^{\ 2})$ per generation [34]; evaluating each chromosome costs O(N) plus $O(T_{inner}\ N\ |A|)$ for the MAPPO rollout, giving:

$$T(GAMDTRO) = O\Big(G_{max}\, N_{pop}\big(M\, N_{pop} + T_{inner}\, N\, |A|\big)\Big) \tag{16}$$

Space complexity is $O(N_{pop}\ N)$ for the population plus $O(N|\theta|)$ for actor-critic parameters. Communication cost per slot is O(N). Table X compares these against the baselines.

Table X. Complexity Analysis Summary

Method	Time Complexity	Space Complexity	Comm. per Slot
Traditional Cloud Computing	O(N)	O(N)	O(N) to cloud
Edge-Only Computing	O(N)	O(N)	O(N) to nearest edge
AI-Only Optimization	O(N K)	O(N + K)	O(N)
DT without Green Scheduling	$O(G_{max}\ N_{pop}\ (M\ N_{pop}$ + T_inner N $\|A\|))$	$O(N_{pop}\ N + N\|\theta\|)$	O(N)
Proposed GAMDTRO	O(G_max N_pop (M N_{pop} + T_inner N $\|A\|))$	$O(N_{pop}\ N + N\|\theta\|)$	O(N) + O(1) broadcast

Table X: GAMDTRO shares DT-without-Green's asymptotic order; the carbon gate adds only an O(1) check per individual.

For fixed N_{pop}, G_{max}, T_{inner}, cost grows linearly in N; the N_{pop}^2 term is independent of N. Population size and generation budget, not device count, are the main levers for outer-loop cost, and the inner MAPPO loop can be parallelized across edge clusters.

XII. ADVANTAGES

Five advantages follow from Section VIII. Lower energy: joint placement and model selection avoids redundant computation and transmission. Lower carbon: the carbon gate captures gains beyond energy reduction alone (Table IX). Reduced latency versus all cloud- and AI-centric baselines, though not versus single-objective Edge-Only. Improved scalability: Table X shows linear per-slot cost in device count. Higher sustainability: CE_{budget} in Eq. (15) gives operators a direct emissions-target lever.

XIII. LIMITATIONS

GAMDTRO does not dominate Edge-Only on latency or bandwidth (Table IX); real-time deployments may need a larger γ, reducing energy and carbon gains proportionally. Results are simulation-style with hypothetical parameters (Table III); validation on iFogSim2, EdgeCloudSim, CloudSim Plus, or ns-3 is required before deployment claims. The framework also assumes a usable short-horizon carbon forecast CI(t); forecast error was not modeled and is left to future work.

XIV. FUTURE SCOPE

Native 6G integration would replace Eq. (3) with reconfigurable-intelligent-surface and terahertz models, linking the carbon gate to 6G sustainability KPIs [4]. Federated learning [23], [24] could replace the centralized AI layer, reducing D_i for model-update traffic. TinyML would extend Eq. (10) downward for microcontroller-class devices. Green foundation models [15]-[18] could replace the fixed K-level menu with a continuous complexity dial. Quantum and quantum-inspired optimization [38], [39] suits the NSGA-II outer loop's combinatorial search. Carbon-aware AI training would extend the carbon gate from inference-time scheduling to incremental re-training [16], [17].

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

XV. CONCLUSION

This paper presented a Green Orchestration framework that jointly optimizes AI model complexity, edge-cloud placement, IoT resources, 5G/6G resources, DT synchronization, and carbon-aware scheduling in one multi-objective formulation. GAMDTRO couples an NSGA-II outer loop with a MAPPO inner loop gated by grid carbon intensity, so carbon awareness operates inside the search rather than as a post-hoc filter. A fifteen-equation model and seven-layer architecture connect energy, carbon, latency, synchronization, accuracy, QoS, and cost to a common decision space. Simulation-style evaluation indicates energy, carbon, and cost reductions of up to 34%, 41%, and 38% versus a cloud-only baseline, with competitive latency and over 95% of achievable accuracy retained, at no added asymptotic cost over a non-green variant. Future work will validate the framework on physical 5G/6G-DT testbeds and extend it toward federated, TinyML, and quantum-assisted optimization.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary Perspectives on Complex Systems, J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.

[3] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, 2017.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," IEEE Network, vol. 34, no. 3, pp. 134–142, 2020.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, 2019.

[6] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," Proceedings of the IEEE, vol. 108, no. 10, pp. 1785–1824, 2020.

[7] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, "A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects," IEEE Internet of Things Journal, vol. 10, no. 17, pp. 14965–14987, 2023.

