Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Interpretable Digital Twin-Safe Scheduling for Deadline-Aware Task Allocation in Dynamic Fog-Edge Computing: A Comparative and Prediction-Validity Evaluation

Nagwa Elmobark

Nagwa.e@sust.ae

Smart University of Science and Technology

Nasser Tamim

Sinai University

Research Article

Keywords: Fog-edge computing, Digital Twin, Deadline-aware scheduling, Safety constraints, Task allocation, Deep reinforcement learning, DQN, DDQN, PPO, IoT

Posted Date: September 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10750114/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Interpretable Digital Twin-Safe Scheduling for Deadline-Aware Task Allocation in Dynamic Fog-Edge Computing: A Comparative and Prediction-Validity Evaluation

Nagwa Elmobark[1]*, Nasser Tamim[2]

1 lecturer, Smart University of Science and Technology, UAE,

Corresponding author: Nagwa.e@sust.ae, eng_nagwaelmobark@yahoo.com

2 Assistant Professor, Faculty of Information Technology, Sinai University

Nasser.tamim@su.edu.eg.

Abstract

Fog-edge computing enables low-latency execution for Internet of Things (IoT) applications by placing computation close to end devices. However, deadline-aware task allocation remains difficult in dynamic fog-edge systems because scheduling decisions must account for heterogeneous node capacity, queue accumulation, workload bursts, node failures and recovery, residual energy, and hard safety constraints. This paper presents a Digital Twin-safe scheduling framework for deadline-aware task allocation in a simulated dynamic fog-edge environment. The framework uses a lightweight analytical Digital Twin to pre-evaluate candidate fog nodes before execution. For each possible action, the Digital Twin estimates the expected response time, deadline-miss risk, post-action queue state, post-action energy state, overload risk, and post-action safety feasibility. A hard safety mask eliminates actions that violate node-availability, energy, or queue-safety constraints before the final scheduling decision is made. The framework is evaluated over 30 independent runs using a common protocol across representative heuristic, safety-aware heuristic, value-based deep reinforcement learning (DQN, DDQN, and Safe DDQN), policy-gradient reinforcement learning (PPO and Safe PPO), standalone Digital Twin, and Digital Twin-DRL fusion schedulers. The strongest numerical result is achieved by the hand-crafted Safe Deadline-Energy-Aware heuristic, with a deadline miss ratio of 2.27% and a mean reward of 4126.77. The Standalone Digital Twin scheduler is highly competitive with this strongest heuristic, achieving a deadline miss ratio of 2.42% and a mean reward of 4112.63, while also providing interpretable action-level predictive diagnostics. Compared with the value-based DRL baselines, the Standalone Digital Twin reduces deadline misses by approximately 88.97% relative to DQN and 87.17% relative to DDQN. The optimized DT-Safe fusion converges to a pure Digital Twin policy (q_weight = 0.0, dt_weight = 1.0), indicating that learned Q-values do not add useful decision guidance under the evaluated configuration. The Digital Twin predictor achieves 97.42% deadline prediction accuracy, a deadline Brier score of 0.0249, 100% overload prediction accuracy, and 100% post-action safety prediction accuracy. The results support the Digital Twin layer as an interpretable and competitive predictive decision-support mechanism for safe deadline-aware fog-edge scheduling.

Keywords: Fog-edge computing; Digital Twin; Deadline-aware scheduling; Safety constraints; Task allocation; Deep reinforcement learning; DQN; DDQN; PPO; IoT.

1. Introduction

Fog-edge computing extends cloud resources toward end devices through intermediate fog and edge nodes. This architecture can reduce round-trip latency and bandwidth pressure for latency-sensitive IoT applications; however, it also introduces a challenging resource-scheduling problem. A scheduler must assign incoming tasks to heterogeneous nodes while simultaneously considering response time, queue state, node availability, residual energy level, and application deadlines [1-5]. In dynamic fog-edge scenarios, this problem is further complicated by workload bursts, stochastic node failures, node recovery, and safety constraints, which can render some node assignments infeasible.

Classical heuristics such as shortest queue, lowest latency, and energy-aware scheduling are attractive because they are inexpensive and interpretable. However, their decision rules are usually fixed and may not capture the effect of an action on future queue evolution, deadline satisfaction, overload, and energy state. Deep reinforcement learning (DRL) has been explored as a data-driven alternative because it can learn policies through repeated interaction with an environment [7-11]. Nevertheless, learned value or policy estimates may face challenges under highly dynamic operating conditions, particularly when the decision is a discrete node-selection action with strict deadline and safety requirements. Digital Twin technology offers a complementary approach. Instead of learning a scheduling policy entirely through trial-and-error, a Digital Twin can maintain an analytical or virtual representation of the current system state and predict the likely consequences of assigning a task to a candidate node. In this work, the Digital Twin is not presented as a black-box replacement for all scheduling mechanisms. It is presented as an interpretable predictive layer that estimates response time, deadline feasibility indicators, post-action queue state, energy depletion, overload risk, and safety feasibility before the scheduling action is applied. The central research question is therefore: can an interpretable Digital Twin predictive layer support safe deadline-aware task allocation under dynamic fog-edge conditions, and how does its performance

compare with strong hand-crafted safe heuristics and DRL baselines? The answer must be stated carefully. The results do not show that the Digital Twin outperforms every heuristic. Instead, they show that the Standalone Digital Twin is competitive with the strongest hand-crafted safe heuristics, demonstrates improved performance compared with the evaluated DRL baselines, and provides action-level predictive validity analysis. The main contributions are as follows:

• A Digital Twin-safe scheduling framework is formulated for deadline-aware task allocation in a dynamic fog-edge environment with heterogeneous nodes, workload bursts, node failures and recovery, queue dynamics, residual energy, and hard safety constraints.

• A lightweight analytical Digital Twin predictor is formalized to estimate candidate-action consequences, including response time, deadline feasibility, post-action queue state, post-action energy state, overload risk, and post-action safety feasibility.

• A hard safety mask is integrated into the scheduling decision process to remove actions that violate node availability, minimum energy, or queue safety limits.

• A common evaluation protocol over 30 independent runs is used to compare simple heuristics, advanced safety-aware heuristics, DQN, DDQN, Safe DDQN, PPO, Safe PPO, Standalone Digital Twin scheduling, and Digital Twin-DRL fusion approaches.

• The results are interpreted without overclaiming: the Digital Twin is not the best numerical scheduler overall, but it is competitive with the strongest safe heuristic and provides stronger predictive interpretation and validity analysis compared with the evaluated DRL baselines.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the problem formulation, environment, Digital Twin model, safety mechanism, baselines, and evaluation protocol. Section 4 reports the comparative results and prediction-validity analysis. Section 5 discusses implications and limitations. Section 6 concludes the paper.

2. Related Work

2.1 Task Scheduling in Fog-Edge Computing

Task scheduling in fog-edge computing is commonly formulated as a multi-objective resource allocation problem that must balance latency, energy consumption, load distribution, deadline satisfaction, and resource availability [19-23]. Metaheuristic approaches and optimization-based methods can achieve strong solutions in structured environments; however, they may introduce considerable computational overhead and become less suitable for real-time scheduling under stochastic task arrivals, dynamic workloads, and node failures. Lightweight heuristics remain widely used because they are transparent, computationally efficient, and easy to implement. Nevertheless, their main limitation is that their decision rules are often local and reactive, relying primarily on current system conditions rather than predicting the future consequences of scheduling actions.

2.2 Deep Reinforcement Learning (DRL) for Fog Scheduling

DRL schedulers attempt to learn scheduling policies by interacting with a simulated or real environment. Value-based approaches such as DQN and DDQN estimate action values for candidate nodes and then select the highest-valued action [9-10], [24-26]. Actor-critic and policy-gradient methods such as PPO have also been applied to computation offloading and resource allocation problems [27- 28]. However, the effectiveness of PPO-based policy optimization depends strongly on factors such as reward scaling, rollout configuration, exploration strategy, action-space characteristics, and environment dynamics. In this paper, PPO and Safe PPO are evaluated as representative DRL baselines under a discrete node-selection configuration, rather than being considered generally unsuitable for fog-edge scheduling problems.

2.3 Safety-Aware Scheduling and Constrained Reinforcement Learning

Safety-aware scheduling is important because an assignment can be infeasible even when it appears latency-efficient. A node may be inactive, energy-depleted, or close to unsafe queue saturation. Safe reinforcement learning methods can incorporate constraints through reward penalties, constrained optimization, shielding, or action filtering [29-31]. This work adopts hard action masking because it is transparent, easy to verify, and computationally inexpensive: unsafe nodes are removed from the candidate action set before making the final decision. This design also enables separating the contribution of safety filtering from predictive, deadline-aware decision-making.

2.4 Digital Twin-Assisted Resource Management

Digital Twin models provide virtual representations of physical or cyber-physical systems for monitoring, prediction, simulation, and decision support [12-15], [34-37]. In edge and fog systems, a Digital Twin can support resource allocation by estimating the potential consequences of candidate decisions before they are applied. Prior studies have investigated Digital Twin-enabled collaborative scheduling, resource allocation, and computation offloading. The present study focuses on a specific gap: action-level prediction validity for safe deadline-aware task allocation. Therefore, the Digital Twin is evaluated not only through the scheduling performance it supports, but also through the accuracy and calibration of its predictions.

2.5 Evaluation Rigor

A persistent challenge in fog-edge scheduling research is the lack of unified evaluation protocols. Reported improvements may depend on different task distributions, baselines, random seeds, reward formulations, or evaluation metrics. Following the benchmarking motivation of the authors' previous work [16], this study adopts a common simulator configuration, repeated independent runs, a diverse baseline set, and multiple evaluation metrics, including response time, deadline miss ratio, deadline satisfaction ratio, safety violation ratio, overload ratio, energy behavior, active-node availability, and cumulative reward.

3. Methodology

The overall method used in this study is shown in Figure 1. The framework starts with the creation of a dynamic fog-edge environment and the task-generation model and then introduces the state representation, safety constraint, and hard action masking. It then assesses heuristic and deep reinforcement learning (DRL) scheduling baselines, the Standalone Digital Twin predictor, and the Digital Twin-guided Safe DDQN fusion policy. The learning-based schedulers are guided by a multi-objective reward function, and all methods are evaluated through a multi-run evaluation protocol using performance, safety, energy, and prediction-validity metrics.

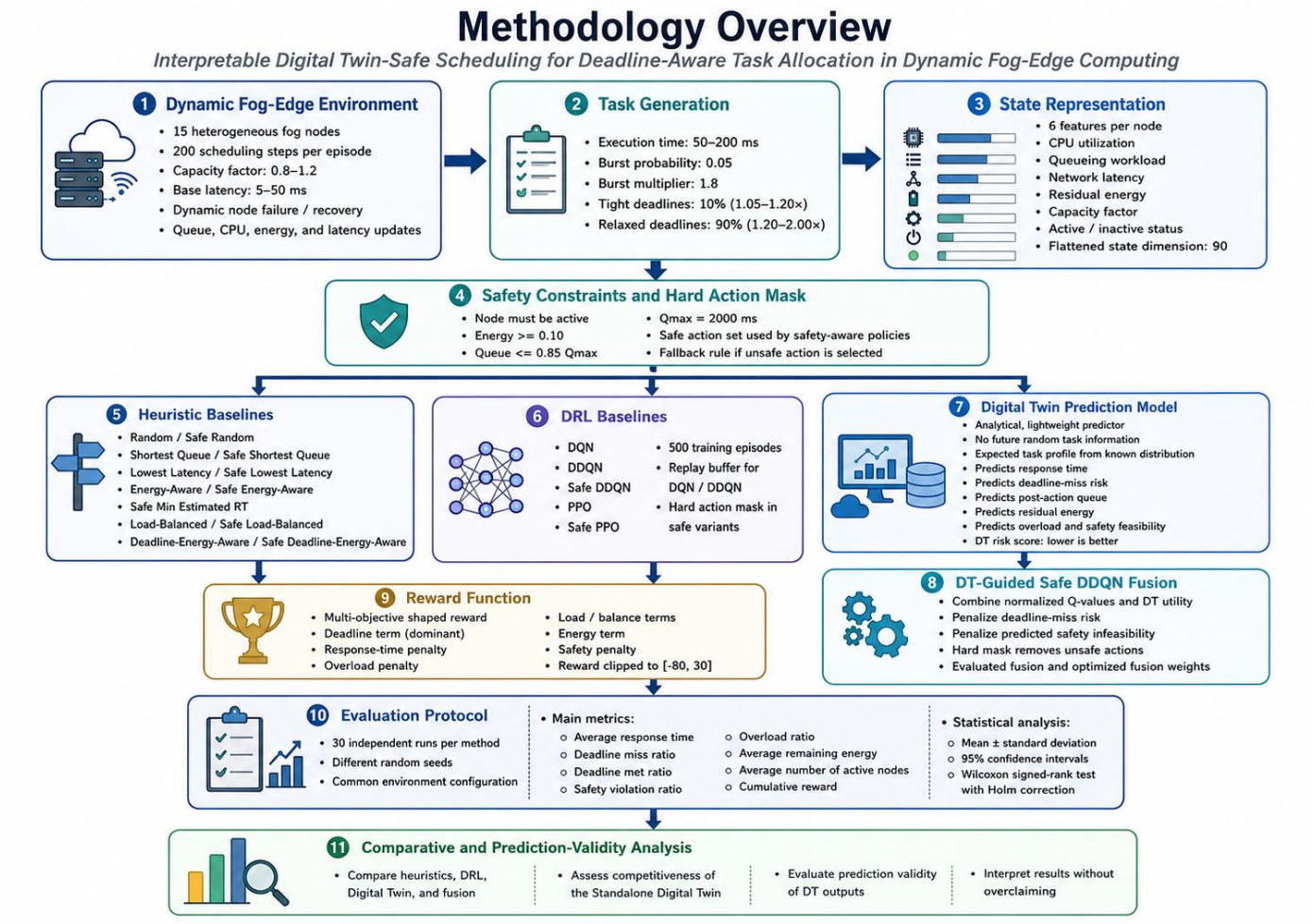


Figure 1. Overall methodology of the proposed Digital Twin-safe scheduling and comparative evaluation framework.

