International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 08 Augustl-2026 | Impact Factor: 3.5

AERAS: Adaptive AI-Driven Resource Allocation for Mission-Critical Tactical Edge Computing

Capt Sahil

How to Cite this Article:

Sahil, C. (2026). AERAS: Adaptive AI-Driven Resource Allocation for Mission-Critical Tactical Edge Computing. International Journal of Creative and Open Research in Engineering and Management, 02(8), 1-9. https://doi.org/10.55041/ijcope.v2i8.107

https://doi.org/10.55041/ijcope.v2i8.107

***Abstract*—Mission-critical tactical edge systems require re-source allocation that maintains low latency, energy efficiency and SLA compliance under bursty workloads and heteroge-neous nodes. This paper presents AERAS (Adaptive Attention-Enhanced Reinforcement Allocation System), which combines GRU-based workload forecasting, attention-weighted task–node encoding and PPO-based multi-objective scheduling for closed-loop edge orchestration. The proposed framework jointly opti-mizes latency, energy consumption, throughput and SLA satis-faction through a normalized reward design and is evaluated across high-load ISR, drone-swarm coordination and cyber-defence alert scenarios. Simulation results show that AERAS consistently outperforms Round Robin, First Fit and DQN baselines. In the high-load ISR case, it achieves 0.62 s mean latency and 4.8 J/task energy, while also reaching 94.0% and 90.5% SLA satisfaction in the drone-swarm and cyber-defence scenarios, respectively. These results indicate that predictive and attention-guided reinforcement learning is a practical approach for mission-aware tactical edge scheduling.**

***Index Terms*—Edge computing; tactical systems; adaptive scheduling; reinforcement learning; workload forecasting; atten-tion mechanism; defence AI; PPO.**

I. Introduction

Mission-critical tactical systems increasingly rely on dis-tributed edge computing to process latency-sensitive data from ISR platforms, autonomous vehicles, mobile command assets and cyber-defence sensors. However, intermittent connectivity, heterogeneous edge nodes, bursty task arrivals and residual-energy constraints make static orchestration and purely reac-tive scheduling ineffective in many operational scenarios.

A substantial body of work has addressed edge task schedul-ing and resource allocation using heuristic, optimization-based and deep reinforcement learning (DRL) methods. While heuristic approaches are simple to deploy, they of-ten lack adaptability under dynamic workload conditions. Optimization-based methods can provide strong performance, but they typically depend on accurate system models and are less suitable for highly variable tactical environments. DRL-based schedulers improve adaptability by treating scheduling as a sequential decision problem; nevertheless, most existing methods remain reactive, rely only on the current observed state, or optimize a limited subset of objectives such as delay or energy alone [1]–[6], [11], [13]. Recent surveys further indicate that prediction-aware and deployment-aware scheduling remain open research directions in heterogeneous edge infrastructures [8], [9].

To address these limitations, this paper presents AERAS (Adaptive Attention-Enhanced Reinforcement Allocation Sys-tem), an AI-driven framework that integrates GRU-based workload forecasting, attention-based state encoding and PPO-driven multi-objective scheduling in a closed-loop tactical edge setting. The proposed framework jointly optimizes latency, energy consumption, throughput and SLA satisfaction while accounting for residual-energy constraints and heterogeneous edge resources. Its effectiveness is evaluated across high-load ISR, drone-swarm coordination and cyber-defence alert scenarios.

The main contributions of this paper are as follows:

- A tactical edge scheduling framework that combines short-horizon workload forecasting, attention-weighted task–node encoding and PPO-based multi-objective con-trol.
- A residual-energy-aware reward formulation and normal-ization strategy for stable policy learning under bursty and non-stationary workloads.
- A complete hardware–software–simulation implementa-tion pathway for deployable evaluation in mission-critical edge environments.
- A scenario-based experimental assessment covering la-tency, energy, SLA satisfaction, scalability and robust-ness.

The remainder of this paper is organized as follows. Sec-tion II reviews related work and positions AERAS against recent DRL, attention-based and multi-objective scheduling approaches. Section III presents the problem formulation and methodology, followed by the system architecture, implemen-tation, experimental evaluation and conclusion in Sections IV–XI.

II. Related Work and Research Positioning

Resource scheduling in edge computing has received signifi-cant attention because heterogeneous edge nodes must support latency-critical applications under dynamic workloads. Prior surveys highlight scalability, uncertainty and realistic deploy-ment under resource constraints as the main open issues [8], [9].

