Future Technology
Open Access Journal
ISSN 2832-0379

August 2026| Volume 05 | Issue 03 | Pages 85-96

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.5.3.8

Future Publishing LLC

Article

Federated reinforcement learning for energy-aware load balancing in edge-fog-cloud IoT continuum

Si Liu, Midhun Chakkaravarthy*

School of AI Computing and Multimedia, Lincoln University College, Malaysia

ARTICLE INFO

Article history:
Received 12 December 2025
Received in revised form
22 March 2026
Accepted 27 April 2026

Keywords:
Federated reinforcement learning,
Load balancing, Edge-fog-cloud continuum,
Energy efficiency, Internet of Things

*Corresponding author
Email address:
midhun@lincoln.edu.my

DOI: 10.55670/fpll.futech.5.3.8

ABSTRACT

Energy efficiency remains a major challenge in deploying IoT systems, especially in scenarios requiring large numbers of devices while balancing computational requirements and operational lifetimes. This paper proposes a federated reinforcement learning framework for adaptive load balancing in the edge-fog-cloud continuum that optimizes energy efficiency and supports diverse quality of service requirements. The proposed framework addresses the limitations of traditional centralized machine learning approaches that require collecting sensitive operational information and transmitting it to cloud servers for centralized analysis. This increases the risk of privacy violations and introduces communication overheads that limit the responsiveness of IoT systems. The proposed framework employs a federated reinforcement learning approach, enabling edge nodes to collaboratively learn an optimal load-balancing policy without transmitting operational information. The proposed framework uses a context-aware reward function that optimizes multiple objectives based on temporal patterns, device energy levels, and application criticality. This enables the proposed framework to adapt its optimization objectives and balance energy efficiency and performance maximization. The proposed framework introduces a new action-space pruning mechanism that accelerates the optimization process by leveraging domain knowledge of possible load-balancing patterns. The proposed framework uses a distributed experience replay buffer to reduce trial-and-error in reinforcement learning. The proposed framework demonstrates its effectiveness in optimizing energy efficiency through a series of experiments in a real-world IoT environment and a centralized machine learning approach. The proposed framework demonstrates that distributed machine learning approaches can outperform centralized ones for optimizing energy efficiency in IoT systems.

1. Introduction

The widespread deployment of Internet of Things (IoT) infrastructures has dramatically changed the landscape of distributed computing systems, with unprecedented amounts of heterogeneous data requiring processing in environments with constrained latency, energy, and quality of service (QoS) [1]. The inherent tension between the computational requirements of IoT devices and their operational lifetimes in smart-city, industrial-automation, and healthcare-monitoring environments has made energy efficiency a major performance bottleneck in IoT infrastructures [2]. The conventional IoT architecture of collecting raw IoT data and processing it in a centralized manner for model development and decision-making in cloud environments has been found wanting in this new IoT paradigm in terms of prohibitive communication overheads, operational data privacy issues, and the inability of this paradigm to support the low-latency requirements of IoT applications [3]. These issues have created significant interest in the use of a continuum architecture for IoT infrastructures, in which tasks are distributed across multiple layers to achieve better proximity and priority [4]. Notably, reinforcement learning and its deep variants have been identified as particularly promising paradigms for the adaptive management of resources in such multi-level infrastructures, owing to their ability to enable intelligent agents to learn optimal decision policies through direct interaction with dynamic and uncertain operating conditions [5]. Previous studies have shown that deep reinforcement learning can effectively address task offloading, load balancing, and resource allocation in fog/cloud IoT infrastructures, significantly reducing end-to-end latency and energy consumption compared to traditional

heuristic approaches [6]. Dynamic task offloading schemes based on DRL in 5G edge/cloud continua have also been proposed to demonstrate the feasibility of efficient and timely decision-making in heterogeneous network infrastructures [7]. Systematic reviews of reinforcement learning-based computation offloading schemes have consolidated a large body of evidence supporting the effectiveness of DRL in achieving greater adaptability than traditional approaches, particularly in non-stationary scenarios [8]. Surveys of federated cloud-edge-fog offloading infrastructures have also identified a growing recognition of the need to leverage multiple computational levels to meet the diversity of IoT application requirements, by developing intelligent scheduling policies that coordinate edge, fog, and cloud resources [9].

Despite these advances, a fundamental limitation pervades the extant RL-based offloading literature: the widespread use of centralized learning paradigms, in which a centralized entity aggregates global state information to learn the optimal policy [10]. Such centralization is not only a scalability bottleneck with increasingly large numbers of devices but also implies a continuous need to transmit raw operational data from devices to a remote server, which poses serious data governance and privacy concerns in sensitive deployment environments [11]. Energy-efficient task scheduling frameworks for Industrial IoT environments have recently addressed tier-specific resource heterogeneity using multi-objective reinforcement learning paradigms [12], yet still rely on centralized learning or idealized communication models that may not hold in the presence of network disruptions. In the context of cluster-level reviews of RL-based fog offloading paradigms, the limitations in the robustness of such paradigms in the presence of fault conditions have been emphasized in the literature [13]. In the context of vehicular edge cloud environments, the importance of balancing the load on the edge nodes was recently emphasized in terms of its positive effect on the energy efficiency and convergence stability of the offloading paradigm, with privacy considerations remaining a secondary aspect [14], a fact that is becoming increasingly problematic in the context of increasingly scaled IoT environments.

Federated learning provides a principle-based solution to resolve the trade-off by allowing multiple clients in a distributed environment to collectively learn a shared model without sharing data or performing any aggregation at any particular client or node [15]. Comprehensive surveys on the security and privacy of federated learning in edge IoT environments have shown that it significantly reduces the attack surface of data centralization while enabling scalable model convergence across heterogeneous devices [16]. Significantly, the theoretical foundations of privacy-preserving federated learning for edge computing, including differential privacy, secure aggregation, and hierarchical aggregation, have been well developed, with strong theoretical guarantees that show the overall architecture's robustness to inference and poisoning attacks [17]. When applied to medical IoT environments, federated learning-based privacy-preserving FL using edge computing has demonstrated the practical feasibility of lightweight secret-sharing approaches that achieve comparable accuracy without compromising individual device data sharing [18]. Similarly, reviews of industrial IoT environments have shown that federated learning using edge intelligence provides responsiveness and fault-tolerant characteristics that are essential in heterogeneous device ecosystems, attributes that are equally important in general-purpose IoT continua [19].

The natural combination of FL with RL, i.e., federated reinforcement learning (FRL), has been gaining more attention as it is considered capable of solving issues of privacy preservation, distributed learning, and adaptive optimization at the same time [20]. By allowing edge devices to learn and improve local decision-making strategies through interaction with their environment and then share model parameters without sharing observations or rewards, FRL models benefit from the privacy-preserving capabilities of FL models while maintaining the ability of DRL models to perform decision-making. Nevertheless, existing FRL models for managing resources in IoT environments do not fully take advantage of the inherent characteristics of the edge-fog-cloud continuum, as they do not account for the dynamic nature of device battery states, domain knowledge of possible patterns of distributing loads to speed up the learning process, and robustness against possible faults during training. Moreover, the connection between intelligent learning of distributed decision-making strategies and centralized optimization in energy-constrained environments remains unclear, making it unclear whether FRL models can outperform centralized models or can be considered approximations with lower privacy requirements.

To bridge the identified gaps, the paper proposes a federated reinforcement learning-based framework for adaptive, energy-efficient load balancing across the edge-fog-cloud IoT continuum. The proposed system enables edge nodes to collaborate to design optimal load-balancing strategies by training local models and exchanging parameters to balance the load without sharing the actual operational data. A context-aware reward function dynamically adjusts the optimization objectives based on time-variant usage patterns, battery conditions, and application criticality, thereby smoothly transitioning between energy efficiency and performance maximization objectives as system conditions change. A new action-pruning technique is also proposed to improve convergence rate by leveraging domain knowledge to restrict the search space to the most likely load-balancing strategies, thereby reducing exploration costs via a distributed experience replay buffer. The effectiveness of the proposed FRL-based load balancing is evaluated through extensive experiments, including fault-injection experiments, to demonstrate its energy efficiency and robust QoS performance even in the presence of network faults, thereby advancing the state of the art in energy-efficient and privacy-preserving IoT management.

2. System model and problem formulation

Consider a three-tier edge-fog-cloud IoT continuum comprising a set of battery-powered edge devices $\mathcal{N} = \{1,2,\dots,N\}$, a set of fog nodes $\mathcal{F} = \{1,2,\dots,F\}$, and a remote cloud server, as illustrated in Figure 1. Edge devices continuously generate computational tasks and must decide at each time slot t whether to process them locally, offload them to an associated fog node, or escalate them to the cloud tier. Fog nodes serve as intermediate aggregators that collect locally trained model parameters from edge devices, perform partial aggregation, and relay compressed updates to the cloud server, which maintains the global federated model. This hierarchical architecture eliminates the need to transmit raw data beyond the edge tier, thereby preserving operational privacy while enabling collaborative intelligence across the continuum.