3.1 Problem Formulation

3.1 Problem Formulation

The scheduling problem is modeled as a sequential decision-making problem in a dynamic fog-edge environment. At each decision step t, a single incoming computational task must be assigned to one of N fog nodes. The nominal action space is discrete:

$$\mathcal{A} = \{1,2,\dots,N\}, \qquad N = 15 \tag{1}$$

Each action corresponds to assigning the incoming task to one fog node. The scheduler first selects a requested node at each decision step. The environment executes a different node only when the requested action violates the current safety constraints and the fallback mechanism is activated.

Each node i is represented by six normalized state features: CPU utilization, normalized queueing workload, normalized network latency, residual energy, normalized processing-capacity factor, and active/inactive status. The node-level state vector and the global state can be expressed as:

$$\mathbf{x}_{i,t} = [\text{cpu}_{i,t},\ \tilde{q}_{i,t},\ \tilde{l}_{i,t},\ e_{i,t},\ \tilde{c}_i,\ \text{active}_{i,t}], \qquad s_t = [\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \ldots, \mathbf{x}_{N,t}] \qquad (2)$$

The queueing workload is normalized by the 2000 ms maximum queue capacity, latency is normalized by 200 ms, and the capacity factor is mapped from its sampled range [0.8, 1.2] to [0, 1]. With N = 15 and six features per node, the flattened state dimension used by the DQN, DDQN, and PPO baselines is 90.

$$\tilde{q}_{i,t} = \text{clip}\left(\frac{q_{i,t}}{Q_{max}}, 0,1\right), \quad \tilde{l}_{i,t} = \text{clip}\left(\frac{l_{i,t}}{200}, 0,1\right), \quad \tilde{c}_i = \text{clip}\left(\frac{c_i - 0.8}{0.4}, 0,1\right)$$

The overall scheduling problem can be represented as a Markov Decision Process (MDP), where the elements represent the state space, discrete node-selection action space, stochastic transition mechanism, multi-objective reward function, and learning discount factor, respectively.

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$$

Each incoming task has an execution-time requirement and a deadline. The environment updates node availability, checks safety, applies fallback when required, executes the task, updates node states, and returns the reward and successor state.

The notation used throughout the scheduling formulation is summarized in Table 1.

Symbol	Meaning
N	Number of fog nodes (15)
$q_{i,t}$	Current queueing workload of node i at step t (ms)
$l_{i,t}$	Current network latency of node i at step t (ms)
$e_{i,t}$	Residual energy of node i, normalized to [0,1]
c_i	Processing-capacity factor of node i
$\text{active}_{i,t}$	Binary node-availability flag
τ_t	Execution-time requirement of the incoming task
d_t	Deadline of the incoming task
a_t	Requested scheduling action
a_t^{exec}	Action actually executed after safety/fallback checking
$RT_{i,t}$	Actual or predicted response time for node i
DMR	Deadline miss ratio
SVR	Safety-violation ratio

Table 1. Main notation used in the scheduling formulation.

3.2 Dynamic Fog-Edge Environment

The simulator contains 15 heterogeneous fog nodes. At reset, each node receives a capacity factor uniformly sampled from [0.8, 1.2] and a base latency uniformly sampled from 5 ms to 50 ms. Queueing

workload and CPU utilization are initialized to zero, residual energy is initialized to one, and all nodes start as active. Each episode contains 200 scheduling steps.

Task execution time is sampled from [50 ms, 200 ms]. Bursty traffic is represented by a burst probability of 0.05 and a burst multiplier of 1.8; therefore, when a burst occurs, the sampled execution time is multiplied by 1.8. Deadlines are heterogeneous: 10% of tasks are assigned tight deadlines using a multiplier sampled from [1.05, 1.20], while the remaining tasks use a relaxed multiplier sampled from [1.20, 2.00]. Node availability is dynamic: an active node fails with a probability of 0.01 per step, whereas an inactive node recovers with a probability of 0.05 per step. Recovery changes only the node-availability status and does not restore residual energy.

For the node that actually executes task t, the effective execution time is adjusted by the node processing-capacity factor:

$$\tau_{i,t}^{eff} = \frac{\tau_t}{\max(c_i, \varepsilon)} \tag{3}$$

In Eq. (3), the numerator is the original task execution-time requirement, the capacity factor adjusts execution speed, and a small positive constant prevents division by zero. Consequently, nodes with a capacity factor above one execute the task faster than the reference rate, whereas nodes with a capacity factor below one require a longer effective execution time.

The simulator first records the queueing workload of the executed node before adding the incoming task. After the task is enqueued, all active nodes process part of their queued workloads. The queue update can be written as:

$$q_{i,t}^{after} = \max\left(0,\, q_{i,t}^{before} + \mathbf{1}(a_t^{exec} = i)\tau_{i,t}^{eff} - \rho c_i \,\mathrm{active}_{i,t}\right) \tag{4}$$

where ρ = 80 is the base processing amount per decision step and 1(·) is an indicator function. The effective execution time is added only to the executed node, whereas each active node reduces its queue according to its processing capacity. The max operator prevents negative queueing workloads.

After the queue update, CPU utilization is derived from the queue ratio and network latency is adjusted deterministically according to congestion:

$$\mathrm{cpu}_{i,t}^{after} = \mathrm{clip}\left(\frac{q_{i,t}^{after}}{Q_{\max}}, 0, 1\right), \qquad l_{i,t}^{after} = l_i^{base}\left(1 + 1.5\frac{q_{i,t}^{after}}{Q_{\max}}\right)$$

The actual response time used by the simulator is therefore based on the pre-assignment queueing workload, the congestion-adjusted latency after the queue update, and the effective task execution time:

$$RT_{i,t} = q_{i,t}^{before} + l_{i,t}^{after} + \tau_{i,t}^{eff} \tag{5}$$

Thus, the response time includes the time spent waiting for previously queued work, the current communication delay after congestion adjustment, and the time required to execute the incoming task.

Residual energy decreases only for the node that executes the task. The implemented energy update is:

$$e_{i,t}^{after} = \mathrm{clip}\left[e_{i,t}^{before} - 0.0005\left(\frac{\tau_{i,t}^{eff}}{200}\right)\left(0.5 + \mathrm{cpu}_{i,t}^{after}\right),\ 0,\ 1\right] \tag{6}$$

The effective execution-time term is normalized relative to a reference duration of 200 ms, while the second multiplicative term combines a baseline energy cost with the effect of post-action CPU utilization. The coefficient 0.0005 is a simulation-specific energy-depletion parameter, and the clip operator constrains normalized residual energy to the interval [0, 1].

A deadline miss is recorded whenever the realized response time exceeds the generated task deadline. An overload event is recorded when the post-action queue of the executed node reaches or exceeds the 1500 ms overload threshold.

$$DM_t = \mathbf{1}(RT_t > d_t), \qquad O_t = \mathbf{1}\left(q_{a_t^{exec},t}^{after} \geq Q_{\mathrm{overload}}\right)$$

3.3 Safety Constraints and Hard Action Mask

A node i is considered safe at decision step t only if it satisfies three conditions: it is operational, its residual energy is at least 0.10, and its queueing workload does not exceed 85% of the maximum queue capacity. With a maximum queue capacity of 2000 ms, the corresponding safe queueing-workload limit is 1700 ms. The hard safety mask is defined as:

$$M_i(s_t) = \mathbf{1}\left[\mathrm{active}_{i,t} = 1 \ \wedge\ e_{i,t} \geq 0.10 \ \wedge\ q_{i,t} \leq 0.85 Q_{\max}\right] \tag{7}$$

Safety-aware scheduling methods select only among nodes that satisfy the safety condition in Eq. (7). When fallback is required and at least one safe node is available, the safe node with the minimum current queueing workload is selected. If no fully safe node exists, the active node with the minimum queueing workload is selected. If all nodes are inactive, the node with the globally minimum queueing workload is used as a worst-case fallback.

$$a_t^{fallback} = \begin{cases} \arg\min\limits_{i \in \mathcal{A}_t^{safe}} q_{i,t}, & \mathcal{A}_t^{safe} \neq \varnothing, \\ \arg\min\limits_{i:\mathrm{active}_{i,t}=1} q_{i,t}, & \mathcal{A}_t^{safe} = \varnothing \text{ and an active node exists}, \\ \arg\min\limits_{i \in \mathcal{A}} q_{i,t}, & \text{otherwise.} \end{cases}$$

For safe learning policies, action masking occurs during policy-level action selection. However, the environment updates node failure and recovery states at the beginning of the subsequent environment step

before re-evaluating whether the requested action is safe. Consequently, an action that was safe during policy selection can become unsafe immediately before execution if the selected node fails. The safety-violation indicator is recorded for the originally requested action, even if the environment then redirects execution using the fallback mechanism. This ordering explains why a small nonzero safety-violation ratio may still be observed under safety-aware policies.

3.4 Reward Function

The reward function is a shaped multi-objective signal used for learning and as a supplementary performance indicator. It rewards deadline satisfaction and higher residual energy while penalizing deadline misses, response time, overload, selection of heavily loaded nodes, load imbalance, and unsafe actions. The implemented reward is summarized as:

$$R_t = R_{\text{deadline}} + R_{\text{rt}} + R_{\text{overload}} + R_{\text{load}} + R_{\text{balance}} + R_{\text{energy}} + R_{\text{safety}} \qquad (8)$$

The deadline component is the dominant term. The deadline margin is defined as the task deadline minus the realized response time. If this margin is nonnegative, the scheduler receives a positive base reward with a bounded bonus proportional to the normalized margin; otherwise, it receives a penalty that grows with the relative deadline-miss magnitude:

$$R_{\text{deadline}} = \begin{cases} 15\left[1 + \min\left(\frac{m_t}{\max(d_t, 10^{-6})}, 0.5\right)\right], & m_t \geq 0, \\ -15\left[1 + \frac{|m_t|}{\max(d_t, 10^{-6})}\right], & m_t < 0. \end{cases}$$

The remaining reward components use the same quantities calculated by the environment. Response time is normalized by 500 ms, overload is penalized in proportion to the post-action queue ratio, and selection of an already loaded node is penalized quadratically. A load-balancing bonus depends on the standard deviation of queueing workloads across all nodes. Higher residual energy receives a small positive contribution, while an explicit penalty of 20 units is applied whenever the requested action violates the safety mask.

$$R_{\text{rt}} = -\min(RT_t/500, 1), \qquad R_{\text{load}} = -3\left(q_t^{before}/Q_{\max}\right)^2$$

$$R_{\text{balance}} = 1.5(1 - \sigma(q_t)/Q_{\max}), \qquad R_{\text{energy}} = 0.5e_t^{after}, \qquad R_{\text{safety}} = -20\,SV_t$$

The final reward is clipped to the interval [−80, 30] to limit extreme values and improve DRL training stability.

$$R_t^{final} = \text{clip}(R_t, -80, 30)$$

3.5 Analytical Digital Twin Predictive Layer

The Digital Twin is a lightweight analytical predictor. It does not execute env.step() while scoring candidate actions and therefore does not access the actual future randomly generated task. Instead, it estimates an expected task profile from the known task-generation distribution and evaluates all candidate actions before the selected action is applied.

