Research Article

Joint Security- and Fault-Aware Scheduling for Critical-Infrastructure Edge AI

Daniel R. Mercer[1,*]
[1] Department of Computer Science, StanfordStanford University, 94305, USA
* Corresponding author: Daniel R. Mercer
Email: DanielRMercer@outlook.com

***Abstract*:** Currently, edge AI is used in critical infrastructure for anomaly detection, state estimation, predictive maintenance, and emergency response. However, if the scheduler treats network attacks and computational failures as irrelevant events, it can lead to situations where minimizing latency results in unreliable outputs, delayed timing, and unusability in real-world operations. Therefore, this paper proposes a joint security-aware and fault-aware scheduling framework and develops a phased evaluation scheme. This framework combines policy-based eligibility screening, risk-sensitive ranking, selective and diversity-aware replication, bounded migration, result verification, and authorized model degradation. This allows it to improve continuity without permanently achieving full redundancy.

Keywords: *Critical-infrastructure edge AI, Security-aware scheduling, Fault-tolerant computing, Real-time inference; Heterogeneous edge systems, Resilient scheduling.*

1. Introduction

Critical infrastructure increasingly depends on distributed AI inference for anomaly detection, state estimation, predictive maintenance, and emergency response. Edge deployment can shorten communication paths and preserve limited autonomy during network disruption, but it also distributes trust across resource-constrained devices whose hardware condition and cyber exposure vary over time. A result can remain statistically plausible while becoming operationally harmful if it is late, corrupted, or produced by an untrusted node.

Scheduling research has already established several foundations for this problem. Hao's latency aware multicore study treats energy and timing as coupled variables and demonstrates the value of software and hardware coordination [24]. The contribution is important because resilience actions such as replication and verification consume energy that cannot be ignored. Hao's later low overhead framework further identifies the scheduler's own computation as part of the real time cost [2]. This is especially relevant when security alerts, health indicators, and shared risk relationships enlarge the decision state.

Security and reliability are usually studied separately. Security mechanisms isolate suspicious resources, whereas fault tolerance uses replication, migration, checkpointing, or degradation. Their independent operation can conflict. Isolation reduces the capacity available for recovery, while replication to a compromised resource can create several timely but untrustworthy outputs. The central premise of this paper is that the two risks should remain distinguishable in evidence yet coordinated in action.

The research problem is how to schedule heterogeneous, deadline constrained inference when a node may be healthy, degraded, compromised, or merely suspected. Neither intrusion detection nor fault diagnosis is assumed to be perfect. The scheduler must act while evidence is incomplete, because waiting for diagnostic certainty may itself violate a protection deadline.

Four questions guide the study. How can security risk, hardware reliability, task criticality, timing urgency, and recovery cost be represented without prohibitive overhead? When should the scheduler replicate, migrate, verify, degrade, or reject work? How should contradictory security and fault evidence affect eligibility? Can an adaptive policy remain safe under unfamiliar attack and failure combinations?

The principal objective is a joint scheduling framework that coordinates trusted placement, diversity aware replication, migration, verification, and controlled degradation. The study focuses on inference tasks across heterogeneous edge nodes. It consumes alerts from external monitoring systems rather than claiming to replace intrusion detection. The threat model covers denial of service, manipulated state reports, compromised execution, and corrupted results, while the fault model covers transient, intermittent, and permanent failures.

Hao's critical infrastructure work is the closest foundation because it embeds primary backup execution, criticality, and model degradation within graph based reinforcement learning [20]. Its notable contribution is to prioritize critical task continuity over raw utilization. The present work extends that principle by asking whether a backup is sufficiently trusted and diverse. QoS aware edge allocation in telemedicine also shows that resource decisions should reflect asymmetric application consequences rather than processor use alone [23].

2. Related Work and Theoretical Foundations

2.1 Real Time, Energy, and Criticality Aware Scheduling

Average latency is insufficient for critical systems because a small tail of late protection tasks can be operationally decisive. Energy is equally ambiguous when disconnected from timely completion. Studies [24] and [2] provide a useful progression from cross layer energy and latency coordination to lightweight online micro scheduling. Their methods do not directly model adversarial trust, but they establish that resilience cannot be evaluated without implementation cost.

Dynamic prioritization for smart city edge AI offers a further contribution by incorporating evolving workload structure and service urgency [5]. The present framework adds physical consequence and integrity demand, since an urgent task assigned to an untrusted resource has not been protected merely by completing early.

2.2 Verification, Security Evidence, and Threat Identification

Zhang et al. use feature disentanglement with a variational autoencoder and a classifier to distinguish genuine signatures from skilled forgeries [6]. Although offline signature verification differs from node attestation, the study is methodologically relevant because it addresses scarce adversarial examples and small interclass separation. Its contribution supports the broader argument that verification confidence should remain explicit rather than be reduced to an unquestioned binary label.