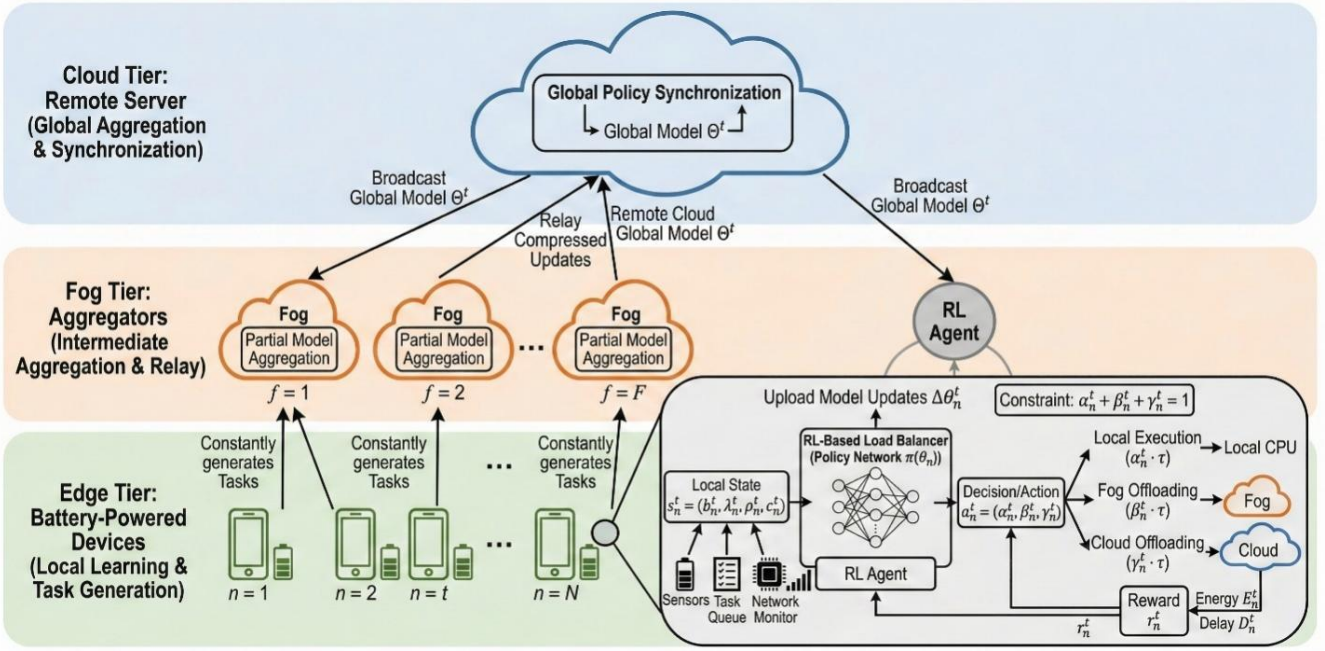

Figure 1. Three-tier edge-fog-cloud IoT continuum architecture

Each edge device n is characterized at time t by a state vector $s_n^t = (b_n^t,\ \lambda_n^t,\ \rho_n^t,\ c_n^t)$, where $b_n^t \in [0,1]$ denotes the normalized battery level, λ_n^t is the current task arrival rate, ρ_n^t represents the local CPU utilization, and c_n^t captures the channel quality to the nearest fog node. The global system state is defined as $S^t = \{s_n^t\}_{n=1}^N$, which represents the collection of all local state vectors and is introduced here for theoretical exposition of the optimization objective in Eq. (5) only. In practice, each device nn n observes its own local state snts_n^t snt exclusively and never accesses the states of other devices, ensuring that no raw operational data is transmitted beyond the edge tier. The joint action space specifies workload allocation fractions across tiers. For device n at time t, the action $a_n^t = (\alpha_n^t, \beta_n^t, \gamma_n^t) \in [0,1]^3$ is a continuous allocation vector denoting the proportions of the current workload assigned to local execution, fog offloading, and cloud offloading, respectively, subject to the simplex feasibility constraint:

$$\alpha_n^t + \beta_n^t + \gamma_n^t = 1, \quad \alpha_n^t,\ \beta_n^t,\ \gamma_n^t \geq 0 \tag{1}$$

The energy consumed by device n at time slot t is modeled as the sum of local computation energy and transmission energy:

$$E_n^t = \kappa \cdot (\rho_n^t)^3 \cdot \alpha_n^t \cdot \tau + P_n^{tx} \cdot (\beta_n^t + \gamma_n^t) \cdot \tau \tag{2}$$

where κ is the effective switched capacitance of the processor, τ is the duration of a time slot, and P_n^{tx} is the transmission power. The end-to-end task completion delay for device n is expressed as:

$$D_n^t = \alpha_n^t \cdot d_n^{loc} + \beta_n^t \cdot (d_n^{fog} + d_n^{fog,comp}) + \gamma_n^t \cdot (d_n^{cloud} + d_n^{cloud,comp}) \tag{3}$$

where the three delay components are explicitly defined as follows. The local computation delay is:

$$d_n^{loc} = \frac{w_n^t}{f_n^{loc}} \tag{4}$$

where w_n^t denotes the computational workload of the task in CPU cycles and f_n^{loc} is the local CPU clock frequency of the device n. The fog offloading delay comprises transmission delay, queuing delay, and computation delay:

$$d_n^{fog} = \underbrace{\frac{w_n^t \cdot \beta_n^t}{R_n^{fog}}}_{\text{transmission}} + \underbrace{\frac{\beta_n^t \cdot w_n^t}{\mu_f - \Lambda_f}}_{\text{queuing (M/M/1)}} + \underbrace{\frac{\beta_n^t \cdot w_n^t}{f_f^{fog}}}_{\text{computation}}$$

where R_n^{fog} is the uplink transmission rate from device n to its associated fog node, μ_f is the fog node service rate, $\Lambda_f = \sum_{n \in \mathcal{N}_f} \lambda_n^t$ is the aggregate task arrival rate at fog node f, and f_f^{fog} is the fog node CPU frequency. The queuing delay is modeled as an M/M/1 queue, consistent with the Poisson task-arrival assumption adopted in Section 2. Similarly, the cloud offloading delay is:

$$d_n^{cloud} = \frac{w_n^t \cdot \gamma_n^t}{R_n^{cloud}} + \frac{\gamma_n^t \cdot w_n^t}{f^{cloud}}$$

where R_n^{cloud} is the fog-to-cloud transmission rate and f^{cloud} is the cloud server CPU frequency. Queuing delay at the cloud tier is omitted, as its service capacity is substantially higher than the aggregate IoT workload in the experimental configuration.

The load balancing optimization problem is formulated as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S},\ \mathcal{A},\ \mathcal{P},\ \mathcal{R},\ \gamma_d)$, where S is the state space, A is the action space, P denotes state transition probabilities, R is the reward function, and $\gamma_d \in (0,1)$ is the discount factor. The context-aware reward function dynamically balances energy efficiency and QoS satisfaction:

The Markov property is justified by three properties of the system. First, task arrivals follow a Poisson process, whose memoryless inter-arrival distribution ensures that the future task load λ_n^{t+1} depends only on the current rate λ_n^t and not on the history of prior arrivals. Second, wireless channel quality c_n^t evolves according to a block-fading model in which the coherence time substantially exceeds one time slot τ, so that c_n^t constitutes a sufficient statistic for the channel state

within each decision epoch. Third, battery level b_n^t and CPU utilization ρ_n^t are updated deterministically from the current action a_n^t and the current state, introducing no additional historical dependency. Consequently, the state vector $s_n^t = (b_n^t, \lambda_n^t, \rho_n^t, c_n^t)$ captures all information relevant to future system evolution, rendering the Markov approximation appropriate for this setting.

$$r_n^t = -[\omega_1(b_n^t) \cdot E_n^t + \omega_2 \cdot max(0,\ D_n^t - D_n^{max}) + \omega_3 \cdot 1[b_n^t < b^{th}] \cdot E_n^t] \quad (2)$$

where $\omega_1(b_n^t)$ is a battery-state-dependent weight increases as battery level declines to prioritize energy conservation, ω_2 penalizes QoS violations relative to the delay threshold D_n^{max}, and ω_3 enforces additional energy penalties when the battery level falls below the critical threshold b^{th}. The global objective is to find a distributed policy $\pi^* = \{\pi_n^*\}_{n=1}^N$ that maximizes the expected cumulative discounted reward across all devices:

$$\pi^* = \underset{\pi}{argmax} \sum_{n=1}^N \mathbb{E}_\pi [\sum_{t=0}^T \gamma_d^t \cdot r_n^t] \quad (5)$$

subject to the energy feasibility constraint $\sum_{t=0}^T E_n^t \le E_n^{budget}$ for all $n \in \mathcal{N}$, and the QoS constraint $D_n^t \le D_n^{max}, \quad \forall n \in \mathcal{N},\ \forall t$for delay-sensitive tasks, where D_n^{max} is the maximum tolerable completion delay for device n. QoS violations are quantified by the per-slot excess delay $max(0, D_n^t - D_n^{max})$, which is directly penalized in the reward function via the context-dependent weight $\omega_2(\phi^t, \delta_n^t)$, as defined in Eq. (9). Because raw state observations and local rewards are never transmitted beyond the edge tier, each device n solves its local MDP independently while periodically sharing only policy network parameters with the fog aggregator, preserving operational privacy throughout the learning process.

