Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

A Novel Edge AI Framework for Selective Subtask Offloading in UAV-Assisted Multiuser Multiserver Mobile Edge Computing

Suhel Ahamed *

*Corresponding author, Department of Information Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh, India. E-mail: suhelahamed@ieee.org

Amit Kumar Dewangan

Department of Information Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh, India. E-mail: amit.nitrr@gmail.com

Abhishek Jain

Department of Information Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh, India. E-mail: ajain.nit@gmail.com

Nishant Behar

Department of Computer Science and Engineering, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, India. E-mail: nishant.itggv@gmail.com

Pankaj Shankar Shrivastava

Department of Electronics and Communication Engineering, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, India. E-mail: pss1170@gmail.com

Journal of Information Technology Management, 2026, Vol. 18, Issue 4, pp. 44-58.
Published by the University of Tehran, College of Management
doi: https://doi.org/10.22059/jitm.2026.108914
Article Type: Research Paper

Received: June 11, 2026
Received in revised form: July 24, 2026
Accepted: August 19, 2026
Published online: October 01, 2026

Abstract

Mobile edge computing (MEC) integrated with unmanned aerial vehicles (UAVs) enables low-latency, energy-efficient computation for resource-constrained ground users; however, selective subtask offloading in multiserver, multiuser environments remains a challenging combinatorial optimization problem. A novel IT management framework is needed to address the challenges of modern smart computing environments and support efficient, adaptive operational decision-making. This paper proposes an Edge AI framework for selective subtask offloading built on the public IEEE Dataport MSMU-CO dataset; we use the term oracle to denote the minimum-cost multi-commodity flow (MCMF) solution, which serves as

an offline near-optimal benchmark for generating training labels and evaluating policies. Starting from 800 MCMF-generated multi-server, multiuser scenarios, each user task is decomposed into K=3subtasks, oracle labels are derived under an explicit latency-energy cost model, and the data are augmented with UAV-supported MEC geometry, including aerial and ground servers, 3D positions, line-of-sight (LoS) probability, and effective uplink rates. To ensure realistic deployment conditions, a deliberate feature leakage guard eliminates anchor-derived geometries, preventing data leakage from post-optimization target assignments and maintaining causal inference realism. A compact multilayer perceptron (MLP) is then trained to predict per-subtask offloading decisions solely from task, channel, and localized UAV context features. Evaluated on a test set containing 1,780 users and 5,340 subtasks, the proposed model achieves an accuracy of 0.842, precision of 0.793, recall of 0.870, an F1 score of 0.830, and an AUC of 0.926 under the cost model used for label generation. Furthermore, we benchmark our approach against a greedy linear relaxation baseline (LR Greedy) based on Huang et al. (2018, 2020). Experiments show that both the proposed policy and the greedy baseline significantly reduce deadline violations and mean system costs compared with an All-Local execution strategy, effectively bridging the performance gap toward the offline oracle labels in varying UAV-assisted MEC environments.

Keywords: Mobile Edge Computing; IT Management Policy, Computation Offloading, Unmanned Aerial Vehicles, Edge Intelligence, Subtask Decomposition, Deep Learning

Introduction

A growing class of mobile workloads such as augmented reality (AR), real-time object detection, and on-device inference demands more computation and energy than typical user-end devices can efficiently provide. MEC addresses this mismatch by offloading computation, control, and storage from remote clouds to edge nodes near the data source, reducing latency and device-side energy consumption (Mao et al., 2017). In multiuser, multiserver, multitask management environments, the primary challenge is determining which tasks or task components should be executed locally and which should be offloaded to specific servers while optimizing a joint latency-energy objective.

This offloading problem is purely combinatorial. Under binary offloading, the number of offloading decisions grows exponentially with the number of users, and the problem remains NP-hard even in single-server MEC scenarios (Chen et al., 2016). Existing approaches such as game-theoretic methods, Lyapunov-based control, and convex or mixed-integer relaxations often require iterative solvers or restrictive assumptions that limit their practical deployment in dense and heterogeneous network topologies. Learning-based methods such as DDLO (Distributed Deep Learning-based Offloading) and DROO (Deep Reinforcement learning-based Online Offloading) partly address this issue by replacing repeated optimization with

fast neural inference while retaining near-optimal performance (Huang et al., 2018; Huang et al., 2020).