The expected task execution requirement incorporates both the mean non-burst execution time and the probability of workload bursts:

$$E[\tau] = \left(\frac{\tau_{\min}+\tau_{\max}}{2}\right)[(1 - p_{\text{burst}}) + p_{\text{burst}}\beta_{\text{burst}}] \tag{9}$$

Using the implemented values of 50 ms and 200 ms for the execution-time limits, a burst probability of 0.05, and a burst multiplier of 1.8, the expected task requirement in Eq. (9) is used to construct the Digital Twin predictions. The expected tight and relaxed deadlines are obtained using the mean multipliers of [1.05, 1.20] and [1.20, 2.00], respectively.

For each candidate node i, the Digital Twin estimates a capacity-adjusted effective execution time and a response time using the current queueing workload and current latency:

$$E\left[\tau_{i,t}^{eff}\right] = \frac{E[\tau]}{\max(c_i,\varepsilon)}, \qquad RT_{i,t}^{DT} = q_{i,t} + l_{i,t} + E\left[\tau_{i,t}^{eff}\right] \tag{10}$$

The predicted deadline-miss risk is modeled as a distribution-weighted mixture of two binary miss indicators corresponding to the expected tight and relaxed deadlines:

$$P_{\text{miss}}^{DT}(i) = p_{\text{tight}}\mathbf{1}\left(RT_{i,t}^{DT} > d_{\text{tight}}^{E}\right) + \left(1 - p_{\text{tight}}\right)\mathbf{1}\left(RT_{i,t}^{DT} > d_{\text{relaxed}}^{E}\right) \tag{11}$$

The Digital Twin also estimates the post-action queueing workload. In the implemented predictor, the processing term is applied directly during candidate-action scoring and is not multiplied by the current active-status flag:

$$q_{i,t}^{DT} = \max\left(0,\, q_{i,t} + E\left[\tau_{i,t}^{eff}\right] - \rho c_i\right) \tag{12}$$

Predicted CPU utilization is derived from the predicted queueing workload using the same queue-capacity normalization as the simulator. The predicted residual-energy update uses the expected task requirement from Eq. (9), rather than the capacity-adjusted expected effective execution time from Eq. (10):

$$e_{i,t}^{DT} = \text{clip}\left[e_{i,t} - 0.0005\left(\frac{E[\tau]}{200}\right)\left(0.5 + \text{cpu}_{i,t}^{DT}\right), 0,1\right] \tag{13}$$

$$\text{cpu}_{i,t}^{DT} = \text{clip}\left(\frac{q_{i,t}^{DT}}{Q_{\max}}, 0,1\right)$$

The predicted overload indicator is:

$$O_{i,t}^{DT} = \mathbf{1}\left(q_{i,t}^{DT} \geq Q_{\text{overload}}\right) \quad (14)$$

The predicted post-action safety indicator combines current node availability with the predicted energy and queueing-workload states:

$$S_{i,t}^{DT} = \mathbf{1}\left[\text{active}_{i,t} = 1 \ \wedge \ e_{i,t}^{DT} \geq E_{\min} \ \wedge \ q_{i,t}^{DT} \leq 0.85 Q_{\max}\right] \quad (15)$$

For candidate ranking, the Digital Twin normalizes predicted response time by 500 ms and predicted queueing workload by the maximum queue capacity. Its final risk score assigns the greatest weight to deadline-miss risk, followed by response time and queueing workload, while safety infeasibility, overload, and energy depletion receive smaller penalties. For compact notation in Eq. (16), R_i and Q_i denote the normalized response-time and queue terms, D_i the predicted deadline-miss risk, S_i the predicted safety indicator, O_i the predicted overload indicator, and E_i the predicted residual energy.

$$R_i = RT_{\text{norm},i}, \quad Q_i = Q_{\text{norm},i}, \quad D_i = P_{\text{miss}}^{DT}(i)$$

$$S_i = S_{i,t}^{DT}, \qquad O_i = O_{i,t}^{DT}, \qquad E_i = e_{i,t}^{DT}$$

Using this notation, lower Digital Twin scores indicate more desirable candidate actions:

$$Score_{i,t}^{DT} = 0.30R_i + 0.35D_i + 0.15Q_i + 0.10(1 - S_i) + 0.05O_i + 0.05(1 - E_i) \quad (16)$$

$$RT_{\text{norm},i} = \text{clip}\left(\frac{RT_{i,t}^{DT}}{500}, 0, 3\right), \qquad Q_{\text{norm},i} = \text{clip}\left(\frac{q_{i,t}^{DT}}{Q_{\max}}, 0, 1\right)$$

The Standalone Digital Twin scheduler first searches among actions that are currently safe and predicted to remain safe after execution. If such candidates exist, it selects the candidate with the lowest Digital Twin score. If none satisfy both conditions, the search is relaxed to currently safe candidates. If no currently safe candidate exists, all actions remain eligible for Digital Twin ranking; if the selected action is unsafe when it reaches the real environment, the environment-level fallback mechanism is applied before execution.

3.6 Digital Twin-Guided Safe DDQN Fusion

The fusion policy combines the Q-value signal of a trained Safe DDQN agent with the Digital Twin utility. Q-values are min-max normalized so that higher values indicate more desirable candidate actions. Digital Twin scores are also min-max normalized and inverted so that lower predicted risk produces higher utility.

$$Q_{\text{norm},i} = \frac{Q_i - Q_{\min}}{Q_{\max} - Q_{\min}}, \qquad DTU_i = 1 - \frac{Score_i^{DT} - Score_{\min}^{DT}}{Score_{\max}^{DT} - Score_{\min}^{DT}}$$

The fusion utility is defined as:

$$U_i = w_q Q_{\text{norm},i} + w_{DT} DTU_i - \lambda_d P_{\text{miss}}^{DT}(i) - \lambda_s (1 - S_i^{DT}) \quad (17)$$

The two main weights control the relative contributions of the Safe DDQN Q-value signal and Digital Twin utility, while the two penalty coefficients control additional deadline-risk and safety-risk penalties. Higher fusion-utility values indicate better candidate actions.

A strict hard mask retains only actions that are currently safe and also predicted to remain safe after assignment. If no action satisfies this strict condition, the policy relaxes the mask to nodes that are currently safe. If no currently safe node exists, the environment fallback action is used. The evaluated DT-guided configuration assigns a weight of 0.40 to the Safe DDQN signal and 0.60 to Digital Twin utility; both additional risk-penalty coefficients are set to 0.20.

A fusion-weight sensitivity analysis was performed by varying the Q-value weight while setting the Digital Twin weight to its complement. The best tested setting assigned zero weight to the Q-value signal and unit weight to Digital Twin utility. This removes the learned Safe DDQN Q-value contribution and makes the fusion utility Digital Twin-driven. Nevertheless, this optimized fusion should not be described as mathematically identical to the Standalone Digital Twin scheduler because the fusion formulation still contains the additional deadline-risk and safety-risk penalties and uses its own strict masking and fallback sequence.

3.7 Baselines and Learning Configuration

The baseline set includes Random, Shortest Queue, Lowest Latency, Energy-Aware scheduling, and safety-aware variants of these policies. Stronger hand-crafted heuristics are also included, namely Minimum Estimated Response Time, Load-Balanced scheduling, Deadline-Energy-Aware scheduling, and their safe variants. These advanced heuristics provide more rigorous non-learning baselines than simple queueing or latency rules.

The Minimum Estimated Response Time heuristic estimates task execution using the midpoint of the base execution-time range and combines this estimate with current queueing workload, current latency, and node capacity. The Load-Balanced heuristic uses a manually weighted score consisting of normalized queueing workload (0.50), CPU utilization (0.30), and normalized latency (0.20). The Deadline-Energy-Aware heuristic combines estimated response time, queue load, residual energy, and, for the non-safe variant, an inactive-node penalty. The safe versions restrict selection to nodes satisfying the current safety mask.

$$E[\tau]_{heuristic} = (\tau_{\min} + \tau_{\max})/2$$

The value-based DRL baselines are DQN, DDQN, and Safe DDQN. Their Q-network receives the 90-dimensional flattened state and outputs 15 action Q-values. The implemented architecture contains three hidden linear layers with 128, 128, and 64 units, respectively. Layer normalization is applied after each of

the first two hidden linear layers, followed by ReLU activation; the 64-unit third hidden layer also uses ReLU. The final output layer is linear and contains 15 units, with no activation applied to the Q-values:

$$90 \rightarrow 128 \rightarrow 128 \rightarrow 64 \rightarrow 15$$

DQN/DDQN training uses 500 episodes with 200 steps per episode, a replay-buffer capacity of 100,000 transitions, a mini-batch size of 64, a discount factor $\gamma = 0.99$, a learning rate of 3×10^{-4}, and weight decay of 1×10^{-5}. Training starts after the buffer contains 500 transitions. The target network is updated every five episodes. Epsilon-greedy exploration starts at 1.0, is multiplied by 0.995 after each episode, and is bounded below by 0.05. Gradient clipping uses a maximum norm of 10. DDQN uses the policy network to select the next action and the target network to evaluate it, whereas standard DQN uses the target network for both selection and evaluation of the maximum next-state value.

PPO and Safe PPO use a discrete actor-critic architecture with 500 episodes and 200 steps per episode. Their configuration uses $\gamma = 0.99$, GAE $\lambda = 0.95$, clipping parameter $\varepsilon = 0.20$, eight PPO update epochs, mini-batches of 64 samples, entropy coefficient 0.01, value-loss coefficient 0.50, learning rate 3×10^{-4}, weight decay 1×10^{-5}, and gradient clipping with maximum norm 10. The shared actor-critic representation contains two 128-unit layers with LayerNorm and ReLU activations; separate actor and critic heads then use 64-unit hidden layers. Safe PPO applies the hard action mask to the actor logits before sampling or greedy action selection.

3.8 Evaluation Protocol

The final unified comparison evaluates each selected scheduling method over 30 runs under the same environment configuration and the same evaluation-seed sequence. Each run contains 200 scheduling steps. The reported metrics include average response time, deadline-miss ratio, deadline-met ratio, safety-violation ratio, overload ratio, average remaining energy, average number of active nodes, and cumulative reward. The main comparison table reports the mean and standard deviation of the corresponding per-run metrics.

For a run containing T scheduling steps, the deadline-miss ratio, deadline-met ratio, safety-violation ratio, and overload ratio are calculated as:

$$DMR = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\,(RT_t > d_t), \qquad DMR_{met} = 1 - DMR$$

$$SVR = \frac{1}{T}\sum_{t=1}^{T} SV_t, \qquad OR = \frac{1}{T}\sum_{t=1}^{T} O_t$$

Cumulative reward is the sum of the step-level rewards obtained during the 200-step run. Average response time and the remaining resource metrics are computed as arithmetic means over the corresponding steps within each run and are then summarized across the 30 evaluation runs.

For the implemented paired inferential analysis, the optimized DT-Safe Fusion configuration that assigns zero weight to the Q-value signal and unit weight to Digital Twin utility is treated as the proposed reference method. It is compared with DQN, DDQN, Safe DDQN, Shortest Queue, Safe Shortest Queue, Safe Minimum Estimated Response Time, Safe Deadline-Energy-Aware scheduling, the Standalone Digital Twin, and the DT-Guided Safe DDQN configuration using 0.40 and 0.60 weights. Runs are matched by run number before paired testing.