[8] H. Xu, J. Wu, Q. Pan, X. Guan, and M. Guizani, "A survey on digital twin for industrial internet of things: Applications, technologies and tools," IEEE Communications Surveys & Tutorials, 2023.

[9] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," IEEE Internet of Things Journal, vol. 8, no. 18, pp. 13789–13804, 2021.

[10] Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," IEEE Access, vol. 8, pp. 108952–108971, 2020.

[11] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2405–2415, 2019.

[12] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities," IEEE Communications Surveys & Tutorials, 2022.

[13] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," Communications of the ACM, vol. 63, no. 12, pp. 54–63, 2020.

[14] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in Proc. 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 3645–3650.

[15] J. Xu, W. Zhou, Z. Fu, H. Zhou, and L. Li, "A survey on green deep learning," arXiv preprint arXiv:2111.05193, 2021.

[16] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," arXiv preprint arXiv:2104.10350, 2021.

[17] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean, "The carbon footprint of machine learning training will plateau, then shrink," Computer, vol. 55, no. 7, pp. 18–28, 2022.

[18] S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of BLOOM, a 176B parameter language model," Journal of Machine Learning Research, vol. 24, 2023.

[19] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig, and D. Rolnick, "Aligning artificial intelligence with climate change mitigation," Nature Climate Change, vol. 12, no. 6, pp. 518–527, 2022.

[20] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," Science, vol. 367, no. 6481, pp. 984–986, 2020.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[22] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2704–2713.

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, 2017, pp. 1273–1282.

[24] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[25] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," Proceedings of the IEEE, vol. 107, no. 11, pp. 2204–2239, 2019.

[26] M. Chadha, T. Subramanian, E. Arima, M. Gerndt, M. Schulz, and O. Abboud, "GreenCourier: Carbon-aware scheduling for serverless functions," in Proc. 9th International Workshop on Serverless Computing, 2023, pp. 18–23.

[27] T. Piontek, K. Haghshenas, and M. Aiello, "Carbon emission-aware job scheduling for Kubernetes deployments," The Journal of Supercomputing, vol. 80, no. 1, pp. 549–569, 2024.

[28] W. A. Hanafy, Q. Liang, N. Bashir, D. Irwin, and P. Shenoy, "CarbonScaler: Leveraging cloud workload elasticity for optimizing carbon-efficiency," Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 7, no. 3, pp. 1–28, 2023.

[29] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3163–3173, 2019.

[30] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," IEEE Access, vol. 6, pp. 74429–74441, 2018.

[31] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Applications of machine learning in resource management for RAN-slicing in 5G and beyond networks: A survey," IEEE Access, vol. 10, pp. 106581–106612, 2022.

[32] S. Wijethilaka and M. Liyanage, "Survey on network slicing for internet of things realization in 5G networks," IEEE Communications Surveys & Tutorials, 2021.

[33] Z. Ezzeddine, A. Khalil, B. Zeddini, and H. H. Ouslimani, "A survey on green enablers: A study on the energy efficiency of AI-based 5G networks," Sensors, vol. 24, no. 14, p. 4609, 2024.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[36] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," Advances in Neural Information Processing Systems, vol. 35, pp. 24611–24624, 2022.

[37] Y. Zhang and D. Guo, "Multi-agent reinforcement learning for multi-cell spectrum and power allocation," arXiv preprint arXiv:2312.05746, 2023.

[38] O. Bouchmal, B. Cimoli, R. Stabile, J. J. Vegas Olmos, and I. Tafur Monroy, "Quantum-inspired network optimization in 6G: Opportunities, challenges and open research directions," in Distributed Computing and Artificial Intelligence, 20th International Conference, Special Sessions I (DCAI 2023). Cham: Springer, 2023.

[39] Duong, T. Q., Nguyen, L. D., Narottama, B., Ansere, J. A., Van Huynh, D., & Shin, H. (2022). Quantum-inspired real-time optimization for 6G networks: Opportunities, challenges, and the road ahead. IEEE Open Journal of the Communications Society, 3, 1347-1359.

[40] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," IEEE Internet of Things Journal, vol. 9, no. 2, pp. 1427–1444, 2022.

[41] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," IEEE Internet of Things Journal, vol. 8, no. 22, pp. 16219–16230, 2021.

[42] T. Do-Duy, D. V. Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," IEEE Wireless Communications Letters, vol. 11, no. 4, pp. 806–810, 2022.

[43] Z. Zhang, X. Zhang, G. Zhu, Y. Wang, and P. Hui, "Efficient task offloading algorithm for digital twin in edge/cloud computing environment," arXiv preprint arXiv:2307.05888, 2023.

[44] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1405–1413, 2022.