UAV-mounted edge servers further extend edge computing beyond fixed terrestrial infrastructure. By carrying computational resources onboard, UAVs can be repositioned on demand to provide LoS-friendly wireless links and flexible coverage to ground nodes (Jeong et al., 2018; Hu et al., 2019). However, once aerial and ground servers coexist, additional spatial factors such as 3D distance, elevation angle, and LoS probability directly affect the feasibility of uplink rate and therefore the offloading decision itself.

This paper focuses on a selective subtask offloading policy. Instead of treating a user task as atomic, each task is divided into multiple subtasks; also, the execution location of each subtask is determined separately. This is very important because fixed signaling and setup overheads can make it preferable to keep lightweight subtasks local while offloading only the computationally heavy subtasks. To study this problem, the framework is built on the publicly available MSMU-CO dataset (Liang et al., 2024), which is extended with UAV-supported MEC semantics and used to evaluate both learned and heuristic offloading policies.

The main contributions are as follows. (i) A reproducible subtask-level dataset is constructed from MSMU-CO using an explicit latency-energy cost model and MCMF-derived oracle labels. (ii) The dataset is augmented with UAV-supported MEC semantics without changing any derived oracle labels. (iii) An Edge AI multilayer perceptron (MLP) with a specific leakage guard is developed so that offloading decisions are predicted only from genuine task, channel, and UAV context features, preserving generalizability. (iv) A literature based LR/DDLO style greedy baseline is added, and all methods are evaluated under a common cost model using latency, energy, cost, deadline violations, and subtask-level classification metrics.

Methodology

Literature Survey

Early foundations of multi-user computation offloading over multi-channel interference by Chen et al. (2016) became a strategic milestone, established the existence of Nash equilibrium, and proposed a distributed algorithm whose efficiency can be bounded with respect to a centralized optimum. Mao et al. (2017) provided a comprehensive MEC survey from the communications viewpoint, systematized the task, wireless channel, and latency energy models that many later studies used, including this work.

For optimization, Bi and Zhang (2018) analyzed a wireless powered MEC system with binary offloading and maximized a weighted sum modeled computation rate via joint optimization of offloading policy and transmission time, solved using bisection and

coordinate descent methods. Huang et al. (2018) introduced the DDLO framework to avoid iterative solving of hard combinatorial problems within each channel coherence interval, which employs parallel DNNs to output near-optimal binary offloading decisions. Huang et al. (2020) proposed DROO, a DRL-based framework that learns these binary mappings from interaction data and cuts decision time by more than an order of magnitude. The LR-Greedy baseline used here borrows the gain ranking and capacity-aware admission ideas that appear in LR/DDLO type schemes, but adapts them to our subtask-level setting and explicit latency-energy objective. Beyond MEC-specific offloading, recent work in the Journal of Information Technology Management illustrates how carefully designed deep models can deliver robust classification performance in resource-constrained or heterogeneous environments (Jain & Raja, 2023; Patnaik et al., 2021). Related work on multi-agent systems also highlights how distributed autonomous agents can coordinate complex decision-making under local information, offering another lens on decentralized resource management problems similar to MEC offloading (Chaharsooghi et al., 2015).

In airborne edge contexts, Jeong et al. (2018) considered a UAV-mounted cloudlet and optimized both bit allocation and UAV trajectory to minimize mobile energy under latency and UAV energy budgets, whereas Hu et al. (2019) jointly optimized offloading decisions and UAV trajectories in a UAV-enabled MEC architecture. More recent work couples UAV trajectory optimization with DRL-based controllers to jointly reduce weighted delay and energy consumption (Wang et al., 2024). These works motivate the UAV geometry—three-dimensional positions, elevation-dependent LoS probability, and distance-dependent rate—that we layer onto the MSMU-CO data. Edge intelligence refers to performing ML inference and even training directly on edge nodes, which can lower latency and improve privacy (Zhou et al., 2019), while decentralized approaches such as federated averaging make on-device training feasible at scale (McMahan et al., 2017). The proposed Edge-AI classifier is intentionally lightweight so that it can be executed at a ground user or onboard a UAV and, in principle, further adapted using decentralized training schemes.

Problem Formulation

We consider a multi-server multi-user MEC system. A scenario contains U users (with U in {10, 20}) and S edge servers (with S in {3, 5}); a subset of the servers are UAV-borne and the remainder are ground MEC nodes. Each user i has one macro task that is decomposed into K = 3 subtasks. For subtask k of user i, we denote the data size by $d_{i,k}$ and the required CPU workload (in cycles) by $c_{i,k}$. The user has local CPU frequency f_i, and each server offers CPU capacity Ft. The offloading decision for each subtask is binary: local execution (label 0) or offloading to the anchor server s* (label 1).