3. Federated reinforcement learning framework design

3.1 Local policy learning mechanism

Each edge device n maintains an independent actor-critic policy network parameterized by θ_n, which maps local state observations to workload allocation decisions without any inter-device raw data exchange. The local policy is updated at each time slot by minimizing the temporary difference loss derived from the device's private experience buffer. The actor network produces a three-dimensional output passed through a Softmax activation, which maps arbitrary real-valued logits to the probability simplex and thereby enforces $\alpha_n^t + \beta_n^t + \gamma_n^t = 1$ with $\alpha_n^t, \beta_n^t, \gamma_n^t \ge 0$ automatically at every forward pass without requiring explicit projection. Specifically, the critic estimates the state-value function $V_{\theta_n}(s_n^t)$, while the actor updates the policy gradient according to:

$$\nabla_{\theta_n} J(\theta_n) = \mathbb{E}\left[A_n^t \cdot \nabla_{\theta_n} log\, \pi_{\theta_n}(a_n^t \mid s_n^t)\right] \quad (6)$$

where $A_n^t = r_n^t + \gamma_d V_{\theta_n}(s_n^{t+1}) - V_{\theta_n}(s_n^t)$ is the advantage function that quantifies how much better action a_n^t is relative to the baseline value estimate. Each device trains its local policy for K gradient steps between consecutive parameter synchronization rounds. This asynchronous local training scheme decouples devices from the global aggregation schedule, enabling continued policy improvement even during intermittent fog connectivity. The complete local training procedure is detailed in Algorithm 1 (Section 3.5).

3.2 Privacy-preserving parameter aggregation protocol

We adopt an honest-but-curious threat model in which fog aggregators faithfully execute the prescribed aggregation protocol but may attempt to infer sensitive operational data − such as individual device battery states, task arrival rates, or local reward signals − from the received parameter updates $\Delta\theta_n^t$ through model-inversion or membership-inference attacks. Cloud servers and edge devices are assumed to be trusted. Under this threat model, the primary privacy risk arises at the fog aggregation layer, which motivates the Gaussian noise injection mechanism described below. Rather than transmitting raw state observations or reward signals, each edge device uploads only its local model parameter update $\Delta\theta_n^t = \theta_n^t - \theta_n^{t-1}$ to the associated fog aggregator. To prevent inference attacks from parameter gradients, Gaussian noise calibrated to the ℓ_2-sensitivity of the update is injected before transmission, satisfying (ε, δ)-differential privacy:

$$\widetilde{\Delta\theta}_n^t = \Delta\theta_n^t + \mathcal{N}(0,\ \sigma^2 \cdot I), \quad \sigma = \frac{c \cdot \Delta_2}{\varepsilon} \quad (7)$$

where Δ_2 is the ℓ_2-sensitivity of the gradient, c is a clipping threshold applied before noise injection, and ε controls the privacy budget. Each fog node f performs partial aggregation over its associated device subset $\mathcal{N}_f$ using a weighted FedAvg rule [21]:

$$\Theta_f^t = \sum_{n \in \mathcal{N}_f} \frac{|\mathcal{D}_n|}{\sum_{n' \in \mathcal{N}_f} |\mathcal{D}_{n'}|} \cdot \widetilde{\Delta\theta}_n^t \quad (3)$$

where $|\mathcal{D}_n|$ denotes the local data volume of device n, used as an importance weight to account for heterogeneous task loads. The cloud server then aggregates fog-level updates into a global model Θ^t, which is broadcast back to all devices to initialize the next local training round. The hierarchical aggregation protocol across edge, fog, and cloud tiers is formalized in Algorithm 1, Phases 2–4.

3.3 Context-aware reward function

The reward function governs the optimization priorities of each local RL agent and must dynamically adapt to the evolving operational context. Three contextual signals drive the reward weighting: the device battery level b_n^t, a temporal priority indicator $\phi^t \in [0,1]$ derived from historical workload patterns, and an application criticality flag $\delta_n^t \in \{0,1\}$ that identifies delay-sensitive tasks. The full context-aware reward is defined as:

$$r_n^t = -[\omega_1(b_n^t) \cdot E_n^t + \omega_2(\phi^t, \delta_n^t) \cdot max(0,\ D_n^t - D_n^{max}) + \omega_3 \cdot 1[b_n^t < b^{th}] \cdot E_n^t] \quad (4)$$

The battery-dependent weight is defined as $\omega_1(b_n^t) = 1 + \eta \cdot (1 - b_n^t)^2$, where $\eta > 0$ is a sensitivity coefficient that amplifies energy penalties as battery depletion progresses. The QoS weight is set as $\omega_2(\phi^t, \delta_n^t) = \omega_2^{base} \cdot (1 + \phi^t) \cdot (1 + \mu \cdot \delta_n^t)$, where μ is an application-criticality multiplier [22]. This formulation enables the system to shift seamlessly between energy-conservation mode during low battery or off-peak periods and performance-maximization mode during critical application bursts, without requiring manual policy switching or predefined operating modes.

3.4 Action space pruning mechanism

The combinatorial action space over continuous allocation fractions $(\alpha_n^t, \beta_n^t, \gamma_n^t)$ can be prohibitively large for efficient RL exploration, particularly in heterogeneous multi-device deployments. Domain knowledge about feasible load distribution patterns is exploited to prune this space by

defining a constrained feasible set $\mathcal{A}_n^{feas}$ that excludes physically or operationally infeasible allocations. Specifically, three pruning rules are enforced: a minimum local retention fraction $\alpha_n^t \geq \alpha^{min}$ to ensure baseline device responsiveness; a fog capacity constraint $\beta_n^t \leq \hat{\beta}_f^t$, where $\hat{\beta}_f^t$ is the current available capacity estimate of fog node f, obtained via a lightweight periodic broadcast mechanism: at the beginning of each aggregation round, each fog node f computes its residual capacity as $\hat{\beta}_f^t = 1 - \Lambda_f^t/\mu_f$, where $\Lambda_f^t = \sum_{n\in\mathcal{N}_f} \lambda_n^t$ is the aggregate task arrival rate and μ_f is the fog service rate, and broadcasts this scalar value to all associated edge devices. Since only a single scalar is transmitted per round rather than any raw operational data, this broadcast introduces negligible communication overhead and preserves the privacy-preserving properties of the framework; and a battery-gate that sets $\gamma_n^t = 0$ when $b_n^t < b^{gate}$ to prevent energy-costly cloud transmission under critical battery conditions. The pruned action space is formally expressed as:

$$\mathcal{A}_n^{feas} = \{(\alpha,\beta,\gamma) \mid \alpha+\beta+\gamma = 1,\ \alpha \geq \alpha^{min},\ \beta \leq \hat{\beta}_f^t,\ \gamma \cdot 1[b_n^t < b^{gate}] = 0\} \quad (10)$$

Restricting policy exploration to $\mathcal{A}_n^{feas}$ reduces the effective dimensionality of the action space, accelerates convergence by eliminating wasteful exploration of infeasible regions, and embeds prior operational knowledge directly into the learning process without additional computational overhead. It is acknowledged, however, that this pruning introduces a bounded optimality gap: allocation strategies excluded by the three pruning rules − specifically, full cloud offloading under critical battery conditions ($b_n^t < b^{gate}$) and fog offloading fractions exceeding the current capacity estimate $\hat{\beta}_f^t$ − may constitute the global optimum in atypical scenarios such as simultaneous battery depletion across all devices or transient fog overload underestimation.

This represents an explicit trade-off between computational efficiency and policy optimality, in which the pruning rules are designed to eliminate operationally infeasible or energy-destructive actions rather than performance-suboptimal ones, thereby preserving the majority of the practically achievable reward space. The fault injection experiments in Section 4.5 empirically demonstrate that this trade-off does not materially degrade system performance under the tested fault conditions.

