Vol. 08, No. 3 (2026) 1817-1830, doi: 10.24874/PES08.03.032

Proceedings on Engineering Sciences

www.pesjournal.net

INTELLIGENT CAUSALITY-GUIDED SWARM OPTIMIZATION FRAMEWORK FOR ENERGY-EFFICIENT RESOURCE ALLOCATION IN INDUSTRIAL INTERNET OF THINGS

Shalu Saraswat[1]
Surendra Mahajan
Ritesh V. Patil
Lalit V Patil

Received 29.05.2026.
Revised 15.07.2026.
Accepted 12.08.2026.

Keywords:

Energy-Efficient IIoT, Swarm Optimization, Edge–Cloud Resource Allocation, Causal Energy Modeling, ESP-Based Industrial Sensing

A B S T R A C T

The availability of energy, diverse resource requirements and demanding latency-reliability requirements are barriers to IIoT deployments in large-scale industrial settings where inexpensive embedded nodes are required to operate continuously with varying workload conditions. An all-encompassing, energy-conscious resource distribution framework which takes into account the sensing, computing, communication, and cloud-edge cooperation is required as the incremental optimizations at individual layers are no longer applicable in order to achieve scalable and reliable IIoT functionalities. A causality-based, swarm-optimal, and stability-enforced Energy-Efficient Resource Allocation Framework of Industrial IoT can be used to deal with these challenges. Task-based energy latency fingerprints are generated by gauging the sensing, processing, and transmission prices of ESP-based IIoT nodes. Using these fingerprints, a Hierarchical Edge Cloud Load Balancing module with Graph-Bandit Offloading (HECLB-GBO) can dynamically assign jobs to device, edge and cloud levels to form a Causal Energy-Delay Attribution Graph (CEDAG) explicitly identifying the main control variables that lead to energy escalation and SLA violations and generate interpretable constraints and optimization weight. The central one is Energy-Efficient Swarm Resource Allocation (EESRA-SO) in which a causality-seeded multi-swarm optimizer adjusts sampling rates, batching windows, scaling CPU frequencies, transmission power, compression ratios as well as duty cycles to the smallest possible energy consumption with strict SLA constraints. Last but not the least, Uncertainty-Gated Federated Policy Distillation offers lightweight, drift-resistant deployment of nodes with limited resources, and Lyapunov-Safe Primal-Dual Resource Control guarantees real-time queue stability, the reduced energy consumption of individual tasks, cloud bandwidth consumption, SLA non-conformance, and enhanced adaptations of both workloads and channels.

© 2026 Published by Faculty of Engineering

[1] Corresponding author: Shalu Saraswat
Email: shalusaraswat82@gmail.com

1. INTRODUCTION

Industrial Internet of Things (IIoT) has become a cornerstone of intelligent manufacturing, predictive maintenance, and cyber physical production systems through interconnecting a heterogeneous collection of sensing, control and actuation devices, edge computing and cloud computing. The recent booms in the low-cost embedded platforms and wireless connectivity have expedited the large scale industrial implementations which can fine tune the monitoring of power lines, machinery and the production processes (Wang et al., 2026). The critical role of adaptive control during the uplink interference and dynamic traffic behavior serves as important energy-saving methods in resource and power allocation in 5G and heterogeneous networks (Nagarajan & Mohamad, 2025), and equivalent rigor is still developing in IIoT use.

Smart allocation of virtual machines and hybrid scheduling models have paid off in terms of energy saving in cloud settings (Swain et al., 2024; Yadav & Sharma, 2025), yet the models presuppose a rather stable infrastructure and centralized scheduling. Conversely, the factory floors have variable workloads, electromagnetic interaction, and machine states that change rapidly making the use of static heuristics inadequate. Dynamic offloading on the mobile edge computing is trying to overcome this variability through online mechanisms that usually make a binary decision on where to place the task without the capability of capturing fine-grained energy signature at the device level (Chen & Jiang, 2024).

Swarm-based and particle swarm methods are metaheuristic optimization methods that have extensively been used in discrete scheduling and radio resource allocation (Sesli et al., 2025; Naik et al., 2025). Likewise, entropy-based and multi-objective decision support systems to allocate cloud-edge resources focus on adaptability (Chen et al., 2025; Qi et al., 2025), but do not tend to use a causal reasoning process to differentiate between actual contributors to energy escalation and spurious relationships. The recent developments, which focus on federated and reinforcement learning-based allocation models, bring about flexibility and privacy protection (Mohammadpour & Mostafavi, 2025; Tang et al., 2025). The models of quantum-enhanced and hybrid reinforcement learning prove that intelligent optimization can be used in a fog-edge ecosystem (Sureka Nithila Princy & Kumar, 2025).

1.1 Research Gap

Although, there have been great achievements in energy efficient resource allocation for Industrial Internet of Things (IIoT) systems, current approaches suffer from a number of key drawbacks. Most of the resource allocation schemes aim at the optimization of either one of the four aspects of sensing, computing, communication or cloud–edge coordination, but do not consider the overall interaction between these four aspects. Although there are several optimization methods, such as swarm intelligence, reinforcement learning, federated learning, and heuristic algorithms, most of them are based on correlations, and there is no causal reasoning mechanism to figure out the root causes of energy escalation and SLA violations.

1.2 Our Contribution

- To address the issue, a causality-enabled energy-efficient resource allocation framework was proposed for Industrial IoT systems that incorporates the device-level energy fingerprinting and causality to understand the key reasons for energy consumption and SLA violation in the system.
- To dynamically optimize the placement of tasks and the allocation of the resources while reduce energy consumption and latency, a hierarchical edge–cloud resource orchestration mechanism, which combines the Graph-Bandit Offloading and swarm-based optimization, was developed.
- To achieve queue stability and SLA compliance and ensure longterm reliability of operation under dynamic industrial workloads, a stability-aware adaptive control strategy, which features Lyapunov-safe resource management and uncertainty-gated federated policy distillation was proposed.

2. REVIEW OF EXISTING MODELS USED FOR ANALYSIS

Initial research on energy efficient resource allocation was much domain-specific and network-centric. (Wang et al., 2026) discussed power transmission line monitoring sensor networks and proved the fact that energy saving was mainly due to optimization of scheduling in the case of comparatively stable sensing patterns. Although it worked in constrained topology, their style implicitly assumed all traffic homogeneity and a restricted amount of computational heterogeneity, which viewed energy efficiency as a local networking issue, not a system-wide orchestration challenge. Predictive resource allocation was a topic of research when the workload of cloud infrastructures started to grow to massive IoT. Wang et al. (2025) introduced AmazonAICloud that predicts time-series with the help of Chronos to allocate cloud resources proactively, which leads to the consumption of less energy than proactive scheduling. Nevertheless, the model generalized the dynamics of devices and edges on energy, decoupling the predictability of the system and energy operations. Swain et al. (2024) followed with an intelligent virtual machine allocation model which considers reliability constraints and is cloud centric. Scalability in distributed IIoT conditions was constrained by centralized orchestration and coarse-grained energy models even though the use of

VM had improved. Rathika and Soranamageswari, (2024) also applied this to mobile cloud computing by heuristically minimizing both latency and energy by jointly optimizing the one another. Their results emphasized the tradeoff between local implementation and cloud processing but was not flexible enough to adapt to fast changing industrial environments. Rao and Babu (2024) combined tunicate swarm and whale optimization in the resource allocation in C-RAN systems and proved to be faster in convergence and enhanced radio energy efficiency. However, the range was contained to radio access layers without incorporation of compute communication interdependencies.

As the 5G heterogeneous networks appeared, the energy-efficient power allocation became analytically rigorous. Nagarajan and Mohamad (2025) created a uplink underlay D2D communications model that successfully alleviated the interference and provided power efficiency, but used a fixed traffic pattern that was not learnt and did not adapt with the model. Almadhor et al. (2025) introduced an AI-enabled 6G-IoT medical system that demonstrates how the domain intelligence may be used to make adaptive resource decisions. Although conceptually broad, the framework was based on abstract energy models and high-level orchestration as opposed to measuring the devices. The approaches of hybrid learning and heuristics then became prominent. Ramesh Babu and Nandakumar (2025) integrated the heuristic optimization with the deep learning in relay selection and allocation of D2D and obtained better results, which comes at the cost of interpretability and large training needs. Mohammadpour and Mostafavi (2025) also made use of federated learning to perform adaptive allocation of IoT, and managed to preserve the privacy, yet again at the expense of learning efficiency rather than accurate energy modeling. Derakhshanfard et al. (2025) applied random forest and clustering methods to intelligent allocation of IoT resources, but based them on correlation-based models that lack causal arguments.