Communication Model

Consistent with the MSMU-CO abstraction, the uplink configuration fixes the bandwidth at B = 80 MHz, the noise power at N0 = 7.96 × 10−13 W, and transmit power Ptx = 0.3 W. With h the best channel power gain of the user, the received signal saturates the link, giving a bounded signal-to-interference-plus-noise ratio (SINR) and the corresponding Shannon rate:

$$SINR = Ptx\ h2 / (N0 + Ptx\ h2) \quad (1)$$

$$R = B \log2 (1 + SINR) \quad (2)$$

Using this bounded SINR expression avoids the unrealistically large throughputs produced by the usual high-SINR approximation and yields more plausible sub-Gbps rate values. A one-time signaling and setup delay Tsetup = 0.20 s is added for any offloaded subtask; this fixed overhead makes it non-obvious whether a light subtask should be sent to the edge or kept local.

Computation, Latency, and Energy Models

For a UAV or a ground anchor node, the server capacity is shared by the remote execution according to a resource ratio. The MCMF resource ratio is the effective server share when the user has a valid anchor and a positive ratio; otherwise, it is 1/U. The per subtask completion time and energy are expressed as:

$$\text{offload: } ti,k = di,k / R + Tsetup + ci,k / (Ft \cdot sharei) \quad (3)$$

$$\text{local: } ti,k = ci,k / fi \quad (4)$$

$$\text{offload: } ei,k = Ptx (di,k / R) + Ptx\ Tsetup \quad (5)$$

$$\text{local: } ei,k = \kappa\ fi\ 2\ ci,k \quad (6)$$

where the CPU energy coefficient is $\kappa = 1 \times 10{-}28$ and the server capacity is Ft = 33.6 GHz. Aggregating over the K subtasks, the user-level latency and energy are the sums of their subtask contributions, $Li = \Sigma k\ ti,k$ and $Ei = \Sigma k\ ei,k$.

Cost Function

$$Ji = \alpha i \cdot Li + (1 - \alpha i) \cdot Ei \quad (7)$$

Each user weights latency against energy with a personalized coefficient αi in [0, 1] drawn from the base MCMF data. Users with large αi values (latency-dominated preferences) tend to retain light subtasks locally to avoid the setup delay, and this interaction is what generates

truly selective oracle patterns in the data. The system objective is to choose the per-subtask offloading decisions that minimize Σi Ji subject to server resource availability.

Proposed Edge-AI Selective Subtask Offloading (EAISSO) Framework

The baseline dataset layer is generated with the MSMU-CO tool (Liang, 2024), which produces multi-server multi-user scenarios and solves the assignment with a minimum-cost maximum-flow formulation. We generated 800 scenarios spanning the four regimes U in {10, 20} crossed with S in {3, 5}, yielding 12,000 user rows; in the base solution, roughly 45.1% of users receive an MCMF server anchor while the rest stay local.

For subtask decomposition, each macro task is divided into K = 3 subtasks by sampling normalized random weights wk and assigning wk. Di bits and wk. Ci cycles to subtask k for user i. Subtasks are typed by descending CPU-cycle rank (heavy-compute, pre-processing, post-processing), giving 36,000 subtask rows in total. For every user, we evaluate up to three candidate offloading patterns under the cost model of Section 3 and select the minimum-cost pattern as the oracle. The patterns are P1 (all subtasks offloaded to s*), P2 (the heaviest ⌈K/2⌉ subtasks offloaded, the rest local), and P3 (all subtasks local). When the user has no anchor (s* = -1), only the All-Local pattern is feasible. Under this labeling procedure, about 55.7% of subtasks are tagged as local and 44.3% as offloaded. The oracle is dominated by the full-offload pattern P1 (35,217 subtasks), with the selective pattern P2 occurring for only 783 subtasks (about 2% of users); no All-Local oracle pattern survives once a feasible anchor exists. The data are split by scenario identifier into 70/15/15 train/validation/test partitions, giving 25,170, 5,490, and 5,340 subtask rows across 560, 120, and 120 scenarios, respectively. Figure 1 depicts the Proposed Edge-AI Selective Subtask Offloading (EAISSO) Framework execution pipeline.