Four outcomes are included in the implemented paired statistical analysis: average response time, deadline-miss ratio, safety-violation ratio, and cumulative reward. The normality of the paired differences is assessed using the Shapiro-Wilk test. Both a paired-samples t-test and the Wilcoxon signed-rank test are calculated for each paired comparison. Cohen's paired standardized effect size is also reported, and statistical significance is assessed at $\alpha = 0.05$.

$$d_z = \frac{\overline{D}}{s_D}, \qquad D_r = X_r^{proposed} - X_r^{baseline}$$

The Digital Twin predictive layer is also evaluated directly by comparing predicted outcomes with the corresponding outcomes observed after real environment execution. Continuous prediction errors for response time, post-action queueing workload, and residual energy are summarized using mean absolute error (MAE), root mean squared error (RMSE), and prediction bias. For a continuous variable y, these measures are:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right|, \qquad RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}, \qquad Bias = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)$$

For deadline-risk prediction, the Digital Twin probability is converted to a binary deadline-miss label using a threshold of 0.5. Deadline prediction accuracy and Brier score are then calculated. Accuracy is also reported for overload prediction and post-action safety feasibility. These prediction-validity measures are used to determine whether the Digital Twin provides reliable action-level predictive information in addition to its scheduling performance.

$$BS = \frac{1}{n}\sum_{j=1}^{n}\left(P_{miss,j}^{DT} - DM_j\right)^2$$

Claims of superiority are restricted to the observed metrics and statistical comparisons. In particular, the final interpretation recognizes that the strongest advanced safe heuristic may remain marginally superior on some numerical outcomes, while the Digital Twin-based methods provide competitive scheduling performance together with an explicit predictive layer for response time, deadline risk, overload, energy, and safety feasibility.

4. Results

4.1 Overall Performance Comparison

Table 2 summarizes the final unified comparison across eleven scheduling methods. The Safe Deadline-Energy-Aware heuristic achieves the best numerical performance, with the lowest deadline-miss ratio (0.0227), the highest deadline-met ratio (0.9773), and the highest mean cumulative reward (4126.77). The Standalone Digital Twin ranks third numerically, with a deadline-miss ratio of 0.0242 and a mean cumulative reward of 4112.63. Although the Safe Deadline-Energy-Aware heuristic remains the numerical leader, the difference in deadline-miss ratio between this method and the Standalone Digital Twin is not statistically significant after Holm correction (adjusted $p = 0.0520$). Small statistically significant paired differences are observed in average response time and cumulative reward; however, their absolute magnitudes are limited to approximately 0.05 ms and 14.14 reward units, respectively. No statistically significant differences are found between the Standalone Digital Twin and Safe Min Estimated RT for average response time, deadline-miss ratio, or cumulative reward. Therefore, the appropriate interpretation is that the Standalone Digital Twin is statistically competitive with the strongest safe heuristics, rather than that it outperforms all heuristic methods.

Table 2. Final unified performance comparison across eleven scheduling methods. Values are means across 30 independent runs.

Rank	Method group	Policy	Avg RT (ms)	DMR	Met ratio	Safety viol.	Overload	Reward
1	Advanced safe heuristic	Safe Deadline-Energy-Aware	132.04	0.0227	0.9773	0.0088	0.0000	4126.77
2	Advanced safe heuristic	Safe Min Estimated RT	132.10	0.0232	0.9768	0.0100	0.0000	4118.15
3	Digital Twin	Standalone Digital Twin	132.09	0.0242	0.9758	0.0095	0.0000	4112.63
4	Optimized fusion	Optimized DT-Safe Fusion (q=0.0, dt=1.0)	132.09	0.0242	0.9758	0.0095	0.0000	4112.63
5	Optimized Digital Twin	Optimized Deadline-Energy DT Safe Scheduler	132.93	0.0283	0.9717	0.0090	0.0000	4075.25
6	Hybrid fusion	DT-Guided Safe DDQN (q=0.4, dt=0.6)	137.63	0.0500	0.9500	0.0093	0.0000	3863.39
7	Value-based DRL	DDQN	159.37	0.1887	0.8113	0.0110	0.0000	2638.14
8	Value-based DRL	DQN	161.68	0.2193	0.7807	0.0128	0.0000	2401.77
9	Safe value-based DRL	Safe DDQN	162.87	0.2195	0.7805	0.0112	0.0000	2381.11
10	Policy-gradient DRL	PPO	935.90	0.6182	0.3818	0.4472	0.3890	-8539.52
11	Safe policy-gradient DRL	Safe PPO	1077.96	0.8325	0.1675	0.0123	0.3907	-10176.90

The Digital Twin methods clearly outperform the value-based DRL baselines in this evaluated setting. Compared with DQN and DDQN, the Standalone Digital Twin reduces the deadline miss ratio from 0.2193 and 0.1887 to 0.0242, corresponding to relative reductions of approximately 88.97% and 87.17%, respectively. The DT-guided Safe DDQN fusion with q_weight = 0.4 and dt_weight = 0.6 improves over the value-based DRL baselines but remains weaker than the pure Digital Twin and the strongest safe heuristics. This result indicates that partial fusion with learned Q-values did not provide an additional benefit under the evaluated dynamic configuration.

Policy-gradient methods form a separate low-performing group. PPO achieves a deadline miss ratio of 0.6182 and Safe PPO reaches 0.8325. These results should not be generalized to all PPO-based resource allocation problems; they indicate that, under this discrete node-selection configuration and reward design, the evaluated PPO agents did not learn a reliable scheduling policy.

The performance of the scheduling methods in terms of the ratio of deadlines missed is shown in Figure 2. The figure shows visually the main performance difference that has been reported in Table 2, with lower values representing increased deadline compliance.

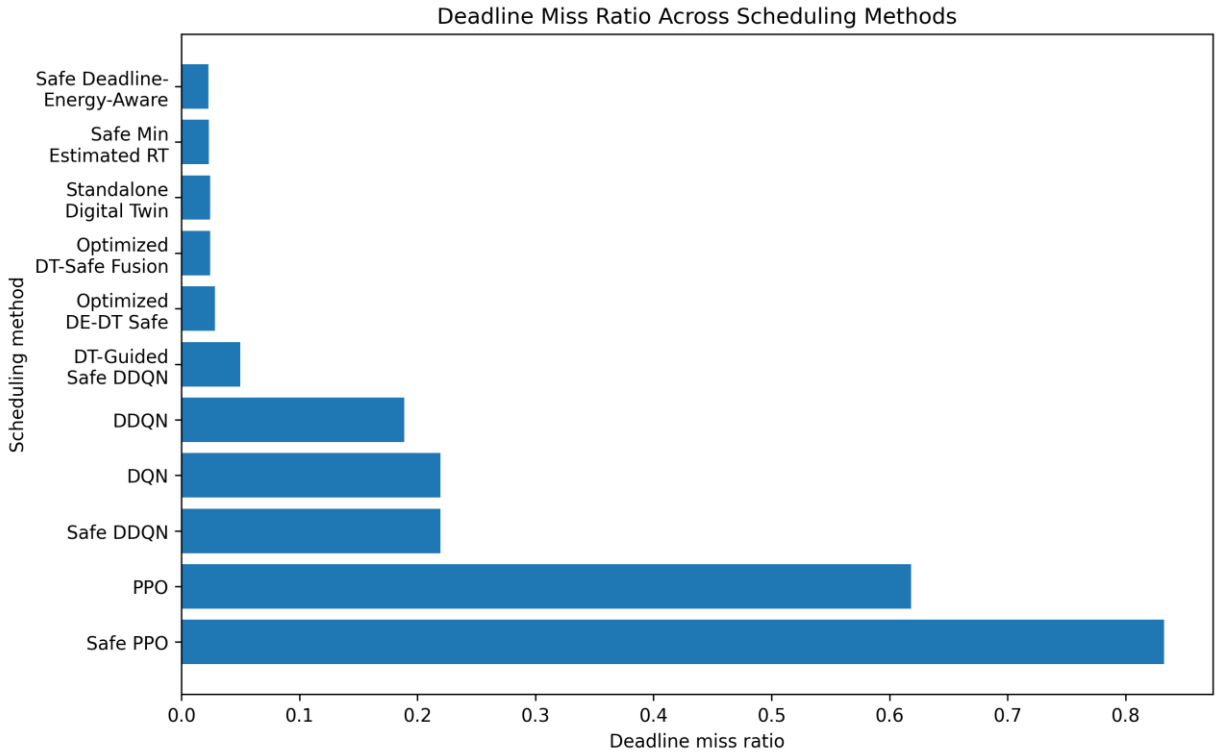


Figure 2. Ranked deadline miss ratio across scheduling methods.

The three top performing methods all perform very similar, as demonstrated in Figure 2. Safe Deadline-Energy-Aware, Safe Min Estimated RT, and the Standalone Digital Twin are tightly clustered and have very low deadline miss ratios, ranging from 0.0227 to 0.0242. With the optimized fusion parameters set to q = 0.0 and dt = 1.0, the numerical results from both the optimized and Standalone Digital Twin match. The DT-guided Safe DDQN fusion is positioned in between the value-based DRL baselines and the best safe heuristics, achieving significantly better results than the value-based DRL baselines. PPO and Safe PPO are evident high miss outliers in the evaluated configuration

4.2 Response-Time Analysis

The best-performing heuristic and Digital Twin-based schedulers have tightly clustered response-time values of approximately 132–138 ms. Safe Deadline-Energy-Aware achieves the lowest numerical average response time at 132.04 ms, followed closely by the Standalone Digital Twin and the optimized DT-Safe fusion at 132.09 ms. The value-based DRL methods are slower, with response times ranging from [159.37: 162.87] ms. PPO and Safe PPO have substantially larger response times of 935.90 ms and 1077.96 ms, respectively, consistent with their high deadline-miss ratios and overload behavior. Figure 3 shows the average response time across all scheduling methods and highlights the differences among the best heuristic and digital twin-based schedulers, the value-based DRL methods, and the policy-gradient baselines.

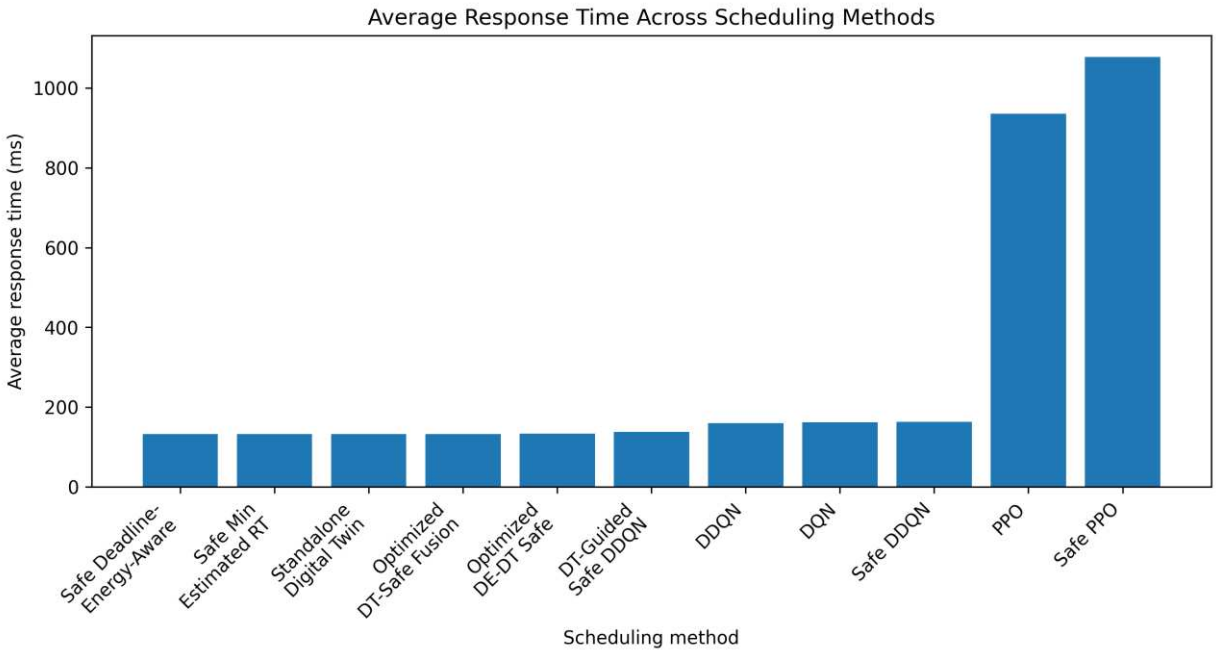


Figure 3. Average response time across scheduling methods.

As seen in Figure 3, the best performing heuristics and schedulers based on the Digital Twin have very similar numerical response times, with an average of around 132–138 ms. The Standalone Digital Twin and optimized DT-Safe fusion have almost the same numerical response time. Value-based DRL methods are grouped, with somewhat longer response times, while PPO and Safe PPO are "high latency" outliers. This is similar to the higher deadline miss and overload ratio observed with them.