Heuristic and optimization-based methods are simple and interpretable, but they often rely on fixed rules or strong assumptions and perform poorly under bursty traffic.

Learning-based schedulers improve adaptability, yet many remain reactive and rely only on the current system state [5], [11], [13], [14].

Deep reinforcement learning has become a strong approach for edge scheduling, with DQN-, actor–critic- and PPO-based methods showing gains in delay, energy and deadline satis-faction [5], [6], [13]. Multi-objective RL further shows that

latency, energy and cost can be balanced more effectively when task preferences vary over time [6], [7].

Attention and transformer-based encoders improve modeling of task dependencies and heterogeneous node relationships [10]. However, these methods usually focus on current-state modeling and do not combine short-horizon workload forecasting with residual-energy-aware tactical scheduling [7], [13].

Accordingly, AERAS is positioned as a closed-loop tactical edge scheduling framework that integrates GRU-based forecasting, attention-weighted state encoding and PPO-driven multi-objective allocation for mission-oriented edge environments.

III. Problem Statement and Methodology

Consider a tactical edge network with N heterogeneous compute nodes $\mathrm{N} = \{n_1, n_2, \ldots, n_N\}$ and mission tasks $\mathrm{T} = \{t_1, t_2, \ldots\}$ arriving over discrete time steps. Each task t_k is represented by $\langle c_k, d_k, p_k, \delta_k \rangle$, where c_k denotes CPU cycle demand, d_k the data volume, p_k the priority class and δ_k the deadline. Each node n_i is described by $s_i = \langle u_i, q_i, m_i, b_i, e_i \rangle$, capturing utilization, queue length, memory, bandwidth and residual energy.

The scheduler selects an action $a_k \in \mathrm{N} \cup \{\text{cloud}\}$ to minimize the composite cost

$$J = \sum_k (\alpha \ell_k + \beta \epsilon_k - \gamma \phi_k + \lambda v_k), \tag{1}$$

where ℓ_k is task latency, ϵ_k is energy consumption, $\phi_k \in \{0, 1\}$ indicates SLA satisfaction and v_k is the SLA violation penalty. Since task arrivals and node states vary over time, the problem is non-stationary and requires adaptive multi-objective scheduling.

AERAS addresses this problem through a four-stage closed-loop pipeline: observe, forecast, encode and act. The scheduler first observes task and node telemetry, then uses a GRU to predict short-horizon workload evolution over H decision steps. An attention layer next forms a relevance-weighted representation of task–node interactions and a PPO actor–critic finally selects the allocation action while the critic stabilizes policy learning.

The GRU stage is given by

$$h_t = \mathrm{GRU}(S_t, h_{t-1}), \tag{2}$$

with the workload forecast

$$\hat{w}_{t+1:t+H} = f_{\mathrm{pred}}(h_t). \tag{3}$$

The attention mechanism computes

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}, \qquad e_{ij} = v^\top \tanh(W s_j + U q_i), \tag{4}$$

and the context vector

$$c_i = \sum_j \alpha_{ij} s_j, \tag{5}$$

which is passed to the PPO actor as the encoded state.

The PPO update uses the clipped objective

$$L_{\mathrm{CLIP}}(\vartheta) = \mathbb{E}_t \left[\min \left(r_t(\vartheta)\hat{A}_t, \; \mathrm{clip}(r_t(\vartheta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{6}$$

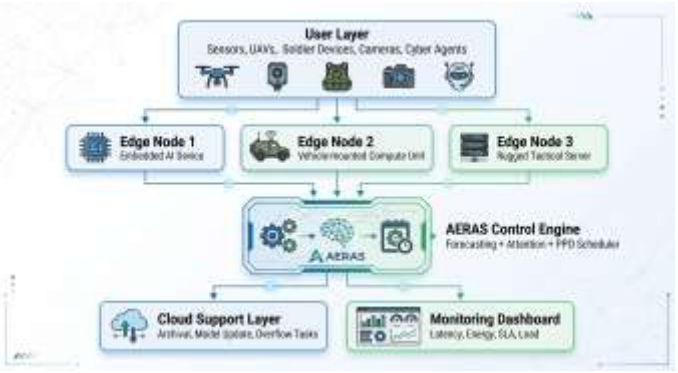

Fig. 1. Three-tier AERAS architecture: user layer (task generation), edge control layer (adaptive scheduler with GRU+attention+PPO), and cloud fallback layer.

