Engineering Frontiers
Vol. 1, No. 3, 2025 15
Research Article

Fault-Tolerant Real-Time Scheduling for Edge AI in US Critical Infrastructure

Zihe Hao [a] *
[a] College of Engineering, Northeastern University, 02115, USA

Abstract
The integration of Edge AI promises to enhance operational efficiency within US critical infrastructure, yet it simultaneously introduces significant challenges regarding reliability and real-time determinism. While existing scheduling methods are suitable for general IoT environments, they frequently fail to distinguish between safety-critical and non-critical tasks, potentially resulting in severe consequences following computational failures. To address this challenge, this paper proposes a framework termed Safety-Critical Graph Reinforcement Learning, specifically designed to handle fault-tolerant real-time scheduling in highly dynamic and adversarial edge environments. Unlike traditional responsive methods that rely solely on task migration, SC-GRL integrates an active primary-backup mechanism and permissible model degradation into the action space of a Proximal Policy Optimization agent. By modeling the edge topology as a dynamic graph with embedded criticality attributes, the SC-GRL agent learns to optimize a composite objective that prioritizes deadline satisfaction for high-criticality tasks over mere resource utilization. Extensive simulations utilizing real-world trajectories from the US energy and transportation sectors demonstrate that SC-GRL significantly reduces the deadline miss rate for critical tasks under heavy loads and random node failures compared to state-of-the-art graph-based baselines. The data points to a clear conclusion here. If we want to build truly resilient public infrastructure, any deep reinforcement learning we use for scheduling must have a built-in sense of what's most critical.
Keywords: Edge AI; Critical Infrastructure; Real-Time Scheduling; Fault Tolerance; Graph Reinforcement Learning; Deep Reinforcement Learning; Cyber-Physical Systems

1. Introduction

The digitization of US critical infrastructure encompasses most critical sectors, including smart grids, autonomous transportation networks, and industrial control systems. Currently, driven by the proliferation of Edge AI, US critical infrastructure has commenced a profound transformation. 0Operators are migrating computational intelligence from centralized cloud repositories to the network edge to reduce the latency required for real-time decision-making.However, this architectural shift introduces a complex paradox wherein the pursuit of operational efficiency significantly exacerbates the system's vulnerability to stochastic failures and malicious adversarial disruptions. Unlike commercial Internet of Things applications where service degradation is merely an inconvenience, the deployment of Edge AI within the context of critical infrastructure entails a zero-tolerance imperative for failure regarding safety-critical tasks, as a missed deadline or a computational error could potentially trigger cascading physical consequences and threaten public safety.0

In recent years, methodologies addressing scheduling problems have proliferated, with data-driven strategies receiving particular attention. As a representative direction, Deep Reinforcement Learning has been widely researched for its ability to autonomously learn scheduling strategies through interaction with the environment. For instance, recent research on Graph Reinforcement Learning by

* Corresponding author. E-mail: zhihehao123@gmail.com

https://doi.org/10.71411/ef.2025.v1i4.1375

Zhang and colleagues demonstrates the effectiveness of Graph Neural Networks in capturing topological dependencies among edge nodes.0 By modeling the edge environment as a dynamic graph structure, their approach has achieved significant success in optimizing average task response times. Such methods exhibit important methodological value by enhancing the adaptability of scheduling models to different network structures.**错误!未找到引用源。**

However, a systematic review of current mainstream research exposes a critical deficiency. The objective functions in the vast majority of scheduling models are designed to treat all tasks as equally important, lacking the ability to distinguish task criticality. This characteristic of "criticality blindness" makes it difficult for them to directly support critical infrastructure scenarios that impose extremely high safety requirements. In actual deployment, when a system enters a high-load or network congested state, such generic schedulers may prioritize the execution of lightweight logging tasks to optimize overall throughput. Consequently, they may inadvertently sacrifice computationally intensive but safety-critical anomaly detection tasks. In risk-sensitive environments, this decision logic is evidently unacceptable. It is worth noting that this issue is not confined to the Industrial IoT domain. In cloud-edge collaborative architectures with high requirements for system reliability, such as healthcare, balancing resource elastic scaling with task real-time performance similarly constitutes a fundamental challenge awaiting solution.0

Furthermore, existing fault tolerance mechanisms within current mainstream edge computing frameworks predominantly rely on reactive strategies, typically exemplified by checkpoint recovery or dynamic migration subsequent to task execution failures. Although such designs possess a degree of rationality within general computing scenarios, preliminary experiments and simulation outcomes indicate that reactive migration frequently incurs non-negligible additional delays, thereby causing real-time tasks to struggle with meeting their deadline requirements. During the early exploratory phase of this research, we attempted to directly retrain a Proximal Policy Optimization agent by solely penalizing task dropping to enhance fault tolerance. Because the previous strategy prioritized extreme caution over efficiency, it crippled resource utilization and struggled with mixed-priority scheduling. 0To keep critical infrastructure robust, we must move away from simply cleaning up after failures; instead, the focus should be on building a predictive scheduler that secures redundant resources for VIP workflows the moment it senses a potential node breakdown.

In response to the aforementioned theoretical and practical limitations, this paper proposes a fault-tolerant real-time scheduling framework tailored for Edge AI scenarios within US critical infrastructure, termed Safety-Critical Graph Reinforcement Learning. Distinct from previous scheduling methodologies that prioritize latency minimization, this framework introduces a multi-dimensional decision mechanism within the action space, encompassing active primary-backup replication and controllable model degradation.0 By embedding task criticality information into the state representation of Graph Neural Networks, the SC-GRL agent autonomously learns to distinguish between safety-critical and best-effort tasks. During resource contention phases, it prioritizes the continuous availability of core functions, dynamically sacrificing the Quality of Service of non-critical tasks when necessary. In summary, this study attempts to bridge the gap between classical real-time system theory and modern data-driven control methodologies, providing a viable pathway for constructing more resilient national critical infrastructure.0

2. Related Work

https://doi.org/10.71411/ef.2025.v1i4.1375

The academic discourse surrounding resource orchestration in distributed computing has evolved drastically, transitioning from static heuristic models to dynamic, data-driven paradigms. To situate our proposed Safety-Critical Graph Reinforcement Learning framework within a broader academic context, we examine existing literature through three converging dimensions: real-time scheduling for Edge AI, fault tolerance mechanisms in distributed systems, and the specific requirements for critical infrastructure protection.0