4.3 Safety Violation and Overload Analysis

Most heuristic, value-based, and Digital Twin-based schedulers maintain low safety-violation ratios and zero overload. PPO is the main exception, with a safety-violation ratio of 0.4472 and an overload ratio of 0.3890. Safe PPO reduces the direct safety-violation ratio to 0.0123 through hard action masking, but its overload ratio remains high at 0.3907. These descriptive results indicate that hard safety masking alone is insufficient when the learned policy does not make effective deadline- and congestion-aware allocation decisions.

Figure 4 compares the safety-violation and overload ratios across the evaluated scheduling methods and highlights the different effects of hard action masking on direct safety violations and congestion-related outcomes.

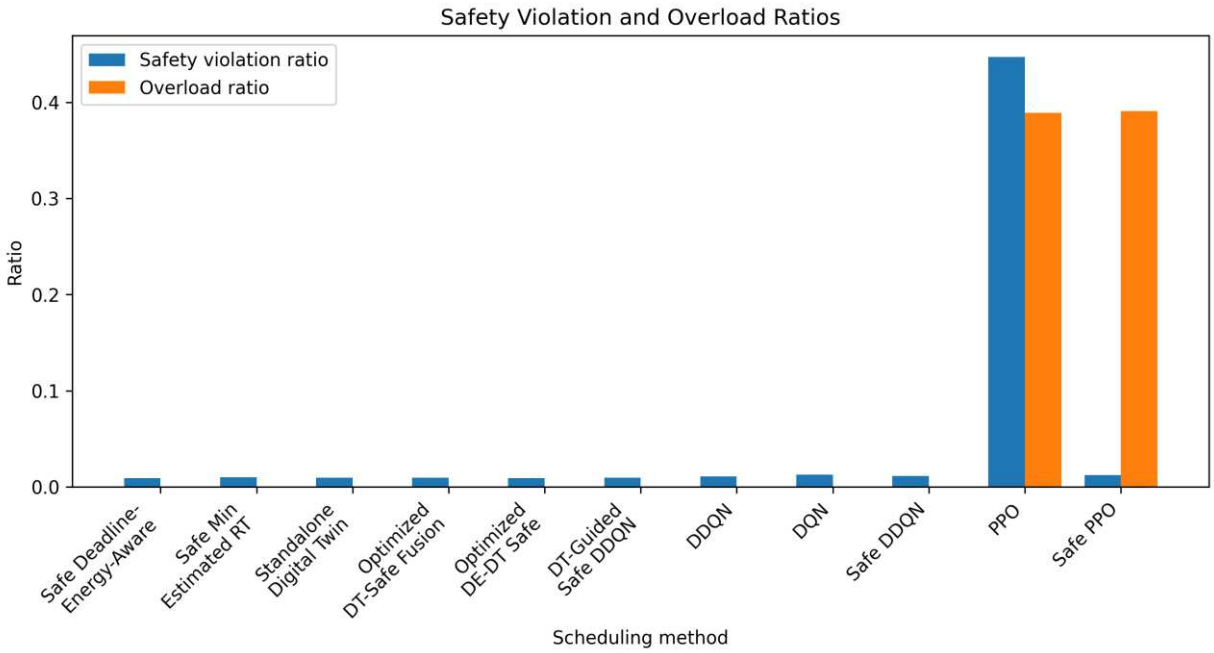


Figure 4. Safety violation and overload ratios across scheduling methods.

Figure 4 shows that most heuristic, value-based, and Digital Twin-based schedulers remain clustered near zero for both safety violations and overload. PPO is a clear outlier on both measures. Safe PPO substantially reduces direct safety violations because unsafe actions are masked during action selection; however, its overload ratio remains high. This visual pattern indicates that hard action masking can prevent immediate unsafe node selections, but it does not by itself guarantee effective queue management or congestion-aware scheduling.

Among the policies included in the paired inferential analysis, the Standalone Digital Twin does not differ significantly in safety-violation ratio from Safe Deadline-Energy-Aware, Safe Min Estimated RT, DDQN, DQN, Safe DDQN, Safe Shortest Queue, or the DT-Guided Safe DDQN fusion after Holm correction. Significant safety improvements are observed only relative to the non-safe Random and Shortest Queue policies. No inferential overload differences are identified among these eleven policies because all record an overload ratio of zero across the paired evaluation runs. PPO and Safe PPO are not included in these inferential comparisons because complete paired per-run observations were not available for these policies.

4.4 Reward Analysis

Reward follows the same overall pattern as deadline and response-time performance. Safe Deadline-Energy-Aware obtains the highest numerical mean reward at 4126.77. The Standalone Digital Twin and optimized DT-Safe fusion both obtain 4112.63, indicating that the optimized fusion produced the same reported mean reward as the Standalone Digital Twin under the evaluated configuration. The value-based DRL baselines receive moderate positive reward values ranging from 2381.11 to 2638.14, whereas PPO and Safe PPO receive large negative rewards because of their higher deadline-miss ratios, overload levels, and response times.

The mean cumulative reward results of all scheduling methods are given in figure 5, and the consistency between the reward results and the results from the deadline-miss results, the response-time results and the overload results is shown.

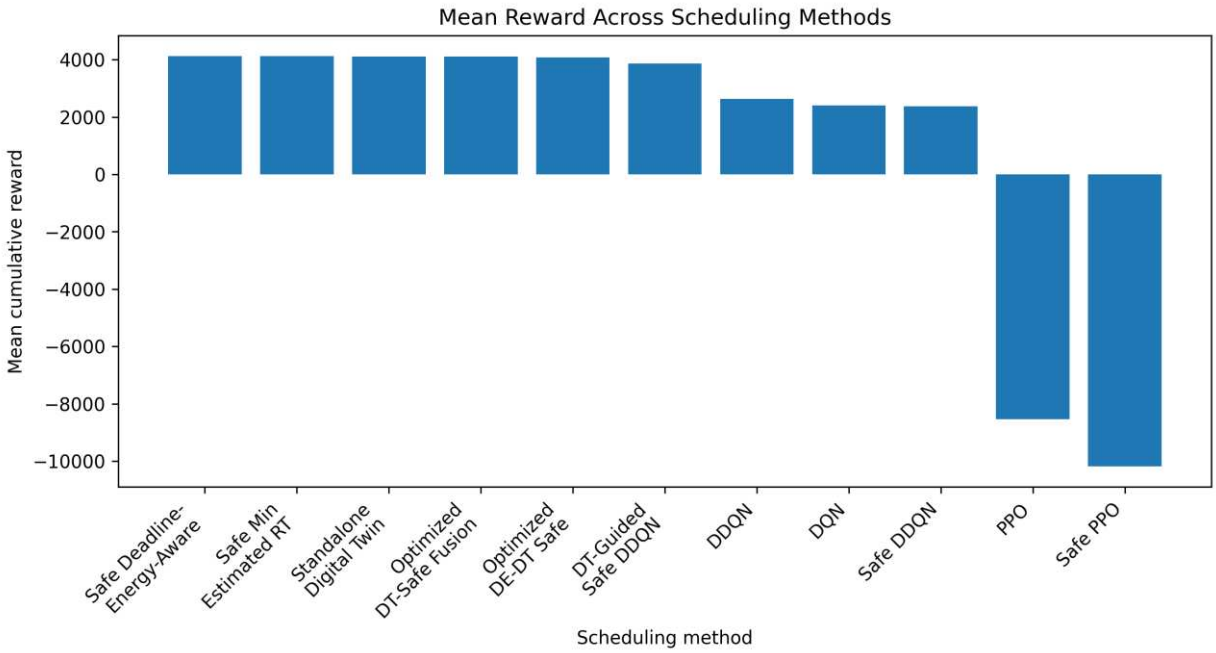


Figure 5. Mean cumulative reward across scheduling methods.

The strongest safe heuristics and the Digital Twin based schedulers belong to the highest reward group of schedulers as seen in figure 5. Safe Deadline-Energy-Aware records the maximum mean reward among all the average rewards and the Standalone Digital Twin and optimized DT-Safe fusion record the same mean reward. The value-based DRL methods get moderate positive rewards while PPO and Safe PPO are obvious outliers with negative rewards. This is similar to the reward function, which heavily punishes late submissions, slow feedback, overload, and unsafe behavior.

4.5 Descriptive Dispersion and Confidence Intervals

The uploaded per-run statistical tables enable strengthening of the results section by including dispersion and confidence intervals for the policies whose 30-run raw results are available. Table 3 reports mean ± SD and 95% confidence intervals for the principal schedulers. The two leading safe heuristics and the Standalone Digital Twin have overlapping confidence intervals for DMR and closely overlapping confidence intervals for response time and reward, supporting the interpretation that the Digital Twin is statistically competitive with the strongest safe heuristics rather than universally superior to them.

Table 3. Descriptive statistics for principal schedulers with uploaded 30-run raw results.

Policy	N	Avg RT mean ± SD	Avg RT 95% CI	DMR mean ± SD	DMR 95% CI	Reward mean ± SD	Reward 95% CI
SDEA	30	132.04 ± 6.28	[129.79, 134.29]	0.0227 ± 0.0224	[0.0147, 0.0307]	4126.77 ± 248.54	[4037.83, 4215.71]
Safe Min RT	30	132.10 ± 6.32	[129.84, 134.36]	0.0232 ± 0.0223	[0.0152, 0.0311]	4118.15 ± 252.32	[4027.86, 4208.44]
Standalone DT	30	132.09 ± 6.34	[129.83, 134.36]	0.0242 ± 0.0242	[0.0155, 0.0328]	4112.63 ± 265.74	[4017.53, 4207.72]
Optimized DT-Safe	30	132.09 ± 6.34	[129.83, 134.36]	0.0242 ± 0.0242	[0.0155, 0.0328]	4112.63 ± 265.74	[4017.53, 4207.72]
DT-Guided Safe DDQN	30	137.63 ± 7.98	[134.78, 140.49]	0.0500 ± 0.0385	[0.0362, 0.0638]	3863.39 ± 362.31	[3733.74, 3993.04]
DDQN	30	159.37 ± 11.49	[155.26, 163.48]	0.1887 ± 0.0677	[0.1644, 0.2129]	2638.14 ± 608.07	[2420.54, 2855.73]
DQN	30	161.68 ± 11.67	[157.50, 165.85]	0.2193 ± 0.0619	[0.1972, 0.2415]	2401.77 ± 540.69	[2208.29, 2595.25]

Safe DDQN	30	162.87 ± 19.25	[155.98, 169.76]	0.2195 ± 0.1114	[0.1796, 0.2594]	2381.11 ± 1039.58	[2009.10, 2753.12]

The PPO, Safe PPO, and Optimized Deadline-Energy DT Safe Scheduler rows in the overall comparison are therefore interpreted descriptively in this revision. No standard deviations, confidence intervals, or p-values are reported for those methods unless their corresponding raw paired runs are provided.

4.6 Paired Statistical Comparison Against the Standalone Digital Twin

Table 4 reports paired Wilcoxon signed-rank tests with Holm correction, using the Standalone Digital Twin as the reference method. Positive relative-change values indicate that the Standalone Digital Twin is better than the comparator for that metric; negative values indicate that the comparator is numerically better. For DMR, the Standalone Digital Twin is significantly better than DT-Guided Safe DDQN, DDQN, DQN, and Safe DDQN after Holm correction. Its DMR is slightly worse than Safe Deadline-Energy-Aware and Safe Min Estimated RT, but these DMR differences are not significant after Holm correction ($p = 0.0520$ and $p = 0.1834$, respectively). In terms of reward, Safe Deadline-Energy-Aware is significantly better than the Standalone Digital Twin, but only by 0.34%. Safe Min Estimated RT is not significantly different from the Standalone Digital Twin on reward. These results reinforce the revised claim: the Digital Twin is not the top numerical method, but it is statistically competitive with the strongest safe heuristics and significantly stronger than the evaluated value-based DRL baselines.

Table 4. Paired Wilcoxon-Holm comparisons using Standalone Digital Twin as the reference scheduler.

