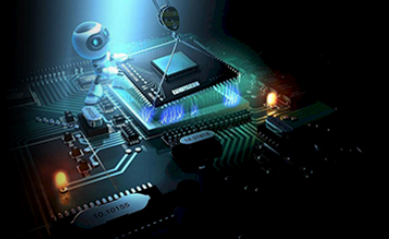


International Journal of Engineering in Computer Science



ISSN Print: 2663-3582
ISSN Online: 2663-3590
IJECS 2026; 8(4): 10-17
www.computersciencejournals.com/ijeecs
Received: 06-07-2026
Accepted: 01-09-2026
Published: 17-10-2026

The names and affiliations of all authors are provided below, following the references.

Lightweight explainable AI for real-time resource allocation in heterogeneous edge computing networks

Deena Nath Gupta¹, Ashish Shukla², Navneet Tyagi³, Satya Pal Singh⁴, Shivoham Tiwari⁵, Divya⁶, Monu Kumar⁷, Meduri Sridhar Sarma⁸, Aruna Malik⁹, Vinay Shankar Dubey¹⁰

DOI: <https://www.doi.org/10.33545/26633582.2026.v8.i4a.317>

ABSTRACT

Heterogeneous edge networks put computing close to users and sensors, but with strict latency, energy and reliability requirements, resource managers need to decide where workloads execute, where to allocate radio resources and compute resources, and when to move the workload to another node. Learning-based policies are able to adjust to varying demand, but they have been hard to audit and can be too complex to explain or infer by resource-limited devices. This review brings together the basic concepts of edge computing technologies, optimization-based allocation, reinforcement learning approaches, explainable AI and efficient inference research. It marks a gap in co-design: Allocation performance, explanation fidelity, and explanation cost are frequently considered a separate and isolated concern. We present an explainable resource-allocation architecture, including a non-blocking explain path and a fast control path, which is light weight. The fast path contains compact state representations, the constraints, and the feasibility shield; the explanation path contains the most important state factors, the constraints that were still kept active, the alternatives that were discarded, and the counterfactuals that were feasible and could be acted on. An evaluation protocol derived from a review is given for comparing the latency, deadline miss, energy, fairness, policy cost, explanation fidelity and operator usefulness. The paper also does not make any claim of experimental validation of the proposed architecture. It ends with research questions concerning temporal explanations, coordination with multiple agents, uncertainty, privacy, and reproducible benchmarks.

Keywords: Edge intelligence, Mobile edge computing, Resource allocation, Task offloading, Explainable artificial intelligence, Reinforcement learning, Real-time systems, Heterogeneous networks

1 Introduction

The idea behind edge computing is to get the computing and storage closer to sensors, mobile users, vehicles, and industrial equipment. It's not a tiny cloud, but it is about minimizing the distance between where data is created and where it is used and the service is executed. There are variations in the types of processors, the availability of the accelerators, the amount of memory, the availability of energy, radio conditions, and administrative control of the nodes. There's also a variation in workloads, such as sustained inference for a camera stream, a hard deadline for a control workload, or partial offloading of a mobile application. Thus, computation, communication, storage and placement of edge resources need to be coordinated across the ever-changing collection of devices (Shi *et al.*, 2016) [1]–(Mach *et al.*, 2017) [3].

Previous research had shown that edge performance is not only compute scheduling based. Previous work had shown that edge performance is not compute scheduling only based. Offloading, allocation and provisioning are clearly distinguished in surveys, and, optimization studies utilize both wireless resources and computational resources (Luo *et al.*, 2021) [4], (Chen *et al.*, 2016) [9]–(Yang *et al.*, 2018) [16]. Later learning-based methods tackle the challenge of making repeated decisions for dynamic and/or combinatorial scenarios, such as multi-user decisions (Xiong *et al.*, 2020) [19]–(Liu *et al.*, 2019; Liu *et al.*, 2023) [21, 31]–(Baek *et al.*, 2021) [33], vehicular decisions (Chen *et al.*, 2019) [22]–(Yan *et al.*, 2020) [24] and task graph decisions (Bi *et al.*, 2021) [25]. While their results are sufficient for the adaptive allocation to be a good design direction, there are other requirements in a production system. A controller has to comply with a period, within which the service is to be executed, hard capacity and safety limitations, and the reasons for choosing an allocation, when operators inquire about the reason for a missed service target.

Explainability is important because the allocation of edges to services has impact on the observable outcomes of services, such as delay, energy consumption, fairness, and task completion. A deep policy could select a nearby node because the path is clogged, or decline a fast node accelerator due to its filled queue, or opt to keep work local because the energy of the transmissions outweighs the energy of the faster node accelerators. When the controller only identifies an action identifier, operators cannot easily tell where a rea

Corresponding Author:

Deena Nath Gupta
Associate Professor,
Department- Computer
Science and Engineering,
College name- Rayat Bahra
University, V.P.O, Tehsil,
Sahauran, Kharar, Punjab
140104, India.

sonable course of action or trade-off is coming from, or if it's a stale state estimate, a model failure or an infeasible action. The Explainable AI (XAI) provides means to explain the factors behind the decisions and the action of the models (Ribeiro *et al.*, 2016) [37] and (Arrieta *et al.*, 2020) [38] and recent studies have used explainability in Federated Resource Management and Federated Edge Intelligence (Patni *et al.*, 2024) [34] – (Nguyen *et al.*, 2025) [36]. The strands are interrelated although not routinely measured in the same manner as allocation policies, using real-time and resource measures. This paper is a critical integrative review of that place of intersection. It brings four contributions to the systems field: (1) a systems-level formulation that considers allocation, explanation, and meeting deadlines as related issues; (2) a taxonomy that correlates the allocation methods with the kind of explanation evidence they can provide; (3) a light-weight architecture that separates the time-critical control loop from the richer explanatory information; and (4) an evaluation protocol that yields measurable outcomes of performance, fidelity, and usability. Architecture is a design proposal that is found from literature, not tested algorithm. The paper is designed for guidance in implementation and experimentation, albeit not as a means to ensure trustworthiness and safe behaviour by itself.