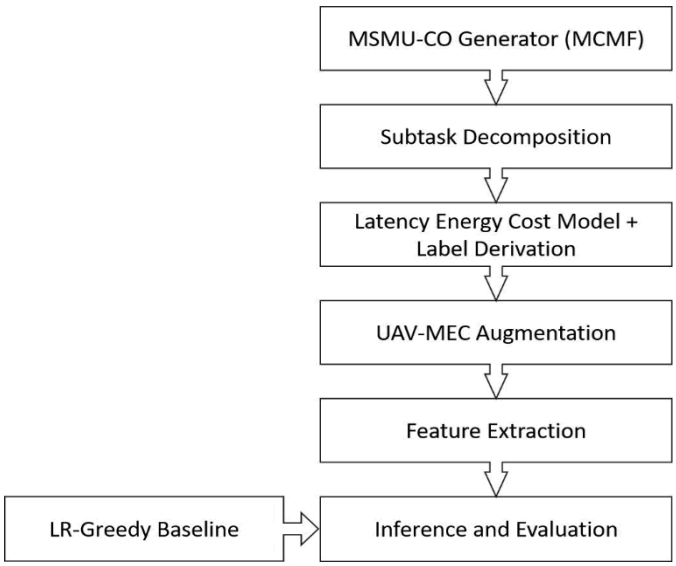

Figure 1. Proposed Edge-AI Selective Subtask Offloading (EAISSO) Framework execution pipeline

UAV MEC Augmentation

On top of the subtask dataset, we overlay a UAV-supported MEC environment using a fixed random seed for reproducibility. Within each scenario, one or two servers are designated as UAVs and the rest as ground nodes. User and server locations are drawn inside a 1,000 m x 1,000 m area; ground nodes and users lie at altitude 0, whereas UAVs hover between 80 m and 150 m. For each offloaded subtask, we join the user position to its assigned (oracle anchor) server and compute the horizontal distance, three-dimensional distance, and elevation angle θ. The Line of Sight (LoS) probability for UAV links is modeled via an urban-micro-style sigmoid function of the elevation angle:

$$PLoS = 1 / (1 + A \cdot \exp(-B (\theta deg - A))) \quad (8)$$

with A = 9.61 and B = 0.16, whereas ground links are treated as deterministic LoS. The effective channel gain combines a free space like term β_0 / r_{3D}^{η} (with reference gain β0 = 10−3 and path-loss exponent η = 2.2) with the LoS probability, attenuating the non LoS share, and the resulting effective uplink rate follows Eq. (2). Each row also records scenario level UAV context such as the number of UAV servers, the minimum distance from the user to any UAV, and a coverage flag (within a 600 m radius). This adds 15 columns to the dataset. Importantly, this spatial augmentation does not modify the oracle labels or server anchors; this invariance is verified programmatically on all dataset splits. Among offloaded subtasks, about 38.5% anchor to a UAV server and the remainder to ground nodes.

Feature Extraction

An important property of this dataset is that the oracle labels are perfectly determined by whether the user has an MCMF anchor; all a user's subtasks share the ground node anchor decision, and the selective pattern P2 occurs only for about 2% of the users. The assigned server features (server coordinates, distances, elevation, LoS probability, effective rate, the UAV server flag, and the MCMF resource ratio) are defined only for offloaded rows and are missing for local execution rows. Including them yields a meaningless 100% accuracy. To preserve integrity and validity and make the task a genuine prediction problem, these anchor-derived columns and the local indicator are excluded from the model inputs. The classifier therefore must decide from task level features (data size, CPU cycles, local CPU frequency, latency weight, channel gain, server count, and the global compute constants), subtask level features (weight, data size, cycles, and type), the user's own position, and scenario level UAV context (number of UAV servers, minimum distance to any UAV, and the coverage flag). This leaves 21 valid input features. The excluded columns are retained only for the cost evaluation, which legitimately needs the true anchor.

Evaluation and Benchmarks

The proposed Edge-AI model (EAISSO) is a compact multilayer perceptron that processes 21 standardized features via two hidden layers (256 and 128 ReLU units, respectively) with 0.15 dropout, and outputs a single logit for the offloading decision. Inputs are standardized with statistics fitted on the training split. Training minimizes a binary cross-entropy loss with logits, using a positive-class weight to counter the label imbalance, the Adam optimizer at learning rate 10-3, a batch size of 1,024, and early stopping on the validation loss with patience 5 over a maximum of 30 epochs. Training converges quickly: the validation accuracy rises from about 0.815 at epoch 1 to roughly 0.841 by epoch 29, when early stopping triggers. The best checkpoint is retained for evaluation.