Comparator	Metric	DT mean	Comp. mean	Rel. Δ%	Holm p	Cohen dz
SDEA	DMR	0.0242	0.0227	-6.62	0.0520	0.461
SDEA	Avg RT	132.0942	132.0410	-0.04	0.0190	0.396
SDEA	Reward	4112.63	4126.77	-0.34	0.0150	-0.483
Safe Min RT	DMR	0.0242	0.0232	-4.32	0.1834	0.301
Safe Min RT	Avg RT	132.0942	132.1007	0.00	1.0000	-0.043
Safe Min RT	Reward	4112.63	4118.15	-0.13	0.2707	-0.231
Optimized DT-Safe	DMR	0.0242	0.0242	0.00	1.0000	NA
Optimized DT-Safe	Avg RT	132.0942	132.0942	0.00	1.0000	NA
Optimized DT-Safe	Reward	4112.63	4112.63	0.00	1.0000	NA
DT-Guided Safe DDQN	DMR	0.0242	0.0500	51.67	1.72e-05	-1.017
DT-Guided Safe DDQN	Avg RT	132.0942	137.6343	4.03	1.86e-08	-1.276
DT-Guided Safe DDQN	Reward	4112.63	3863.39	6.45	1.86e-08	1.205
DDQN	DMR	0.0242	0.1887	87.19	1.72e-05	-2.646
DDQN	Avg RT	132.0942	159.3675	17.11	1.86e-08	-2.765
DDQN	Reward	4112.63	2638.14	55.89	1.86e-08	2.728
DQN	DMR	0.0242	0.2193	88.98	1.72e-05	-3.219
DQN	Avg RT	132.0942	161.6760	18.30	1.86e-08	-2.745
DQN	Reward	4112.63	2401.77	71.23	1.86e-08	3.258

Safe DDQN	DMR	0.0242	0.2195	88.99	1.72e-05	-1.886
Safe DDQN	Avg RT	132.0942	162.8679	18.89	1.86e-08	-1.754
Safe DDQN	Reward	4112.63	2381.11	72.72	1.86e-08	1.834

4.7 Digital Twin Prediction Validity

Table 5 assesses whether the Digital Twin provides a useful predictive layer rather than serving only as a rule-based scheduler. The response-time prediction achieves a mean absolute error of 36.79 ms and an RMSE of 45.71 ms. The post-action queue MAE is 29.19 ms, while the post-action energy MAE is 0.000115. The Digital Twin achieves a deadline-prediction accuracy of 0.9742, with a deadline Brier score of 0.0249, indicating a low probabilistic prediction error. It also achieves an overload-prediction accuracy of 1.0000 and a post-action safety-prediction accuracy of 1.0000. The fallback ratio is 0.0102, indicating that the corrective fallback mechanism is required in only a small proportion of scheduling decisions.

Table 5. Digital Twin prediction-validity metrics across 30 independent runs.

Metric	Value	Interpretation
Response Time MAE (ms)	36.7900	Average absolute error of response-time prediction
Response Time RMSE (ms)	45.7100	Root-mean-square response-time prediction error
Queue MAE (ms)	29.1900	Average absolute error of post-action queue prediction
Energy MAE	0.000115	Average absolute error of post-action energy prediction
Deadline Prediction Accuracy	0.974200	Accuracy of the predicted deadline-miss label
Deadline Brier Score	0.024900	Probabilistic prediction error for the deadline-miss estimate; lower values indicate more accurate probability predictions.
Overload Prediction Accuracy	1.0000	Accuracy of predicted overload events
Post-Action Safety Accuracy	1.0000	Accuracy of post-action safety feasibility prediction
Fallback Ratio	0.010200	Fraction of decisions corrected by the fallback safety mechanism

The classification results of the Digital Twin for deadline misses, overload events, and post-action safety feasibility are shown in Figure 6. It complements Table 5, which gives numerical values to the accuracy achieved in the three predicted outcomes, by visually comparing the accuracy achieved across the three outcomes.

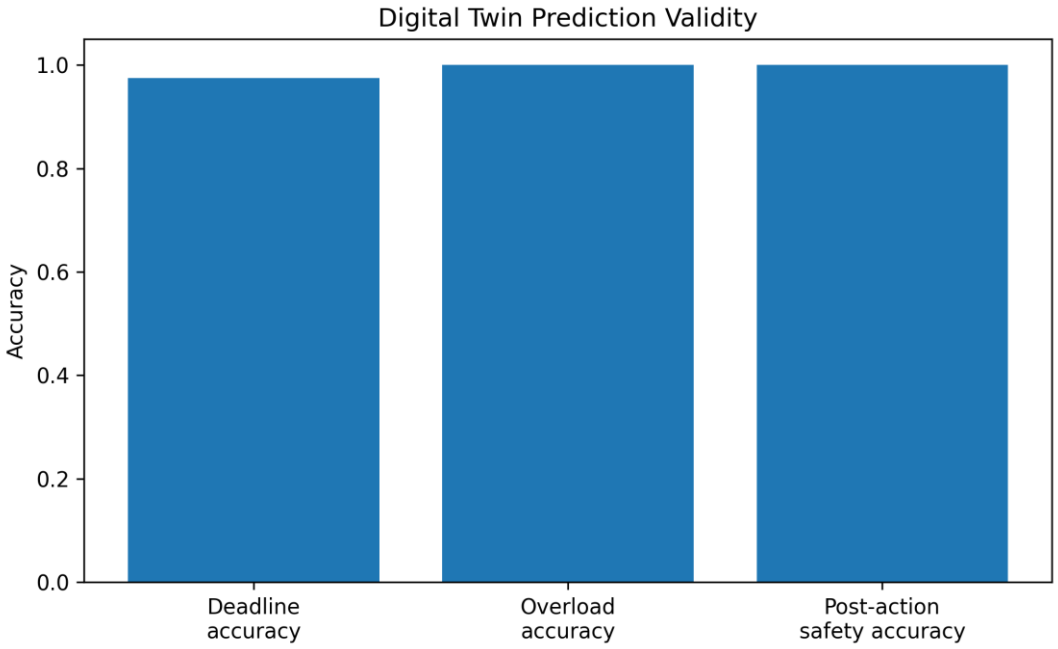


Figure 6. Digital Twin prediction validity across deadline, overload, and post-action safety metrics.

As seen in figure 6, the Digital Twin is demonstrating high classification accuracy for the assessed outcomes. The accuracy of the deadline-miss prediction is 0.9742, and the accuracy of the overload and post-action safety predictions are both 1.0000, respectively, in the evaluated runs. The results show that the label of the predicted and observed word is highly agreeing in meaning. It is important to note, however, that the best overload and safety accuracy should be considered in conjunction with the frequency of positive and negative events since accuracy may be affected by class imbalance, if the predicted events are rare.

Figure 7 is a focus on the continuous prediction errors for response time and post-action queueing workload. The errors are used to provide a more detailed evaluation of the Digital Twin than classification accuracy as they quantify the numerical difference between the predicted and measured system responses.

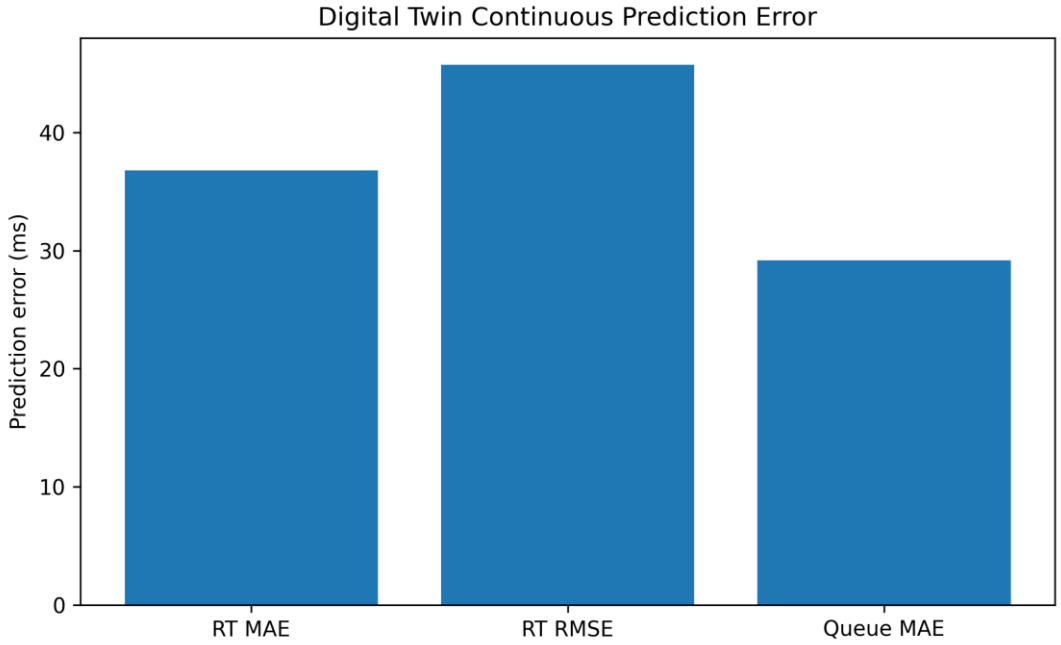


Figure 7. Digital Twin continuous prediction errors for response time and queue estimation.

As shown in figure 7, the Digital Twin gives reasonably close numerical estimates of the post-action state of the system with a response-time MAE of 36.79 ms, a response-time RMSE of 45.71 ms and a queue MAE of 29.19 ms. The RMSE is greater than the MAE, indicating that there are more prediction errors larger than the typical absolute error. Despite this, the continuous-error results, combined with the high deadline-prediction accuracy and low fallback ratio, support the view that the Digital Twin is an informative predictive layer, as opposed to just a hard-coded scheduling rule.

4.7 Statistical Significance Analysis

Eleven scheduling-policies were evaluated using complete paired observations over 30 common evaluation seeds and inferential analysis was performed. Each competing method was compared to the Standalone Digital Twin, which was used as the base reference policy. The same seeds were used in all of the policies evaluated, so paired-sample analysis was used for the observations.

The Shapiro–Wilk test was used to test the normality of paired differences. Because several difference distributions for comparisons were not normally distributed, the Wilcoxon signed rank test was used as the main inferential procedure. A Holm correction was performed independently across each of the performance metrics to correct for the FWE due to multiple comparisons. An adjusted significance level of 0.05 was used to assess statistical significance. A paired difference was also used to produce Cohen's d_z as an additional standardized effect size measure.

Table 3 displays the descriptive statistics and 95% confidence intervals of the main methods used for scheduling. The results are presented for the performance as obtained from the 30 evaluation runs, as well as for the average performance.

Table 3. Descriptive statistics and 95% confidence intervals for the principal scheduling methods across 30 paired evaluation runs.

Policy	Avg RT, mean ± SD (95% CI), ms	DMR, mean ± SD (95% CI)	Safety violation, mean ± SD	Reward, mean ± SD (95% CI)
Safe Deadline-Energy-Aware	132.04 ± 6.28 (129.79–134.29)	0.0227 ± 0.0224 (0.0147–0.0307)	0.0088 ± 0.0064	4126.77 ± 248.54 (4037.83–4215.71)
Safe Min Estimated RT	132.10 ± 6.32 (129.84–134.36)	0.0232 ± 0.0223 (0.0152–0.0312)	0.0100 ± 0.0066	4118.15 ± 252.32 (4027.86–4208.44)
Standalone Digital Twin	132.09 ± 6.34 (129.83–134.36)	0.0242 ± 0.0242 (0.0155–0.0328)	0.0095 ± 0.0074	4112.63 ± 265.74 (4017.53–4207.72)
Optimized DT-Safe Fusion	132.09 ± 6.34 (129.83–134.36)	0.0242 ± 0.0242 (0.0155–0.0328)	0.0095 ± 0.0074	4112.63 ± 265.74 (4017.53–4207.72)
DT-Guided Safe DDQN	137.63 ± 7.98 (134.78–140.49)	0.0500 ± 0.0385 (0.0362–0.0638)	0.0093 ± 0.0065	3863.39 ± 362.31 (3733.74–3993.04)
DDQN	159.37 ± 11.49 (155.26–163.48)	0.1887 ± 0.0677 (0.1644–0.2129)	0.0110 ± 0.0074	2638.14 ± 608.07 (2420.54–2855.73)
DQN	161.68 ± 11.67 (157.50–165.85)	0.2193 ± 0.0619 (0.1972–0.2415)	0.0128 ± 0.0104	2401.77 ± 540.69 (2208.29–2595.25)
Safe DDQN	162.87 ± 19.25 (155.98–169.76)	0.2195 ± 0.1114 (0.1796–0.2594)	0.0112 ± 0.0078	2381.11 ± 1039.58 (2009.10–2753.12)

The results for response-time and deadline-miss of the three prevalent schedulers (Safe Deadline-Energy-Aware, Safe Min Estimated RT, and the Standalone Digital Twin) are grouped closely together as shown in Table 3. Their average miss ratio does not exceed 0.025 and the average response time is around 132 ms. Also the confidence intervals of these methods overlap significantly, which means that their descriptive performance is clustered together.