II Review scope and method

The review is organized into the following bodies of peer-reviewed works: edge and mobile-edge foundations and surveys (Shi *et al.*, 2016) [1]– (Xu *et al.*, 2021) [8], optimization and online resource allocation (Chen *et al.*, 2016) [9]– (Liu *et al.*, 2024) [18], learning-based allocation and orchestration (Xiong *et al.*, 2020) [19]– (Baek *et al.*, 2021) [33] and XAI, explainable edge applications, and efficient inference (Patni *et al.*, 2024) [34]– (Deng *et al.*, 2020) [40]. The list of sources covers the period 2015–2025 and the sources are comprised of surveys, journal articles and a base conference paper. We grouped the sources based on the decision problem they are solving, the learning or optimization mechanism they employ and the evidence they give regarding operational cost or explanation. The synthesis focuses on the nodes and workloads that are not uniform, but rather have consideration for nodes that are time sensitive.

This is an integrative critical review, not systematic review or quantitative meta-analysis as per the PRISMA guidelines. The performance measures listed in the cited studies are quite diverse and require a different workload model, network assumptions, objective functions, simulators, and baseline so it is misleading to pool the values of the performance measures listed in the cited studies. The review is not reported as the number of screens counted or is not exhaustive. It has been created with the aim of comparing design assumptions, establishing a set of dimensions that can be reproduced in future empirical research purposes and to develop a framework that is available for future empirical research to validate. In the tables, the designator representative means that the papers within the range cited are representative of research clusters rather than that all the papers within the range are conducted in the same manner or yield the same outcome.

III Edge resource allocation problem

A Heterogeneous system model

The time varying graph of end devices, access points, edge servers, and cloud resources can be used to represent a practical edge system. A capacity vector, that defines the CPU cycles, the accelerator throughput, memory, storage and available energy of each compute node, is defined. A capacity vector, which specifies the number of CPU cycles, the number of accelerator cycles, memory, storage, and available energy of each compute node, is defined. Network links have bandwidth, propagation delay, loss and may be of different costs. Tasks are received with characteristics like input volume, compute necessities, deadline, dependence construction, class of confidentiality and minimum service quality. Tasks come with features like input volume, compute requirements, deadline, dependency structure, class of confidentiality and minimum service quality. There are cer-

tain tasks which can be divided into subtasks, and some which require to be performed as part of a single task or a directed acyclic graph of tasks. Observed tasks and node properties may not be exactly as they are expected: Queues can be delayed, wireless quality can vary and devices can come or go. The architecture surveyed in edge intelligence and resource scheduling research is heterogeneous (Luo *et al.*, 2021; Zhou *et al.*, 2019; Xu *et al.*, 2021) [4, 6, 8] which is the core of the architecture.

B Decision variables and objectives

B Decision variables and objectives.

The controller can choose a serving node at every decision epoch, determine whether to offload or not, set the CPU or accelerator share, reserve bandwidth and transmit power, order tasks and decide whether to replicate or migrate a service. A weighted sum of latency and energy are typically minimized in the classical formulations under the constraint of radio, compute and task (Chen *et al.*, 2016) [9]– (Yang *et al.*, 2018) [16]. Throughput, Quality of experience, utilization, cost, reliability and fairness are other objectives (Wu *et al.*, 2021; Liu *et al.*, 2024; Alqerm *et al.*, 2021) [17, 18, 23]. There can be bad compromises that are hidden in a single weighted sum, such as a small reduction in the average latency that can be accompanied by a large number of deadline misses for one service class. A good design will therefore identify what outcomes are hard constraints, what outcomes are optimization goals and what outcomes are monitored as distributional safeguards.

C A review level formulation

Let y_t be the observed state of the system, a_t be the allocation action and z_t be the uncertain task and network conditions. One general goal over horizon T is to minimize the expected total of weighted task latency, energy, cost when the task is not completed on time, cost of migrating a task from its origin node to destination node, and cost of contention among tasks for a given node, while satisfying per-node capacity constraints and per-task feasibility constraints. A lightweight XAI system adds an extra explanation budget B_e that could be defined by time, memory or energy for the explanation generation and a quality criterion for the evidence that is associated with a decision. However, we do not assume that the quality of explanation is automatically expressed by a scalar reward and instead formulate this as a constrained design problem. A policy is a function $a_t = \pi(x_t)$, whereas an explanation function produces $e_t = g(x_t, a_t, \pi, c_t)$ containing active constraints and the version of the model. These proposed equations form a design vocabulary, which isn't an empirical model that's fitted to a specific network.

The use of cash and/or food to determine how resources are prioritized.

A optimization and heuristic control

A optimization and heuristic control.

Optimization-based allocation gives explicit objectives, constraints. Large problems may be more tractable by decomposition and/or convex relaxation, or be provably more efficient given certain assumptions using online heuristics (Sardellitti *et al.*, 2015) [10]– (Yang *et al.*, 2018) [16], (Bi *et al.*, 2021) [25]. They have the advantages of interpretability of variables and clear relationship between constraints and decisions. This allows them to be used as a baseline and for explanation - the controller may report a bandwidth limit, a queue constraint or an energy trade-off. Their limitations are they aren't successful if the state is partially obscured, their model isn't correct, or the number and type of edge nodes increases. Solving a large optimization problem repeatedly might exceed a real-time decision budget, particularly if the number of tasks, and link conditions (or the rate of change) increase quicker than the solver can converge.