To benchmark the learned policy against a classical optimization heuristic, we implement an LR/DDLO style greedy baseline inspired by the linear-relaxation and DDLO designs of Huang et al. for multi-server multi-user MEC networks (Huang et al., 2018, 2019). For each user, the method computes the cost of keeping all subtasks local versus offloading all of them to the anchor s*, and defines the gain as the cost reduction from offloading. Users with a feasible anchor and positive gain are sorted in descending order of gain and greedily admitted to their server while a normalized resource-demand budget remains below a capacity threshold of 1.2; users that cannot be admitted, or that have no anchor or non-positive gain, stay local. All decisions are then re-costed with the same cost model used for the oracle labels.

Results and Discussion

All methods are evaluated on the same held-out test split of 1,780 users and 5,340 subtasks across 120 scenarios, and every method is scored with the identical cost model of Section 3. The Edge-AI model is trained once on the training split; neither the model nor the datasets are modified when the LR-Greedy baseline is added. We compare six methods: Oracle, Proposed Edge-AI (EAISSO), All-Local, All-MCMF, Random, and LR-Greedy. We report per-user latency, energy, and cost (means ± standard deviation), the deadline-violation rate, and, for the methods that produce per-subtask decisions, subtask-level classification metrics against the oracle label. The deadline is set to 1.5 x the median oracle latency, which evaluates to 3.257 s. Experiments were run on a CPU-only environment using PyTorch and scikit-learn; the compact model and the closed-form cost evaluation make the entire pipeline inexpensive, consistent with the goal of edge deployability.

Table 1. Test-set performance of selective subtask offloading methods

Method	Latency (s)	Energy (J)	Cost	Deadline Viol.	Subtask Acc.
Oracle	2.606 ± 1.763	50.748 ± 65.980	30.439 ± 44.958	0.212	1.000
Edge-AI (EAISSO)	2.708 ± 1.822	56.058 ± 66.827	33.193 ± 45.353	0.235	0.842
All-Local	4.447 ± 3.469	76.199 ± 63.204	40.887 ± 42.453	0.529	_
All-MCMF	2.610 ± 1.762	50.743 ± 65.984	30.440 ± 44.957	0.213	_
Random	3.516 ± 2.448	63.689 ± 62.446	35.707 ± 43.125	0.403	_
LR-Greedy	3.176 ± 2.731	54.874 ± 65.140	32.005 ± 44.319	0.296	0.881

Note. Cost is the per-user latency–energy objective of Eq. (7). "Subtask Acc." is classification accuracy against the oracle label; baselines without per-subtask predictions are marked " _ ".

The oracle achieves the lowest cost (30.439) and, notably, All-MCMF nearly matches it (30.440), because the oracle is dominated by full-offload patterns once a feasible anchor exists. EAISSO attains a mean cost of 33.193, which is 9.0% above the oracle but 18.8% below All-Local (40.887) and 7.0% below Random (35.707). LR-Greedy lands at 32.005, between the oracle and EAISSO on cost.

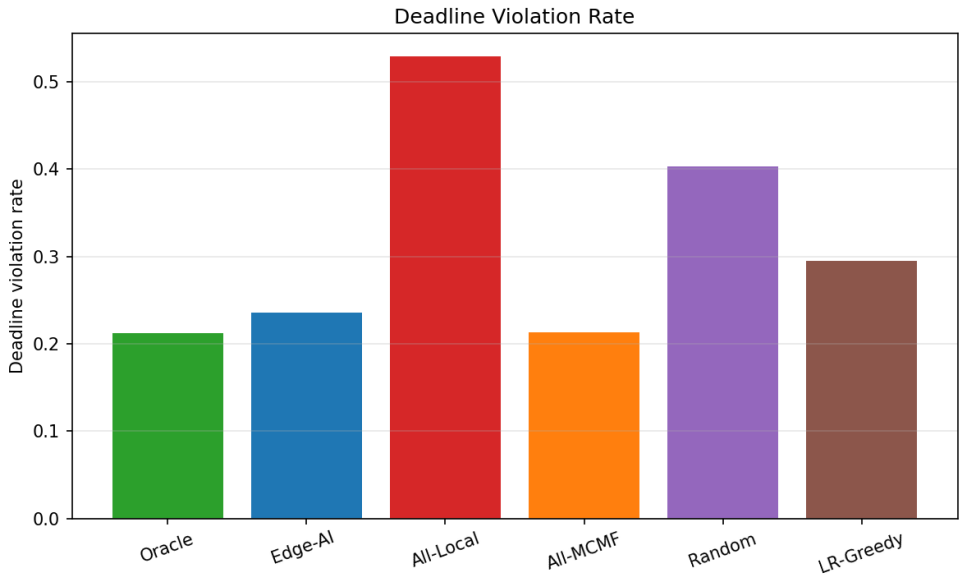

Figure 2. Deadline-violation rate by method, where deadline = 1.5 x median oracle latency (3.257 s)