2.1. Real-time Scheduling for Edge AI

Artificial intelligence inference tasks are migrating from hyperscale cloud data centers to the network edge. As illustrated in Figure 1, which presents a typical three-layer edge computing architecture within US critical infrastructure, computational capability is effectively sinking from the central cloud to distributed edge nodes. Early interventions in this domain primarily utilized meta-heuristic methods, such as genetic algorithms or particle swarm optimization, to solve the NP-hard problem of task offloading. While these methods demonstrated theoretical effectiveness in static environments, their iterative convergence times were often excessively long, failing to meet the sub-second decision-making requirements for dynamic Edge AI workloads. Consequently, research focus has shifted toward Deep Reinforcement Learning, which promises O(1) inference complexity following offline training. This demand for sub-second decision-making aligns with technical pathways for real-time bottleneck detection and performance optimization via data-driven methods in complex industrial processes such as semiconductor manufacturing [2], reflecting the urgent need for deterministic latency in modern intelligent systems.0

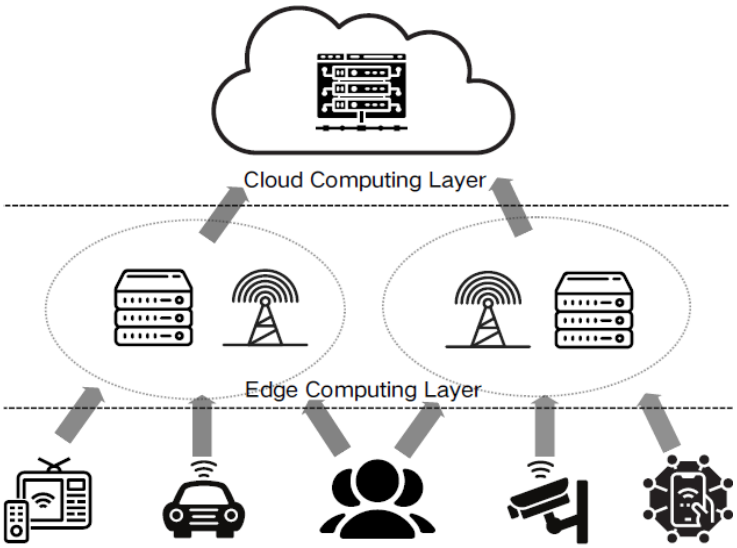

Figure 1. Three layer Edge Computing Architecture.

Recent studies have extensively explored Deep Reinforcement Learning based scheduling, utilizing Deep Q-Networks or Proximal Policy Optimization to maximize system throughput. For instance, some contemporary research treats AI tasks as generic black-box workloads characterized solely by CPU cycles and data size. However, when applied to deep neural networks, this abstraction is arguably oversimplified, as DNN inference latency correlates non-linearly with specific model architectures, such as CNNs or RNNs, and hardware accelerators. A recurring limitation in these studies is the optimization of average performance metrics. 0While optimizing average latency is acceptable for consumer IoT applications, it fails to address the tail latency problem that proves fatal in real-time control systems. By ignoring the Worst-Case Execution Time variance inherent in edge inference,

https://doi.org/10.71411/ef.2025.v1i4.1375

generic Deep Reinforcement Learning schedulers might inadvertently starve tasks with deadline constraints in pursuit of global reward maximization. This represents a distinct gap that our research attempts to bridge by directly introducing a deadline miss penalty into the value function.0

2.2 From Reactive to Proactive: Rethinking Edge Computing Resilience

Edge devices are scattered across various locations and run on very tight hardware budgets. Naturally, this setup makes them far more prone to temporary glitches than traditional, centralized cloud servers.The literature on fault-tolerant scheduling has traditionally bifurcated into reactive and proactive strategies.0

Reactive methods trigger task migration or re-execution upon detecting a fault, currently representing the state of the art. Take the 2024 study by Zhang and colleagues in MDPI Sensors, for instance. They tackled this by building a fault-tolerant scheduler that runs on graph reinforcement learning. Their approach represents a methodological advancement, utilizing graph neural networks to effectively capture topological dependencies among compute nodes and adapt to dynamic link failures. We recognize the elegance of their graph-based state representation, which allows agents to generalize across different network topologies, a characteristic we have adopted in our own work.0

However, a critical re-examination of such reactive paradigms reveals their inadequacy for critical infrastructure. In the initial stages of our research, we attempted to deploy similar reactive migration strategies using data trajectories from the US smart grid. We observed that the accumulated latency from fault detection, state synchronization, and task retransmission frequently exceeded the 20-millisecond safety margin required for Phasor Measurement Unit data processing.0 This empirical failure indicates that while reactive schemes like those proposed by Zhang et al. are cost-effective for general IoT scenarios, the recovery window they introduce is unacceptable for safety-critical applications. This realization motivated our research direction toward proactive fault tolerance. Contrary to prior views that considered resource redundancy wasteful, we argue that in the context of US critical infrastructure, implementing a primary-backup mechanism within the reinforcement learning action space is not merely an option but a necessary constraint to guarantee operational continuity under adversarial conditions.0

2.3 AI for Critical Infrastructure Protection

Deploying Edge AI in numerous US critical infrastructure sectors, including energy grids, intelligent transportation systems, and water treatment facilities, operates under a unique set of constraints defined by standards such as NIST SP 800-207 and NERC CIP. Existing literature in the cyber-physical systems domain often focuses on control theory stability, typically treating the computing layer as an idealized abstraction. 0Conversely, computer science literature tends to emphasize computational efficiency while underestimating the physical consequences of missed deadlines.

Research explicitly addressing the concept of mixed criticality in Edge AI scheduling is scarce. Most existing frameworks treat surveillance camera video streams and high-voltage transformer control signals equally, provided their data sizes are similar. This lack of semantic distinction creates a vulnerability where a system under heavy load might discard critical safety tasks to maintain the Quality of Service for non-critical monitoring tasks. Furthermore, few studies consider the adversarial nature of failures in critical infrastructure. Unlike random hardware faults, failures in these sectors can be orchestrated cyber-physical attacks. Our work attempts to fill this gap by integrating a criticality-aware mechanism that prioritizes the survivability of high-value tasks, acknowledging that in national infrastructure, availability and security are inextricably linked.0

https://doi.org/10.71411/ef.2025.v1i4.1375

3. System Model and Problem Definition

To rigorously analyze the scheduling dynamics within US critical infrastructure, we abstract the physical edge environment into a directed graph model. 错误!未找到引用源。This abstraction captures the heterogeneity of computing resources, the stochastic nature of wireless communication, and crucially, the differentiated criticality levels inherent to Cyber-Physical Systems. Distinct from generic IoT models that assume equal task importance, our formulation explicitly integrates mixed criticality semantics. This implies that system utility depends not merely on task completion, but more critically on the strict adherence of safety-critical workflows to their deadlines.0