C Social networks and the impact on communities

Reinforcement learning models sequential resource decisions as a policy, which takes in action a state and then gets a reinforcement. The published works use deep reinforcement learning to represent

nonlinear relationships, and adapt to the dynamics of workloads; some examples include IoT offloading, virtual edge systems, vehicle networks, task graphs, augmented reality and task orchestration (Xiong *et al.*, 2020) [19]–[24], (Chen *et al.*, 2021) [27]– (Sheng *et al.*, 2021) [29], (Liu *et al.*, 2023) [31]– (Baek *et al.*, 2021) [33]. Multi-agent policies are policies which are spread across devices, or edge nodes, and federated approaches aim to coordinate learning without centralizing all data (Liu *et al.*, 2020; Yu *et al.*, 2021) [20, 26]. These techniques may enhance adaptability, but their operational characteristics are a function of state design, reward scaling, exploration control, training data, and relationship between the simulated network and deployed network. A high-reward policy can still result in inappropriate actions, unstable policy when distribution shift, or hard-to-explain policies to a network operator.

A lightweight constraint, C Real-time control is a flexible, yet powerful constraint. C Real-time control is a flexible yet powerful constraint

The low average inference latency is not the only thing that is real-time allocation. A control decision has to be completed before the time of scheduling or application and the tail of the distribution of the latency matters. The performance of the models is in competition for the same CPU, accelerator, memory and energy that is used to execute user tasks. The cost of distributed decisions and large action spaces are explicitly considered as system concerns in the development of lightweight reinforcement learning for vehicular edge allocation (Wu *et al.*, 2025) [30]. Research on efficient inference demonstrates that af-

ter deployed, it is important to verify the accuracy and robustness of compression and hardware acceleration (Sze *et al.*, 2017; Deng *et al.*, 2020) [39, 40] to reduce the cost of the model. The objective is the entire decision process for the proposed class of systems, namely, telemetry acquisition, construction of the state, inference of the policy, feasibility checks, dispatch, and possibly any work of explanation that prevents action. Explanation design needs to be designed within that pathway and budgeted as part of that pathway.

V Explainability for edge allocation

The purpose of XAI methods is to respond to various queries and thus cannot be considered to be the same. Global explanations explain the overall policy behaviour, for example, which state features are affecting allocations over a workload distribution. Local explanations provide explanations of one decision, i.e., why the task i was given in node j at time t ? Feature-attribution methods provide estimates of the values of inputs that most affected an output; surrogate models can help approximate a policy with a simpler local rule; counterfactuals can help identify state changes that might lead to a different action; and traces of the rules or constraints can be used to show which explicit conditions admitted or denied candidate actions. The LIME method is a primitive illustration of local model-agnostic explanation (Ribeiro *et al.*, 2016) [37] and more comprehensive XAI taxonomies explore ideas, opportunities and limitations (Arrieta *et al.*, 2020) [38]. When it comes to allocate resources, it's far better to have action level evidence than a general statement that "CPU load was important."

Table 1. Research clusters and their implications for the proposed design

Research cluster	Representative evidence	Strength for allocation	Remaining issue for an explainable real-time design
Optimization and on-line control	[9]–[18], [25]	Explicit objectives, constraints, and interpretable feasibility conditions	Model assumptions and solver cost can limit adaptation under partial observation and rapid state change
Deep and multi-agent reinforcement learning	[19]–[24], [27]–[29], [31]–[33]	Learns nonlinear and sequential decisions across varied edge applications	Policy behavior, generalization, invalid actions, and operator-facing rationale need separate evaluation
Federated and multi-timescale learning	[20], [26]	Coordinates distributed decisions while addressing data locality	Coordination overhead, privacy leakage through traces, and explanation across agents
Lightweight learning and orchestration	[30], [32]	Reduces action or execution cost for constrained edge settings	Compression and reduced action spaces must preserve deadline performance and explanation fidelity
XAI and explainable edge applications	[34]–[38]	Provides explanation concepts and task-specific edge or resource-management examples	Few shared benchmarks for multi-resource placement actions and hard deadlines
Efficient inference	[39], [40]	Compression and hardware-aware execution can reduce model cost	Needs post-compression calibration, safety checks, and explanation-cost accounting

A helpful allocation explanation should have a question from the operations. It can declare that the chosen node had fulfilled the deadline estimate, accelerator memory condition and didn't find the uplink to be congested; it can also specify that there was a competing node that had a queue estimate higher than a threshold. If the estimated uplink delay was less than a given value, then the next best node would be available, and we would have a trade-off that would go further if we lowered the uplink delay. These explanations must assume that the controller has some state and constraints from the decision time. It is risky to try to create an explanation at a later point in time, based on current telemetry, as it may be a description of another network state. The following should, therefore, be recorded with each explanation record: temporal provenance, model version, input timestamp, and uncertainty.

The direct evidence base is coming into being. In the field of resource management, explainable AI has been used in hierarchical federated learning (Patni *et al.*, 2024) [34]; for aerial surveillance, an explainable edge framework has been created for the application of aerial surveillance (Zhu *et al.*, 2025) [35]; and for industrial inspection, an explainable edge framework is also being developed for the use of industrial inspection (Nguyen *et al.*, 2025) [36]. While these works

demonstrate the possibility of edge-side explanation in a task-specific context, they do not provide a widely accepted way of using explanation to explain real-time placement and resource actions across a number of heterogeneous nodes. It is noted that there is a difference in structure between explaining the prediction of a classifier and explaining a joint action over node placement, CPU share, radio capacity and deadline constraints. A resource manager must be given explanations that maintain this action structure and reveal which constraints are binding.