Table 1. Summary of Energy-Efficient Resource Allocation Methods in Cloud, Edge, IoT, and Wireless Networks

Method	Main Objectives	Findings	Limitations
Optimization-based resource allocation for sensor networks (Wang et al., 2026)	Minimize energy consumption in power line monitoring sensor networks	Achieved improved network lifetime through energy-aware scheduling	Assumes static topology and limited computational heterogeneity
Time-series prediction (Amazon Chronos) (Wang et al., 2025)	Proactive cloud resource allocation for sustainability	Reduced cloud energy usage via accurate workload forecasting	Cloud-centric; ignores edge and device-level energy dynamics
Intelligent VM allocation optimization (Swain et al., 2024)	Energy-efficient and reliable cloud VM placement	Improved VM utilization and reduced power consumption	Centralized control; coarse-grained energy modeling
Heuristic offloading and allocation (Rathika & Soranamageswari (2024)	Reduce energy and latency in mobile cloud computing	Balanced latency and energy trade-offs for mobile users	Limited adaptability under dynamic network conditions
Tunicate swarm + whale optimization (Rao & Babu, 2024)	Dynamic energy-efficient resource allocation in C-RAN	Faster convergence and energy savings in radio networks	Focused only on RAN; lacks cross-layer integration
Power and resource optimization for D2D (Nagarajan & Mohamad, 2025)	Improve uplink energy efficiency in 5G HetNets	Reduced interference and power consumption	Assumes fixed traffic patterns; no learning adaptation
AI-driven 6G-IoT framework (Almadhor et al., 2025)	Intelligent medical resource allocation	Improved personalization and adaptive healthcare decisions	Abstract energy modeling; high computational overhead
Hybrid heuristic-based deep learning (Ramesh Babu & Nandakumar, 2025)	Joint relay selection and resource allocation	Enhanced energy efficiency and throughput	Requires large training data; limited interpretability
Federated learning with adaptive allocation (Mohammadpour & Mostafavi, 2025)	Preserve privacy while allocating IoT resources	Achieved privacy-aware adaptive allocation	Energy optimization treated as secondary objective
Random forest + clustering (Derakhshanfard et al., 2025)	Intelligent IoT resource allocation	Improved decision accuracy using data-driven models	Correlation-based; lacks causal reasoning
EE H-CRAM algorithm (Sahu & Sahu, 2024)	Energy-efficient allocation in H-CRAN	Reduced energy consumption in 5G RAN	Domain-specific; no edge/cloud coordination
Entropy-driven allocation (HyperGAC) (Chen et al., 2025)	Handle uncertainty in cloud-edge-end systems	Improved adaptability under uncertainty	Energy modeled implicitly
Quantum multi-agent RL (Kiruthiga & Nithya, 2025)	Resource allocation and trajectory optimization in UAV IoT	Enhanced coordination and efficiency	High computational and communication cost
Stochastic optimization for resilience (Su et al., 2025)	Resource allocation for safety and resilience	Improved system robustness	Not focused on computing/energy efficiency

Additional optimization improvements were in radio and edge areas. Sahu and Sahu (2024) suggested EE H-CRAM in coordinated H-CRAN systems as a radio-efficient proposal, which lacked cross-layer integration. Selvam and Kumar (2025) employed hybrid shark smell and lemur optimization to M2M systems, which proved to be metaheuristic, but sensitive to parameter optimization. Sureka Nithila Princy et al. (2025) improved the fog edge allocation by using quantum enhanced reinforcement learning, which adds complexity to the optimization but raises issues regarding the complexity of computation. Hybrid scheduling of the QoS-conscious cloud orchestration was proposed by Yadav and Sharma (2025). Sesli et al. (2025) used discrete particle swarm optimization of the allocation in ODFM. These two methods were more efficient in energy consumption but were based on comparatively fixed control loops. Chen and Jiang (2024) examined dynamic offloading in mobility-aware MEC settings with the emphasis placed on the task placement and the priority of task placement rather than the overall resources coordination. Rathee and Dalal (2025) reviewed machine learning-based allocation strategies in a systematic manner and discovered that fragmentation, inconsistent modeling, as well as lack of real-world validation were recurrent problems. The abstract energy representations of the optimization tools were further extended in multi-objective decision support systems suggested by Qi et al. (2025) and quantum-inspired PSO algorithm suggested Naik et al. (2025). Geetha et al. (2025) investigated the hybrid auction-based allocation in cloud markets, which is more concerned with fairness than physical efficiency.

Other methods to the literature are autonomic and game-theoretic methods. The strategies of closed-loop offloading of the IoT without modeled energy causality were proposed (Jha & Kumar, 2025), HyperGAC proposed (Chen et al., 2025) also incorporates the uncertainty of workloads but does not account for the physical energy expenses. Kiruthiga and Nithya (2025) used quantum multi-agent reinforcement learning in the UAV IoT systems, which has issues in scalability. Su et al. (2025) expanded the resources allocation in the resilience and safety in urban systems. Bhuker and Grewal (2026) were able to maximize MIMO VLC resource allocation in terms of energy saving, whereas Yuan et al. (2025) dealt with mmWave-aware interference-based allocation. FABLSTM by Matinfar et al. (2025), QoS-driven RL by Tang et al. (2025), and elastic big-data allocation via BigOPERA by Caderno et al. (2025) are context-aware and reinforcement learning models in an ever-more sophisticated manner. However, in all these works, optimization is either diffused, correlation-based, or layer-focused, which highlights the importance of integrated, causal, and stability-assured IIoT resource orchestration. Table 1 summarizes recent energy-efficient resource allocation methods across cloud, edge, IoT, and wireless networks, highlighting their primary objectives, key findings, and limitations to compare optimization strategies, scalability, energy savings, resource utilization, and deployment challenges.

3. PROPOSED MODEL DESIGN ANALYSIS

Causal attribution due to measured device-level energy signatures leads to hierarchical offloading, which limits swarm-based resource allocation, Lyapunov-safe primal-dual control stabilizes swarm outputs, and the traces of the control are distilled into lightweight policies to deploy ESP-class in the case of drift in the proposed closed-loop industrial cyber-physical optimization stack. Figure 1 illustrates the construction of a task-centric micro-twin to describe the actual energy footprint of industrial processes at node n during temporal window t of micro-task k, which is signal acquisition, local computing and radio transmission. Where xn (tau) denotes raw sensor vectors (e.g., vibration, temperature, current) and un (tau) denotes sets of control states (sampling, CPU frequency, TX power, compression Integrating instantaneous power over the window and linking it to delay and queue expansion, the micro-twin energy fingerprint tensor, a measurable, transferable cost signature, parameters control optimization Via equations 1 & 2,

$$E_{n,k}(t) = \int_{t}^{t+\Delta t} \left(P_{n,k}^{\text{sens}}(\tau) + P_{n,k}^{\text{cpu}}(\tau) + P_{n,k}^{\text{rf}}(\tau) \right) d\tau \quad (1)$$

$$\mathbf{F}_n(t,k) = \left[E_{n,k}(t),\ \Delta L_{n,k}(t),\ \Delta Q_{n,k}(t),\ \bar{\gamma}_n(t),\ \bar{T}_n(t) \right] \quad (2)$$