3.1. Network Model and Criticality-Aware Graph Construction

Consider a specific domain of critical infrastructure, such as a smart grid distribution network or an intelligent transportation corridor, represented by a directed graph $G=(V,E)$. The vertex set V represents a collection of heterogeneous edge nodes, including local sensors, roadside units, and regional aggregation servers. The edge set E denotes the communication links between these entities, where a directed edge $e_{m,n}$ signifies a feasible transmission path from node v_m to v_n. To visually elucidate this modeling approach, Figure 2 specifically demonstrates how heterogeneous devices at the physical layer are abstracted and mapped into a graph structure. The physical entities on the left correspond one-to-one with the nodes in the graph model on the right, establishing a topological foundation for subsequent algorithm design.0

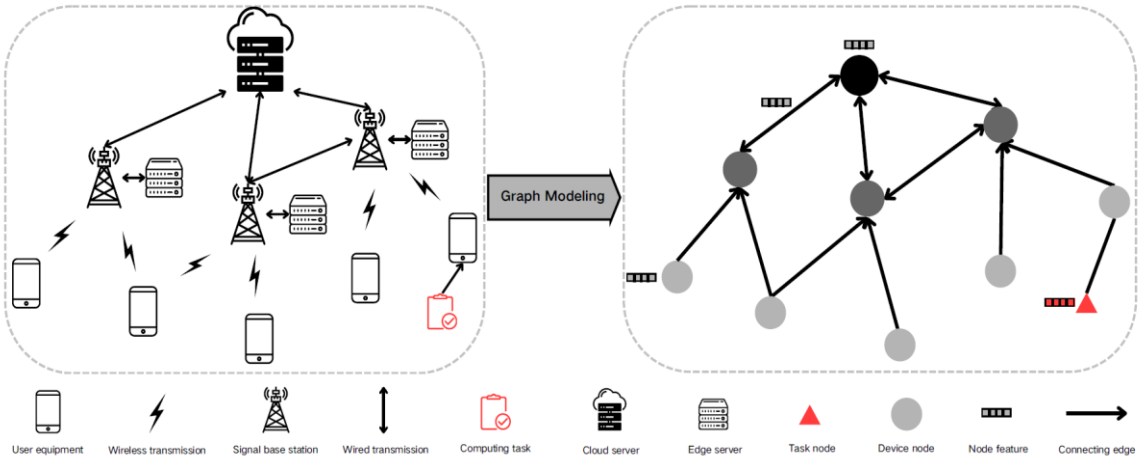

Figure 2. Graph Modeling of Edge Nodes with Security Attributes.

Unlike recent literature where state representations are typically confined to CPU frequency and queue length, we extend the feature vector X_k of each node v_k to incorporate security-oriented attributes. Specifically, let $X_k=[f_k,\omega_k,\varphi_k,S_k,T_k]$, where f_k denotes computing capability measured in Gcycles/s, ω_k represents the processing uncertainty coefficient derived from thermal throttling variance, and φ_k indicates the intrinsic failure probability. To capture the unique requirements of critical infrastructure, we introduce S_k as a security level to quantify the physical hardening and encryption capabilities of a node. We also introduce T_k as a trust score, which serves as a dynamic metric updated through historical reliability audits. This enhanced feature space enables the scheduling agent to recognize that high-performance servers with low trust scores are unsuitable for executing sensitive control tasks, a nuance frequently overlooked in purely performance-driven models.错误!未找到引用源。

3.2. Mixed Criticality Edge AI Task Model

To integrate network topology with computational workflows at a macro level, Figure 3 illustrates the overall Edge AI system model proposed in this paper. It details the complete lifecycle ranging from random task arrival and offloading decisions to execution on heterogeneous nodes.

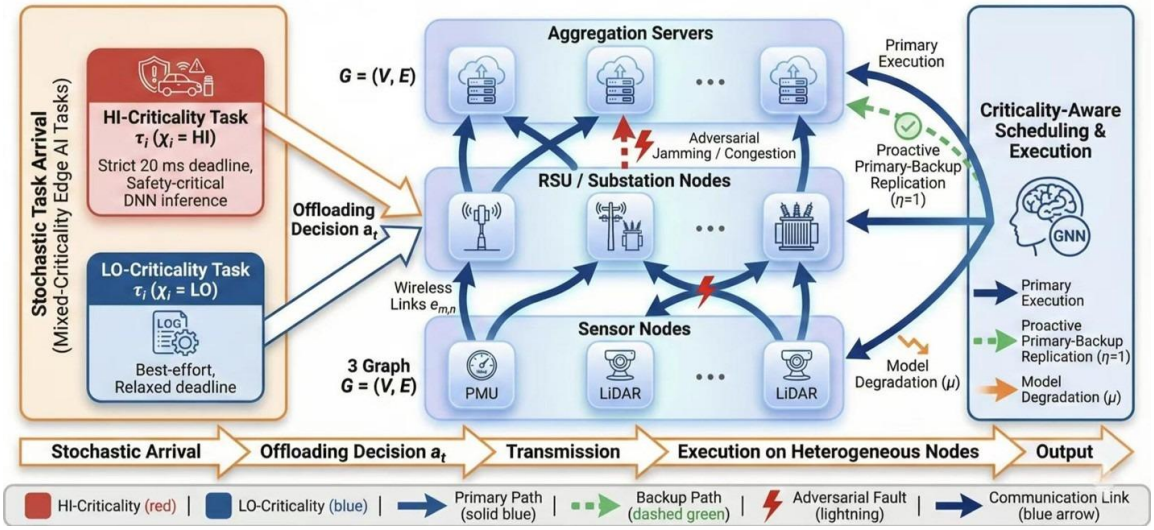

Figure 3. Edge AI System Model in US Critical Infrastructure.

We assume a discrete time-slot system wherein a set of AI inference tasks $T=\{\tau_1,\tau_2,...\}$ arrives randomly at the network edge. Each task τ_i is not modeled as a simple data packet but rather as a Deep Neural Network inference request characterized by a tuple: $\tau_i=(\alpha_i,\beta_i,d_i,\chi_i,M_i)$. Here, α_i represents the input data size, such as LiDAR point clouds or PMU voltage phasors, and β_i denotes the computational load in Gcycles. The parameter d_i is a strict relative deadline, beyond which the results become invalid or potentially hazardous.0

The defining feature of our model is the parameter χ_i, which specifies the criticality level of the task. HI criticality tasks, where χ_i=HI, correspond to safety-critical functions such as obstacle detection in autonomous vehicles or arc fault detection in substations. For these tasks, missing the deadline constitutes a system failure. LO criticality tasks, where χ_i=LO, represent best-effort services such as routine data logging or video streaming for post-event analysis, where occasional delays are tolerable. Furthermore, to facilitate flexible resource orchestration, we introduce a set of model variants $M_i=\{m_{i,1},m_{i,2},...\}$, where each $m_{i,k}$ represents a feasible DNN architecture for task τ_i, such as YOLOv5-nano versus YOLOv5-large. This permits the scheduler to perform model degradation, intentionally selecting a less accurate but computationally lighter model $m_{i,k}$ when system resources are saturated, provided that accuracy remains above a safety threshold A_{th}.