Figure 2 illustrates the deadline violation rates. All-Local violates the deadline for 52.9% of users, Random for 40.3%, and LR-Greedy for 29.6%, whereas Edge-AI (EAISSO) (23.5%) approaches the oracle and All-MCMF (about 21.2% - 21.3%). Compared to All-Local, the proposed Edge-AI method EAISSO cuts the deadline-violation rate by roughly 56%.

Subtask-level classification

Table 2 reports how closely the per-subtask decisions match the oracle beyond the cost view.

Table 2. Subtask-level classification metrics against the oracle label on the test set

Method	Accuracy	Precision	Recall	F1 / AUC
Edge-AI (EAISSO)	0.842	0.793	0.870	0.830 / 0.926
LR-Greedy	0.881	0.987	0.741	0.847 / 0.863

Edge-AI reaches an accuracy of 0.842 with balanced precision (0.793) and recall (0.870), an F1 of 0.830, and an AUC of 0.926, indicating a well-balanced policy. LR-Greedy is more conservative: it has a higher accuracy (0.881) and very high precision (0.987) but lower recall (0.741), showing that it offloads when the gain is clear and capacity permits, at the cost of missing some beneficial offloads.

Latency - energy trade-offs

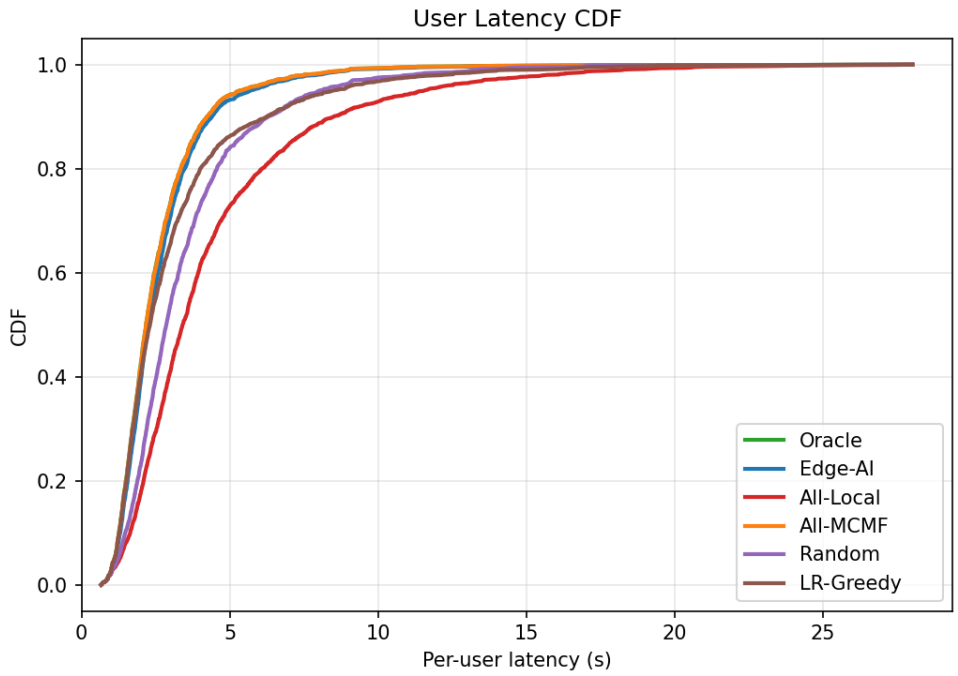

Figure 3. Cumulative distribution of per-user latency for the six offloading policies

Figure 3 shows the cumulative distribution of per-user latency. The oracle, All-MCMF, and Edge-AI curves are all near the low-latency end of the CDF, while LR-Greedy is shifted slightly to the right. In contrast, Random and especially All-Local show far-right shifts, with All-Local suffering the worst completion times because every task is forced onto weak local CPUs.