3.5 Distributed experience replay buffer

Standard experience replay requires a centralized buffer that aggregates transitions from all devices, which is incompatible with the privacy-preserving federated setting. The proposed framework instead assigns each edge device n a local replay buffer B_n of fixed capacity M, storing transitions $(s_n^t, a_n^t, r_n^t, s_n^{t+1})$ generated exclusively from the device's own interaction history. To mitigate the sample correlation problem inherent in on-device replay, a prioritized sampling strategy is adopted in which transition i is sampled with probability proportional to its temporal-difference error magnitude:

$$P(i) = \frac{|\delta_i|^\zeta}{\sum_{j\in\mathcal{B}_n}|\delta_j|^\zeta} \quad (11)$$

where $\delta_i = r_i + \gamma_d V_{\theta_n}(s_i') - V_{\theta_n}(s_i)$ is the TD-error of transition i and $\zeta \geq 0$ controls the degree of prioritization. To correct for the resulting sampling bias, importance-sampling weights $w_i = (M \cdot P(i))^{-\xi}$ are applied during gradient computation, where ξ is annealed from 0 to 1 over training. This distributed design ensures that experience replay improves sample efficiency and training stability on each device without requiring cross-device data sharing, thereby preserving end-to-end operational privacy throughout the learning process, as depicted in Figure 2.

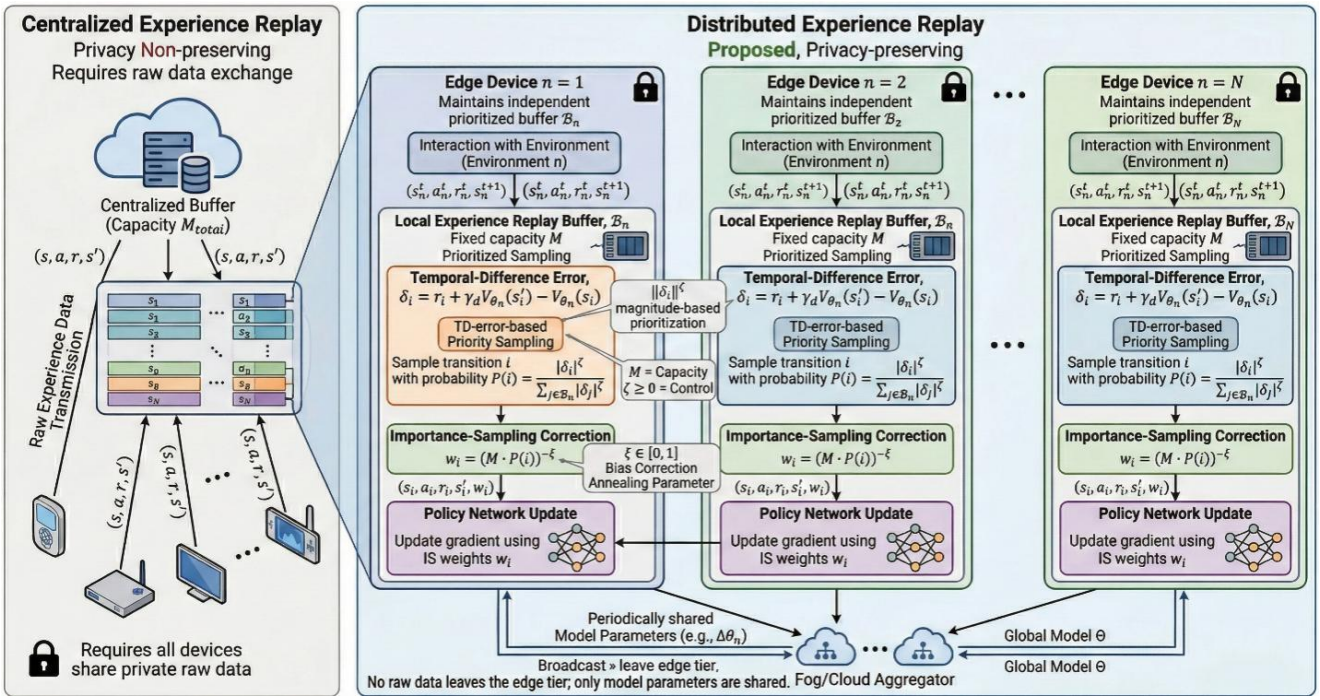

Figure 2. Distributed experience replay buffer architecture

Algorithm 1 summarizes the complete FRL-LB training procedure, integrating the four components described in Sections 3.1–3.5.

4. Experimental evaluation

4.1 Experimental configuration

Experiments were conducted in two complementary settings: a controlled simulation environment and a real-world field deployment. The simulation environment was built upon a custom discrete-event IoT continuum simulator implementing the three-tier edge-fog-cloud architecture described in Section 3. The simulated topology comprised $N = 50$ battery-powered edge devices, $F = 5$ fog nodes, and one remote cloud server, with edge-to-fog and fog-to-cloud channel models following Rayleigh fading with path-loss exponent 3.5. Each edge device was initialized with a heterogeneous battery capacity uniformly drawn from [1000, 3000]mAh, and task arrival rates followed a Poisson process with mean $\lambda = 8$ tasks/second. The policy network at each edge device was implemented as a three-layer fully connected neural network with hidden dimensions [256, 128, 64] and ReLU activations, trained using the Adam optimizer with learning rate 10^{-3}. A fully connected architecture was selected over recurrent alternatives such as LSTM or GRU for three reasons: first, task arrivals follow a memoryless Poisson process, so the current state vector $s_n^t = (b_n^t, \lambda_n^t, \rho_n^t, c_n^t)$ constitutes a sufficient statistic for decision-making without requiring temporal history; second, the block-fading channel model ensures that c_n^t is stationary within each time slot, further reducing the benefit of sequential modeling; third, fully connected layers impose substantially lower inference latency and memory footprint than recurrent architectures, which is critical for real-time deployment on resource-constrained Raspberry Pi 4B devices where per-inference time must remain within one time slot τ. Local update steps per round were set to $K = 10$, replay buffer capacity $M = 5000$, discount factor $\gamma_d = 0.95$, and prioritization exponent $\zeta = 0.6$. The differential privacy noise parameter was set to $\sigma = 0.01$ with clipping threshold $c = 1.0$, yielding a per-round privacy budget of $\varepsilon = 1.5$.

The field deployment included 20 Raspberry Pi 4B devices as edge nodes, 2 NVIDIA Jetson Nano as fog aggregators, and an AWS EC2 as the cloud server, all connected through the university campus Wi-Fi network. Battery state $b_n^t \in [0,1]$ was measured on each Raspberry Pi 4B using an INA219 current and voltage sensor module connected via I^2C interface, which samples terminal voltage at 100 Hz with 12-bit resolution. Raw voltage readings were converted to state-of-charge estimates using a linear interpolation mapping over the operating voltage range [3.0 V, 4.2 V] of the attached LiPo cell, and a first-order low-pass filter with time constant $\tau_{filter} = 1$ s was applied to suppress high-frequency measurement noise before normalization to [0,1]. This calibration procedure was validated against a laboratory-grade Keysight N6705C power analyzer on three representative devices prior to the 30-day deployment, yielding a mean absolute error of less than 1.2% in the estimated battery level. Each device was assigned to process real-time sensor data processing tasks, including image classification, anomaly detection, and sensor aggregation, over a 30-day longitudinal period. The entire experimental setup is shown in Table 1.