In addition, there are several aspects of explanation quality. Fidelity asks if the explanation is correct, to mean the policy or explicit decision logic is available. Stability is asking the question of whether small, irrelevant changes in the State give very misleading accounts. Completeness: Does the important decision factors and constraints are represented? Usefulness has to do with whether an explanation is useful to the operator to diagnose the problem with a service or select a corrective action. Confidence should reflect uncertainty in state estimates, and in the explanation method. These attributes may come into conflict: A concise explanation may be easily readable but lack context, or a very detailed explanation may be costly to run on a computer and hard to examine quickly. Therefore, the following explana-

tion metrics should be reported separately in a paper from the allocation reward and execution latency.

Table 2. Explanation methods for resource-allocation decisions

Method	Decision evidence it can provide	Edge cost and principal caveat
Constraint and rule trace	Feasible set, binding capacity, deadline test, and reason an action was rejected	Low cost when constraints are explicit; may not explain learned preference among feasible actions
Feature attribution	State features associated with a selected action or action score	Can be costly or unstable under correlated telemetry; attribution is not causal proof
Local surrogate	Approximate local decision boundary or compact rule around one state	Requires sampling and fidelity checks; local validity may not extend to nearby operating regimes
Counterfactual explanation	Small feasible state change that would select a different action	Must respect coupled network dynamics and hard constraints; search can be expensive
Temporal trace	Sequence of state changes, allocations, and outcomes around a service event	Requires time-synchronized logging, retention controls, and clear separation of observation from inference

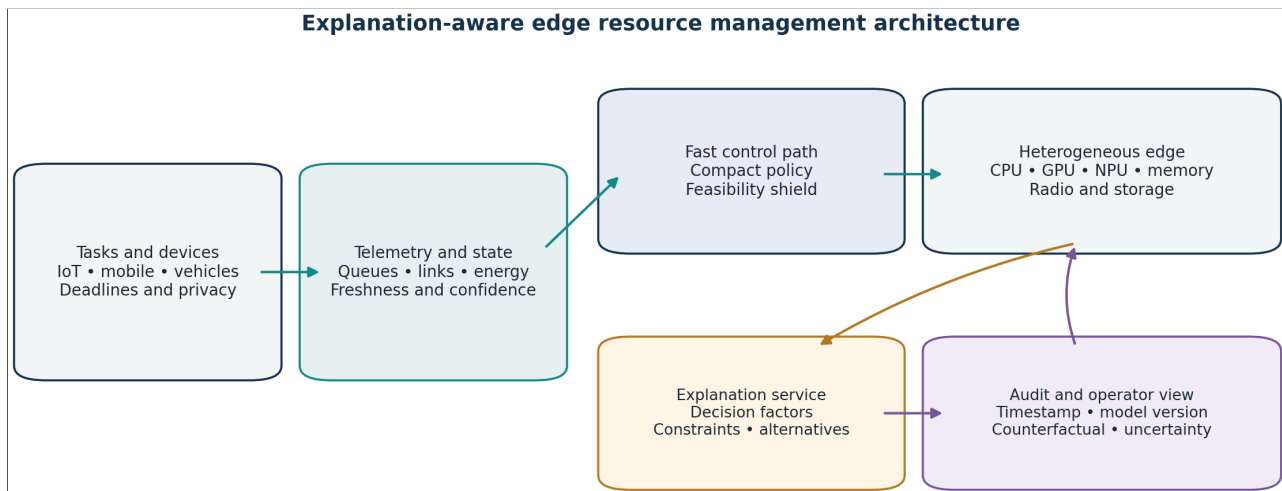


Figure 1. Explanation-aware edge resource management architecture

VI Proposed lightweight explainable framework

The framework comprises of six logical components: Telemetry and state layer, compact allocation policy, feasibility shield, dispatch controller, explanation service, audit store. Task descriptors, queue and capacity summaries, link measurements, energy state, service deadlines and age or confidence of each measurement are included in telemetry. The state layer converts these (and other) inputs into a smaller size representation and labels stale or missing values. The policy chooses one of the possible action templates, such as execute locally, place on edge, cloud-fallback or adjust resource-share. A feasibility shield performs deterministic checks prior to dispatch such as capacity, memory, policy and deadline constraints. The separation is layered so that the learned policy can recommend adaptive actions and known invalid allocations are thwarted by explicit controls.

The architecture has a separation of a fast control path from an explanation path. The fast path validates the state, compactly infers policy, checks the constraints and dispatches the actions. Must satisfy a decision budget at a high percentile and not just as an average. The minimal explanation record should be produced using existing policy outcomes and constraint traces and not prevent dispatch. A more in-depth explanation can be created "after the incident" or during an audit or an operator. The requirement for a lightweight deployment is to be applied to the whole service with buffers, feature transformations, explanation models and logging. Some model compression techniques (e.g., (Sze *et al.*, 2017) [39] and (Deng *et al.*, 2020) [40]) and low-dimensional action representations (e.g., (Wu *et al.*, 2025) [30]) are possible, but should be tested for fidelity and deadline behavior on the target hardware.