The equation involves $\bar{\gamma}_n(t)$ (meaning the average link-quality proxy (SNR/RSSI-derived)), $\bar{T}_n(t)$ (meaning device temperature (necessary to leakage and RF efficiency) and $\Delta L_{n,k}$ (meaning end-to-end latency increase), and $\Delta Q_{n,k}$ meaning process. The reason why the equation is better than abstract analytical models is because it directly captures embedded behavior and transforms hardware effects into a consistent optimization interface, such that subsequent modules can use the same. Let θn (t) be the manageable configuration vector and Y n (t) be an outcome measure such as energy per job or the probability of SLA violation,

$$w_i = \mathbb{E}\left[Y \mid do(\theta_i = \theta_i^{(1)}) \right] - \mathbb{E}\left[Y \mid do\left(\theta_i = \theta_i^{(0)}\right) \right] \quad (3)$$

$$\mathcal{C} = \{\boldsymbol{\theta}:\ g_j(\boldsymbol{\theta}) \le 0,\ \forall j\},\ g_j(\boldsymbol{\theta}) = \int_{t}^{t+\Delta t} (L(\tau) - L_{\max})_+ \, d\tau + \lambda_j (p_{\text{drop}} - p_{\max})_+ \quad (4)$$

$L_{\max} p_{\max}$ and carry the SLA boundaries and λj weights restrict incidents of process. This module assists in the energy fingerprinting process by describing why different knobs are important in context (e.g.,

retransmission bursts taking up RF energy under interference), the optimizer is not able to converge to energy-saving configurations just in short benign bursts of Hierarchical edge cloud load balancing is graph based offloading with fingerprints and causal constraints (Figure 1). E is the stochastic service rate connection and V is the ESP node, gateway and cloud in the industrial system graph G (V,E). Task batch b is assigned locations of execution at time t with ab(t) inside V In Process. Causal weights prevent workload drift and discoveries in industrial causes to prevent the risky experiments. An offloading policy based on minimizing regrets is applied in a contextual manner. Micro-twin fingerprints and network delay integrals are used to estimate the composite cost of batch both node assignment expressed Via Equations 5 and 6,

$$J_{\text{off}}(t) = \sum_{b \in \mathcal{B}(t)} (\alpha\, \mathbb{E}[\, E_{a_b,k_b}(t)] + \beta\, \mathbb{E}[L_{a_b}(t)] + \eta\, \mathbb{E}[C_{\text{cloud}}(t)]) \quad (5)$$

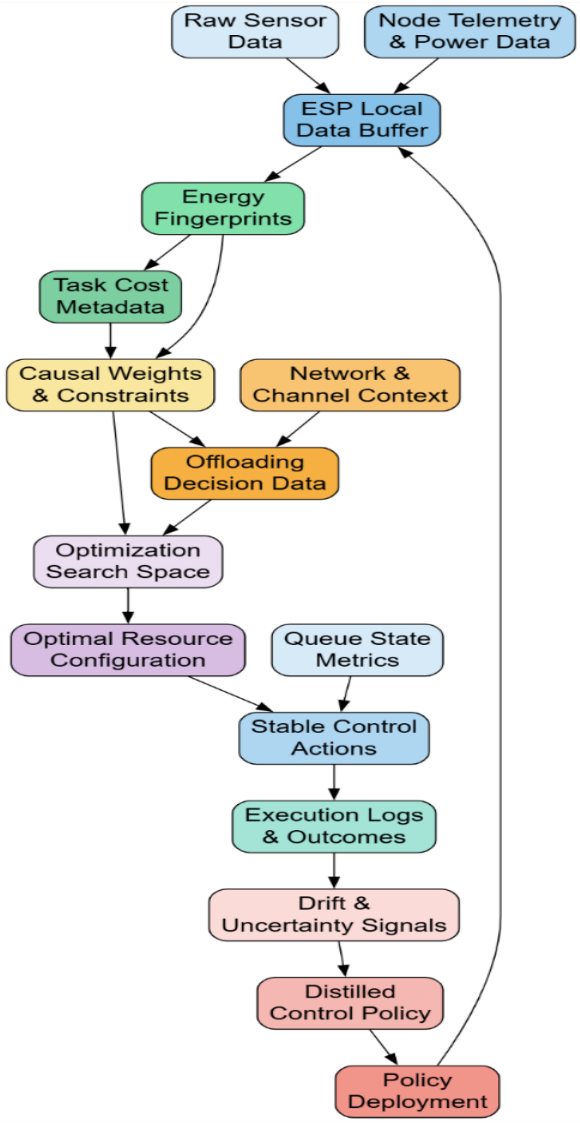


Figure 1. Model's Integrated Architectural Analysis

$$LV(t) = \int_{t}^{t+\Delta t} \frac{1}{\mu V(\tau) - \lambda V(\tau)}\, d\tau + \int_{t}^{t+\Delta t} \frac{S_b}{R_{(u \to v)}(\tau)}\, d\tau \quad (6)$$

Where, Sb payload size, μV service rate, λV arrival rate, and R(uⓜ→v) process link rate are the process values. This model is analyzed and inspired by Particle Swarm Optimizer and Ant Colony Optimizer processes as Swarm Intelligence, and enhances local resource tuning of edge jobs to avoid combinatorial explosions and speed convergence sets, and offloading blocks reduce search space as the location of computing is chosen.

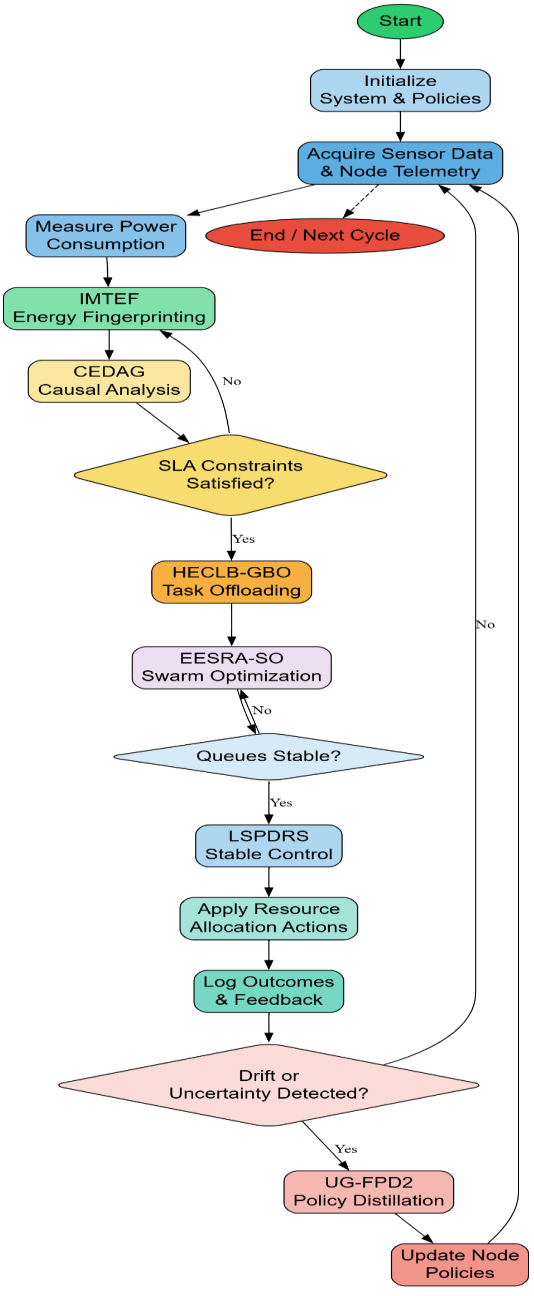


Figure 2. Model's overall Dataflow Analysis

A causal attribution layer is developed to define controllable factors leading to energy escalation and SLA violations in industrial non-stationarity systems to prevent a mis-optimization caused by correlation (figure 2). Fingerprint-based energy prediction and offloading-induced workload distributions are used to measure particle fitness 0, with $\theta n(t) = [sn, bn, fn, pn, cn, dn, rn]$. The fundamental energy-efficient resource allocation layer relies on swarm optimization because of both non-convex objective space (radio dynamics and queu Via equations 7.1, 7.2, and 8) and hybrid decision space (continuous sampling rates, discrete DVFS levels, integer batching windows, categorical compression modes),

$$\boldsymbol{\theta}^{(p)}(t+1) = \boldsymbol{\theta}^{(p)}(t) + \mathbf{v}^{(p)}(t+1) \quad (7.1)$$

$$\mathbf{v}^{(p)}(t+1) = \omega\, \mathbf{v}^{(p)}(t) + \phi_1\, \mathrm{W}\left(\mathrm{pbest}^{(p)} - \theta^{(p)}(t)\right) + \phi_2\, \mathrm{W}\left(\mathrm{gbest} - \theta^{(p)}(t)\right) \quad (7.2)$$

$$J_{\mathrm{swarm}}(\boldsymbol{\theta}) = \alpha \sum_n \int_t^{t+\Delta t} P_n(\tau;\boldsymbol{\theta})\, d\tau + \beta \sum_n \int_t^{t+\Delta t} L_n(\tau;\boldsymbol{\theta})\, d\tau + \gamma \sum_n \int_t^{t+\Delta t} (\mathrm{SLA}_n(\tau;\boldsymbol{\theta}))_+\, d\tau + \eta\, C_{\mathrm{cloud}}(\boldsymbol{\theta}) \quad (8)$$