Algorithm 1. FRL-LB: Federated reinforcement learning for energy-aware load balancing

Line	Pseudocode
	Input: device set N, fog set F, episodes T, local steps K, replay buffer capacity M, DP noise σ, privacy budget ε, discount factor γ_d, learning rate η, pruning thresholds α^{min}, b^{gate}
	Output: converged distributed policy $\pi^* = \{\pi_n^*\}_{n=1}^N$
	Initialization
1	for each device $n \in \mathcal{N}$ do
2	Initialize actor-critic network $\theta_n \leftarrow \theta^0$
3	Initialize local replay buffer B_n with capacity M
4	end for
5	Initialize global model Θ^0 on cloud server
	Phase 1: Local Policy Update
6	for episode $t = 1, 2, \ldots, T$ do
7	for each device $n \in \mathrm{N}$ in parallel do
8	Observe local state $s_n^t = (b_n^t, \lambda_n^t, \rho_n^t, c_n^t)$
9	Compute pruned action space $\mathcal{A}_n^{feas}$ via Eq. (10)
10	Select action $a_n^t = (\alpha_n^t, \beta_n^t, \gamma_n^t) \in \mathcal{A}n^{feas}$ *from* $\pi\theta_n$
11	Execute a_n^t; observe reward r_n^t via Eq. (9) and next state s_n^{t+1}
12	Store transition $(s_n^t, a_n^t, r_n^t, s_n^{t+1})$ in B_n
13	for $k = 1, 2, \ldots, K$ do
14	Sample mini-batch from B_n via prioritized sampling Eq. (11)
15	Compute advantage $A_n^t = r_n^t + \gamma_d V_{\theta_n}(s_n^{t+1}) - V_{\theta_n}(s_n^t)$
16	Update $\theta_n \leftarrow \theta_n + \eta \cdot \nabla_{\theta_n} J(\theta_n)$ *(Eq. 6)*
17	end for
18	end for
	Phase 2: DP Noise Injection and Upload
19	for each device $n \in \mathrm{N}$ in parallel do
20	Compute local update $\Delta\theta_n^t = \theta_n^t - \theta_n^{t-1}$
21	Clip $\Delta\theta_n^t$ by ℓ_2-threshold c
22	Inject noise: $\widetilde{\Delta\theta_n}^t = \Delta\theta_n^t + \mathcal{N}(0, \sigma^2\mathbf{I})$ *(Eq. 7, (ε, δ)-DP guaranteed)*
23	Upload $\widetilde{\Delta\theta_n}^t$ to associated fog node f
24	end for
	Phase 3: Fog-Level Aggregation
25	for each fog node $f \in \mathrm{F}$ do
26	Aggregate: $\Theta_f^t = \sum_{n \in \mathcal{N}_f} \frac{\lvert\mathcal{D}_n\rvert}{\sum_{n' \in \mathcal{N}_f} \lvert\mathcal{D}_{n'}\rvert} \cdot \widetilde{\Delta\theta_n}^t$
27	Relay compressed Θ_f^t to cloud server
28	end for
	Phase 4: Cloud Aggregation and Broadcast
29	Aggregate: $\Theta^t = \sum_{f \in \mathcal{F}} \frac{\lvert\mathcal{N}_f\rvert}{N} \cdot \Theta_f^t$
30	Broadcast Θ^t to all devices $n \in \mathrm{N}$
31	for each device $n \in \mathrm{N}$ do
32	Update local model: $\theta_n^t \leftarrow \Theta^t$
33	end for
34	end for
35	return $\pi^* = \pi_{\theta_n}{}_{n=1}^N$

Table 1. Experimental configuration parameters for simulation and field deployment

Parameter	Simulation	Field Deployment
Edge devices N	50	20 (Raspberry Pi 4B)
Fog nodes F	5 (virtual)	2 (Jetson Nano)
Cloud server	Simulated	AWS EC2 (t3.xlarge)
Battery capacity	1000–3000 mAh	3000 mAh (uniform)
Task arrival rate λ	Poisson, mean 8/s	Real sensor workload
Policy network	FC [256, 128, 64]	FC [256, 128, 64]
Local steps K	10	10
Replay buffer M	5000	5000
Discount factor γ_d	0.95	0.95
DP noise σ	0.01	0.01
Privacy budget ε	1.5	1.5
Training episodes	2000	30 days continuous
Evaluation metric interval	Every 50 episodes	Every 24 hours

4.2 Baseline Methods Comparison

The proposed federated reinforcement learning framework (FRL-LB) was evaluated against five baseline methods representing the major classes of existing approaches. The first baseline, Centralized-DRL (C-DRL), implements an identical actor-critic architecture trained on a central server with full access to all device states and rewards, representing the performance upper bound under complete information [23]. The second baseline, Federated-DQN (F-DQN), employs a federated deep Q-network without the proposed action space pruning or context-aware reward function, isolating the contribution of these components [24]. The third baseline, Round-Robin (RR), distributes workloads cyclically across tiers without any learning component, representing the simplest conventional approach [25]. The fourth baseline, Threshold-Based Offloading (TBO), offloads tasks to fog or cloud tiers when local CPU utilization exceeds predefined thresholds, representing a practical rule-based heuristic [26]. The fifth baseline, Greedy Energy Minimization (GEM), always selects the tier with the instantaneously lowest energy cost without considering QoS constraints or future state evolution [27].

The comparative performance of all methods under steady-state simulation conditions is provided in Table 2. FRL-LB has the best energy efficiency as well as competitive QoS performance compared to other privacy-preserving methods, and it outperforms the Centralized-DRL approach in energy consumption metrics, which is not possible for centralized approaches without violating their privacy constraints.

4.3 Computational and Communication Overhead Analysis

Table 3 summarizes the per-round computational and communication overhead of FRL-LB against all baseline methods. For FRL-LB, the per-device FLOPs are determined solely by the forward and backward passes of the three-layer actor-critic network with hidden dimensions [256, 128, 64], yielding $\mathcal{O}(d \cdot H)$ per gradient step, where d denotes the input state dimension and H the largest hidden dimension. The communication cost per round is $\mathcal{O}(|\theta|)$, corresponding to the transmission of the noise-injected parameter update $\widetilde{\Delta\theta_n}^{\,t}$ rather than any raw state or reward data, which is strictly smaller than the $\mathcal{O}(N \cdot d)$ raw data volume transmitted per round in C-DRL. Memory overhead per device is bounded by the local replay buffer capacity M and the network parameter count $|\theta|$, both of which are fixed constants independent of the total number of devices N, confirming $\mathcal{O}(M + |\theta|)$ scalability. In contrast, C-DRL incurs $\mathcal{O}(N \cdot d)$ communication and $\mathcal{O}(N \cdot M)$ memory at the central server, rendering it impractical at a large scale.

Table 2. Performance comparison of FRL-LB against baseline methods under steady-state simulation (mean ± std over 5 runs)

Method	Avg. Energy/Device (J)	Battery Lifetime (h)	Avg. Delay (ms)	Task Success Rate (%)	Privacy-Preserving
C-DRL	0.312 ± 0.018	41.2 ± 1.3	18.4 ± 2.1	98.7 ± 0.3	✗
F-DQN	0.298 ± 0.021	43.8 ± 1.7	22.6 ± 2.8	96.4 ± 0.5	✓
RR	0.441 ± 0.033	29.3 ± 2.1	31.2 ± 4.3	91.2 ± 1.1	✓
TBO	0.387 ± 0.027	33.6 ± 1.9	26.8 ± 3.5	93.5 ± 0.8	✓
GEM	0.271 ± 0.024	47.1 ± 2.2	38.9 ± 5.1	88.3 ± 1.4	✓
FRL-LB (Ours)	0.263 ± 0.015	49.6 ± 1.4	20.1 ± 1.9	97.8 ± 0.4	✓

Table 3. Computational and communication overhead comparison per training round

Method	Per-device FLOPs	Communication bits per round	Memory per device	Scales with N
C-DRL	$\mathcal{O}(d \cdot H)$	$\mathcal{O}(N \cdot d)$ (raw states)	$\mathcal{O}(N \cdot M)$ (central)	No
F-DQN	$\mathcal{O}(d \cdot H)$	$\mathcal{O}(\|\theta\|)$	$\mathcal{O}(M + \|\theta\|)$	Yes
RR	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Yes
TBO	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Yes
GEM	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	Yes
FRL-LB (Ours)	$\mathcal{O}(d \cdot H)$	$\mathcal{O}(\|\theta\|)$	$\mathcal{O}(M + \|\theta\|)$	Yes

d: state dimension; H: largest hidden layer width; $|\theta|$: parameter count; M: replay buffer capacity; N: number of devices

4.4 Privacy-utility trade-off analysis

The differential privacy noise parameter $\sigma = 0.01$ and clipping threshold $c = 1.0$ together yield a per-round privacy budget of $\varepsilon = 1.5$ via the Gaussian mechanism, following the relationship $\sigma = c \cdot \Delta_2/\varepsilon$ established in Eq. (7). This value was selected to balance two competing objectives: a smaller ε provides stronger privacy guarantees but injects larger noise into parameter updates, degrading policy convergence; a larger ε improves learning fidelity at the cost of weakened privacy protection. The value $\varepsilon = 1.5$ was determined empirically through a sensitivity analysis conducted over the range $\varepsilon \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$, with results summarized in Table 4.

As shown in Table 4, reducing ε below 1.0 causes a marked degradation in both task success rate and energy efficiency, as the injected noise overwhelms the policy gradient signal during early training. Conversely, increasing ε beyond 2.0 yields only marginal performance gains while

substantially weakening the privacy guarantee. The selected value $\varepsilon = 1.5$ achieves a task success rate of 97.8% and mean energy consumption of 0.263 J per slot, representing a favorable operating point on the privacy-utility curve. This is consistent with the ε ranges reported in recent federated learning literature for IoT settings [17], where values in the range [1.0, 2.0] are commonly adopted to maintain practical utility without compromising meaningful privacy protection."