where

$$r_t(\vartheta) = \frac{\pi_\vartheta(a_t|s_t)}{\pi_{\vartheta_{\mathrm{old}}}(a_t|s_t)}. \tag{7}$$

The scalar reward is

$$r_t = -\alpha \ell_t - \beta \epsilon_t + \gamma \phi_t - \lambda v_t, \tag{8}$$

and the normalized terms are

$$\tilde{\ell}_t = \frac{\ell_t}{\delta_t}, \qquad \tilde{\epsilon}_t = \frac{\epsilon_t}{E_{\max}}, \tag{9}$$

leading to

$$r_t = -\alpha \tilde{\ell}_t - \beta \tilde{\epsilon}_t + \gamma \phi_t - \lambda v_t. \tag{10}$$

Overall, the methodology converts scheduling from a reactive allocation problem into a predictive multi-objective control problem. By combining workload forecasting, attention-guided state modeling and PPO-based policy optimization, AERAS is designed to support heterogeneous tactical edge environments while balancing latency, energy efficiency, throughput and SLA compliance.

IV. System Architecture

The AERAS architecture follows a three-tier tactical edge model (Fig. 1). The user layer generates mission workloads from ISR sensors, UAV telemetry sources and cyber-monitoring agents, while the edge control layer hosts heterogeneous compute nodes and the AERAS scheduler for telemetry ingestion and placement decisions. The cloud layer provides overflow support for non-urgent tasks. This design follows established multi-tier edge architectures [1], [4] while adding prediction-guided closed-loop control [8].

V. Hardware Implementation

A. Hardware Components

AERAS uses four hardware layers: workload-generating devices, heterogeneous edge compute nodes, a communication fabric and a supervisory orchestration node. The workload-generating layer includes surveillance cameras, radar interfaces, UAV telemetry sources, battlefield IoT sensors and cyber-monitoring agents. Since residual energy is treated as a scheduling variable, node-level power telemetry must be collected in real time through dedicated measurement modules.

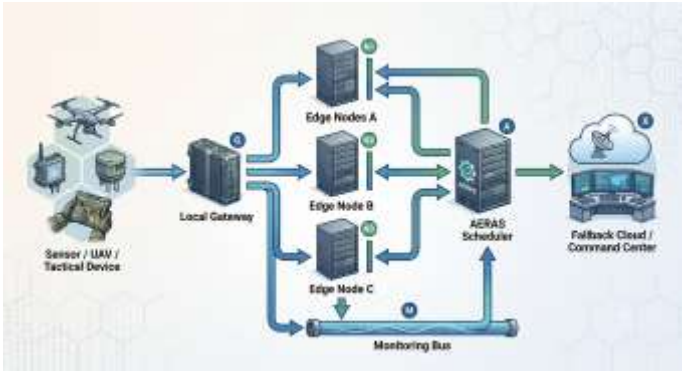

Fig. 2. Hardware deployment pathway: device-generated tasks flow through local gateways to heterogeneous edge nodes, which report telemetry to the AERAS scheduler; the scheduler issues allocation decisions and may redirect overflow to cloud fallback.

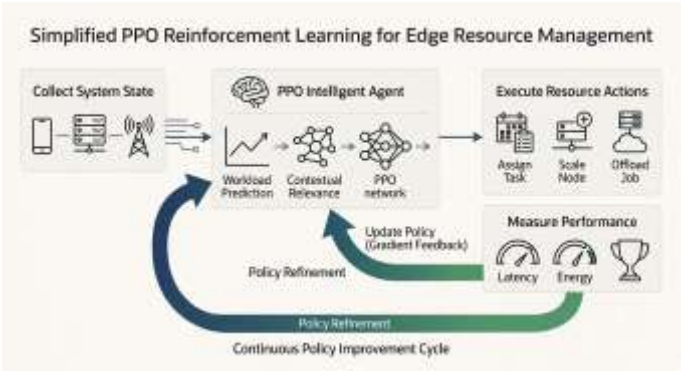

Fig. 3. Algorithmic flow of AERAS: observe → GRU forecast → attention encode → PPO act → execute → reward → policy update.