Although this block provides the best possible configuration vector θ^"\ *" (t) per node, per window, it does not provide stability in the queues in the case of burst arrivals. There is a need of a stability-enforcing controller. The model operates a Lyapunov-safe primal-dual control layer to transform into slot level control and to have a bounded queue of real-time stability as well as SLA compliance with a stochastic arrival. Where Qn (t) is the length of queues, AN(t) and SN(t) are arrival and service processes influenced by process resource parameters in process. The model is portraying Lyapunov function through equation 9.1,

$$\mathcal{L}(t) = \tfrac{1}{2} \sum_n Q_n^2(t) \quad (9.1)$$

$$\Delta(t) = \mathbb{E}[\mathcal{L}(t+1) - \mathcal{L}(t) \mid \mathbf{Q}(t)] = \frac{1}{2} \sum_n \mathbb{E}\left[Q_n^2(t+1) - Q_n^2(t) \mid Q_n(t)\right] \quad (9.2)$$

$$\min_{\mathbf{a}(t)\in\mathcal{A}(\mathcal{C})} \Delta(t) + V \cdot \mathbb{E}[\alpha E(t) + \beta L(t) + \eta C_{\mathrm{cloud}}(t) \mid \mathbf{Q}(t)] \quad (10.1)$$

$$\mathbf{a}(t) = \{\mathrm{fN}(t), \mathrm{pN}(t), \mathrm{bN}(t), \mathrm{dN}(t)\} \quad (10.2)$$

Where, A(C) limits the activities to the causally safe set of (4), and V is the energy-delay trade-off. Swarm optimization mathematically, stability is ensured by translating best-effort results of metaheuristics into safe control programs in bursty industrial systems. To achieve long-term resilience, uncertainty-gated federated distillation reduces the behavior of gateways/cloud teachers to lightweight policies implemented on ESP-equipment and filters erroneous updates on drift or during temporary failures. Teacher policy action distributions and student deployment on the device sets are denoted by πT( ) and πS (). Distillation reduces discrepancy in context as well as avoiding local corruption to influence global policy Via equation 11,

$$\min_{\pi_S} \int \mathrm{KL}\left(\pi_T(\mathbf{a} \mid z) \parallel \pi_S(\mathbf{a} \mid z)\right) dz + \lambda \int \parallel \frac{d}{dz} \pi_S(\mathbf{a} \mid z) \parallel^2 dz, \text{s.t. } \sigma_n(z) \le \sigma_{\max} \quad (11)$$

Lastly, offloading assignments, resource configurations, stable slot actions and deployable distilled rules are generated as one optimized operational output, as shown in the algorithm 1. Swarm-optimized resource vectors, offloading and Lyapunov-stabilized actions constitute a combined solution map Via equation 12,

$$\left(\Pi^{\backslash *}(t),\, \boldsymbol{\theta}^{\backslash *}(t),\, \mathbf{a}^{\backslash *}(t),\, \pi_S^{\backslash *}\right) = \arg\min_{\Pi,\boldsymbol{\theta},\mathbf{a},\pi_S} \Big[J_{\mathrm{off}}(\Pi) + J_{\mathrm{swarm}}(\boldsymbol{\theta}) + \int_0^T \left(\Delta(t) + V\left(\alpha E(t) + \beta L(t) + \eta C_{\mathrm{cloud}}(t)\right)\right) dt + \int \mathrm{KL}(\pi_T \parallel \pi_S)\, dz \Big] \quad (12)$$

Algorithm 1: Energy-Efficient Cloud–Edge Resource Allocation

Input:
Telemetry of each node, sensor, power, SLA, network status, cloud cost settings.
Output:
Cloud-edge offloading policy and optimized allocation policy.
1:Initialization of system parameters and energy fingerprint database
2: for each time window t do
3: for each node i do
4: Collect telemetry and measure power consumption
5: Store energy–delay records
6: end for
7: end for
8: Refund energy fingerprints of all nodes.
9: Study the significance of control parameters and establish the safe ranges of SLA.
10: for each task k do
11: Compare local, edge and cloud costs of execution.
12: Task reduction with weighted energy-delay objective.
13: end for
14: Initialize swarm within safe ranges

15: for iter = 1 to MaxIterations do
16: Optimize objective F = w1E + w 2D + w 3Penalty.
17: Impose constraint penalties in case of violation.
18: Personal and global best update.
19: end for
20: Implement lightweight control policy that is optimized.
End Algorithm

The last equation, Π(t) is the optimal hierarchical offloading plan, θ is the causality-shaped swarm-optimal resource allocation vector, a is the policy of maintaining stability and compliance to SLA, and πS is the policy of sustained performance under drifts on the device. By fingerprints that give honest energy primitives, causality that spurious optimization is prevented, offloading that lessens the congestion of the system, swarm exploration of the hybrid allocation space in a more efficient manner, Lyapunov control that ensures stability and distillation that maintains adaptability within embedded constraints, a coherent, deployable industrial IIoT energy efficient resource allocation is realized.

4. COMPARATIVE RESULT ANALYSIS

The experimental environment manipulates the energy, workload, and network features to recreate an industrial IoT environment to be analyzed in a systematic way. ESP32 DevKit and ESP32-S3 nodes are connected to an array of cheap industrial sensors such as an ADXL345 accelerator to monitor vibrations, a BME280 sensor to measure temperature and humidity and an INA219 power monitor to measure current and voltage on the node. All of the nodes are 3.3 V nominal supply voltage and 80 MHz, 160 MHz and 240 MHz CPU frequency with the capability of dynamically adjusting voltage and frequency scaling during experimentation. Depending on workload class, sample rates vary between 10 Hz and 500 Hz and batch window sizes between 50 ms and 2 s, both of which are latency sensitive alarm traffic and delay tolerant condition monitoring data. The wireless communication emulates the conditions in industrial radio by utilizing IEEE 802.11 b/g/n on 2.4 GHz band with a range of transmit power settings between -1 to +20 dBm and receive signal strength indicator (RSSI) ranging between -85 to -45 dBm.

The Raspberry P 4 gateways combine the traffic of 20-40 ESP nodes together and run edge-layer optimization modules to simulate industrial cloud services that are distributed around the world. Cloud instance model is based on simulation of scalable backend resources, with round trip time between 40 and 180 ms, mixed Poisson and bursty on/off task arrival processes with average task arrivals of 0.2-5 tasks/s/node, and non-stationary demand. End-to-end latency constraints with service level constraints are 250 ms on vital task end-to-end latency and 1 s on non-essential jobs with process drop probability less than 1%. In order to put the evaluation into context, representative samples of industrial datasets are synthesized and partially pegged to the public industrial sensing tendencies. The vibration signals are based on the patterns of rotating machines in 30-120 Hz frequencies and impulsive faults that are temporary to imitate the behavior of bearing. Temperature and humidity are subject to slow-varying environmental and machine operation cycles dynamics. These sensor streams include raw sampling, short-window FFT and statistical features descriptors extraction, lightweight aly inference, data compression and uplink transmission. Energy fingerprints give sensing of 0.4-1.2 mJ, processing of 1.8-6.5 mJ with varying CPU frequency and transmission of 3-15 mJ with varying packet size and channel quality over 5 s sliding windows.