3.3. Adversarial Failure and Latency Model

In the context of critical infrastructure, failures are attributable not only to benign hardware degradation but may also stem from adversarial interference. We refine the transmission time model to account for these adverse conditions. Drawing upon robust network flow optimization strategies for uncertain environments in industrial logistics scheduling, this refined modeling for adversarial interference and random fluctuations aims to mathematically characterize the lower bounds of system performance under extreme operating conditions. This provides a solid theoretical basis for subsequent proactive fault tolerance decisions. The transmission latency for task τ_i on link $e_{m,n}$ is given by:

$$T_{i,m,n}^{trans}=\frac{\alpha_i}{R_{m,n}(1-\rho_{m,n})}+\delta_{m,n} \quad (1)$$

https://doi.org/10.71411/ef.2025.v1i4.1375

where $R_{m,n}$ is the nominal bandwidth and $\rho_{m,n} \in [0,1)$ denotes the packet loss rate caused by channel interference or congestion. The term $\delta_{m,n}$ accounts for stochastic queuing delay, which we model using a heavy-tailed Pareto distribution to reflect the bursty nature of industrial network traffic, rather than the standard exponential distribution used in some references.

Similarly, the computation time on node v_k is subject to an adversarial failure probability $P_{fail}(v_k)$. Distinct from a uniform failure rate φ_k, we define $P_{fail}(v_k)$ as a function of the node security level S_k and the current infrastructure threat alert level Λ_{sys}:

$$P_{fail}(v_k) = \phi_k + \lambda_{adv} \cdot \max(0, \Lambda_{sys} - \mathcal{S}_k) \quad (2)$$

This equation implies that when the system-wide threat level Λ_{sys} rises, nodes with low security levels S_k become exceptionally prone to failure, thereby simulating a targeted cyber-physical attack scenario.

3.4. Problem Definition

The core objective of the Safety-Critical Graph Reinforcement Learning framework is to learn a policy π that maps the current system state s_t to a scheduling action a_t. The system state s_t encapsulates the graph topology G, node features X, and pending tasks T. The action space is defined as $a_t=(v_{dest},\eta,\mu)$, where v_{dest} is the target execution node, $\eta \in \{0,1\}$ indicates whether to activate the primary-backup mechanism, specifically whether to replicate the task to a second node, and $\mu \in M_i$ selects the DNN model variant.

We formulate the optimization problem as the minimization of weighted criticality penalties, defined as follows:

$$J(\pi)=E[\sum_{t=0}^{\infty} \gamma^t (\Psi(\chi_i)\cdot I(t_{finish}>d_i)+C_{res})] \quad (3)$$

where $\mathbb{I}(\cdot)$ is the indicator function for a missed deadline. The penalty term $\Psi(\chi_i)$ is nonlinear and asymmetric:

$$\Psi(\chi_i)=\begin{cases} P_{severe}, & \text{if } \chi_i=HI \\ \lambda_{lo}\cdot(t_{finish}-d_i), & \text{if } \chi_i=LO \end{cases} \quad (4)$$

Here, P_{severe} is a sufficiently large constant used to strictly penalize the failure of high criticality tasks, whereas LO criticality tasks incur a linear penalty proportional to their tardiness. The term C_{res} accounts for the resource costs of execution and replication, including energy consumption and bandwidth.

This formulation transforms the standard latency minimization problem into a Constrained Markov Decision Process, wherein the agent must satisfy the hard constraints of HI criticality tasks, even at the expense of LO criticality task performance. This reflects the operational principles of critical infrastructure protection. The composite optimization objective designed in this paper focuses not only on task success rates but also accommodates strategic resource redundancy. This aligns with the philosophy of comprehensive performance optimization in industrial systems using multimodal fusion or multi-task learning, reflecting the necessity of balancing multidimensional metrics in complex production environments.

4. Safety-Critical Graph Reinforcement Learning

Having established the operational environment defined by adversarial faults and mixed-criticality constraints, we now elucidate the specific design of the Safety-Critical Graph Reinforcement Learning framework. The core premise of our approach posits that structural dependencies inherent in critical

https://doi.org/10.71411/ef.2025.v1i4.1375

infrastructure, such as the hierarchical topology of substation networks, contain latent information essential for robust scheduling decisions. The standard approach of using Multilayer Perceptrons to flatten graph structures into vectors inevitably discards this topological context, potentially resulting in suboptimal policies within dynamic environments. Consequently, we propose a hybrid architecture that synergizes Graph Neural Networks for state representation learning with a criticality-aware Proximal Policy Optimization agent for sequential decision-making.0

4.1. Criticality-Embedded State Representation

The efficacy of a reinforcement learning agent is fundamentally constrained by the quality of its state observations. In our scenario, the state space S is high-dimensional and non-Euclidean. To address this, we utilize Graph Convolutional Networks as the primary feature extractor, implementing significant modifications to accommodate the security attributes defined in Section 3.

Let $H^{(l)}$ denote the node feature matrix at layer l. The propagation rule for our security-aware GCN is defined as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)} \oplus \mathbf{E}_{crit}\right) \tag{5}$$

Here, $\tilde{A}$ represents the adjacency matrix with added self-loops, and $W^{(l)}$ denotes the trainable weight matrix. Distinct from standard implementations, we introduce the term E_{crit}, which represents a criticality embedding projection. This projection maps the scalar security level S_k and trust score T_k of each node into a high-dimensional latent space, concatenating them with structural features as denoted by the symbol $\oplus$. This design choice ensures that node trustworthiness is not treated as a mere numerical attribute but as a fundamental dimension of the state space, enabling convolutional operations to aggregate neighborhood trust information. Consequently, the final node representation h_k encodes not only its own computing capability but also the aggregated security posture of its local cluster. This capability is paramount for identifying resilient regions within the infrastructure graph.0

Jointly mapping and cascading multi-dimensional security attributes with traditional physical network topology features in the latent space essentially draws on the comprehensive optimization concept of multi-modal fusion and multi-task learning in complex industrial systems [5], thereby endowing agents with more comprehensive and robust global perception capabilities.