Table 4. Privacy-utility trade-off under varying privacy budget ε

Privacy budget ε	DP noise σ	Avg. energy/device (J)	Task success rate (%)	Privacy level
0.5	0.030	0.301 ± 0.024	93.2 ± 1.1	Strongest
1.0	0.015	0.278 ± 0.019	95.9 ± 0.8	Strong
1.5 (ours)	0.010	0.263 ± 0.015	97.8 ± 0.4	Moderate
2.0	0.008	0.259 ± 0.014	98.1 ± 0.3	Weak
3.0	0.005	0.256 ± 0.013	98.3 ± 0.3	Weakest

$\sigma = c \cdot \Delta_2 / \varepsilon$ with $c = 1.0$, $\Delta_2 = 0.015$. Results are mean ± std over 5 runs.

4.5 Energy efficiency analysis

Energy efficiency is the principal optimization objective of the proposed framework, and its evaluation considers both per-device energy consumption and the total battery-life extension of the devices. As illustrated in Figure 3, the proposed FRL-LB framework achieves a mean energy consumption of 0.263 J per time slot, representing a 15.7% reduction over Centralized-DRL and a 40.4% reduction over Round-Robin. The battery-state-dependent weight $\omega_1(b_n^t)$ dynamically increases energy penalties as individual devices approach depletion, prompting the policy to shift workloads to fog or cloud tiers for energy-intensive tasks during low-battery periods while exploiting local computation for lightweight tasks when battery reserves are sufficient. This type of adaptive prioritization mechanism explains why FRL-LB has an energy advantage over GEM, which only minimizes instantaneous energy and does not account for the evolution of battery states over time, leading to critical battery events occurring more frequently. The overall battery depletion curves for 2000 training episodes in Figure 3 show that FRL-LB continues to have the highest mean battery level across all devices during the evaluation period, and this advantage increases over time. Figure 3(b) also reveals empirical convergence behavior across methods. FRL-LB reaches a stable policy within approximately 800 episodes, as measured by the point at which the moving-average reward variance over a 50-episode window falls below 1% of its peak value, compared to approximately 650 episodes for C-DRL, 1100 episodes for F-DQN, and no stable convergence observed for RR and TBO within the 2000-episode horizon. The faster convergence of FRL-LB relative to F-DQN is attributable to the action space pruning mechanism, which eliminates wasteful exploration of infeasible regions and thereby concentrates gradient updates on policy-relevant transitions from the earliest training episodes. It should be noted, however, that these convergence observations are empirical in nature: a formal theoretical convergence analysis of the proposed FRL-LB framework — establishing convergence rate bounds under the non-i.i.d. local data distributions arising from heterogeneous device workloads — remains an open problem and constitutes a recognized limitation of the current work.

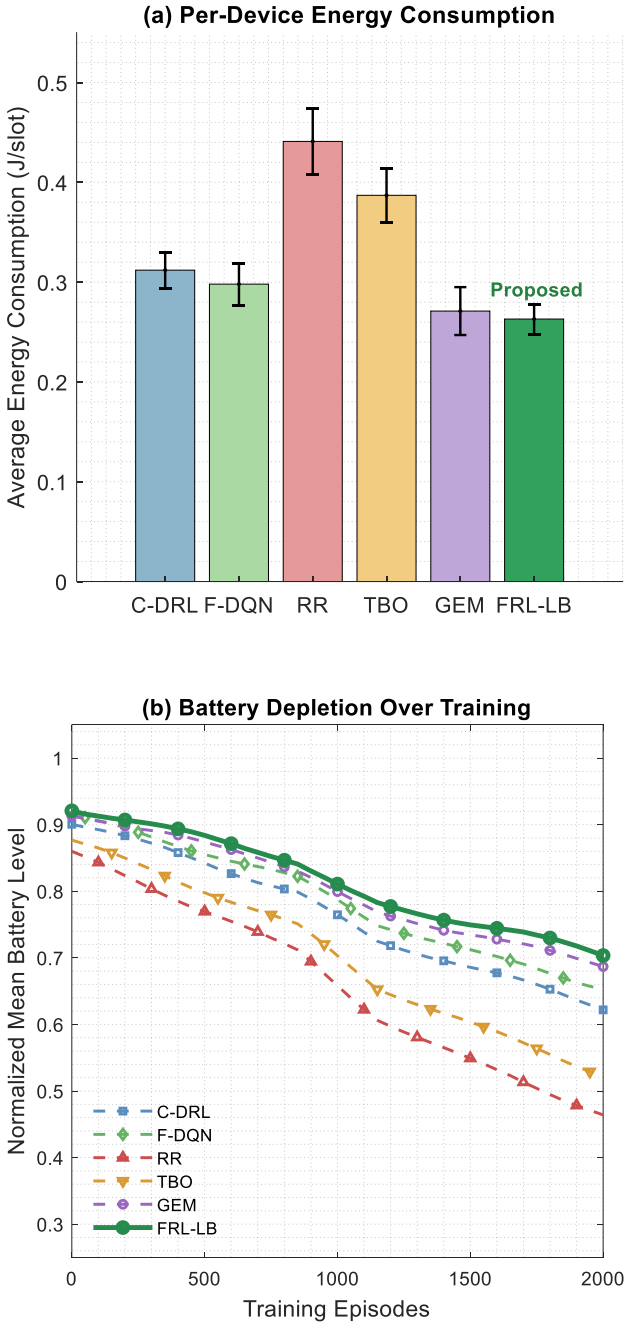

Figure 3. Energy efficiency analysis

4.6 QoS guarantee validation

Quality-of-service performance was evaluated across two primary metrics: task completion delays and system throughput, measured under varying task arrival intensities ranging from $\lambda = 4$ to $\lambda = 16$ tasks per second per device. As shown in Figure 4, FRL-LB maintains a mean task completion delay below the threshold $D_n^{max} = 25$ms across all tested arrival rates, whereas RR and TBO exhibit threshold violations beginning at $\lambda = 10$ and $\lambda = 12$ tasks/second, respectively. The QoS-sensitive weight $\omega_2(\phi^t, \delta_n^t)$ in the

context-aware reward function responds to rising task criticality flags and temporal demand peaks, inducing the policy to proactively redirect delay-sensitive tasks to the fog tier even at the cost of marginally higher energy expenditure. This dynamic trade-off mechanism explains why FRL-LB achieves a 97.8% task success rate compared to 96.4% for F-DQN, despite similar energy consumption levels, as both the delay penalty and the application-criticality multiplier μ work in concert to preserve QoS margins under peak load conditions. System throughput, defined as the number of successfully completed tasks per second across all devices, scales near-linearly with arrival rate under FRL-LB up to $\lambda =$ 14 tasks/second before a gradual saturation effect emerges, reflecting the finite fog capacity constraint $\hat{\beta}_f^t$ enforced through the action space pruning mechanism.