Real time threat identification for financial APIs under federated learning provides another cross domain reference [7]. Its distributed evidence perspective is useful, but financial traffic and infrastructure control have different timing and physical consequences. Ren's risk adaptive reinforcement learning for anti money laundering review similarly illustrates dynamic prioritization under evolving evidence [8], although its decision horizon is less stringent than edge control.

2.3 Fault Tolerance, Affinity, and Structure Aware Learning

Fault tolerant scheduling in [20] motivates criticality aware backup and graceful degradation. Understanding the underlying failure modes in communication-centric edge infrastructure is equally important, as different fault types—transient, intermittent, or permanent—impose distinct recovery strategies and timing constraints [1]. Task affinity research further demonstrates that migration can disturb cache locality and introduce timing jitter [9]. This is a significant constraint on security response: moving a task away from a suspicious node is not free, and a destination should be judged by trust, health, timing, and affinity together.

Structure aware deep reinforcement learning represents tasks and heterogeneous cores relationally rather than as unrelated state entries [10]. That contribution is particularly valuable when nodes share communication paths, software images, power sources, or attack surfaces. The present framework preserves these relationships while limiting adaptive actions through mandatory safety rules.

2.4 Cross Domain Optimization, Resilience, and Deployment

Liu's submodular allocation framework models heterogeneous slots and directed spillover effects [11]. Its recommendation domain is not direct evidence for processor security, yet the relational insight is useful because assigning one replica changes the value of other candidates within the same shared risk group. Adaptive vehicle dispatch under time varying demand [13] and machine learning based logistics optimization [12] also demonstrate responsiveness to changing demand. Their larger time scales limit direct transfer, but their methodological emphasis on dynamic reallocation remains relevant.

Predictive multidimensional safeguards for billion scale platforms offer a notable layered resilience perspective [14]. Industrial multi objective process regulation [15] and collaborative energy and emissions specifications [16] show, in different operational domains, that efficiency, compliance, and safety constraints can require simultaneous rather than sequential optimization. In the domain of power networks, edge AI has also been deployed for real-time blackout prevention, where the cost of delayed or misrouted inference can be catastrophic, further motivating the need for schedulers that jointly consider timing and reliability [4]. These studies do not validate edge scheduling, but they help position the problem as constrained systems engineering.

Knowledge graph based audit response systems demonstrate how heterogeneous evidence can be organized for traceable decision support [17]. Research on commercialization of AI security technologies adds a socio technical boundary: technical performance alone may not secure adoption when regulation, capital, and organizational demand are misaligned [18]. Government guided commercialization in the space sector [19] is more remote from scheduling, yet it reinforces a limited deployment lesson concerning institutional capacity and trained operators. It is cited only for this implementation boundary, not as technical evidence for the proposed algorithm.

Existing studies contribute energy aware scheduling, lightweight decisions, criticality based backup, affinity awareness, structured learning, verification, and dynamic risk assessment. A gap remains in coordinating these elements when the cause of abnormal behavior is uncertain. The proposed framework retains separate security and health evidence, makes replication trust and diversity aware, and places adaptive ranking within explicit constraints.

3. System Model and Problem Formulation

3.1 Architecture, Tasks, and Evidence

The system contains heterogeneous edge nodes, monitoring services, a joint risk assessor, and an online scheduler. Tasks carry arrival time, deadline, estimated demand, criticality, integrity requirement, data sensitivity, and permissions for replication, migration, verification, or degraded execution. Security evidence includes intrusion scores, attestation freshness, communication anomalies, and result disagreement. Health evidence includes heartbeat loss, execution deviation, error reports, and restart history.

The two evidence channels are not collapsed immediately. An early design using one abnormality score obscured whether migration was driven by possible compromise or progressive failure. The revised design retains provenance and confidence, then combines evidence when actions are evaluated.

3.2 Threat, Fault, and Shared Risk Models

The threat model includes denial of service, manipulated utilization reports, compromised execution, and corrupted outputs. These threats are particularly acute in industrial IoT and edge computing environments, where heterogeneous devices, legacy protocols, and distributed attack surfaces create unique security challenges that have been extensively surveyed in recent literature [3]. The fault model includes transient, intermittent, and permanent failure. Nodes also belong to shared risk groups defined by common power, communication, software, administration, or location. Two replicas within one group are not treated as independent protection.

Table 1 specifies auditable evidence rather than invented performance results. Every experiment should preserve task identifiers, timestamps, policy versions, detector settings, attack and fault traces, and model configurations.