The explanation service should output a concise structured record for each action: decision identifier and time of the action; task and node

identifiers or privacy-preserving aliases; placed and allocated resources; top reason(s) for the decision; constraints in place; one or more rejected alternatives with a reason for rejection; uncertainties; freshness of the data; version of the model and policy; and, if it can be computed reliably, a counterfactual threshold. The record should be accurate and clearly differentiate between observed evidence and inferred contribution. The state facts observed include "uplink congestion estimate was high," for example, the model attributions include "congestion caused this placement." It is important to keep these categories separate, in order to lower the chances of a plausible explanation being confused with causal proof.

Compact action parameterization can be used to design a lightweight policy. The controller is not required to check all combinations of node, CPU share, bandwidth share and task ordering, but can do so after masking out any infeasible nodes, and choosing a placement and allocate a resource from a small number of levels found to be admissible. An offline teacher policy can be compressed to an edge-deployable model by a distillation stage and calibration and regression tests are used to verify that the compressed policy still satisfies the constraint adherence and relevant decision behaviour properties. In an optimization mode, or a lower confidence or stale telemetry, the hybrid controller may fall back to a heuristic. The design is based on the complementary strengths of optimization and learning (Chen *et al.*, 2016) [9]–(Yang *et al.*, 2018) [16] and (Xiong *et al.*, 2020) [19]–(Baek *et al.*, 2021) [33] but takes a different approach to compact inference than most of these previous works, in which it is an afterthought.

An explanation pathway should be selective. All decisions must have a very small perturbation trace, enough so that one can be audited, but not be required to run for each control step. They may occur due to a deadline being missed, a low confidence state, an unusual action, crossing a fairness threshold, and/or requesting an operator. A coun

terfactual does not have to be computed over a restricted set of candidate changes, but rather by search over a restricted subset of the state space. If the system can't satisfy the fidelity and/or time bound, it should report that it is not able to provide an explanation and keep the action trace instead of reporting false precision. Explanation sampling should not systematically exclude rare, but important events.

In the framework, it is not assumed that the operator will give the authorization to all allocations. Policy review may be at an alert or as part of policy management, with the majority of real-time decisions automated. The operator interface should display the selected action,

the constraints that will make the action be successful, and uncertainty in a uniform manner, as well as a few alternatives. Auditing view should be able to replay the events in a temporal manner, not based on current measurements, but based on the decision-time state. Any changes to the policy or constraints should be versioned and be assessed as other policy changes. It is important to note that "explainability" is a subset of "observability" and "governance", it does not replace capacity planning, access control, safety analysis or strict validation.

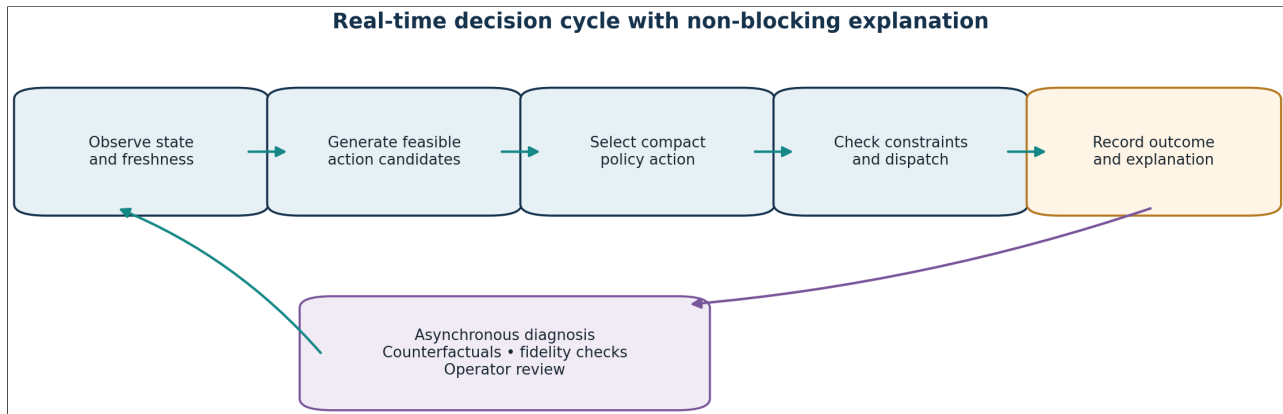


Figure 2. Real-time decision cycle with a non-blocking explanation path

VII Evaluation protocol

A valid assessment should be done according to at least four baselines: a conventional optimization or heuristic allocator, a learning policy with no explanation, a post-hoc explanation pipeline with the same learning policy, and the lightweight explainable design. If appropriate, a multi-agent or federated baseline can help to separate the effects of distributed training and coordination (Liu *et al.*, 2020; Yu *et al.*, 2021; Wu *et al.*, 2025) [20, 26, 30]. All policies will have the same task arrivals, initial resource state, action constraints, and test traces. Training and evaluation workload and network conditions should be separated and generalization tested with held out workload and network conditions. If a simulation-only result, the result should be clearly indicated and, prior to making deployment claims, followed by a hardware-in-the-loop or edge testbed evaluation.

Mean and median latency for a decision or task, 95th percentile and 99th percentile latency for a decision or task, ratio of tasks missed deadlines, ratio of tasks successfully completed, energy consumption per completed task, throughput of tasks, usage of resources, usage of resources per user/service class, number of migration/offload operations performed, and fairness across users/service classes. Policy overhead shall be measured and reported for inference time, peak memory, model size, CPU/accelerator use and energy consumption of the controller. These metrics are different from a policy that makes average service better to one that is better at the tail of service in real-time. The results should be broken down by node type, task class, load level and connectivity state since the overall result may mask a failure mode that can be focused on in a specific node/service.