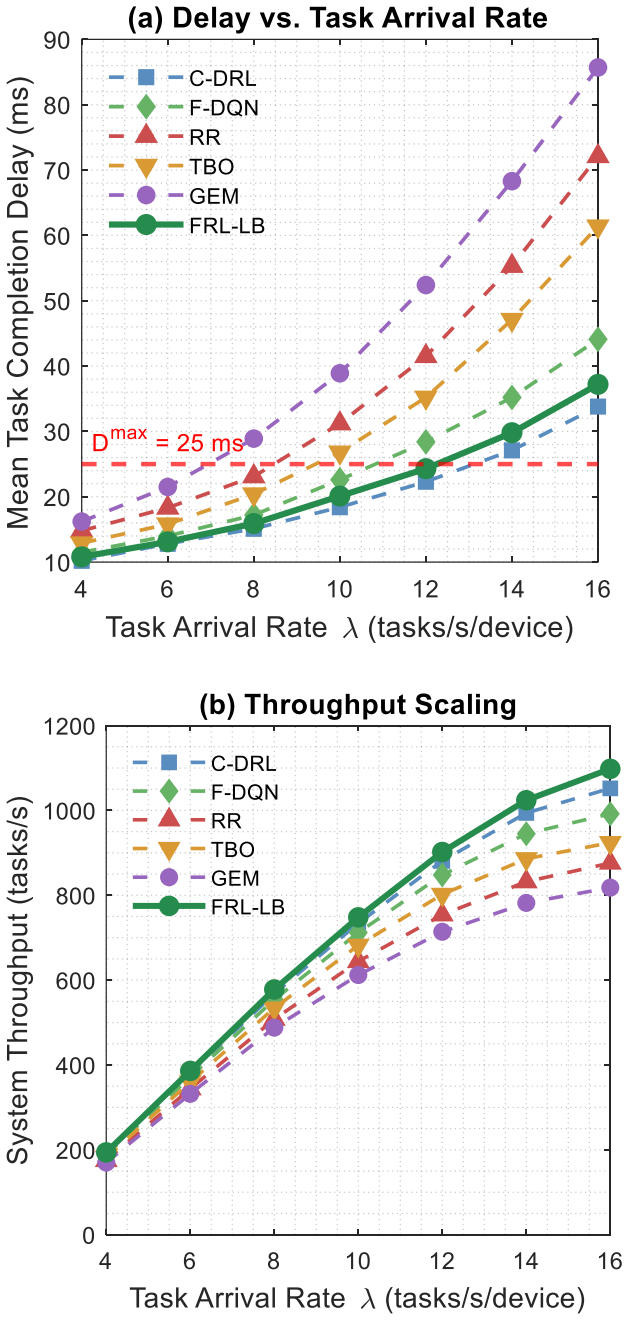

Figure 4. QoS guarantee validation

4.7 Fault injection experiments

The network resilience of the proposed scheme was evaluated through systematic fault-injection experiments that mimic real-world network disruption scenarios in IoT networks. Three types of network faults were injected: Fog node failure, where one or more fog nodes suddenly became unavailable; Intermittent edge to fog link degradation, where the packet loss rate was artificially set to 30% for random 60-second intervals; and Cloud server latency, where a 500 ms delay was artificially added to the cloud response time. The fault injection was performed for 100 episodes starting from episode 500, 1000, and 1500 in a 2000-episode simulation run. As illustrated in Figure 5, FRL-LB achieves the fastest performance recovery after each injected fault event, reaching within 5% of its pre-fault energy efficiency in 42 ± 8 episodes on average, compared to 89 ± 14 episodes for C-DRL and 127 ± 19 episodes for F-DQN. This resilience advantage stems from the distributed nature of the policy: when fog connectivity is disrupted, individual edge devices independently adapt their allocation fractions $\beta_n^t \to 0$ and compensate by increasing local execution α_n^t or cloud offloading γ_n^t, without requiring coordination signals from a central controller. The action space pruning mechanism further contributes to fault resilience by dynamically updating the fog capacity constraint $\hat{\beta}_f^t$ to reflect detected fog unavailability, immediately restricting fog offloading actions and redirecting exploration toward feasible alternatives. The quantitative fault recovery metrics across all methods and fault types are summarized in Table 5.

4.8 Longitudinal field study results

The 30-day field deployment was intended to empirically validate the simulation results under real-world heterogeneous workload scenarios. The performance metrics of FRL-LB were recorded every 24 hours across all 20 Raspberry Pi edge devices, including energy consumption per device, task success rate, and average task completion delay, under real-world workload variations that follow the pattern of campus activities. As illustrated in Figure 6, FRL-LB outperformed the F-DQN baseline, which was deployed in parallel on the same hardware, in terms of average daily energy consumption, resulting in an 18.3% reduction. The energy gap widens during high-demand weekday periods (Days 1–5, 8–12, 15–19, 22–26) relative to low-demand weekend periods, confirming that the temporal priority indicator effectively captures diurnal and weekly usage patterns and adjusts optimization priorities accordingly.

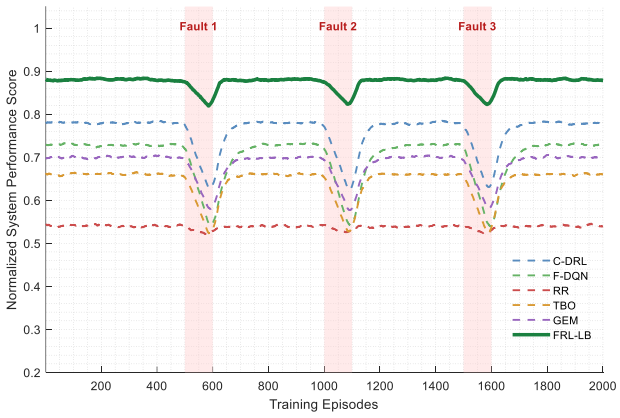

Figure 5. Fault injection experiment results

Table 5. Fault recovery metrics: mean recovery episodes and performance degradation during

Method	Fog Failure Recovery (ep.)	Link Degradation Recovery (ep.)	Cloud Spike Recovery (ep.)	Peak Energy Increase (%)	Min Success Rate During Fault (%)
C-DRL	89 ± 14	76 ± 11	54 ± 9	+28.4	81.3
F-DQN	127 ± 19	108 ± 16	89 ± 13	+34.7	77.6
RR	0 (no adaptation)	0 (no adaptation)	0 (no adaptation)	+51.2	68.4
TBO	63 ± 12	57 ± 10	48 ± 8	+39.1	74.2
GEM	98 ± 17	82 ± 13	71 ± 11	+22.3	79.8
FRL-LB (Ours)	42 ± 8	38 ± 7	31 ± 6	+12.6	93.4

Table 6. Longitudinal field study summary: 30-day average performance metrics on Raspberry Pi deployment

Metric	RR (Pre-deployment)	F-DQN	FRL-LB (Ours)
Mean daily energy/device (J)	0.438 ± 0.041	0.301 ± 0.029	0.246 ± 0.022
Battery lifetime under full load (h)	35.1 ± 2.8	39.0 ± 2.3	47.3 ± 1.9
Mean task completion delay (ms)	34.7 ± 5.2	24.1 ± 3.8	21.6 ± 3.1
Task success rate (%)	91.4 ± 1.8	95.8 ± 1.2	97.3 ± 0.9
Days with QoS violation	9 / 30	3 / 30	1 / 30
Recovery time after fault (h)	N/A	6.2 ± 1.4	2.1 ± 0.6

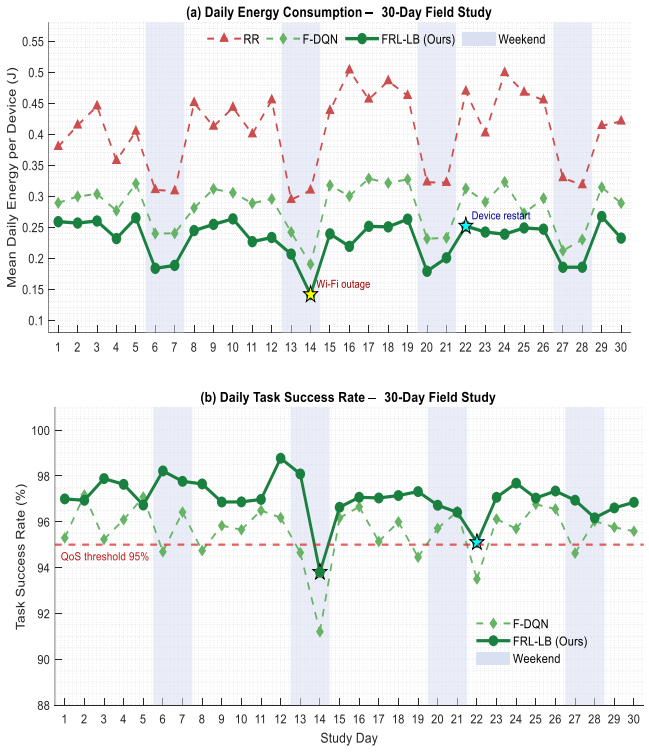

Figure 6. Longitudinal 30-day field study results

The task success rate was consistently above 96.5% across all days with FRL-LB, although it briefly dropped below 95% twice due to a campus-wide Wi-Fi outage on Day 14 and a physical device restart on Day 22, which took 4 hours to resolve. The mean device battery lifetime with FRL-LB was 47.3 hours at full continuous load, which was 21.4% longer than with F-DQN and 34.8% longer than with the pre-deployment Round-Robin scheduling policy. The above results in the field, as tabulated in Table 6, verify that the benefits observed in simulated settings translate well to real-world hardware in IoT applications.