TABLE 1. Observable evidence and verification procedures

Field	Meaning	Source	Verification
Task timing	Arrival, start, and completion	Scheduler and runtime logs	Recalculate with monotonic timestamps
Node assignment	Selected core, accelerator, or node	Decision log	Cross check operating system runtime
Security alert	Suspicion and detector	Intrusion or attestation service	Retain rule, score, and timestamp

Field	Meaning	Source	Verification
	confidence		
Health alert	Evidence of degradation or failure	Heartbeat and device error logs	Repeat check or compare with injection trace
Replication	Replica destinations and risk groups	Scheduler log	Match task ID and independence attributes
Recovery	Detection through acceptable result	Scheduler, network, and verification logs	Reconstruct the complete event chain
Energy	Energy during the evaluation interval	On-device monitor or power meter	Calibrate and synchronize sampling
Ground truth	Injected attack or fault condition	Experiment controller	Compare monitoring output with injection record

Evaluation should combine periodic, variable rate, and burst intensive inference with controlled attacks and faults. The same traces are replayed across deadline oriented, trust aware, reliability aware, replication based, and unconstrained adaptive baselines. Measures include critical deadline satisfaction, accepted compromised output, recovery latency, false isolation, energy per acceptable result, migration cost, replication overhead, and scheduler decision time.

Training, validation, and testing remain separate for adaptive policies. Test scenarios include attack and failure combinations absent from training. Controlled injection provides ground truth but may simplify naturally evolving compromise, so conclusions remain conditional on the stated threat model.

4. Proposed Joint Scheduling Framework

Priority reflects deadline proximity, remaining work, waiting time, criticality, integrity demand, and execution uncertainty. Resource state distinguishes trusted, monitored, restricted, isolated, degraded, and failed conditions. A task remains intrinsically urgent even when its preferred resource becomes unsafe; risk modifies eligibility and action cost, not the task's importance.

Mandatory policy first removes resources that violate data sensitivity, compatibility, hard timing, or critical trust constraints. Moderate risk produces conditional eligibility with verification rather than automatic exclusion. Trusted placement then considers completion time, affinity, energy, queue interference, and future recovery capacity [20].

Replication is selective rather than permanent. High consequence tasks under elevated uncertainty receive diverse execution when deadline slack and capacity permit. Diversity considers physical node, communication path, software environment, and shared risk group. Result reconciliation uses application appropriate agreement because nondeterministic models may not produce bitwise identical output [21].

Queued work can be reassigned when trust or health deteriorates. Active migration is used only when state can be recovered and transfer cost is lower than restart. Approved smaller models, reduced input resolution, or early exits may preserve a critical deadline, but every degradation decision records the model version and application authorization.

A structure aware policy ranks feasible action templates after screening. Feedback reflects critical completion, integrity exposure, recovery delay, energy, migration, replication, and decision cost. Reward design is not assumed to be straightforward. Excessive security penalty can waste monitored capacity, while weak penalty can expose sensitive work [22].

Independent rules prohibit critical tasks on restricted nodes, enforce mandatory diversity, block unapproved degradation, and impose a decision time budget. If the adaptive policy fails to return a valid action, a deterministic fallback uses the most trusted feasible resource. Rule activations are logged so that apparent performance can be separated from safety intervention.

Reproducible reporting discloses hardware, operating system, monitor configuration, workload trace, random seed, attack and fault injection, ambient conditions, and every failed run. Cross platform replication remains necessary before broad generalization.

5. Conclusion

The proposed framework treats resilience as an emergent property of timing, trust, health, diversity, verification, and governance rather than as a single fault recovery statistic. Experimental analysis must determine whether reduced exposure results from genuinely coordinated risk reasoning or merely from greater reservation and task rejection, whether shorter recovery depends on spare capacity, and whether deadline improvement persists when detector accuracy falls. Ablation should remove trust screening, diversity control, adaptive replication, degradation, and fallback separately, while sensitivity analysis varies burstiness, correlated failure, adversarial manipulation, and the share of critical tasks. The wider implication is that edge AI scheduling for critical infrastructure cannot be separated cleanly into performance, security, and reliability layers, yet integration should not erase uncertainty or turn unrelated domain evidence into direct technical proof. The cited work collectively contributes useful foundations in scheduling, verification, resilience, optimization, and adoption, but the specific joint framework still requires empirical validation. Further research is needed on coordinated adversaries, scheduler compromise, hardware aging, distributed control authority, and long-term operational certification.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The author(s) declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] *Bi, S., Yuan, X., Hu, S., Li, K., Ni, W., Hossain, E., & Wang, X. (2024). Failure analysis in next-generation critical cellular communication infrastructures. arXiv preprint arXiv:2402.04448.*

[2] *Hao, Z. (2026). Low-Overhead Scheduling for Real-Time AI Workloads on Multi-Core Edge Chips. International Journal of Advance in Applied Science Research, 5(3), 15-25.*