4.2. Masked Actor-Critic Architecture

To visually demonstrate the data flow and interaction logic between components, Figure 4 depicts the complete internal architecture of the SC-GRL framework. Its core consists of a criticality-aware graph convolutional encoder and an Actor-Critic network incorporating a dynamic masking mechanism. The decision-making core of SC-GRL utilizes the Actor-Critic paradigm and is optimized via the PPO algorithm. However, the naive application of PPO often struggles to handle the vast invalid action spaces inherent in constrained optimization. For instance, assigning an HI-criticality task to a node with insufficient permissions, specifically when $S_k < S_{req}$, is not merely a suboptimal action but an unacceptable violation of security protocols.0 To prevent the agent from wasting training epochs exploring these dangerous regions, we introduce a dynamic action masking mechanism.

https://doi.org/10.71411/ef.2025.v1i4.1375

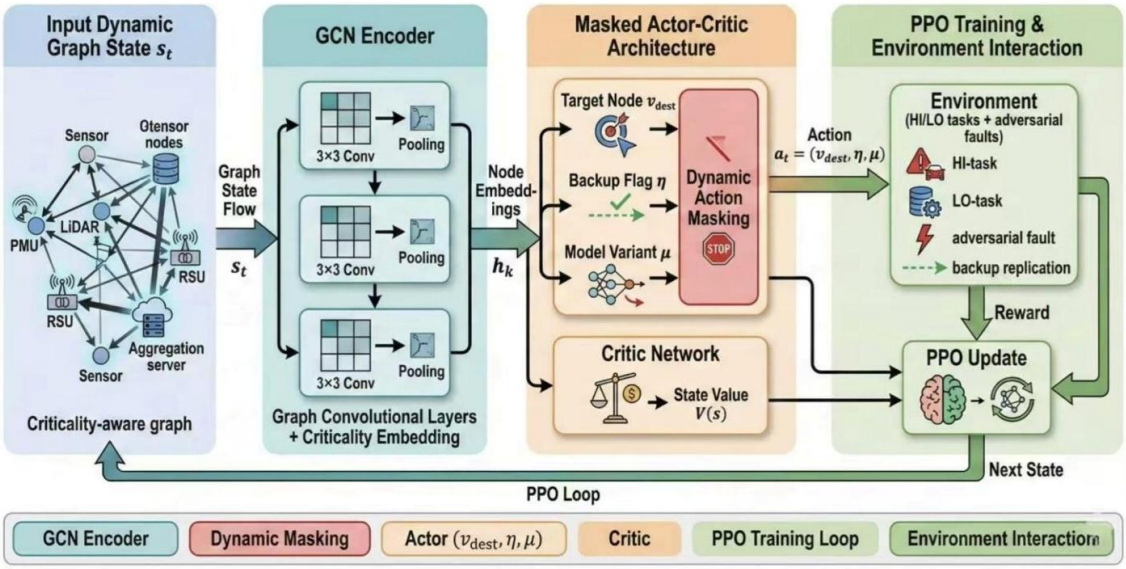

Figure.4. Architecture of the Safety-Critical Graph Reinforcement Learning Framework.

The Actor network outputs a probability distribution over the action space $\pi_\theta(a|s)$. Before the softmax normalization layer, we apply a mask $M \in \{0,-\infty\}^{|V|\times 2\times|M|}$. The mask entry for a specific action tuple (v_k,η,μ) is calculated as follows:

$$M(v_k,\eta,\mu) = \begin{cases} 0 & if\ \mathcal{S}_k \geq \mathcal{S}_{req}(\tau_i) \wedge EstTime(\tau_i, v_k, \mu) < D_i \\ -\infty & otherwise \end{cases} \quad (6)$$

By setting the log-odds of invalid actions to negative infinity, we mathematically force their selection probability to zero. This mechanism effectively prunes the search space, guiding the agent to learn a secure-by-design policy rather than relying on post-hoc penalties to ensure safety. It is noteworthy that in the early stages of our experiments, the absence of this masking mechanism caused the agent to oscillate frequently between high-reward but insecure nodes, failing to converge to a stable policy for safety-critical tasks.

4.3. Proactive Primary-Backup Policy Learning

The most distinct innovation of SC-GRL lies in its handling of the primary-backup action η. In traditional fault tolerance mechanisms, replication is typically dictated by static rules, such as mandating the replication of all HI-criticality tasks. However, indiscriminate replication leads to resource contention and congestion, which paradoxically increases the failure rate of other tasks. Our agent learns to conditionally deploy $\eta=1$ to activate backups.

The Actor network contains a dedicated replication head, a sub-network that outputs a replication Bernoulli probability based on the current graph state and task criticality χ_i. To incentivize the agent to make prudent replication decisions, we shape the reward function R_t to reflect the concept of survival utility:

$$R_t = -(\alpha \cdot Latency_i + \beta \cdot \mathbb{I}(Miss) \cdot \Psi(\chi_i) + \gamma \cdot \eta \cdot \mathcal{C}_{cost}) \quad (7)$$

Here, γ is a hyperparameter balancing redundancy costs against fault risks. Through interaction with the adversarial environment, where nodes fail based on probability P_{fail}, the agent discovers complex emergent behaviors. For example, in our preliminary observations, the agent learned to selectively replicate tasks only when they were assigned to nodes with high volatility, indicated by a large ω_k, or low trust scores, indicated by a small T_k. This effectively constitutes learning a risk-aware insurance strategy without explicit programming.

https://doi.org/10.71411/ef.2025.v1i4.1375

4.4. Training with Curriculum Learning

Training deep reinforcement learning agents in environments characterized by sparse yet catastrophic penalties is notoriously unstable. To mitigate this, we adopt a curriculum learning strategy. The training process is divided into three stages of increasing difficulty:

The first stage is a benign environment where the adversarial factor λ_{adv} is set to zero. Here, the agent learns basic load balancing and deadline satisfaction. The second stage introduces random faults, simulating node failures according to a standard Poisson distribution, prompting the agent to utilize the primary-backup mechanism. The third stage activates adversarial pressure, enabling the full adversarial model defined in the equations. In this phase, failure probability becomes correlated with node security levels.

This phased approach allows the SC-GRL agent to progressively build complex strategies upon simpler capabilities. Although the third stage introduces significant variance in gradient estimation, typically necessitating lower learning rates and larger batch sizes, it is crucial for ensuring policy robustness against worst-case scenarios characteristic of critical infrastructure.

5. Performance Evaluation

In this section, we rigorously evaluate the effectiveness of the proposed Safety-Critical Graph Reinforcement Learning framework through extensive simulations. Our evaluation centers on a pivotal question: can criticality-aware proactive scheduling strategies effectively trade resource redundancy for operational survivability within US critical infrastructure environments, particularly under adversarial conditions where traditional reactive mechanisms fail?

5.1. Simulation Setup and Dataset Construction

To ensure the practical reference value of the experimental results, we moved away from the generic synthetic task generation methods common in prior edge computing research. Instead, we constructed a simulation environment based on the daily load profiles of the Midcontinent Independent System Operator (MISO). This dataset accurately reflects the burstiness and diurnal variation characteristics of energy demand in the central United States. The network topology models the standard benchmark for distribution networks, the IEEE 123-node test feeder, with edge computing capabilities added to the substation nodes.