B. Hardware Deployment Flow

Fig. 2 illustrates the hardware deployment pathway. Modularity is a design requirement so that sensing, scheduling and execution can be independently profiled and upgraded essential in tactical systems where nodes may be repositioned or degraded during operation.

VI. Algorithm Implementation

A. Core Scheduling Logic and GRU Forecasting

Let the system state at step t be $\mathbf{S}_t = [\mathbf{s}_{t,1}, \ldots, \mathbf{s}_{t,N}]$. The GRU hidden state is updated as:

$$\mathbf{h}_t = \mathrm{GRU}(\mathbf{S}_t, \mathbf{h}_{t-1}) \tag{11}$$

producing a predicted workload vector $\hat{\mathbf{w}}_{t+1:t+H}$ for the next H steps.

B. Attention-Based State Encoding

The attention mechanism scores each task-node pair:

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{j'} \exp e_{ij'}}, \quad e_{ij} = \mathbf{v}^\top \tanh \mathbf{W}\mathbf{s}_j + \mathbf{U}\mathbf{q}_i \tag{12}$$

where $\mathbf{q}_i$ is the task feature vector, $\mathbf{s}_j$ is the node state and $\mathbf{v}, \mathbf{W}, \mathbf{U}$ are learnable parameters. The context vector $\mathbf{c}_i = \sum_j \alpha_{ij}\mathbf{s}_j$ provides the encoded input to the PPO actor.

C. PPO Policy Update

The PPO clipped objective is:

$$L^{\mathrm{CLIP}}(\vartheta) = \mathbb{E}_t \left[\min \left(r_t(\vartheta)\hat{A}_t,\ \mathrm{clip}(r_t(\vartheta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{13}$$

where $r_t(\vartheta) = \pi_\vartheta(a_t|s_t)/\pi_{\vartheta_{\mathrm{old}}}(a_t|s_t)$ is the probability ratio, $\hat{A}_t$ is the Generalized Advantage Estimate (GAE) and $\epsilon = 0.2$ prevents destabilizing policy updates.

D. Reward Design

The multi-objective scalar reward at step t is:

$$r_t = -\alpha \ell_t - \beta \varepsilon_t + \gamma \phi_t - \lambda v_t \tag{14}$$

with $(\alpha, \beta, \gamma, \lambda) = (0.4, 0.2, 0.3, 0.1)$ as default tactical weights; these can be mission-profile-adapted.

Fig. 3 presents the closed-loop algorithmic flow. The pseudocode is shown in Algorithm 1.

Algorithm 1 AERAS Scheduling Loop

Require: Edge node set N, task stream, trained GRU, attention, PPO
Initialise $\mathbf{h}_0 \leftarrow \mathbf{0}$, episode buffer $B \leftarrow \emptyset$
for each decision step t **do**
  Observe $\mathbf{S}_t$ (task queue + node telemetry)
  $\mathbf{h}_t \leftarrow \mathrm{GRU}(\mathbf{S}_t, \mathbf{h}_{t-1})$; compute $\hat{\mathbf{w}}_{t+1}$
  Compute attention weights $\{\alpha_{ij}\}$ via (12)
  Form context $\mathbf{c}_t \leftarrow \sum_j \alpha_{ij}\mathbf{s}_j$
  $a_t \leftarrow \pi_\theta(\cdot \mid \mathbf{c}_t)$ ▷ PPO actor
  Execute a_t; observe $\ell_t, \varepsilon_t, \phi_t, v_t$
  $r_t \leftarrow (14)$; store (s_t, a_t, r_t, s_{t+1}) in B
  if $|B| = T_{\mathrm{update}}$ **then**
    Update θ by maximising (13) over B; clear B
  end if
end for

E. GRU-Based Workload Forecasting

Let $S_t = [s_{t,1}, s_{t,2}, \ldots, s_{t,N}]$ denote the system state at time step t, where each $s_{t,i}$ contains node-level telemetry and task-queue features. The GRU module maps the input sequence $\{S_{t-H+1}, \ldots, S_t\}$ to a forecast of near-future workload demand over horizon H:

$$h_t = \mathrm{GRU}(S_t, h_{t-1}), \tag{15}$$

$$\hat{w}_{t+1:t+H} = f_{\mathrm{pred}}(h_t), \tag{16}$$

where h_t is the hidden state and $f_{\mathrm{pred}}(\cdot)$ is a linear prediction head. In the implementation used in this work, the GRU predictor uses one recurrent layer with a hidden dimension of 64, followed by a fully connected projection layer. Dropout is applied during training to reduce overfitting. The forecast horizon is selected as $H \in \{3, 4, 5\}$ decision steps, depending on the mission profile.