The description, SD and CI for Standalone Digital Twin are equal to those of Optimized DT-Safe Fusion. This is consistent with the optimized fusion that sets q_weight = 0.0 and dt_weight = 1.0 to cancel out the contribution of the learned Q-value signal. The value-based DRL methods, on the other hand, have confidence intervals that are pushed towards longer response times, a higher ratio of deadlines missed, and a lower cumulative reward.

The main paired Wilcoxon comparisons between the Standalone Digital Twin and the competing policies are summarized in Table 4. p-values have been Holm-adjusted. The difference, Δ, is defined as the mean of the Standalone Digital Twin – the mean of the competing policy. Therefore, negative differences signal a Digital Twin's benefits in terms of response time and the percentage of missed deadlines, while positive differences signal a Digital Twin's benefits on cumulative reward.

Table 4. Holm-adjusted paired Wilcoxon comparisons between the Standalone Digital Twin and competing scheduling policies.

Competitor	Avg RT Δ (ms)	Adjusted p	DMR Δ	Adjusted p	Reward Δ	Adjusted p	Cohen's d_z for DMR
Safe Deadline-Energy-Aware	+0.053	0.0190	+0.0015	0.0520	−14.14	0.0150	+0.461
Safe Min Estimated RT	−0.006	1.0000	+0.0010	0.1834	−5.52	0.2707	+0.301
Optimized DT-Safe Fusion	0.000	1.0000	0.0000	1.0000	0.00	1.0000	—
DT-Guided Safe DDQN	−5.540	<0.0001	−0.0258	<0.0001	+249.23	<0.0001	−1.017
DDQN	−27.273	<0.0001	−0.1645	<0.0001	+1474.49	<0.0001	−2.646

DQN	−29.582	<0.0001	−0.1952	<0.0001	+1710.86	<0.0001	−3.219
Safe DDQN	−30.774	<0.0001	−0.1953	<0.0001	+1731.52	<0.0001	−1.886
Safe Shortest Queue	−25.060	<0.0001	−0.1378	<0.0001	+1236.39	<0.0001	−1.693
Shortest Queue	−25.049	<0.0001	−0.1375	<0.0001	+1709.95	<0.0001	−1.680
Random	−33.072	<0.0001	−0.1993	<0.0001	+2314.18	<0.0001	−3.760

As shown in Table 4, the results indicate that the Standalone Digital Twin has a significant performance advantage over the value-based DRL and basic heuristic baselines evaluated. The Standalone Digital Twin has significantly reduced response times, deadline-miss ratios, and significantly increased cumulative rewards compared to DQN, DDQN, and Safe DDQN, with all adjusted p-values below 0.0001.

Also, large effect sizes when items are missed from the time limit. The absolute Cohen's d_z values are 3.219, 2.646, and 1.886, respectively, against DQN, DDQN, and Safe DDQN. These values imply not only an improvement that is statistically significant over the value-based DRL baselines but also significant standardized paired differences.

DT-Guided Safe DDQN fusion also significantly outperforms the fixed-weight fusion in terms of response time, deadline-miss ratio, and cumulative reward for all adjusted p-values < 0.0001. The deadline-miss effect size against the fixed-weight fusion is big in absolute value with a Cohen's d_z of 1.017. The result suggests that performance was hurt by assigning a positive weight to the learned Q-value signal in this setting.

Comparisons with strong safe heuristics must be given a more conservative reading. Safe Deadline-Energy-Aware achieves a lower (numerically lower) deadline-miss ratio than the Standalone Digital Twin. But the Holm-corrected Wilcoxon p-value for this difference is 0.0520, which is just above the chosen significance level. Hence, no statistically significant difference in the deadline-miss difference was found between the two methods.

A statistically significant improvement of the Smallest Significant Difference (SSD) is seen when comparing the Standalone Digital Twin with the Safe Deadline-Energy-Aware in terms of average response time and cumulative reward. However, these values are capped at about 0.05ms and 14.14 reward units, respectively. Although Safe Deadline-Energy-Aware is still the number one option, the gap between it and Standalone Digital Twin is not significant in practice.

After Holm correction, there is no statistically significant difference between Standalone Digital Twin and Safe Min Estimated RT for response time, deadline-miss ratio or cumulative reward. This outcome is statistically identical to the two best safe heuristics, indicating that the Standalone Digital Twin is a viable option.

The Optimized DT-Safe Fusion is statistically equal to the Standalone Digital Twin in all evaluated metrics due to the equality of the paired observations generated by the two policies. This is not to be interpreted as another Hybrid Improvement. Rather, it indicates the optimization result in which the learned Q-value component is given zero weight and the fusion decision is completely driven by the Digital Twin.

After safety violation correction, the Standalone Digital Twin is very similar to Safe Deadline-Energy-Aware, Safe Min Estimated RT, Safe DDQN, Safe Shortest Queue, and the DT-Guided Safe DDQN fusion. Compared to the non-safe Random and Shortest Queue policies, the values for significant safety improvements are seen, which yields p-values of less than 0.0001.

The eleven statistically evaluated policies do not show any inferential differences, as all of these overload ratios are zero for both runs. The inferential comparisons below do not include PPO and Safe PPO due to the lack of complete paired per-run observations for these policies. The results they produce are therefore only descriptive comparisons.

In summary, the inferential analysis corroborates three key findings. First, the Standalone Digital Twin offers significantly better performance in terms of statistically significant improvement and practically large improvement, over the evaluated DQN, DDQN, Safe DDQN, basic heuristic, and fixed-weight fusion baselines. Second, the Digital Twin is statistically competitive with the best safe heuristics for deadline-miss performance; there is no significant difference in deadline-miss performance after the Holm correction. Thirdly, the Optimized DT-Safe Fusion does not take any hybrid gain as it sets the weight of the learned Q-value component to zero and yields the same paired result as the Standalone Digital Twin.

4.8 Corrected Interpretation of the Main Findings

The revised interpretation is deliberately conservative. First, the Standalone Digital Twin is not the best numerical method overall; the Safe Deadline-Energy-Aware heuristic remains the numerical leader. However, the difference between the two methods in deadline-miss ratio is not statistically significant after Holm correction, and the statistically significant differences in response time and cumulative reward have small absolute magnitudes.

Second, the Standalone Digital Twin significantly outperforms the evaluated DQN, DDQN, Safe DDQN, basic heuristic, and fixed-weight fusion baselines for the principal performance outcomes supported by complete paired observations. It also achieves substantially better descriptive performance than PPO and Safe PPO, although no inferential claims are made for these policy-gradient methods because complete paired per-run observations were not available.

Third, the optimized fusion result should not be presented as evidence of an additional hybrid gain. The optimal fusion weights assign zero contribution to the learned Q-value component, and the optimized fusion produces identical paired observations to the Standalone Digital Twin.

Fourth, the poor PPO and Safe PPO results should be interpreted as configuration-specific rather than as a general limitation of policy-gradient methods.

5. Discussion

The main contribution of this work is not simply the introduction of a new scheduler with the highest numerical rank. Rather, it is the development of an interpretable predictive decision-support layer capable of evaluating candidate node assignments before execution. This distinction is important because the strongest safe heuristic also uses well-designed local information and achieves the best numerical performance. The value of the Digital Twin therefore lies in combining near-best scheduling performance with action-level predictions of response time, deadline-miss risk, queue evolution, energy depletion, overload, and post-action safety feasibility.

The statistical results reinforce this interpretation. Although Safe Deadline-Energy-Aware remains the numerical leader, the difference between this heuristic and the Standalone Digital Twin in deadline-miss ratio is not statistically significant after Holm correction. The statistically significant differences observed in average response time and cumulative reward are small in absolute magnitude, amounting to approximately 0.05 ms and 14.14 reward units, respectively. In addition, no statistically significant differences are found between the Standalone Digital Twin and Safe Min Estimated RT for average response time, deadline-miss ratio, or cumulative reward. These findings indicate that the Standalone Digital Twin is statistically competitive with the strongest safe heuristics for the central deadline-compliance outcome.

The results also indicate that Digital Twin-based predictive reasoning was more effective than the evaluated learned Q-value signals under the studied configuration. The Standalone Digital Twin significantly outperforms DQN, DDQN, and Safe DDQN in average response time, deadline-miss ratio, and cumulative reward after Holm correction. The associated deadline-miss effect sizes are large, showing that the observed differences are not only statistically significant but also substantial in standardized paired terms.

The DT-guided fusion results provide further support for this interpretation. When the learned Q-value component receives a positive weight, the fixed-weight DT-Guided Safe DDQN fusion performs significantly worse than the Standalone Digital Twin across response time, deadline-miss ratio, and cumulative reward. The optimized fusion assigns q_weight = 0.0 and dt_weight = 1.0, indicating that the learned Q-value signal does not provide additional useful guidance under the evaluated configuration. The optimized fusion also produces identical paired performance observations to the Standalone Digital Twin. Nevertheless, this outcome should be interpreted as full dependence on Digital Twin information within the fusion formulation rather than assumed mathematical equivalence unless the complete scoring, masking, fallback, and action-selection rules are also identical.

The policy-gradient results should be interpreted with care. PPO and Safe PPO perform poorly in the descriptive comparison under the evaluated discrete node-selection setup, with high deadline-miss ratios, long response times, and substantial overload. However, complete paired per-run observations were not available for these methods, and no inferential claims are therefore made regarding their differences from the Digital Twin. These results do not demonstrate that PPO is unsuitable for all fog-edge scheduling or offloading problems. Alternative reward scaling, longer training, different rollout lengths, entropy schedules, network architectures, or continuous and hybrid action formulations may produce different outcomes. The current findings should therefore be framed as evidence that the evaluated PPO configuration is weak for this particular discrete, deadline-constrained, and dynamic node-selection problem.

The safety analysis provides an additional distinction between feasibility enforcement and scheduling quality. Hard action masking substantially reduces direct safety violations by preventing unsafe node selections. However, the Safe PPO results show that action masking alone does not guarantee effective congestion management or deadline-aware scheduling. Among the methods included in the paired statistical analysis, the Standalone Digital Twin does not differ significantly in safety-violation ratio from the principal safe heuristic, safe DRL, or DT-guided methods after Holm correction. Significant safety improvements are observed only relative to the non-safe Random and Shortest Queue policies.

A practical implication of these findings is that strong safe heuristics should always be included in the baseline set. Comparing a Digital Twin or DRL scheduler only with weak heuristics can inflate the apparent contribution of the proposed method. In this study, the inclusion of Safe Deadline-Energy-Aware and Safe Min Estimated RT provides a more realistic benchmark and leads to a more accurate conclusion. The Standalone Digital Twin is statistically competitive with the strongest safe heuristics for deadline-miss performance and significantly stronger than the evaluated value-based DRL and fixed-weight fusion baselines, while providing additional interpretability through prediction-validity metrics.

Overall, the findings suggest that the main advantage of the proposed Digital Twin is not absolute numerical dominance over every scheduling strategy. Its contribution lies in providing a competitive, transparent, and safety-aware predictive mechanism that explains the expected consequences of candidate scheduling decisions before execution. This combination of near-best performance, statistical competitiveness with strong safe heuristics, and action-level interpretability represents the principal contribution of the framework.

6. Limitations and Threats to Validity

This study is simulation-based. The environment includes node failures, node recovery, workload bursts, energy depletion, and queue dynamics, but it is not a real fog-edge deployment. Therefore, the results should be validated on larger simulators, trace-driven workloads, or real testbeds before making deployment-level claims.

The Digital Twin is analytical rather than data-trained. This makes it interpretable and efficient, but it also means that its accuracy depends on how well the analytical assumptions approximate the simulated system. In environments with more complex network behavior, cold-start effects, service interference, or multi-hop communication, the Digital Twin equations may require calibration or learning-based correction.

Complete paired 30-run results were available for only a subset of the evaluated scheduling methods. For these methods, the analysis reports standard deviations, 95% confidence intervals, Wilcoxon signed-rank tests with Holm correction, and the selected paired effect-size measures. Complete paired per-run results were not available for PPO, Safe PPO, or the Optimized Deadline-Energy DT Safe Scheduler. These methods are therefore retained as descriptive mean-value comparisons only, and no inferential claims are made for them.

The DRL baselines are evaluated under a fixed training budget and fixed hyperparameter configuration. More extensive hyperparameter tuning may improve DQN, DDQN, PPO, or Safe PPO. Therefore, the results support claims about the evaluated baselines under the specified configuration, not about the theoretical limits of those algorithms.