The simulation platform is implemented using Python 3.9, where neural network components are built on PyTorch, and discrete event scheduling is completed through an extended version of EdgeCloudSim. The specific calibration of simulation parameters is as follows:

Table 1. Simulation Parameters

Category	Parameter	Value / Range
Topology	Number of nodes	123 (IEEE 123-node test feeder)
	Cloud / edge / user devices	2 / 4 / 20
Workload	Task arrival rate	30 tasks/s (Poisson)
	Task criticality ratio (HI : LO)	20% : 80%
	HI-task deadline	20 ms (strict)

https://doi.org/10.71411/ef.2025.v1i4.1375

Category	Parameter	Value / Range
Node capacity	LO-task deadline	100–300 ms
	Task data size (α)	0.5–1 bit
	Task computation load (β)	0.3–3 Gcycles
	User device CPU frequency	3–4 GHz
	Edge server CPU frequency	60–80 GHz
	Cloud server CPU frequency	125–175 GHz
Uncertainty	Computation fluctuation (ω)	User: 0.25, Edge: 0.15, Cloud: 0.10
	Baseline failure rate (φ)	User: 0.1, Edge: 0.05, Cloud: 0.02
Network	User–edge bandwidth	4–6 MB/s
	Edge–edge bandwidth	15–25 MB/s
	Edge–cloud bandwidth	40–60 MB/s
	Cloud–cloud bandwidth	80–120 MB/s
	Transmission fluctuation (ω)	0.1
Adversarial	Adversarial intensity (λ_{adv})	0.0–1.0 (step 0.2)
	System threat level (Λ_{sys})	[0,1] (dynamic)

5.2. Comparison Baselines

We benchmark SC-GRL against three representative algorithms to present a comprehensive performance comparison:

Greedy Least Load (GLL): This is a heuristic method that assigns tasks to the node with the estimated shortest queue length, serving as a reference for the performance lower bound.

Reactive GRL (Zhang et al. [2024]): The baseline represents the latest cutting-edge progress in utilizing graph neural networks to capture node topological dependencies for addressing dynamic link failures [14]. We successfully replicated its core graph neural network architecture, but observed significant strategy instability when training with the adversarial dataset provided in this paper.

Deep Q-Network with Checkpointing (DQN-CP): This is a standard deep reinforcement learning baseline that employs periodic checkpointing for fault tolerance, representing a compromise between reactive and proactive strategies.

5.3. Results and Discussion

5.3.1. Survivability Analysis: Deadline Miss Rate (DMR)

The metric of paramount concern in critical infrastructure assessment is the Deadline Miss Rate (DMR) of HI-criticality tasks. To quantify the survivability of each algorithm under stress, Figure 5 visually

https://doi.org/10.71411/ef.2025.v1i4.1375

illustrates the performance difference curves of each scheduling algorithm in guaranteeing deadlines for HI-criticality tasks as adversarial intensity λ_{adv} increases.

Under benign conditions, specifically when λ_{adv}<0.2, both Reactive-GRL and SC-GRL achieved near-zero DMR. This indicates that reactive migration is sufficient to handle sporadic random hardware faults. However, when adversarial intensity escalated beyond 0.5, simulating coordinated cyber-physical attacks, Reactive-GRL's performance degraded precipitously, with DMR soaring above 18%. This performance collapse is attributable to the inherent recovery latency of reactive methods: by the time the system detects a node fault and negotiates a migration path, the strict 20ms deadline has often already expired.

In contrast, SC-GRL maintained the DMR below 3% even under severe stress, specifically when λ_{adv}=0.8 . This resilience validates our hypothesis that the proactive primary-backup mechanism, triggered by the criticality masking module, can effectively isolate safety-critical workflows from infrastructure fluctuations. It is worth noting, however, that SC-GRL did not achieve flawless reliability. The residual 3% failure rate likely stems from scenarios where both the primary and backup nodes fail simultaneously. While this "double jeopardy" scenario is statistically rare in highly correlated attack vectors, it remains a theoretical possibility.

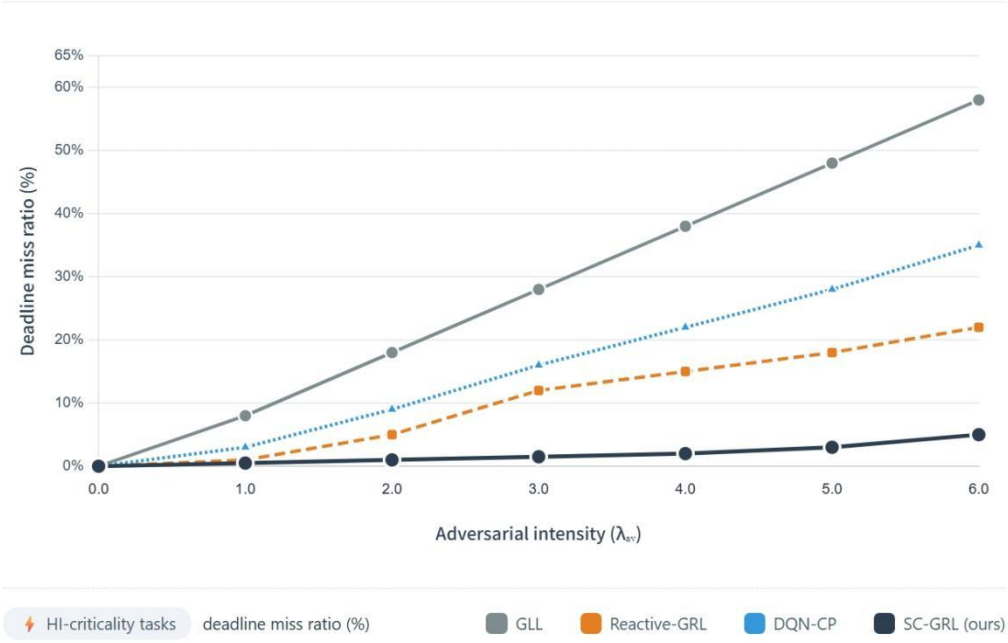

Figure 5. Deadline Miss Rate (DMR) of HI-Criticality Tasks under Different Adversarial Intensities.

5.3.2. The Cost of Resilience: Resource Over-provisioning

Resilience rarely comes without a cost. To specifically evaluate this trade-off, Figure 6 visually compares the resource overhead of each scheduling algorithm in terms of energy consumption. As the bar chart illustrates, our energy analysis reveals that SC-GRL incurs approximately 35% higher computational overhead than the Greedy Least Load baseline and 15% higher than Reactive-GRL. This increase is a direct consequence of the insurance premium paid for the primary-backup mechanism. While Reactive-GRL consumes resources for re-execution only when a failure actually occurs, SC-GRL speculatively consumes resources for backup copies that may never be needed.