F. Attention-Based State Encoding

The attention mechanism scores each task–node pair as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}, \tag{17}$$

$$e_{ij} = v^\top \tanh(W s_j + U q_i), \tag{18}$$

where q_i is the task feature vector, s_j is the node state vector and v, W and U are learnable parameters. The attention output is the context vector

$$c_i = \sum_j \alpha_{ij} s_j, \tag{19}$$

which is passed to the PPO actor as the encoded state representation. The attention block uses single-head additive attention with projection dimension 64. This design prioritizes the most relevant task–node interactions under heterogeneous and bursty workloads.

G. PPO Policy Update

The PPO objective is defined as

$$L_{\text{CLIP}}(\vartheta) = \mathbb{E}_t\left[\min\left(r_t(\vartheta)\hat{A}_t,\ \text{clip}(r_t(\vartheta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{20}$$

where

$$r_t(\vartheta) = \frac{\pi_\vartheta(a_t|s_t)}{\pi_{\vartheta_{\text{old}}}(a_t|s_t)}, \tag{21}$$

and $\hat{A}_t$ is the generalized advantage estimate (GAE). The clipping coefficient is set to $\epsilon = 0.2$ in the default configuration. The actor–critic model is optimized using Adam with learning rate 3×10^{-4}, batch size 64 and five update epochs. The discount factor γ and GAE parameter λ_{GAE} are set to 0.99 and 0.95, respectively.

H. Reward Design and Normalization

The scalar reward at time step t is

$$r_t = -\alpha \ell_t - \beta \varepsilon_t + \gamma \phi_t - \lambda v_t, \tag{22}$$

where ℓ_t is task latency, ε_t is the energy consumed per task, $\phi_t \in \{0, 1\}$ indicates SLA satisfaction and v_t is the SLA violation penalty. The default weight vector is $(\alpha, \beta, \gamma, \lambda) = (0.4, 0.2, 0.3, 0.1)$. To ensure balanced optimization, each term is normalized before aggregation:

$$\tilde{\ell}_t = \frac{\ell_t}{\delta_t}, \qquad \tilde{\varepsilon}_t = \frac{\varepsilon_t}{E_{\max}}, \tag{23}$$

where δ_t is the task deadline and $E_{\max}$ is the maximum reference energy per task. The normalized reward becomes

$$r_t = -\alpha \tilde{\ell}_t - \beta \tilde{\varepsilon}_t + \gamma \phi_t - \lambda v_t. \tag{24}$$

This normalization stabilizes PPO training and prevents any single objective from dominating policy updates.

I. Computational Complexity Analysis

Let N denote the number of edge nodes and d the feature dimension of the joint task–node representation. At each decision step, the GRU forecasting stage incurs $O(Hd^2)$ complexity for horizon H, the attention encoding stage incurs $O(Nd)$ for score computation and context aggregation and the PPO policy evaluation incurs $O(d^2)$ for the forward pass through the actor–critic network. Therefore, the total per-step inference complexity is

$$O(Hd^2 + Nd + d^2), \tag{25}$$

which remains suitable for deployment on Jetson-class edge devices and lightweight supervisory nodes.

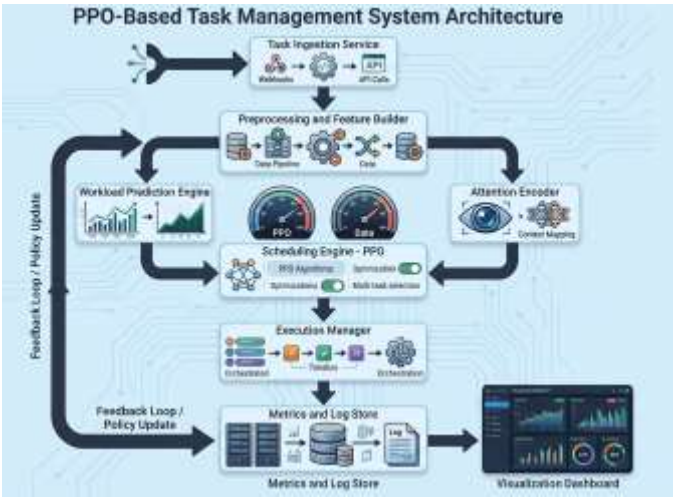

Fig. 4. Software architecture flowchart: task preprocessing → GRU forecasting → attention encoding → PPO scheduling → execution → metric logging and online policy update.