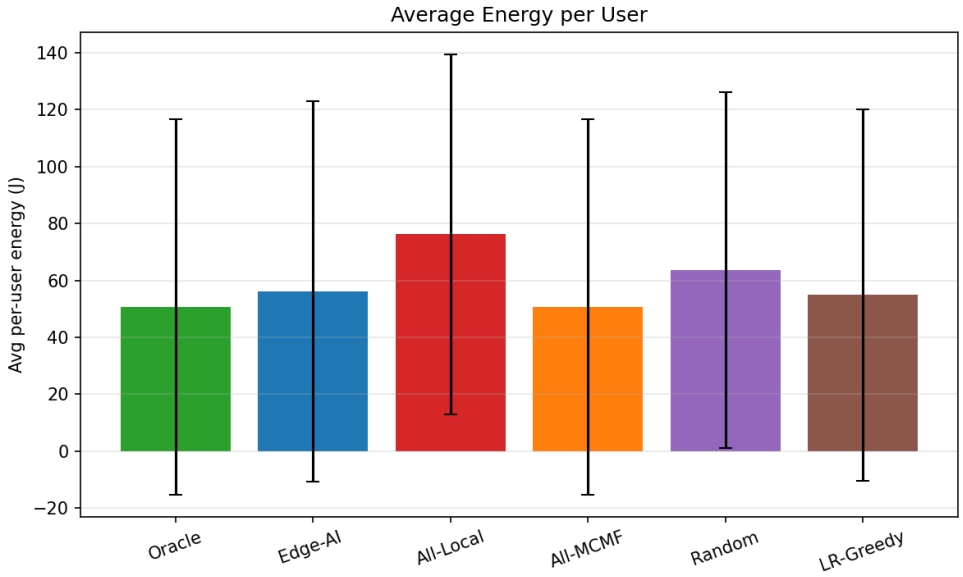

Figure 4. Average energy per user by method (error bars show one standard deviation)

Figure 4 reports mean per-user energy; All-Local is the most energy-consuming (76.199 J), while the oracle and All-MCMF are the least (about 50.7 J); Edge-AI (56.058 J) and LR-Greedy (54.874 J) are in between. Taken together, the learned and greedy policies recover most of the oracle's latency benefit without its full information requirements.

UAV versus ground anchors

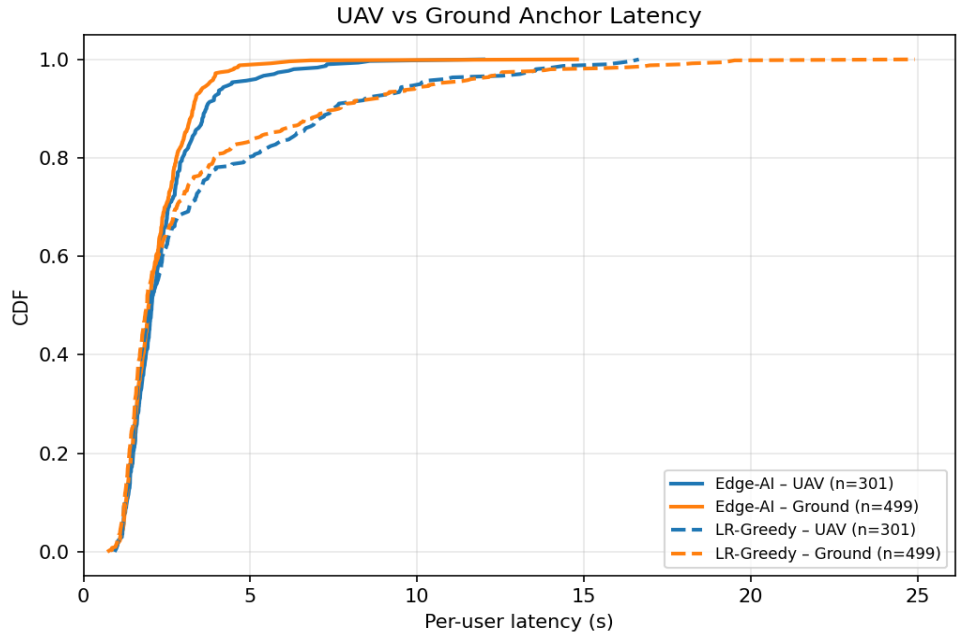

Figure 5. Per-user latency CDFs for offloaded users, categorized by UAV versus ground anchor

Figure 5 categorizes the latency CDF of offloaded-anchor users by whether their anchor is a UAV or a ground server, for both Edge-AI (solid) and LR-Greedy (dashed).

Ground anchors generally deliver slightly lower latency than UAV anchors because the deterministic LoS terrestrial link avoids the probabilistic LoS penalty and the longer 3D distance to a hovering UAV. The gap is modest; however, confirming that UAV servers are a viable offloading target in this environment, particularly for users within UAV coverage. The LR-Greedy curves exhibit a heavier tail than Edge-AI for both anchor types, but with its conservative admission policy leaving some high-cost users local.