Explanation evaluation should include generation latency and cost of the resources used in addition to fidelity, stability, completeness and usefulness of the operator. Evaluation may be based on the fidelity of the explanation in following policy changes – or on the agreement of a faithful local surrogate on perturbations sampled – as well as on the constraints being checked directly against the recorded decision trace. Stability can be measured in the face of controlled perturbations which should not be used to make the decision. Prediction of a counterfactual action change should be based on counterfactual feasibility, or on the distance from the observed action, rather than its quality. Usefulness of an operator depends on the task based study, for example, the ability of the participants to understand the problem (why the deadline was not met) or to select a safe corrective action. Just a subjective satisfaction rating is not a good indicator of value in operation. Ablation studies need to factor in only one or two of the following: compact state design, action masking, feasibility shield, explanation triggering, and model compression. Bursty arrivals, partial observability, device churn, wireless interference, sensor or telemetry delay, and task mix changes constitute some of the stressors that should be included in stress tests. Reliability tests should add missing state variables or corrupt state variables, and then verify if the policy works without the uncertainty flags and with them. The system needs to be assessed on the normal and rare high impact events. Reporting the random seeds, workload generators, hardware specifications, explanation settings, and policy parameters allows the results to be easily reproduced and comparison of the methods to be made later. The protocol proposed is summarized in table III.

Table 3. Evaluation dimensions and reporting requirements

Evaluation dimension	Recommended measures	Required reporting
Service performance	p50/p95/p99 task and decision latency, deadline misses, completion rate, throughput	By task class, load, node type, and network condition
Resource cost	Energy per task, controller energy, CPU/GPU use, memory, model size, bandwidth	Separate workload cost from policy and explanation overhead
Allocation quality	Utilization, fairness, migration frequency, queue lengths, constraint violations	State objective weights and hard constraints; include baseline parity

(continued on next page)

Evaluation dimension	Recommended measures	Required reporting
Explanation quality	Fidelity, stability, completeness, counterfactual feasibility, confidence calibration	Measure separately from reward and service latency
Operational usefulness	Fault diagnosis accuracy/time, safe intervention selection, trust calibration	Use task-based operator study; do not rely on satisfaction alone
Robustness and reproducibility	Performance under churn, bursts, stale/missing telemetry, distribution shift	Publish workload traces, seeds, hardware, policy and explanation settings

VIII Research gaps and open questions

The first open problem is that there is no common standard on how to measure explanation-aware allocation. Due to the variety of existing work with respect to the network topology, the task model, the metric and the baseline, cross-paper comparison is limited (Luo *et al.*, 2021) [4], (Djigal *et al.*, 2022) [5]. The benchmark should specify heterogeneous device profiles and arrival processes, deadlines, task dependencies, trace uncertainty, and failure events, and enable researchers to change objective weights and explanation budgets. It should include common interfaces for Optimization, Single-Agent Learning, Multi-Agent Learning and Hybrid Policies. It should also have a means of measuring the effectiveness of the explanation records without relying on allocation results. This would enable distinguishing between the improvement of policy quality and reduction of explanation cost or enhancement of the understanding of the operators.

The second problem is: Temporal Explanation under varying state. A local explanation may only be valid if referring to one decision but not for a series of migrations, oscillations or missed deadlines. Explanations for real systems must link actions over time, and explain which state changes were significant and which were not significant, and they must differentiate policy adaptation from measurement noise. Counterfactuals need to respect dynamics: a change in the bandwidth can change the placement in the queueing and the placement downstream, therefore a static bandwidth one feature counterfactual explanation might be misleading. Evidence is needed to support a temporal causal claim, as opposed to just a feature attribution claim. A practical research avenue can be the logging of compact state-action trajectories, and the subsequent comparison of the controller's predicted outcomes of the logged trajectories with the outcomes of the actual action. A practical research route may be the logging of compact state-action trajectories, and then comparing the outcome of the action with the predicted outcome of the logged trajectories by the controller before exposing the causal language to the operators.

Thirdly, explaining and privacy are incompatible. Edge telemetry could include user behaviors, industrial process state, location or device identity. Although a learning system is federated or decentralized (Liu *et al.*, 2020; Yu *et al.*, 2021) [20, 26] a detailed explanation record can be a new source of sensitive information. Further research will evaluate data minimization, generation of explanations at the local level, retention mechanisms, and aggregation without compromising privacy. It should also explore who has access to explanations and the response to disagreement between the local and network level objectives. Efficiency may be improved by a policy that disadvantages a class of users or services and fairness must be deduced and stated instead of being taken for granted from the sum of the utilities of the policy.

Fourth, explainability needs to be evaluated in the presence of a shift in the distribution and in the presence of adversarial or faulty input. A concise explanation may accurately express a policy that has become inappropriate due to the introduction of a new class of device, radio environment or application. Confidence indicators should be dynamic, not only based on model uncertainty, but also based on the freshness of the data used as input to the model, while out-of-distribution detection should steer data to safe fallbacks. Measuring the explanation fidelity should be done post-quantization, pruning, distillation or hardware acceleration as it can change action boundaries due to compression (Sze *et al.*, 2017) [39], (Deng *et al.*, 2020) [40]. Future

work should also determine the fidelity of explanation records in the case of several agents coordinating and in the case of an online-updated policy.