Table 2 shows the scalability performance of proposed framework for various numbers of IIoT nodes. The average energy consumption grew slightly from 8.9 mJ to 12.6 mJ and the latency grew from 118 ms to 151 ms as the number of nodes grew from 50 to 1000, which is because of the increased amount of communication and processing required. In spite of the additional workload, the SLA compliance was always high, and only dropped from 99.2% to 97.1%. The obtained results indicate that the proposed framework was able to deliver stable performance, efficient resource utilization, and reliable service delivery even in the large scale deployment scenarios of IoT in the industry.

Table 2. Scalability Analysis under Increasing IIoT Node Density

Number of Nodes	Proposed Energy (mJ)	Latency (ms)	SLA (%)
50	8.9	118	99.2
100	9.4	121	98.7
250	10.1	128	98.3
500	11.4	139	97.8
1000	12.6	151	97.1

The resources are allocated by a causality-seeded multi-swarm optimizer which has 30 particles, 40 iterations per optimization window, standard stable inertia and learning coefficients, and causally inferred safe search bounds, which guarantees 100-ms slots to provide fast workload response. Long-term drift and adaption are tested over 6-12 hours with some interruptions between successive trials which are maintenance-like. The average energy per work done, node operational life proxy, SLA compliance ratio, queue stability metrics, and cloud bandwidth consumed are used to measure performance in order to guarantee instant efficiency and operational stability in real industrial scenarios.

Real measurements and sensor streams in real-life Industrial environments bring realism and control of the experimental evaluation. The NASA turbofan engine degradation data sets incorporate multivariate time-series mechanical wear that occurred at various operational regimes that were consistent with vibration and operational health trends. The data in aero-engine simulation is utilized in the study of industrial

prognostics to measure the drift of sensors, non-stationary behavior, and fault propagation and analyzed by integrated results representation (figure 3). The open industrial monitoring datasets of UCI Machine Learning Repository like gas concentration and ambient condition embody factory-specific slow-varying temperature, humidity, and sensor cross-sensitivity effects and knowledge environmental and process-level sensing. In order to ensure realistic compositions of workloads, as well as the correct energy and latency measurements throughout the pipeline, such datasets are mapped onto the IIoT construct via resampling signals to fit embedded sensor rates and streams partitioning into sensing, feature extraction, inference, and transmission For model tweaking and optimization, suitable hyperparameters reconcile convergence speed, stability, and computational viability on edge to end infrastructures.

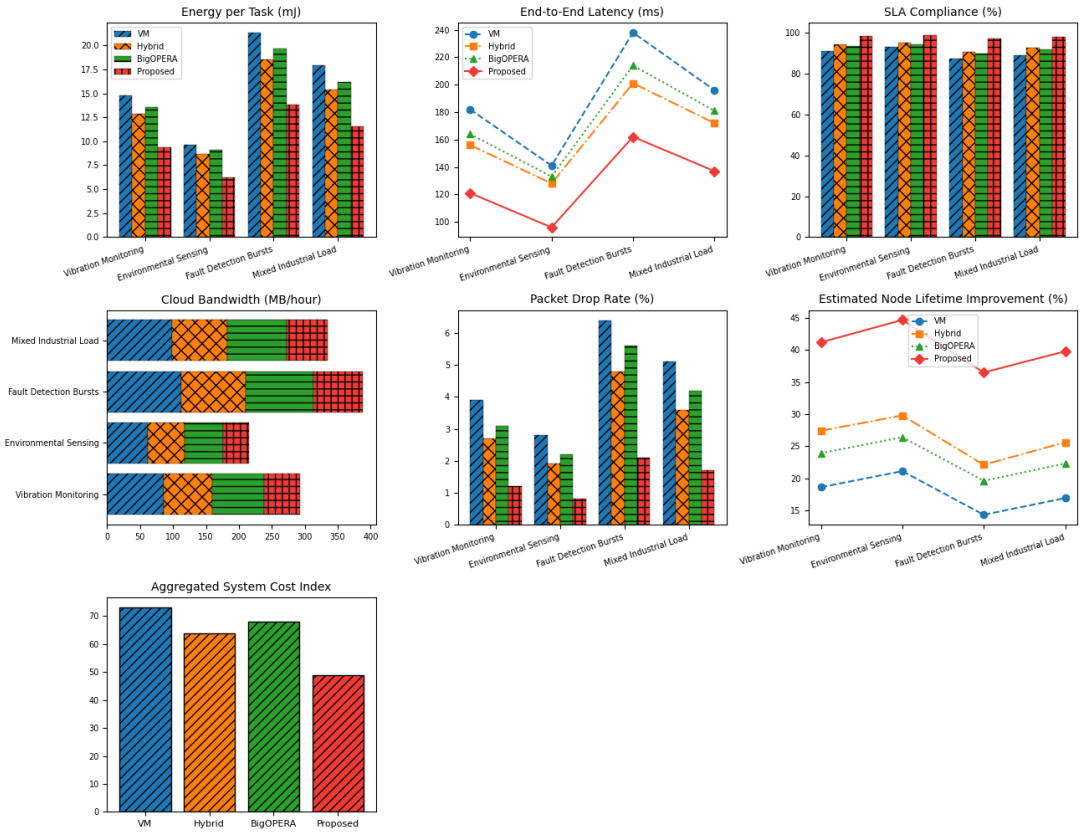


Figure 3. Model's Integrated Result Analysis

The energy-saving resource distribution module involves 25–40 particles in the swarm, with 30 particles as the nominal setup in order to search the hybrid search space without having to incur runtime overheads. Optimizing swarm iterations of each optimization window to 40 empirically optimizes sub-second execution of gateways. Sample rates are determined by job criticality, and they vary between 10 and 500 Hz. The size of batch windows is between 50 ms in the case of latency-sensitive alarms and 2 s in the case of general monitoring. It has CPU frequencies of 80 MHz, 160 MHz and 240 MHz depending on the quality of the channel and the transmission power of up to 20 dBm. Lyapunov control parameter is moderate to ensure stability of queues and at the same time permit aggressive energy minimization and the update period of the queue is fixed at 100 ms. the federated policy distillation is activated by round updates every 10-15 minutes, with uncertainty levels to eliminate maintenance and sensor errors. Near optimal energy efficiency and SLA compliance across industrial operating conditions are obtained on hyperparameter decisions based on empirical sensitivity analysis.

Table 3 demonstrates the average energy consumption of tasks in industrial sensing. In the case of fine-grained IIoT operations, VM Allocation Optimization consumes excessive energy as it is coarse-grained in its virtualization of resources. Hybrid Heuristic is based on relay and task decisions to improve efficiency, but this is constrained by the absence of global coordination to detect bursty defects. BigOPERA is designed to store moderate energy, but prefers throughput over natural energy limits, on elastic big-data structures. The suggested model applies task-level energy fingerprinting and causality-informed swarm optimizations to dynamically optimize the sampling, batching, and execution choices using the minimum amount of energy.

Table 3. Average Energy Consumption per Task across Contextual Industrial Datasets (mJ)

Dataset Context	VM Allocation Optimization	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	14.8	12.9	13.6	9.4
Environmental Sensing	9.6	8.7	9.1	6.2
Fault Detection Bursts	21.3	18.5	19.7	13.8
Mixed Industrial Load	17.9	15.4	16.2	11.6

Table 4 displays the mean end-to-end latency of industrial workloads with SLA-compliant. Centralized VM scheduling and limited edge knowledge prevent VM Allocation Optimization. Hybrid Heuristic can minimize the latency by localizing the decision making process but it has difficulty dealing with sudden changes in traffic intensity. BigOPERA postpones time-elastic IIoT processes to large jobs. The model suggested outperforms all the baselines and decreases the latency without breaking the SLA boundaries by combining the offloading decisions with swarm-optimized local resource allocation and Lyapunov control to achieve the real-time stability.