VII. Software Implementation

AERAS is implemented in Python using PyTorch for the GRU predictor, attention encoder and PPO agent, with FastAPI for the control dashboard and SimPy for discrete-event simulation. The software stack is modular, comprising task ingestion, node monitoring, prediction, encoding, scheduling, logging and visualization modules, enabling deployment from simulation to physical edge nodes.

A. Software Architecture

Fig. 4 summarizes the closed-loop software pipeline from task preprocessing to policy update.

B. Simulation Design

Three scenarios are evaluated: high-load ISR, drone swarm coordination and cyber-defence alert bursts. Each scenario is run at 50%, 80% and 100% load over 50 episodes with controlled random seeds and the recorded metrics are mean latency, P95 latency, throughput, energy per task, SLA satisfaction, queue length and reward variance.

VIII. End-to-End Workflow

Fig. 5 captures the complete operational loop. The controller ingests telemetry, runs the three-stage AI pipeline (GRU → attention → PPO), issues allocation decisions and updates the policy from observed execution outcomes. Treating scheduling as a recurring decision process rather than a one-time mapping enables AERAS to adapt progressively to evolving tactical conditions, a property essential in contested environments where system state changes rapidly.

IX. Results and Discussion

The proposed AERAS framework was evaluated across three tactical scenarios: high-load ISR, drone-swarm coordination and cyber-defence alert bursts. Each scenario was executed at 50%, 80% and 100% load over 50 episodes with controlled random seeds and the evaluation metrics included

TABLE I
COMPARATIVE ANALYSIS OF AERAS AGAINST REPRESENTATIVE SCHEDULING APPROACHES.

Dimension	**Heuristic scheduling**	**DRL (IoT/generic edge)**	**Recent PPO/offloading**	**AERAS (proposed)**
Decision principle	Predefined rules	Learned policy from state	Learned policy, richer control	**Proactive learned policy**
Workload anticipation	None	Usually reactive	Limited	**Explicit GRU forecasting**
State modelling	Shallow, manual	Task-resource vectors	Multi-factor, tier-aware	**Attention-weighted context**
Learning strategy	None	DQN / DDQN	PPO / policy-gradient	**PPO + GRU + attention**
Objective handling	Single / weak multi-obj.	Moderate	Often multi-objective	**Explicitly multi-objective**
Heterogeneity handling	Limited	Moderate	Moderate–strong	**Strong**
Implementation mapping	Easy, simplistic	Simulation-centric	Simulation-centric	**HW–SW–simulation integrated**
Tactical suitability	Low	Moderate	Moderate–high	**High**
Robustness under bursty load	Low	Moderate	Moderate–high	**High**

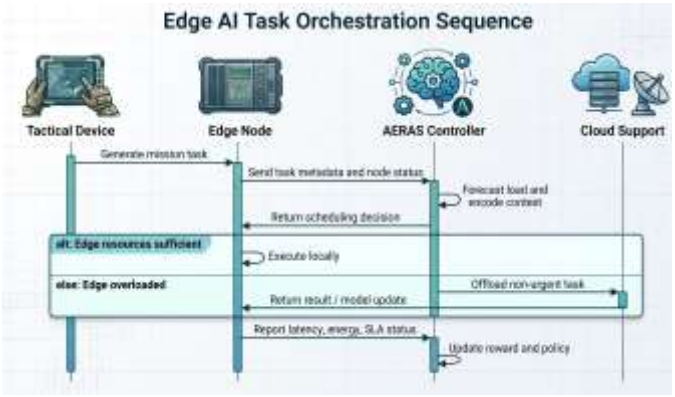

Fig. 5. End-to-end AERAS operational workflow: task generation → telemetry collection → GRU-attention-PPO scheduling → execution feedback → continuous policy improvement.

mean latency, P95 latency, energy per task, SLA satisfaction, queue length and reward variance.