Lastly, an explanation needs to go beyond just its presence during human-centred evaluation. The operators should be evaluated for fault diagnosis time, correctness of intervention, trust calibration, and the capability of finding an incomplete and/or incorrect explanation. The interface shouldn't show uncertainty, yet not be overpowering in attributions. In service critical environments there will be a need to provide an explanation that is concise enough to be reviewed during an incident and detailed enough to be audited afterwards. The requirements imply a tiered explanation presentation – the action explanation for immediate monitoring, and the state, constraint and counterfactual explanations for diagnosis, which can be expanded during that time. The experience gained from edge applications in industry and aerial edge applications (Zhu *et al.*, 2025; Nguyen *et al.*, 2025) [35, 36] provides a good starting point, but a domain-specific operator study is required for resource allocation.

Issues that can arise and are covered in this article include:

There are a number of limitations to this review. It has an integrative and curated source set, not created using a registered systematic-review protocol. The papers cover various edge architectures, applications and goals and are based on simulated environments in many cases. The literature selected is still quite limited when it comes to direct work to combine explainability, heterogeneous resource allocation and hard real-time guarantees together. The framework and evaluation protocol thus delineates a research agenda, not a contract to a specific policy or explanation procedure that can achieve a specific service level objective. To claim a deployment, it is necessary to perform measurements on representative hardware and workloads for the specific deployment.

The quality of the allocation, real-time cost and explanation quality should be considered as coupled but separately measurable properties in a lightweight explainable edge resource manager. A large amount of literature provides theoretical bases for edge scheduling, optimization and adaptive learning (Shi *et al.*, 2016) [1]–(Yi *et al.*, 2023) [28] and efficient inference (Sheng *et al.*, 2021) [29]–(Deng *et al.*, 2020) [40], however, design choices need to be made for their combination. We suggest that the fast path policy should be compact, that feasibility checks should be done deterministically before dispatch, that the explanation service is asynchronous, and that audit records are kept for the dispatch time that captures the state, constraints, alternatives, uncertainty and model version. The same workloads should be used for future evaluations, and the following measures should be reported: tail latency, deadline misses, energy, fairness, cost of controllers, explanation fidelity, and outcomes of operators. In this way, explainability becomes a verifiable component of the edge resource management process and it can keep the difference between a valid explanation and evidence that an allocation is optimal, fair, or safe.

References

1. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*. 2016;3(5):637–646.
2. Mao Y, You C, Zhang J, Huang K, Letaief KB. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials*. 2017;19(4):2322–2358.
3. Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials*. 2017;19(3):1628–1656.

4. Luo Q *et al.* Resource Scheduling in Edge Computing: A Survey. *IEEE Communications Surveys & Tutorials*. 2021;23(4):2131–2165.
5. Djigal H *et al.* Machine and Deep Learning for Resource Allocation in Multi-Access Edge Computing: A Survey. *IEEE Communications Surveys & Tutorials*. 2022;24(4):2449–2494.
6. Zhou Z *et al.* Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. *Proceedings of the IEEE*. 2019;107(8):1738–1762.
7. Chen J, Ran X. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*. 2019;107(8):1655–1674.
8. Xu D *et al.* Edge Intelligence: Empowering Intelligence to the Edge of Network. *Proceedings of the IEEE*. 2021;109(11):1778–1837.
9. Chen X, Jiao L, Li W, Fu X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Transactions on Networking*. 2016;24(5):2795–2808.
10. Sardellitti S, Scutari G, Barbarossa S. Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing. *IEEE Transactions on Signal and Information Processing over Networks*. 2015;1(2):89–103.
11. You C, Huang K, Chae H, Kim BH. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Transactions on Wireless Communications*. 2017;16(3):1397–1411.
12. Mao Y, Zhang J, Song SH, Letaief KB. Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems. *IEEE Transactions on Wireless Communications*. 2017;16(9):5994–6009.
13. Mao Y, Zhang J, Letaief KB. Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices. *IEEE Journal on Selected Areas in Communications*. 2016;34(12):3590–3605.
14. Tran TX, Pompili D. Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks. *IEEE Transactions on Vehicular Technology*. 2019;68(1):856–868.
15. Pham QV, Le LB, Chung SH, Hwang WJ. Mobile Edge Computing With Wireless Backhaul: Joint Task Offloading and Resource Allocation. *IEEE Access*. 2019;7:16444–16459.
16. Yang L, Zhang H, Li M, Guo J, Ji H. Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G. *IEEE Transactions on Vehicular Technology*. 2018;67(7):6398–6409.
17. Wu C, *et al.* Online User Allocation in Mobile Edge Computing Environments: A Decentralized Reactive Approach. *Journal of Systems Architecture*. 2021;113:101904.
18. Liu Y, *et al.* Joint Task Offloading and Resource Allocation in Heterogeneous Edge Environments. *IEEE Transactions on Mobile Computing*. 2024;23(6):7318–7334.
19. Xiong X, Zheng K, Lei L, Hou L. Resource Allocation Based on Deep Reinforcement Learning in IoT Edge Computing. *IEEE Journal on Selected Areas in Communications*. 2020;38(6):1133–1146.
20. Liu X, Yu J, Feng Z, Gao Y. Multi-Agent Reinforcement Learning for Resource Allocation in IoT Networks With Edge Computing. *China Communications*. 2020;17(9):220–236.
21. Liu Y, Yu H, Xie S, Zhang Y. Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks. *IEEE Transactions on Vehicular Technology*. 2019;68(11):11158–11168.
22. Chen X, Zhang H, Wu C, Mao S, Ji Y, Bennis M. Optimized Computation Offloading Performance in Virtual Edge Computing Systems Via Deep Reinforcement Learning. *IEEE Internet of Things Journal*. 2019;6(3):4005–4018.
23. Alqerm I, Pan J. DeepEdge: A New QoE-Based Resource Allocation Framework Using Deep Reinforcement Learning for Future Heterogeneous Edge-IoT Applications. *IEEE Transactions on Network and Service Management*. 2021;18(4):3942–3954.
24. Yan J, Bi S, Zhang YJ A. Offloading and Resource Allocation With General Task Graph in Mobile Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*. 2020;19(8):5404–5419.
25. Bi S, Huang L, Wang H, Zhang YJ A. Lyapunov-Guided Deep Reinforcement Learning for Stable Online Computation Offloading in Mobile-Edge Computing Networks. *IEEE Transactions on Wireless Communications*. 2021;20(11):7519–7537.
26. Yu S, Chen X, Zhou Z, Gong X, Wu D. When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multi-Timescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network. *IEEE Internet of Things Journal*. 2021;8(4):2238–2251.
27. Chen X, Liu G. Energy-Efficient Task Offloading and Resource Allocation via Deep Reinforcement Learning for Augmented Reality in Mobile Edge Networks. *IEEE Internet of Things Journal*. 2021;8(13):10843–10856.
28. Yi M, Yang P, Chen M, Loc NT. A DRL-Driven Intelligent Joint Optimization Strategy for Computation Offloading and Resource Allocation in Ubiquitous Edge IoT Systems. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2023;7(1):39–54.
29. Sheng S, Chen P, Chen Z, Wu L, Yao Y. Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing. *Sensors*. 2021;21(5):1666.
30. Wu D, Wu S, Cui Y, Zhong A, Tang T, Wang R, Lin X. Lightweight Deep Reinforcement Learning for Dynamic Resource Allocation in Vehicular Edge Computing. *Digital Communications and Networks*. 2025;11(5):1530–1542.
31. Liu L, *et al.* Asynchronous Deep Reinforcement Learning for Collaborative Task Computing and On-Demand Resource Allocation in Vehicular Edge Computing. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(12):15513–15526.
32. Yamansavascular B, Baktir AC, Sonmez C, Ozgovde A, Ersoy C. DeepEdge: A Deep Reinforcement Learning-Based Task Orchestrator for Edge Computing. *IEEE Transactions on Network Science and Engineering*. 2023;10(1):538–552.
33. Baek J, Kaddoum G. Heterogeneous Task Offloading and Resource Allocations via Deep Recurrent Reinforcement Learning in Partial Observable Multi-Fog Networks. *IEEE Internet of Things Journal*. 2021;8(2):1041–1056.
34. Patni S, *et al.* Explainable AI Empowered Resource Management for Enhanced Communication Efficiency in Hierarchical Federated Learning. *Computers & Electrical Engineering*. 2024;117:109260.
35. Zhu H, Demirbag U, Aujla GS, Shi L, Zhang P. Explainable Edge AI Framework for IoD-Assisted Aerial Surveillance in Extreme Scenarios. *IEEE Internet of Things Journal*. 2025;12(5):4570–4578.
36. Nguyen HTT, Nguyen LPT, Cao H. XEdgeAI: A Human-Centered Industrial Inspection Framework With Data-Centric Explainable Edge AI Approach. *Information Fusion*. 2025;116:102782.
37. Ribeiro MT, Singh S, Guestrin C. Why Should I Trust You?: Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016:1135–1144.
38. Arrieta AB, *et al.* Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI. *Information Fusion*. 2020;58:82–115.