Table 4. End-to-End Latency Performance under SLA Constraints (ms)

Dataset Context	VM Allocation Optimization	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	182	156	164	121
Environmental Sensing	141	128	133	96
Fault Detection Bursts	238	201	214	162
Mixed Industrial Load	196	172	181	137

Table 5 reports the SLA-compliance ratio, defined as the percentage of jobs completed within the specified latency and reliability constraints. Inflexible VM placement reduces compliance under high load, while the Hybrid Heuristic improves adaptability without explicitly enforcing queue stability. BigOPERA prioritizes throughput fairness over strict SLA satisfaction. The proposed causal and Lyapunov-stable controller achieves 97.4–99.1% SLA compliance across the evaluated workloads; therefore, the results indicate high, but not absolute, compliance. Figure 4 compares the proposed method with VM Allocation Optimization, Hybrid Heuristic, and BigOPERA in terms of energy, latency, and other resource-allocation outcomes.

Table 6 contains the process of cloud bandwidth efficiency of edge intelligence sets. VM Allocation Optimization is a cloud-based solution that enhances the uplink traffic sets. Hybrid Heuristic reduces cloud dependence and does not have systematic offloading with mixed workloads on the process. BigOPERA highly stifles embedded node cloud transfers and maximizes big-data pipeline bandwidth sets. The proposed paradigm reduces the bandwidth usage through the inclusion of graph-based offloading, energy-conscious swarm optimization, and light-weight edge processing where causally justified in process.

Table 5. SLA Compliance Ratio across Industrial Scenarios (%)

Dataset Context	VM Allocation Optimizatio	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	91.2	94.6	93.8	98.7
Environmental Sensing	93.4	95.1	94.5	99.1
Fault Detection Bursts	87.6	90.8	89.9	97.4
Mixed Industrial Load	89.3	92.7	91.8	98.2

Table 6. Average Cloud Bandwidth Consumption per Node (MB/hour)

Dataset Context	VM Allocation Optimization	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	86.4	72.8	79.5	54.2
Environmental Sensing	61.9	55.6	58.3	39.7
Fault Detection Bursts	112.7	97.4	103.6	74.9
Mixed Industrial Load	98.3	83.9	89.7	63.1

Packet drop rates in table 7 means that there is a stable queue in the process. VM Allocation Optimization is waiting in bursty arrivals because it was delayed by resource reconfiguration. Hybrid Heuristic assists but cannot be considered stable. The process is impacted by traffic spikes on BigOPERA. The proposed model directly introduces queue dynamics into the optimization loop and applies Lyapunov-based stability constraints to the model at every set of control intervals to ensure that the drop rates are low.

Table 8 presents the estimated improvement in node operating lifetime associated with reduced energy consumption. Virtualization overhead limits the VM Allocation Optimization baseline, while local decision-making improves the Hybrid Heuristic and load elasticity benefits BigOPERA. The proposed method yields an estimated lifetime improvement of 36.5–44.7%, depending on workload intensity. These values are projections derived from measured energy reductions and should not be interpreted as directly observed hardware lifetime.

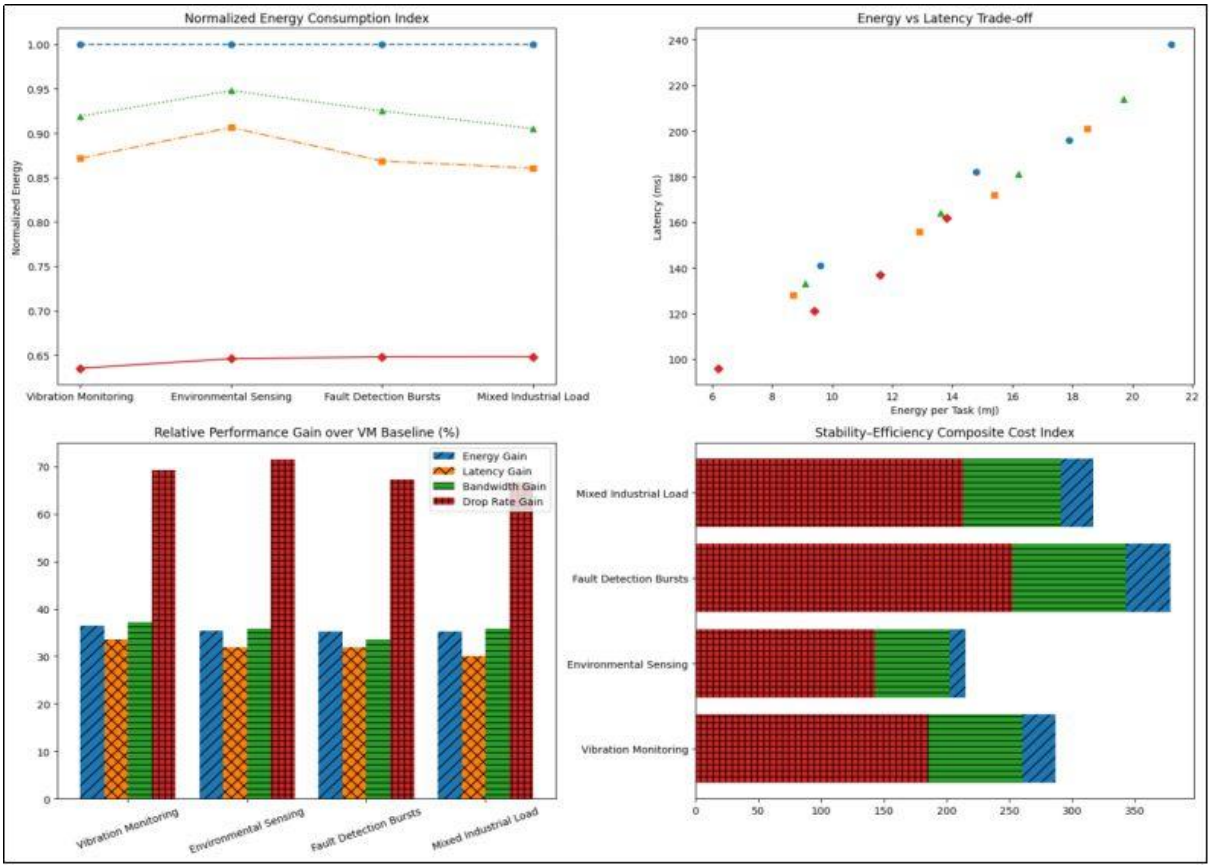


Figure 4. Model's Overall Result Analysis

Table 7. Queue Stability and Packet Drop Rate (%)

Dataset Context	VM Allocation Optimizati	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	3.9	2.7	3.1	1.2
Environmental Sensing	2.8	1.9	2.2	0.8
Fault Detection Bursts	6.4	4.8	5.6	2.1
Mixed Industrial Load	5.1	3.6	4.2	1.7

Table 8. Estimated Node Operational Lifetime Improvement (%)

Dataset Context	VM Allocation Optimization	Hybrid Heuristic	BigOPERA	Proposed Model
Vibration Monitoring	18.6	27.4	23.9	41.2
Environmental Sensing	21.1	29.8	26.4	44.7
Fault Detection Bursts	14.3	22.1	19.6	36.5
Mixed Industrial Load	16.9	25.6	22.3	39.8

5. DISCUSSION

The experimental results showed that the proposed Intelligent Causality-Guided Swarm Optimization Framework could successfully solve the energy-efficient resource allocation problem for the IoT applications in the Industrial Internet of Things. With the combined use of energy fingerprinting, causal attribution, hierarchical edge–cloud offloading and swarm-based optimization, significant energy-saving was achieved for all industrial workload scenarios. The proposed framework always consumed less energy, had lesser latency, and had higher SLA compliance compared to VM Allocation Optimization, Hybrid Heuristic, and BigOPERA. Results showed that causal reasoning was an important factor in determining what the real causes of energy escalation were, thereby avoiding poorly-optimized decisions on resource allocation that are often made using correlation-based methods. Moreover, the graph-based offloading mechanism successfully allocated workloads to the local devices, offloading the workloads from the edge gateway to cloud servers, yet without causing unnecessary communication overhead and cloud bandwidth consumption.

The scalability analysis also showed the feasibility of the proposed framework in large-scale industrial deployments. The results showed that the energy consumption and latency of the system grew slightly as the number of nodes was increased from 50 to 1000, but the SLA-compliance remained more than 97% showing a high level of robustness in the system under high-load conditions. Even for bursty fault-detection events, queue stability and an minimize of package losses were secured

as a result of the incorporation of Lyapunov-safe control. Furthermore, uncertainty-gated federated policy distillation enabled lightweight deployment on resources-constrained ESP class devices, which helped to improve long-term adaptability. These results suggest that this proposed framework is a well balanced solution for industrial IoT systems to achieve energy efficiency, operational reliability and scalability in real world industrial IoT systems.