[3] *Zhukabayeva, T., Zholshiyeva, L., Karabayev, N., Khan, S., & Alnazzawi, N. (2025). Cybersecurity solutions for industrial internet of things–edge computing integration: Challenges, threats, and future directions. Sensors, 25(1), 213.*

[4] *Curtis, M., & Price, C. (2026). Edge AI for Real-Time Blackout Prevention in Robot Power Networks. Available at SSRN 6953398.*

[5] *Hao, Z. (2026). Dynamic Task Prioritization for Edge AI in Smart Cities: Balancing Latency and Energy Efficiency. Journal of Intelligence and Engineering Technology, 1(1), 60-69.*

[6] *Zhang, H., Guo, J., Li, K., Zhang, Y., & Zhao, Y. (2024). Offline Signature Verification Based on Feature Disentangling Aided Variational Autoencoder. arXiv E-Prints. arXiv preprint arXiv:2409.19754.*

[7] *Ren, L. (2025). Real-time threat identification systems for financial api attacks under federated learning framework. Academic Journal of Business & Management, 7(10), 65-71.*

[8] *Ren, L. (2025). Reinforcement learning for prioritizing anti-money laundering case reviews based on dynamic risk assessment. Journal of Economic Theory and Business Management, 2(5), 1-6.*

[9] *Hao, Z. (2025). Task Affinity-Aware Scheduling for Multi-Core Edge Devices in Autonomous Vehicles. Engineering Frontiers, 1(2).*

[10] *Hao, Z. (2026). Structure-Aware Deep Reinforcement Learning for Latency-Minimal Scheduling of Edge AI Inference on Heterogeneous Cores. Journal of Intelligence and Engineering Technology, 1(1), 50-59.*

[11] *Liu, Y. (2026). Heterogeneous Resource Slot Optimization in Multi-Dimensional Recommendation Landscapes: A Submodular Constrained Framework with Cross-Space Spillover Effects. Journal of Progress in Engineering and Physical Science, 5(1), 26-31.*

[12] *Shengtao, L. (2025). Machine Learning-Based Logistics Network Optimization Algorithm. Academic Journal of Computing & Inform0ation Science, 8(5), 46-54.*

[13] *Huang, S. (2025). Real-time adaptive dispatch algorithm for dynamic vehicle routing with time-varying demand. Academic Journal of Computing & Information Science, 8(9), 108-118.*

[14] *Liu, Y. (2026). Cascading Resilience Through Predictive Multi-Dimensional Safeguards: System Stability Architecture for Billion-Scale Concurrent Platforms. Innovation in Science and Technology, 5(1), 35-45.*

[15] *Liu, X. (2025). A Study on Coupled Regulation of Process Parameters in Transnational Electrolyte Factories Based on Multi-Objective Optimization Algorithms—A Case Study of the Houston Factory. Journal of Progress in Engineering and Physical Science, 4(6), 5-14.*

[16] *Xinshun, L. (2026). Research on Technical Specification for Collaborative Optimization of Energy Efficiency and VOC in Continuous Chemical Production. Journal of Academic Research and Advances, 2(1), 41-51.*

[17] *Liu, X. (2025). Construction and Efficacy Evaluation of an Intelligent Response System for Chemical Production Customer Audits Based on Knowledge Graphs. Innovation in Science and Technology, 4(10), 22-28.*

[18] *Meng, S. (2026). Bridging Research and Market Adoption in Artificial Intelligence: an Investment-Driven Framework for Commercializing AI Security Technologies. Academic Journal of Sociology and Management, 4(3), 12-19.*

[19] *Meng, S. (2026). Accelerating Commercial Space Technology Commercialization Through Government-Guided Industrial Investment Funds: a Case Study of CAS Space and China's Emerging Aerospace Ecosystem. Journal of Economic Theory and Business Management, 3(2), 1-5.*

[20] *Hao, Z. (2025). Fault-Tolerant Real-Time Scheduling for Edge AI in US Critical Infrastructure. Engineering Frontiers, 1(4).*

[21] *Koteswaramma, R., & Shaik, M. A. (2026). A novel Artificial Intelligence method for Secure and Fault-Resilient Adaptive Scheduling based on Industrial Internet of Healthcare Things. Advances in Data Science and Adaptive Analysis.*

[22] *Joseph, A. (2026). Optimizing Resource Allocation and Fault Tolerance in Cloud-Based Artificial Intelligence Workloads.*

[23] *Hao, Z., Yin, M., Xu, J., Liu, Z., & Chen, Y. (2026, March). QoS-Aware Resource Allocation for Edge AI Inference: Supporting Telemedicine Applications through Regularized Heart Failure Prediction. In 2026 IEEE 8th International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 477-480). IEEE.*

[24] *Hao, Z. (2026). Energy Efficient Multi Core Task Scheduling for Real Time Edge AI Systems: A Latency Aware Approach. International Journal of Advance in Applied Science Research, 5(3), 1-14.*