The scheduling objective is defined as

$$J = \sum_{k} (\alpha \ell_k + \beta \epsilon_k - \gamma \phi_k + \lambda v_k), \tag{26}$$

where ℓ_k is task latency, ϵ_k is energy consumption, ϕ_k is the SLA satisfaction indicator, v_k is the SLA violation penalty and $\alpha, \beta, \gamma, \lambda$ are the trade-off weights. The reward used for PPO training is

$$r_t = -\alpha \ell_t - \beta \epsilon_t + \gamma \phi_t - \lambda v_t, \tag{27}$$

with normalized terms

$$\tilde{\ell}_t = \frac{\ell_t}{\delta_t}, \qquad \tilde{\epsilon}_t = \frac{\epsilon_t}{E_{\max}}, \tag{28}$$

and normalized reward

$$r_t = -\alpha \tilde{\ell}_t - \beta \tilde{\epsilon}_t + \gamma \phi_t - \lambda v_t. \tag{29}$$

The PPO update follows the clipped objective

$$L_{\text{CLIP}}(\vartheta) = \mathbb{E}_t \left[\min \left(r_t(\vartheta) \hat{A}_t, \; \text{clip}(r_t(\vartheta), 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right], \tag{30}$$

TABLE II
QUANTITATIVE RESULTS FOR THE HIGH-LOAD ISR SCENARIO

Method	Mean lat. (s)	P95 lat. (s)	Energy (J/task)	SLA (%)
Round Robin	1.20 ± 0.08	2.10 ± 0.15	8.0 ± 0.5	71.0 ± 3.2
First Fit	1.05 ± 0.07	1.90 ± 0.13	7.5 ± 0.5	75.5 ± 3.0
DQN	0.82 ± 0.06	1.25 ± 0.10	6.2 ± 0.4	83.0 ± 2.5
AERAS	0.62 ± 0.05	0.95 ± 0.08	4.8 ± 0.3	91.0 ± 1.8

TABLE III
QUANTITATIVE RESULTS FOR THE DRONE-SWARM SCENARIO

Method	Mean lat. (s)	P95 lat. (s)	Energy (J/task)	SLA (%)
Round Robin	0.95 ± 0.06	1.70 ± 0.12	7.2 ± 0.5	78.5 ± 2.8
First Fit	0.88 ± 0.06	1.60 ± 0.11	6.9 ± 0.4	80.2 ± 2.6
DQN	0.70 ± 0.05	1.05 ± 0.09	5.6 ± 0.3	87.0 ± 2.0
AERAS	0.50 ± 0.04	0.78 ± 0.06	4.1 ± 0.2	94.0 ± 1.5

where $r_t(\vartheta) = \pi_\vartheta(a_t|s_t)/\pi_{\vartheta_{\text{old}}}(a_t|s_t)$ and $\hat{A}_t$ is the generalized advantage estimate.

Table II summarizes the ISR scenario results. AERAS achieves a mean latency of 0.62 s, P95 latency of 0.95 s, energy consumption of 4.8 J/task and SLA satisfaction of 91.0%, outperforming Round Robin, First Fit and DQN. The improvement over DQN is particularly notable, with latency reduced from 0.82 s to 0.62 s and energy per task reduced from 6.2 J/task to 4.8 J/task. These gains indicate that forecasting and attention-based state encoding help the scheduler anticipate congestion and prioritize critical task–node interactions more effectively.

Table III shows the results for the drone-swarm scenario. AERAS achieves a mean latency of 0.50 s, P95 latency of 0.78 s, energy consumption of 4.1 J/task and SLA satisfaction of 94.0%, again outperforming all baselines. The improvement over DQN is consistent across all metrics, confirming that the proposed reward design maintains stable policy learning under tight latency constraints.

Table IV presents the cyber-defence alert burst results. AERAS attains 0.68 s mean latency, 1.05 s P95 latency, 5.1 J/task energy and 90.5% SLA satisfaction, indicating

TABLE IV
QUANTITATIVE RESULTS FOR THE CYBER-DEFENCE SCENARIO

Method	Mean lat. (s)	P95 lat. (s)	Energy (J/task)	SLA (%)
Round Robin	1.45 ± 0.10	2.50 ± 0.20	9.2 ± 0.6	65.0 ± 3.6
First Fit	1.30 ± 0.09	2.20 ± 0.18	8.6 ± 0.5	69.5 ± 3.4
DQN	0.95 ± 0.07	1.55 ± 0.12	6.8 ± 0.4	81.0 ± 2.7
AERAS	0.68 ± 0.05	1.05 ± 0.08	5.1 ± 0.3	90.5 ± 1.9

strong performance even under irregular high-priority workloads. Compared with the reactive baselines, the results show that the proposed closed-loop framework better preserves service quality during bursty tactical events.