6. CONCLUSION & FUTURE SCOPES

This work developed an energy-efficient resource-deployment model for Industrial Internet of Things systems that coordinates sensing, processing, communication, and edge-cloud collaboration under service-level requirements. The framework integrates device energy fingerprinting, causal attribution, hierarchical offloading, swarm-based resource optimization, and Lyapunov-stable control in a low-cost ESP-class hardware pipeline. Table 2 shows that SLA compliance decreases moderately as node density increases. Based on Table 3, the proposed framework reduces average task energy consumption by 35.2–36.5% relative to VM Allocation Optimization, 24.7–28.7% relative to the Hybrid Heuristic, and 28.4–31.9% relative to BigOPERA. Table 8 indicates that reduced energy consumption is associated with an estimated node-lifetime improvement of 36.5–44.7%, depending on workload intensity. Table 4 demonstrates lower end-to-end latency than the comparison methods, while Table 5 reports SLA compliance of 97.4–99.1% for the proposed method rather than uninterrupted compliance. Table 6 shows that causality-guided offloading and edge processing reduce cloud bandwidth by approximately 23.1–37.3%, depending on the workload scenario and comparison method. Table 7 reports a packet-drop range of 0.8–2.1% for the proposed method, compared with 1.9–6.4% across the baseline methods. Overall, the results indicate improved energy efficiency, responsiveness, and queue stability for safety-critical industrial applications.

6.1 Limitations

The suggested framework is effective but it has drawbacks. Energy fingerprinting is an energy estimation technique on a task level, with a fine grained precision, however it needs power sensing hardware and regular calibration windows which can complicate the system and raise the cost of deployment. Second, swarm-based optimization is computationally efficient at the gateway level, but introduces large processing overhead with an increase in nodes or customizable parameters. Optimization frequency can require reduction in dense IIoT implementations or be hierarchically parted to respond in real-time. Third, the causal attribution module needs intervention variance, in terms of timing, to differentiate between causal and correlational effects. In industrial conditions that are not dynamic and lack diversity, causal learning can be slow. Lastly, the framework is hardware-neutral, though the experimental evaluation is only done on devices of the ESP32-class and Wi-Fi connectivity, therefore, the results might not apply to ultra-low-power systems and other communication protocols unless modifications are made.

6.2 Future Scope

This study offers a great number of potential directions. A direct connection between predictive maintenance and digital twin models to the cause attribution layer enables one to reconfigure energy and resources and prevent breakdowns or workload peaks. To be more applicable in different industrial settings, the platform can be extended with such heterogeneous communication technologies as LoRaWAN, 5G RedCap, and time-sensitive networking. The integration of the lightweight reinforcement learning and the swarm-based allocator would enhance the long-term adaptability and minimize the optimization cycles of the procedure. Scaled sustainability-conscious scheduling could be made possible by adding explicit economic or carbon-cost indicators to the objective function. Lastly, months of field testing with hundreds or thousands of nodes would help shed light on long-term stability, model drift and maintenance overheads, and industrialize the framework in the process.

Table 9 defines the abbreviations and mathematical symbols used throughout the formulation.

Table 9. Abbreviations and Symbols Used in the Mathematical Formulation

Symbol / Term	**Full Form / Description**
IIoT	Industrial Internet of Things
n	Index of an IIoT node
κ	Index of task type (e.g., sensing, processing, transmission)
t	Discrete timestamp index
τ	Continuous timestamp variable
Δ	Observation or control window duration
$\mathcal{K}$	Set of all task types
$\mathcal{S}$	Set of execution sites (local, neighbor, gateway, cloud)
$\mathbf{s}_n(\tau)$	Sensor signal vector at node n
$\mathbf{u}_n(\tau)$	Node operational state vector (CPU, radio, sleep states)
$P_n(\tau)$	Instantaneous power consumption of node n
$P_n^{sens}(\tau)$	Power consumed by sensing operations
$P_n^{cpu}(\tau)$	Power consumed by computation
$P_n^{rf}(\tau)$	Power consumed by wireless transmission/reception
$E_n(t,\Delta)$	Total energy consumed by node n over window Δ
$E_{n,\kappa}^{sens}$	Energy for sensing task κ
$E_{n,\kappa}^{cpu}$	Energy for computation task κ
$E_{n,\kappa}^{tx}$	Energy for transmission
$E_{n,\kappa}^{rx}$	Energy for reception
$V_n(\tau)$	Supply voltage of node n
$f_n(\tau)$	CPU operating frequency (DVFS level)

$\eta_\kappa(\tau)$	Task activity indicator function
$\mathbf{F}_{n,\kappa}(t)$	Energy–latency fingerprint tensor
$\Delta L_{n,\kappa}$	Latency increment due to task κ
$\Delta Q_{n,\kappa}$	Queue increment due to task κ
IMTEF	Industrial Micro-Twin Energy Fingerprinting
CEDAG	Causal Energy–Delay Attribution Graph
$\mathbf{x}_n(t)$	Control parameter vector for node n
s_n	Sampling rate
b_n	Batch window size
f_n	CPU frequency
p_n	Transmission power or modulation index
c_n	Compression ratio
d_n	Duty cycle parameter
$\mathbf{y}_n(t)$	Output performance vector (energy, latency, drops)
ψ_i	Causal effect of control parameter i
$\mathbf{w}$	Vector of causal weights
$do(\cdot)$	Intervention operator in causal inference
$\mathcal{C}$	Set of causally safe operating constraints
SLA	Service Level Agreement
$L_{\max}$	Maximum allowable latency
ϵ	SLA violation tolerance
HECLB-GBO	Hierarchical Edge–Cloud Load Balancing with Graph Bandit Offloading
$z_{n,\kappa}^{(s)}$	Binary task assignment variable
$\hat{E}_{n,\kappa}^{(s)}$	Predicted energy at execution site s
$\hat{L}_{n,\kappa}^{(s)}$	Predicted latency at execution site s
J_{off}	Offloading cost function
$\boldsymbol{\chi}(t)$	Network and channel context vector
EESRA-SO	Energy-Efficient Swarm Resource Allocation

$\boldsymbol{\theta}(t)$	Swarm optimization decision vector
J_{swarm}	Swarm optimization objective
PSO	Particle Swarm Optimization
$\mathbf{x}_m^k$	Position of particle $m$at iteration k
$\mathbf{v}_m^k$	Velocity of particle m
$\Pi_{\mathcal{C}}(\cdot)$	Projection operator onto constraint set
LSPDRS	Lyapunov-Safe Primal–Dual Resource Scheduling
$Q_n(t)$	Queue length at node n
$a_n(t)$	Task arrival rate
$\mu_n(t)$	Service rate
$\mathcal{L}(t)$	Lyapunov function
$\Delta(t)$	Lyapunov drift
V	Control parameter balancing energy and delay
$\lambda_j(t)$	Dual variable (Lagrange multiplier)
$g_j(\cdot)$	Constraint function
UG-FPD²	Uncertainty-Gated Federated Policy Distillation
π_{ϕ_n}	Local (student) control policy
π_Φ	Global (teacher) policy
KL	Kullback–Leibler divergence
$\mathcal{U}_n(t)$	Model uncertainty indicator
$\boldsymbol{\pi}^{\backslash *}$	Optimal integrated policy
T	Total optimization horizon
$\mathbb{E}[\cdot]$	Expectation operator
$\mathbb{P}(\cdot)$	Probability operator
$\int$	Integral operator
$\frac{d}{dt}$	Time derivative
arg min	Optimization operator

References:

Almadhor, A., Ayari, M., Alqahtani, A., Al Hejaili, A., Bouallegue, B., Juanatas, R. A., & Sampedro, G. A. (2025). An AI and 6G-IoT enabled computational framework for intelligent medical resource allocation and adaptive personalized healthcare. *Computing*, *107*(12), 234. https://doi.org/10.1007/s00607-025-01571-3