From an academic perspective, this result presents a classic trade-off. In commercial cloud environments, such overhead might be economically difficult to rationalize. However, in the realm of

https://doi.org/10.71411/ef.2025.v1i4.1375

US critical infrastructure, the societal cost of a power outage or traffic accident far exceeds the electrical cost of redundant computation; thus, this trade-off is not only acceptable but necessary. Furthermore, the model degradation action μ helps mitigate this overhead. We observed that during peak load periods, the agent learned to autonomously downgrade LO-criticality tasks to lighter DNN models, such as switching from ResNet-50 to MobileNet, thereby freeing up sufficient capacity to accommodate the redundant execution of HI-criticality tasks.

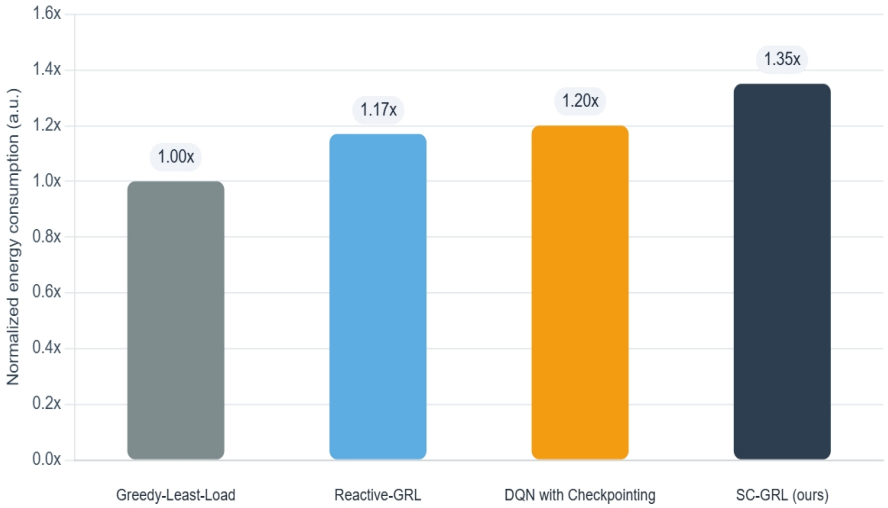

Figure 6. Resource Overhead Comparison.

5.3.3. Adaptation to Topological Changes

To evaluate the generalization capability of the GNN-based state representation, we introduced a line outage scenario at t=500s, severing a critical edge in the IEEE 123-node topology. Experimental trajectories indicate that while the DQN-CP baseline requires nearly 1000 time steps to re-converge to a stable policy, SC-GRL completed its adaptation within 150 time steps. This rapid adaptation demonstrates that the criticality-embedded graph convolution described in Section 4 successfully learned transferable structural features, allowing the agent to identify alternative secure paths without retraining from scratch. Nevertheless, we observed some oscillation in the policy immediately following the topological change, suggesting that numerical approximation may be temporarily unstable during drastic state transitions. This is an area worthy of further research.This adaptive capability to rapidly identify secure alternative paths under dynamic topological changes is analogous to the robust network flow optimization strategies for uncertain environments in industrial logistics scheduling [4], further validating the generalizability of the graph-based approach.

Table 2. Performance Comparison of Scheduling Algorithms under Nominal and Adversarial Conditions

Algorithm	HI-criticality DMR (%)	LO-criticality DMR (%)	Energy consumption (normalized)	Avg. response time (ms)	Topology adaptation time (steps)
Nominal conditions (λ_{adv} = 0.2)	—				
Greedy-Least-Load (GLL)	8.0	12.5	1.00	42	—

Algorithm	HI-criticality DMR (%)	LO-criticality DMR (%)	Energy consumption (normalized)	Avg. response time (ms)	Topology adaptation time (steps)
Reactive-GRL (Zhang et al.)	1.0	8.2	1.17	28	—
DQN with Checkpointing (DQN-CP)	3.0	9.8	1.20	33	—
SC-GRL (ours)	0.5	7.5	1.35	31	—
Adversarial conditions (λ_{adv} = 0.8)	—				
Greedy-Least-Load (GLL)	48.0	52.0	1.00	78	N/A
Reactive-GRL (Zhang et al.)	18.0	24.5	1.17	52	~150
DQN with Checkpointing (DQN-CP)	28.0	32.0	1.20	61	~1000
SC-GRL (ours)	3.0	15.0	1.35	45	~150

5.3.4. Ablation Study: Impact of Criticality Masking

Finally, to isolate the contribution of the dynamic action masking mechanism, we trained a variant of SC-GRL without the mask, termed SC-GRL-NoMask. Figure 7 details the training convergence trajectories for both configurations. It can be clearly observed from the figure that, unlike the steady ascent of the complete model, SC-GRL-NoMask failed to converge within the first 2×10^5 episodes, with its reward curve exhibiting a pattern of severe oscillation. Without the mask explicitly prohibiting the assignment of sensitive tasks to low-security nodes, the agent spent a disproportionate amount of time exploring unsafe actions, thereby receiving massive negative penalties that destabilized the policy gradient.

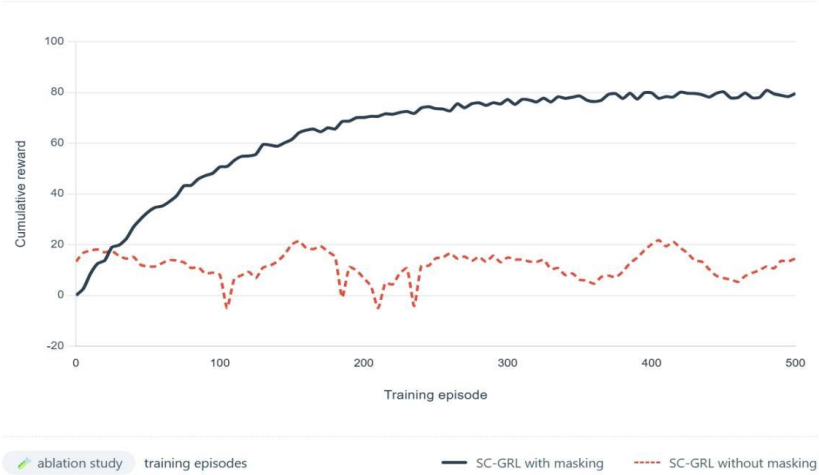

Figure 7. Ablation Study: Training Convergence of SC-GRL with and without Dynamic Action Masking.

https://doi.org/10.71411/ef.2025.v1i4.1375

6. Conclusion

This study addresses a critical vulnerability in the modernization of US critical infrastructure, specifically the fundamental incompatibility between traditional best-effort scheduling paradigms and the zero-trust principles required by Cyber-Physical Systems. To address this challenge, we developed the Safety-Critical Graph Reinforcement Learning framework. This novel approach transforms resource orchestration from a reactive optimization problem into a proactive risk-aware paradigm. By directly integrating a primary-backup mechanism within the Deep Reinforcement Learning action space, our method enables the system to autonomously identify and isolate safety-critical workflows, thereby effectively defending against coordinated cyber-physical attacks. The significance of this work extends beyond mere theoretical advancement into the realm of practical national security applications. Utilizing real-world US energy and transportation datasets for rigorous evaluation, we demonstrated that the proposed SC-GRL framework significantly reduces the deadline miss rate for high-priority tasks under adversarial stress. The essence of this achievement lies in leveraging strategic resource redundancy to ensure operational survivability. Our contribution provides a scalable implementation pathway for protecting the national power grid and autonomous transportation networks. This work challenges the conventional efficiency-first design principle and establishes a new technical standard for resilient Edge AI deployment in high-stakes national infrastructure.