AERAS differs from heuristic, DRL and PPO-based methods through proactive workload anticipation, attention-weighted state modeling and an explicit hardware–software–simulation pathway. The scalability and robustness results indicate stable latency and SLA compliance under higher load, node failures and link degradation, with graceful performance degradation relative to reactive baselines.

Overall, the findings confirm that GRU forecasting, attention-guided encoding and PPO control provide an effective scheduling strategy for mission-critical tactical edge computing. The framework is practical for defence-oriented edge intelligence, while physical deployment remains an important next step.

X. COMPARISON WITH EXISTING WORK

Table I positions AERAS relative to representative scheduling approaches. Heuristic schedulers are simple but perform poorly under variable load and heterogeneous node states [1]. DRL-based methods improve adaptability but remain largely reactive, while PPO-based offloading frameworks show stronger multi-objective control in non-stationary edge settings [5]–[7].

AERAS extends these directions by adding short-horizon forecasting, attention-guided state encoding and an implementation-oriented hardware–software pathway for tactical edge environments [8], [12]. This combination targets mission-oriented scheduling scenarios that are less represented in prior simulation-centric work.

XI. CONCLUSION

This paper presented AERAS, a tactical edge scheduling framework that combines GRU-based workload forecasting, attention-based state encoding and PPO-driven multi-objective control. The proposed method addresses the limitations of reactive scheduling under bursty workloads, heterogeneous edge nodes and residual-energy constraints.Simulation results across ISR, drone-swarm and cyber-defence scenarios show that AERAS consistently improves latency, energy efficiency and SLA satisfaction over Round Robin, First Fit and DQN baselines.

These results confirm that predictive and attention-guided reinforcement learning is well suited to mission-critical edge environments. The main limitation is that the current evaluation is simulation-based and does not yet include large-scale physical deployment.

Future work will extend the framework to federated tactical networks, adaptive reward tuning and validation on real edge hardware with live sensor streams.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things J.*, vol. 3, no. 5, pp. 637–646, 2016.
[2] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.
[3] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things J.*, vol. 5, no. 2, pp. 450–465, 2018.
[4] P. Wang *et al.*, "Joint task assignment, transmission and computing resource allocation in multilayer mobile edge computing systems," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 2872–2884, 2019.
[5] T. Zheng, J. Wan, J. Zhang *et al.*, "Deep reinforcement learning-based workload scheduling for edge computing," *J. Cloud Comput.*, vol. 11, no. 3, 2022.
[6] W. Zhang and H. Ou, "Reinforcement learning based multi objective task scheduling for energy efficient and cost effective cloud edge computing," *Scientific Reports*, vol. 15, art. 41716, 2025.
[7] J. Zhang *et al.*, "Edge computing and caching optimisation based on PPO," in *Proc. IEEE 9th World Forum IoT (WF-IoT)*, 2023.
[8] A. A. Ismail, N. E. Khalifa and R. A. El-Khoribi, "A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study," *Cluster Comput.*, vol. 28, art. 184, 2025.
[9] S. Chen, Q. Li, M. Zhou and A. Abusorrah, "Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm," *Sensors*, vol. 21, no. 3, art. 779, 2021.
[10] N. Gholipour, M. D. de Assuncao, P. Agarwal, J. Gascon-Samson and R. Buyya, "TPTO: A Transformer-PPO based task offloading solution for edge computing environments," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2023.
[11] Y. Tu, H. Chen, L. Yan and X. Zhou, "Task offloading based on LSTM prediction and deep reinforcement learning for efficient edge computing in IoT," *Future Internet*, vol. 14, no. 2, pp. 30–35, 2022.
[12] D. Roy *et al.*, "Next frontier of military communication: Harnessing edge/fog computing in integrated tactical networks," in *Proc. IEEE MILCOM*, 2024.
[13] H. Gao, W. Huang and T. Liu, "PPO2: Location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7599–7612, 2022.
[14] N. Sharma, A. Ghosh, R. Misra and S. K. Das, "Deep meta Q-learning based multi-task offloading in edge-cloud systems," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2583–2598, 2023.