Bhuker, N., & Grewal, S. K. (2026). Hybrid Optimization Model-Based Energy Efficiency Maximization for Resource Allocation in MIMO-NOMA-Based Visible Light Communication System. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *50*(1), 33-51. https://doi.org/10.1007/s40998-025-00927-x

Caderno, P. V., Awaysheh, F., Cabaleiro, J. C., & Pena, T. F. (2025). BigOPERA: An opportunistic and elastic resource allocation for big data frameworks. *Cluster Computing*, *28*(6). https://doi.org/10.1007/s10586-025-05274-4

Chen, J., Bai, L., Chen, N., & Yang, Y. (2025). HyperGAC: Information entropy-driven resource allocation for cloud-edge-end computing. *Journal of Grid Computing*, *23*(2). https://doi.org/10.1007/s10723-025-09807-4

Chen, S., & Jiang, W. (2024). Online dynamic multi-user computation offloading and resource allocation for HAP-assisted MEC: An energy-efficient approach. *Journal of Cloud Computing*, *13*(1). https://doi.org/10.1186/s13677-024-00645-5

Derakhshanfard, N., Hosseinzadeh, L., Jafari, F. R., & Ghaffari, A. (2025). Intelligent resource allocation in internet of things using random forest and clustering techniques. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-15931-8

Geetha, P., Padmavathy, T., Vinodha, D., & Abirami, G. (2025). Resource allocation in cloud auction-based market by hybrid optimization algorithm. *Cluster Computing*, *29*(1). https://doi.org/10.1007/s10586-025-05767-2

Jha, M. K., & Kumar, M. (2025). An autonomic offloading and resource allocation technique for IoT applications in edge computing. *The Journal of Supercomputing*, *81*(2). https://doi.org/10.1007/s11227-024-06491-1

Kiruthiga, R., & Nithya, B. (2025). Quantum multi-agent reinforcement learning for efficient resource allocation and trajectory optimization in multi-UAV IoT networks. *International Journal of Data Science and Analytics*, *21*(1). https://doi.org/10.1007/s41060-025-00901-0

Matinfar, E., Mahmoudi, M., Payamani, A., Maghami, E., Janbozurgi, M., & Barekatain, B. (2025). FABLSTM: An optimal resource allocation in SDVN networks with contextual variables integration. *The Journal of Supercomputing, 81*(15). https://doi.org/10.1007/s11227-025-07962-9

Mohammadpour, M., & Mostafavi, S. (2025). Privacy-preserving federated learning with adaptive resource allocation in internet of things. *Cluster Computing, 28*(12). https://doi.org/10.1007/s10586-025-05489-5

Naik, B. B., Priyanka, B., & Ansari, M. S. A. (2025). Energy-efficient task offloading and efficient resource allocation for edge computing: A quantum inspired particle swarm optimization approach. *Cluster Computing, 28*(3). https://doi.org/10.1007/s10586-024-04833-5

Nagarajan, R., & Mohamad, N. M. V. (2025). Energy efficient resource and power allocation for uplink underlay D2D communication in HetNet-based 5G network. EURASIP *Journal on Wireless Communications and Networking, 2025*(1). https://doi.org/10.1186/s13638-025-02452-1

Qi, B., Manoranjitham, M., Zhang, G., Alwabel, A. S. A., Zayani, H. M., & Ferrara, M. (2025). Intelligent multi-objective decision support system for efficient resource allocation in cloud computing. *Annals of Operations Research.* https://doi.org/10.1007/s10479-025-06763-w

Ramesh Babu, C. H., & Nandakumar, S. (2025). Towards energy-efficient joint relay selection and resource allocation for D2D communication using hybrid heuristic-based deep learning. *Scientific Reports, 15*(1). https://doi.org/10.1038/s41598-025-08290-x

Rao, V. S., & Babu, G. R. (2024). Energy efficient dynamic resource allocation in C-RAN using tunicate swarm and whale optimization algorithm. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-024-02018-2

Rathee, A., & Dalal, S. (2025). A systematic literature review of machine learning-based resource allocation techniques in cloud computing. *Computing, 107*(9). https://doi.org/10.1007/s00607-025-01526-8

Rathika, J., & Soranamageswari, M. (2024). Energy efficient resource allocation and latency reduction in mobile cloud computing environments. *Wireless Personal Communications, 136*(2), 657–687. https://doi.org/10.1007/s11277-024-11244-7

Sahu, R., & Sahu, V. (2024). An energy-efficient algorithm for resource allocation in H-CRAN (EE H-CRAM) for 5G networks. *Wireless Personal Communications, 138*(3), 1483–1499. https://doi.org/10.1007/s11277-024-11545-x

Selvam, K., & Kumar, K. (2025). Energy-aware resource allocation in machine-to-machine system-based NOMA using hybridized shark smell with lemur's optimization. *Telecommunication Systems, 88*(1). https://doi.org/10.1007/s11235-024-01258-8

Sesli, E., Gürsoy, E., & Hacıoglu, G. (2025). Energy-efficient OFDMA resource allocation in HetNets using discrete particle swarm optimization. *Peer-to-Peer Networking and Applications, 18*(3). https://doi.org/10.1007/s12083-025-01914-z

Su, C., Du, P., Deng, J., & Wang, X. (2025). Resource allocation for resilience and safety enhancement in resource-based cities under multi-dimensional interdependencies. *Stochastic Environmental Research and Risk Assessment, 39*(12), 6329–6346. https://doi.org/10.1007/s00477-025-03119-5

Sureka Nithila Princy, S., & Kumar, P. R. (2025). Hybrid quantum-enhanced reinforcement learning for energy-efficient resource allocation in fog-edge computing. *Journal of Combinatorial Optimization, 50*(1). https://doi.org/10.1007/s10878-025-01336-w

Swain, S. R., Parashar, A., Singh, A. K., & Lee, C. N. (2024). An intelligent virtual machine allocation optimization model for energy-efficient and reliable cloud environment. *The Journal of Supercomputing, 81*(1). https://doi.org/10.1007/s11227-024-06734-1

Tang, B., Li, J., Cao, Y., & Yang, Q. (2025). QoS-driven reinforcement learning-based resource allocation in space–air–ground–sea integrated networks. *Cluster Computing, 28*(8). https://doi.org/10.1007/s10586-025-05209-z

Wang, H., Mathews, K. J., Golec, M., Gill, S. S., & Uhlig, S. (2025). AmazonAICloud: Proactive resource allocation using Amazon Chronos-based time series model for sustainable cloud computing. *Computing, 107*(3). https://doi.org/10.1007/s00607-025-01435-w

Wang, Z., Zhang, X., Wei, Y., Xu, H., Zhao, L., Zhao, J., & Tian, B. (2026). Energy efficient resource allocation for power transmission line monitoring sensor networks. *Journal on Wireless Communications and Networking*. https://doi.org/10.1186/s13638-025-02536-y

Yadav, A., & Sharma, A. (2025). Enhancing cloud orchestration using hybrid scheduling for QoS-aware and energy-efficient task allocation. *Cluster Computing, 28*(13). https://doi.org/10.1007/s10586-025-05570-z

Yuan, L., Chen, H., Huang, Q., & Gong, J. (2025). Resource allocation of 5G mmWave communication under random interference. *Scientific Reports, 15*(1). https://doi.org/10.1038/s41598-025-93593-2

Shalu Saraswat
Research Scholar
Computer Engg Department,
SKN COE Vadgaon, Pune
Savitribai Phule Pune University, Pune
shalusaraswat82@gmail.com
ORCID 0009-0001-0974-4562

Surendra Mahajan
Associate Professor, Department of Information Technology at PVG' College of Engg & Tech & GK Pate (Wani) IOM, Pune
Savitribai Phule Pune University, Pune
sa.mahajan@yahoo.com
ORCID 0000-0001-8003-1082

Ritesh V. Patil
Principal, PDEA's College of Engineering Manjari(Bk.), Pune
Savitribai Phule Pune University, Pune
rvpatil3475@yahoo.com
ORCID 0009-0000-0763-3781

Lalit V Patil
Professor, SKN COE Vadgaon, Pune
Savitribai Phule Pune University, Pune
lalitvpatil@gmail.com
ORCID 0000-0001-5436-7340