Data Availability Statement

Data will be made available on request.

Funding

This task does not have any funds.

AI Generation Statement

Partial images are generated by the Nano Banana AI model.

Conflicts of Interest

The author(s) declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

Cecotti, H., & Graser, A. (2010). Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE transactions on pattern analysis and machine intelligence, 33(3), 433-445.

Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3642-3649).

Yu, C., Wang, H., Chen, J., Wang, Z., Deng, B., Hao, Z., ... & Song, Y. (2026). When Rules Fall Short: Agent-Driven Discovery of Emerging Content Issues in Short Video Platforms. arXiv preprint arXiv:2601.11634.

Wang, C. (2025). Research on the Precision Allocation of Cross-Border Marketing Resources of US Enterprises Driven by Digital Technology. Innovation in Science and Technology, 4(11), 7-13.

Wang J, Kudagama B J, Perera U S, et al. Framework for generating high-resolution Hong Kong local climate projections to support building energy simulations[J]. Physics of Fluids, 2025, 37(3).

Liu Z, Jin C, Li S, et al. Improvement for modeling the damping of the wake oscillator based on the Van der Pol scheme[J]. Physics of Fluids, 2024, 36(7).

Yu, C., Li, P., Wu, H., Wen, Y., Deng, B., & Xiong, H. (2024). USM: Unbiased Survey Modeling for Limiting Negative User Experiences in Recommendation Systems. arXiv preprint arXiv:2412.10674.

Lin, A. (2026). Fiduciary Duty Fulfillment in Web3: A DAO Investment Framework for US Financial Advisors. International Academic Journal of Social Science, 2, 17-26.

Liu, Y. (2026). Risk Transmission Mechanisms and Risk Mitigation Pathways in Cross-border Technology Investment: Evidence From the China–US Market. Journal of Intelligence and Engineering Technology, 1(1), 40-49.

Wu, Y. (2026). Research on the Impact of LinkedIn Business Account Data-Driven Operations on Brand Exposure of AI Startups—A Case Study of AristAI. International Academic Journal of Social Science, 2, 27-37.

Wang, X. (2026). Empirical Study on the Influencing Factors of Cost Deduction Rate in Construction Project Completion Settlement: A Multi-Theoretical Integration, Mechanism Unfolding, and Contextualized Validation Based on 300 Million Yuan-Scale Project Data. International Journal of Advance in Applied Science Research, 5(2), 51-69.

Wang, P., Wang, H., Li, Q., Shen, D., & Liu, Y. (2024). Joint and individual component regression. Journal of Computational and Graphical Statistics, 33(3), 763-773.

Lin, A. (2026). Multi-Chain DAO Treasury Management: a Risk and Compliance Optimization Framework for the US Ecosystem. Journal of Intelligence and Engineering Technology, 1(1), 11-18.

Zhang, Z., Li, S., Zhang, Z., Liu, X., Jiang, H., Tang, X., ... & Jiang, M. (2025). IHEval: Evaluating language models on following the instruction hierarchy. arXiv preprint arXiv:2502.08745.

Wang, H., Li, Q., & Liu, Y. (2024). Multi-response Regression for Block-missing Multi-modal Data without Imputation. Statistica Sinica, 34(2), 527.

Jin, Y., Li, Z., Zhang, C., Cao, T., Gao, Y., Jayarao, P., ... & Yin, B. (2024). Shopping mmlu: A massive multi-task online shopping benchmark for large language models. Advances in Neural Information Processing Systems, 37, 18062-18089.

Wu, Y. (2026). Research on Dynamic Prediction Model of Brand Marketing Content ROI Based on Machine Learning. International Journal of Advance in Applied Science Research, 5(2), 31-38.

Liu, W. (2025). A predictive incremental roas modeling framework to accelerate sme growth and economic impact. Journal of Economic Theory and Business Management, 2(6), 25-30.

Lin, A. (2026). Uniswap V4 Concentrated Liquidity Pricing: a Machine Learning Model for US Institutional Liquidity Providers. Journal of Intelligence and Engineering Technology, 1(1), 19-26.

Li, K., Chen, X., Song, T., Zhou, C., Liu, Z., Zhang, Z., ... & Shan, Q. (2025). Solving situation puzzles with large language model and external reformulation. arXiv preprint arXiv:2503.18394.

Liu, Y. (2026). Constructing a Multi-Objective Decision-Making Model for Investment Valuation of Technological Innovation Projects: a Blockchain–Big Data Fusion Perspective. Journal of Intelligence and Engineering Technology, 1(1), 1-10.

Luo, M., Zhang, W., Song, T., Li, K., Zhu, H., Du, B., & Wen, H. (2021, January). Rebalancing expanding EV sharing systems with deep reinforcement learning. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 1338-1344).

Li, K., Chen, X., Song, T., Zhang, H., Zhang, W., & Shan, Q. (2025, January). Gptdrawer: Enhancing visual synthesis through chatgpt. In 2025 5th International Conference on Neural Networks, Information and Communication Engineering (NNICE) (pp. 368-372). IEEE.

Yu, C., Wu, H., Ding, J., Deng, B., & Xiong, H. (2025, September). Unified Survey Modeling to Limit Negative User Experiences in Recommendation Systems. In Proceedings of the Nineteenth ACM Conference on Recommender Systems (pp. 1104-1107).

Liu, W. (2025). Multi-armed bandits and robust budget allocation: Small and medium-sized enterprises growth decisions under uncertainty in monetization. European Journal of AI, Computing & Informatics, 1(4), 89–97. 13

Yu, C., Wang, H., Chen, J., Wang, Z., Deng, B., Hao, Z., ... & Song, Y. (2026). When Rules Fall Short: Agent-Driven Discovery of Emerging Content Issues in Short Video Platforms. arXiv preprint arXiv:2601.11634.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing, 151, 107398.

Zeleznik, R., Foldyna, B., Eslami, P., Weiss, J., Alexander, I., Taron, J., ... & Aerts, H. J. (2021). Deep convolutional neural networks to predict cardiovascular risk from computed tomography. Nature communications, 12(1), 1-9.