7. Conclusion

This paper presented an interpretable Digital Twin-safe scheduling framework for deadline-aware task allocation in dynamic fog-edge computing. The framework combines an analytical Digital Twin predictive layer with hard safety masking to evaluate candidate node assignments before execution. For each candidate action, the Digital Twin estimates response time, deadline-miss risk, post-action queue state, residual energy, overload, and post-action safety feasibility, enabling the scheduler to select a low-risk feasible node.

The results show that the Safe Deadline-Energy-Aware heuristic achieves the best numerical performance overall. However, its deadline-miss ratio is not statistically different from that of the Standalone Digital Twin after Holm correction. Although small statistically significant paired differences are observed in average response time and cumulative reward, their absolute magnitudes are limited to approximately 0.05 ms and 14.14 reward units, respectively. No statistically significant differences are found between the Standalone Digital Twin and Safe Min Estimated RT for average response time, deadline-miss ratio, or cumulative reward. These findings indicate that the Standalone Digital Twin is statistically competitive with the strongest safe heuristics for the principal deadline-compliance outcome.

The Standalone Digital Twin achieves a deadline-miss ratio of 2.42% and a mean cumulative reward of 4112.63 while also providing interpretable action-level predictive diagnostics. It significantly outperforms DQN, DDQN, and Safe DDQN in average response time, deadline-miss ratio, and cumulative reward after Holm correction. The corresponding paired deadline-miss effect sizes are large, indicating that these differences are substantial as well as statistically significant. The Digital Twin also achieves substantially better descriptive performance than PPO and Safe PPO; however, no inferential claims are made for these policy-gradient methods because complete paired per-run observations were not available.

The fixed-weight DT-Guided Safe DDQN fusion improves descriptively on the value-based DRL baselines but remains significantly weaker than the Standalone Digital Twin across average response time, deadline-miss ratio, and cumulative reward. Moreover, the optimized DT-Safe fusion assigns zero weight to the learned Q-value component and produces identical paired performance observations to the Standalone Digital Twin. This finding indicates that the learned Q-value signal does not provide an additional benefit under the studied configuration. Nevertheless, the optimized fusion should be described as fully Digital-Twin-driven within the fusion formulation rather than automatically assumed to be mathematically identical to the Standalone Digital Twin unless their complete scoring, masking, fallback, and action-selection rules are also identical.

The prediction-validity results further support the role of the Digital Twin as an informative predictive layer rather than only a fixed scheduling rule. The model achieves high deadline-prediction accuracy, low probabilistic prediction error, accurate overload and post-action safety classification, and relatively low continuous prediction errors for response time, queue state, and residual energy. These outputs provide transparent action-level explanations that are not directly available from conventional heuristic or DRL policies.

Overall, the proposed Digital Twin layer should be positioned as a competitive and interpretable decision-support mechanism for safe, deadline-aware fog-edge scheduling rather than as a universal replacement for strong hand-crafted safe heuristics. Its main value lies in combining near-best scheduling performance, statistical competitiveness with the strongest safe heuristics, significant improvements over the evaluated value-based DRL baselines, and transparent pre-execution predictions of the expected consequences of each candidate allocation decision.

Future work should evaluate the framework using trace-driven workloads, larger and more heterogeneous fog-edge infrastructures, multiple task classes, and real fog-edge testbeds. Further research should investigate adaptive Digital Twin calibration, online learning, broader DRL hyperparameter optimization, alternative reward formulations, and learning-based correction of the analytical prediction model. Complete paired per-run results should also be retained for all evaluated schedulers, including PPO, Safe PPO, and the optimized Digital Twin variants, to support consistent reporting of dispersion, statistical significance, and effect sizes across every method.

Appendix A. Additional Descriptive Statistics

Table A1 confirms that the three leading methods—Safe Deadline-Energy-Aware, Safe Min Estimated RT, and the Standalone Digital Twin—have closely clustered mean response times, deadline-miss ratios, and cumulative rewards. The Standalone Digital Twin and Optimized DT-Safe rows also have identical reported means and standard deviations. In contrast, the value-based DRL methods show higher response times and deadline-miss ratios, while the non-safe Random and Shortest Queue baselines exhibit

substantially higher variability in safety violations. These descriptive statistics support the performance patterns reported in the main comparison, while inferential conclusions remain limited to methods with complete paired per-run results.

Table A1. Compact descriptive statistics from the uploaded 30-run results file.

Policy	N	Avg RT mean ± SD	DMR mean ± SD	Safety mean ± SD	Reward mean ± SD
SDEA	30	132.04 ± 6.28	0.0227 ± 0.0224	0.0088 ± 0.0064	4126.77 ± 248.54
Safe Min RT	30	132.10 ± 6.32	0.0232 ± 0.0223	0.0100 ± 0.0066	4118.15 ± 252.32
Standalone DT	30	132.09 ± 6.34	0.0242 ± 0.0242	0.0095 ± 0.0074	4112.63 ± 265.74
Optimized DT-Safe	30	132.09 ± 6.34	0.0242 ± 0.0242	0.0095 ± 0.0074	4112.63 ± 265.74
DT-Guided Safe DDQN	30	137.63 ± 7.98	0.0500 ± 0.0385	0.0093 ± 0.0065	3863.39 ± 362.31
DDQN	30	159.37 ± 11.49	0.1887 ± 0.0677	0.0110 ± 0.0074	2638.14 ± 608.07
DQN	30	161.68 ± 11.67	0.2193 ± 0.0619	0.0128 ± 0.0104	2401.77 ± 540.69
Safe DDQN	30	162.87 ± 19.25	0.2195 ± 0.1114	0.0112 ± 0.0078	2381.11 ± 1039.58
Shortest Queue	30	157.14 ± 10.62	0.1617 ± 0.0889	0.1265 ± 0.1224	2402.68 ± 947.65
Safe Shortest Queue	30	157.15 ± 10.58	0.1620 ± 0.0885	0.0075 ± 0.0061	2876.23 ± 714.74
Random	30	165.17 ± 6.88	0.2235 ± 0.0598	0.1495 ± 0.0348	1798.45 ± 452.99

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in Proc. 1st ACM MCC Workshop Mobile Cloud Computing (MCC'12), 2012, pp. 13-16, doi: 10.1145/2342509.2342513.

[2] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," IEEE Commun. Surveys Tuts., vol. 20, no. 3, pp. 1826-1857, 3rd Quart. 2018, doi: 10.1109/COMST.2018.2814571.

[3] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," IEEE J. Sel. Areas Commun., vol. 34, no. 5, pp. 1728-1739, May 2016, doi: 10.1109/JSAC.2016.2545559.

[4] F. Y. Okay and S. Ozdemir, "A fog computing based smart grid model," in Proc. Int. Symp. Networks, Computers and Communications (ISNCC), Hammamet, Tunisia, 2016, pp. 1-6, doi: 10.1109/ISNCC.2016.7746062.

[5] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," IEEE Commun. Surveys Tuts., vol. 23, no. 4, pp. 2131-2165, 4th Quart. 2021, doi: 10.1109/COMST.2021.3106401.

[6] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurr. Comput.: Pract. Exper., vol. 24, no. 13, pp. 1397-1420, 2012, doi: 10.1002/cpe.1867.

[7] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[8] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," J. Commun. Netw., vol. 24, no. 1, pp. 83-98, Feb. 2022, doi: 10.23919/JCN.2021.000041.

[9] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529-533, Feb. 2015, doi: 10.1038/nature14236.

[10] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proc. AAAI Conf. Artif. Intell., 2016, pp. 2094-2100.

[11] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," Future Gener. Comput. Syst., vol. 110, pp. 1098-1115, Sep. 2020, doi: 10.1016/j.future.2019.09.060.

[12] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary Perspectives on Complex Systems. Cham, Switzerland: Springer, 2017, pp. 85-113.

[13] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," Int. J. Adv. Manuf. Technol., vol. 94, no. 9-12, pp. 3563-3576, Feb. 2018, doi: 10.1007/s00170-017-0233-1.

[14] C. Xu, Z. Tang, H. Yu, P. Zeng, and L. Kong, "Digital Twin-driven collaborative scheduling for heterogeneous task and edge-end resource via multi-agent deep reinforcement learning," IEEE J. Sel. Areas Commun., vol. 41, no. 10, pp. 3056-3069, Oct. 2023, doi: 10.1109/JSAC.2023.3310066.

[15] Y. Wu, K. Zhang, and Y. Zhang, "Digital Twin networks: A survey," IEEE Internet Things J., vol. 8, no. 18, pp. 13789-13804, Sep. 2021, doi: 10.1109/JIOT.2021.3079510.

[16] N. Elmobark and A. M. Said, "A data-driven unified taxonomy and benchmarking framework for machine learning-based task scheduling in edge-cloud computing," Research Square, Preprint, Jun. 2026, doi: 10.21203/rs.3.rs-9924991/v1.

[17] N. Elmobark, S. Elhishi, and A. M. Mohammed, "Attention-enhanced double deep Q-network for multi-objective deadline-aware task scheduling in fog computing," Research Square, Preprint, May 2026, doi: 10.21203/rs.3.rs-9267504/v1.

[18] N. Elmobark, S. Elhishi, and A. M. Mohammed, "Dissecting ADDQN: An ablation study for deadline-aware task scheduling in fog computing," Unpublished manuscript, under review.

[19] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," Softw.: Pract. Exper., vol. 47, no. 9, pp. 1275-1296, Sep. 2017, doi: 10.1002/spe.2509.

[20] M. Abd Elaziz, L. Abualigah, and I. Attiya, "Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments," Future Gener. Comput. Syst., vol. 124, pp. 142-154, Nov. 2021, doi: 10.1016/j.future.2021.05.026.

[21] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: A generic cloud computing model for vehicular ad hoc networks," IEEE Wireless Commun., vol. 22, no. 1, pp. 96-102, Feb. 2015.

[22] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," ACM Trans. Internet Technol., vol. 19, no. 1, Art. no. 9, pp. 1-21, Feb. 2019, doi: 10.1145/3186592.

[23] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," J. Netw. Comput. Appl., vol. 201, Art. no. 103333, May 2022, doi: 10.1016/j.jnca.2022.103333.

[24] Z. Wang, M. Goudarzi, M. Gong, and R. Buyya, "Deep reinforcement learning-based scheduling for optimizing system load and response time in edge and fog computing environments," Future Gener. Comput. Syst., vol. 152, pp. 55-69, Mar. 2024, doi: 10.1016/j.future.2023.10.012.

[25] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, "Deep reinforcement learning-based task scheduling in IoT edge computing," Sensors, vol. 21, no. 5, Art. no. 1666, Mar. 2021, doi: 10.3390/s21051666.

[26] H. van Hasselt, "Double Q-learning," in Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 2613-2621.

[27] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," IEEE Internet Things J., vol. 6, no. 3, pp. 4005-4018, Jun. 2019, doi: 10.1109/JIOT.2018.2876279.

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

[29] P. Cao, W. Liu, G. Liu, X. Ji, J. Zhao, and Y. Shi, "A workload-aware adaptive energy management policy for the smart factory using deep reinforcement learning," IEEE Trans. Ind. Inform., vol. 17, no. 12, pp. 7938-7948, Dec. 2021.

[30] M. Alshiekh, R. Bloem, R. Ehlers, B. Koenighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in Proc. AAAI Conf. Artif. Intell., vol. 32, no. 1, 2018, pp. 2669-2678, doi: 10.1609/aaai.v32i1.11797.

[31] J. Garcia and F. Fernandez, "A comprehensive survey on safe reinforcement learning," J. Mach. Learn. Res., vol. 16, no. 1, pp. 1437-1480, 2015.

[32] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, and R. Buyya, "Resource allocation and task scheduling in fog computing and Internet of Everything environments: A taxonomy, review, and future directions," ACM Comput. Surv., vol. 54, no. 11s, pp. 1-38, Jan. 2022, doi: 10.1145/3513002.

[33] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[34] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," White Paper, 2014. [Online]. Available: https://www.researchgate.net/publication/275211047

[35] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," Decis. Support Syst., vol. 145, Art. no. 113524, Jun. 2021, doi: 10.1016/j.dss.2021.113524.

[36] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in Digital Twin networks," IEEE Trans. Ind. Inform., vol. 17, no. 7, pp. 4968-4977, Jul. 2021, doi: 10.1109/TII.2020.3016320.

[37] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-Twin-enabled 6G: Vision, architectural trends, and future directions," IEEE Commun. Mag., vol. 60, no. 1, pp. 74-80, Jan. 2022, doi: 10.1109/MCOM.001.21143.