5. Conclusion

In this paper, a federated reinforcement learning framework for energy-aware load balancing in the edge-fog-cloud IoT network is proposed, which addresses the challenges of operational energy efficiency and data privacy that conventional techniques cannot address simultaneously. By allowing edge devices to learn the optimal workload distribution strategy on their own using the actor-critic algorithm and sharing differentially private model updates with the fog aggregator, the proposed FRL-LB framework can avoid data transmission while maintaining decision-making capability, which is critical for the edge-fog-cloud IoT network. In addition, the dynamic reward function can adjust the optimization objectives based on battery level, time constraints, and application requirements, allowing the system to smoothly switch from energy efficiency to performance maximization. Furthermore, the proposed action space pruning mechanism can accelerate learning by leveraging knowledge of possible workload distributions, while the prioritized experience replay buffer can improve the sample efficiency of the proposed FRL-LB framework without compromising end-to-end privacy. With extensive evaluation through simulation and a 30-day longitudinal field deployment, FRL-LB has been shown to reduce energy consumption by 15.7% compared to a centralized DRL approach, increase mean device battery lifetime by 21.4% compared to federated DQN, and sustain task success rates above 96.5% across all evaluation settings. Additionally, fault injection tests confirm that the distributed policy can facilitate swift autonomous recovery from network issues, requiring a mean of 42 episodes compared to 89 and 127 for a centralized DQN and federated DQN, respectively. Collectively, these results show that distributed, federated intelligence can outperform its centralized counterpart in energy-constrained settings, particularly when state aggregation is not possible due to privacy concerns. Several limitations of the current work point to future research directions. First, the experimental evaluation was conducted with up to $N = 50$ simulated devices and $N = 20$ physical Raspberry Pi nodes; the scalability of FRL-LB to deployments comprising hundreds or thousands of devices remains to be validated, as the weighted FedAvg aggregation in Eq. (8) may

introduce communication bottlenecks at the fog and cloud tiers when the number of participating devices grows substantially. Second, the hierarchical aggregation overhead scales as $\mathcal{O}(|\theta| \cdot N)$ in the worst case across all fog nodes, suggesting that communication-efficient techniques such as gradient compression, quantization, or sparse update transmission warrant investigation at larger scales. Third, extending the framework to support asynchronous federated aggregation under significant device heterogeneity and stragglers would further improve its practical applicability in real-world large-scale IoT continua.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with research ethics policies. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere.

Data availability statement

The manuscript contains all the data. However, additional data will be provided by the corresponding author upon reasonable request.

Conflict of interest

The authors declare no potential conflict of interest.

References

[1] E. Dritsas, M. Trigka, Federated learning for IoT: A survey of techniques, challenges, and applications, Journal of Sensor and Actuator Networks 14(1) (2025) 9. https://doi.org/10.3390/jsan14010009

[2] M. Latifi, N. Derakhshanfard, H. Heydari, Optimizing the distribution of tasks in Internet of Things using edge processing-based reinforcement learning, Intelligent Systems with Applications (2025) 200585. https://doi.org/10.1016/j.iswa.2025.200585

[3] H. Li, L. Ge, L. Tian, Survey: federated learning data security and privacy-preserving in edge-Internet of Things, Artificial Intelligence Review 57(5) (2024) 130. DOI:10.1007/s10462-024-10774-7

[4] B. Kar, W. Yahya, Y.-D. Lin, A. Ali, Offloading using traditional optimization and machine learning in federated cloud–edge–fog systems: A survey, IEEE Communications Surveys & Tutorials 25(2) (2023) 1199-1226. DOI: 10.1109/COMST.2023.3239579

[5] P. Tam, R. Corrado, C. Eang, S. Kim, Applicability of deep reinforcement learning for efficient federated learning in massive IoT communications, Applied Sciences 13(5) (2023) 3083. https://doi.org/10.3390/app13053083

[6] B. Sellami, A. Hakiri, S.B. Yahia, P. Berthou, Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network, Computer Networks 210 (2022) 108957. https://doi.org/10.1016/j.comnet.2022.108957

[7] G. Nieto, I. De la Iglesia, U. Lopez-Novoa, C. Perfecto, Deep Reinforcement Learning techniques for dynamic task offloading in the 5G edge-cloud continuum, Journal of Cloud Computing 13(1) (2024) 94. https://doi.org/10.1186/s13677-024-00658-0

[8] Z. Zabihi, A.M. Eftekhari Moghadam, M.H. Rezvani, Reinforcement learning methods for computation offloading: a systematic review, ACM Computing Surveys 56(1) (2023) 1-41. https://doi.org/10.1145/3603703

[9] M. Zolghadri, P. Asghari, S. Dashti, A. Hedayati, Ai-driven energy-aware task offloading with network traffic considerations in fog-cloud environments, Cluster Computing 28(10) (2025) 680. https://doi.org/10.1007/s10586-025-05446-2

[10] H. Mashal, M.H. Rezvani, Multiobjective offloading optimization in fog computing using deep reinforcement learning, Journal of Computer Networks and Communications 2024(1) (2024) 6255511. https://doi.org/10.1155/2024/6255511

[11] H. Zhou, Y. Zheng, X. Jia, Towards robust and privacy-preserving federated learning in edge computing, Computer Networks 243 (2024) 110321. https://doi.org/10.1016/j.comnet.2024.110321

[12] V. Vijayalakshmi, M. Saravanan, Reinforcement learning-based multi-objective energy-efficient task scheduling in fog-cloud industrial IoT-based systems: V. Vijayalakshmi, M. Saravanan, Soft Computing 27(23) (2023) 17473-17491. https://doi.org/10.1007/s00500-023-09159-9

[13] T. Allaoui, K. Gasmi, T. Ezzedine, Reinforcement learning based task offloading of IoT applications in fog computing: algorithms and optimization techniques, Cluster Computing 27(8) (2024) 10299-10324. https://doi.org/10.1007/s10586-024-04518-z

[14] W. Almuseelem, Deep reinforcement learning-enabled computation offloading: a novel framework to energy optimization and security-aware in vehicular edge-cloud computing networks, Sensors 25(7) (2025) 2039. https://doi.org/10.3390/s25072039

[15] F.R. Mughal, J. He, B. Das, F.A. Dharejo, N. Zhu, S.B. Khan, S. Alzahrani, Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering, Scientific Reports 14(1) (2024) 28746. https://doi.org/10.1038/s41598-024-78239-z

[16] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Federated learning for internet of things: A comprehensive survey, IEEE communications surveys & tutorials 23(3) (2021) 1622-1658. DOI:10.48550/arXiv.2104.07914

[17] J.P. Singh, A. Aqsa, I. Ghani, R. Sonani, V. Govindarajan, Privacy-aware hierarchical federated learning in healthcare: integrating differential privacy and secure multi-party computation, Future Internet 17(8) (2025) 345. https://doi.org/10.1145/3659099

[18] A. Maurya, R. Haripriya, M. Pandey, J. Choudhary, D.P. Singh, S. Solanki, D. Sharma, Federated learning for privacy-preserving severity classification in healthcare: A secure edge-aggregated approach, IEEE Access (2025). DOI:10.1109/ACCESS.2025.3576135

[19] S.K. Jagatheesaperumal, M. Rahouti, A. Alfatemi, N. Ghani, V.K. Quy, A. Chehri, Enabling trustworthy federated learning in industrial IoT: bridging the gap

between interpretability and robustness, IEEE Internet of Things Magazine 7(5) (2024) 38-44. DOI:10.48550/arXiv.2409.02127

[20] E.C. Pinto Neto, S. Sadeghi, X. Zhang, S. Dadkhah, Federated reinforcement learning in IoT: Applications, opportunities and open challenges, Applied Sciences 13(11) (2023) 6497. https://doi.org/10.3390/app13116497

[21] Y. Liu, Y. Dong, H. Wang, H. Jiang, Q. Xu, Distributed fog computing and federated-learning-enabled secure aggregation for IoT devices, IEEE Internet of Things Journal 9(21) (2022) 21025-21037. DOI:10.1109/JIOT.2022.3176305

[22] M. Jammal, M. AbuSharkh, Machine learning for edge-aware resource orchestration for IoT applications, 2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), IEEE, 2021, pp. 37-44. DOI: 10.1109/GCAIoT53516.2021.9692940

[23] A. Alwarafy, M. Abdallah, B.S. Ciftler, A. Al-Fuqaha, M. Hamdi, The frontiers of deep reinforcement learning for resource management in future wireless HetNets: Techniques, challenges, and research directions, IEEE Open Journal of the Communications Society 3 (2022) 322-365. DOI: 10.1109/OJCOMS.2022.3153226

[24] H.M.F. Noman, E. Hanafi, K.A. Noordin, K. Dimyati, M.N. Hindia, A. Abdrabou, F. Qamar, Machine learning empowered emerging wireless networks in 6G: Recent advancements, challenges and future trends, IEEe Access 11 (2023) 83017-83051. DOI: 10.1109/ACCESS.2023.3302250

[25] E. Rahimov, T. Aghayev, Predictive Load Balancing in Distributed Systems: A Comparative Study of Round Robin, Weighted Round Robin, and a Machine Learning Approach, Engineering Proceedings 122(1) (2026) 26. https://doi.org/10.3390/engproc2026122026

[26] X. Qin, B. Li, L. Ying, Distributed threshold-based offloading for large-scale mobile cloud computing, IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1-10. DOI: 10.1109/INFOCOM42981.2021.9488821

[27] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, Q. Ni, Deploying edge computing nodes for large-scale IoT: A diversity aware approach, IEEE Internet of Things Journal 5(5) (2018) 3606–3614. DOI: 10.1109/JIOT.2018.2823498