39. Sze V, Chen YH, Yang TJ, Emer JS. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Proceedings of the IEEE. 2017;105(12):2295-2329.
40. Deng L, Li G, Han S, Shi L, Xie Y. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. Proceedings of the IEEE. 2020;108(4):485-532.

How to cite this article: Gupta DN, Shukla A, Tyagi N, Singh SP, Tiwari S, Divya, et al. Lightweight explainable AI for real-time resource allocation in heterogeneous edge computing networks. *International Journal of Engineering in Computer Science*. 2026;8(4):10-17. doi: <https://www.doi.org/10.33545/26633582.2026.v8.i4a.317>

Authors

Deena Nath Gupta

Associate Professor, Department- Computer Science and Engineering, College name- Rayat Bahra University, V.P.O, Tehsil, Sahauran, Kharar, Punjab 140104, India.

Ashish Shukla

Assistant professor, Department- Computer Application, College name with address-Axis Institute of Higher Education, Rooma, Kanpur, UP, India

Navneet Tyagi

Associate Professor, HOD MCA, Department- MCA, College name with address-Inmantec Institution, Ghaziabad, Uttar Pradesh, India

Satya Pal Singh

Assistant Professor, Department- Information Technology, College name with address-IIMT College of Engineering, Greater Noida, Uttar Pradesh, India

Shivoham Tiwari

Assistant Professor, Department- Information Technology, College name with address- I.T.S College Mohan Nagar Ghaziabad(UG Campus), Uttar Pradesh, India

Divya

Assistant Professor, Department- CSE, College name with address- Guru Tegh Bahadur Institute of Technology, Rajouri Garden, New Delhi, India

Monu Kumar

Assistant Professor, Department- Computer Science Engineering, College name with address--I.T.S Engineering College Greater Noida, Uttar Pradesh, India

Meduri Sridhar Sarma

Assistant Professor, Department- Computer Application, College name with address-Aditya University ,Surampalem, Andhra Pradesh, India

Aruna Malik

Assistant Professor, Department- Computer Science & Engineering, College name with address- Birla Institute of Technology (Mesra), Ranchi, Off Campus Jaipur, Rajasthan, India

Vinay Shankar Dubey

Research Scholar, Department- Mathematics, College name with address-Nehru Gram Bharti University Jamunipur Prayagraj Uttar Pradesh, India