Observations

Three observations emerge from the experimental study. (i) The near-equivalence of All-MCMF and the oracle reveals that, under the chosen parameters and the dominant full-offload regime, offloading every subtask of an anchored user is almost always optimal; the selective pattern P2 matters for only a small minority of latency-sensitive users. This is a property of the dataset and cost model rather than a limitation of the methods, and it sets a demanding bar for any learned policy. (ii) The Edge-AI model performs competitively even though we explicitly removed leakage-prone features from its inputs. By withholding anchor-derived geometry, we force the classifier to infer offloading value from task, channel, and UAV context features alone, which is the realistic situation an edge node faces before an assignment is computed. Achieving 0.842 accuracy and 0.926 AUC in this condition and translating that into a 56% reduction in deadline violations relative to All-Local indicates that the signal in the non-leaking features is genuine and predictive. (iii) LR-Greedy and Edge-AI are complementary. LR-Greedy is precision-oriented (0.987 precision) and slightly cheaper on average cost (32.005 versus 33.193), but its lower recall (0.741) leaves more users local, producing a higher deadline violation rate (0.296 versus 0.235). Edge-AI offloads more aggressively and meets more deadlines. Practically, these two policies could be combined: the greedy rule provides an interpretable, capacity-aware safeguard, and the proposed model exploits fine-grain interactions in the feature space. At a broader IT management level, our findings complement results published by Kashyap et al. (2025) showing that intelligent, adaptive services such as generative AI-driven hyper-personalized health monitoring are better realized through distributed edge platforms than centralized clouds. There are also limitations in these conclusions. The UAV geometry is synthetic and static rather than trajectory optimized, the cost model assumes sequential subtask execution per user, and the dominant full offload policy limits the headroom for selective policies. These are deliberate consequences of reusing the existing dataset and outputs.

Conclusion

We presented a novel Edge-AI framework (EAISSO) for selective subtask offloading in UAV-supported multi-server multiuser MEC, built entirely on the MSMU-CO dataset. The pipeline decomposes tasks into subtasks, derives oracle labels with an explicit latency-energy cost model, augments the data with UAV geometry, and trains a compact MLP under a strict leakage guard. On a held-out test set, the model reaches 0.842 subtask accuracy and 0.926 AUC, cuts deadline violations by roughly 56% relative to All-Local, and operates within 9.0% of the oracle on cost. A literature-inspired LR-Greedy baseline, added without retraining, provides a precision-oriented complement at a mean cost of 32.005.

Future work will replace the static UAV placement with joint trajectory and offloading optimization, extend the binary per subtask decision to fractional offloading and multi-server splitting, incorporate online channel dynamics suited to DRL-based controllers, and explore federated training of the Edge-AI model across distributed edge nodes to improve privacy issues.

Code and Data Availability

The source code and processed dataset used in this study are publicly available at https://github.com/suhelahamed/MSMU_SST_CO

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy, have been completely witnessed by the authors.

Funding

This study was supported by the research facilities of Guru Ghasidas Vishwavidyalaya (a central university), Bilaspur (C.G., India).

References

Bi, S., & Zhang, Y. J. (2018). Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. IEEE Transactions on Wireless Communications, 17(6), 4177–4190. https://doi.org/10.1109/TWC.2018.2821664

Chaharsooghi, S. K., & Taheri, Z. (2015). Developing a multi-issue and flexible negotiation mechanism based on multi-agent systems in the automated interchange. Journal of Information Technology Management, 6(4), 589–606. https://doi.org/10.22059/jitm.2015.52452

Chen, X., Jiao, L., Li, W., & Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Transactions on Networking, 24(5), 2795–2808. https://doi.org/10.1109/TNET.2015.2487344

Hu, Q., Cai, Y., Yu, G., Qin, Z., Zhao, M., & Li, G. Y. (2019). Joint offloading and trajectory design for UAV-enabled mobile edge computing systems. IEEE Internet of Things Journal, 6(2), 1879–1892. https://doi.org/10.1109/JIOT.2018.2878876

Huang, L., Feng, X., Feng, A., Huang, Y., & Qian, L. P. (2018). Distributed deep learning-based offloading for mobile edge computing networks. Mobile Networks and Applications. Advance online publication. https://doi.org/10.1007/s11036-018-1177-x

Huang, L., Feng, X., Zhang, L., Qian, L., & Wu, Y. (2019). Multi-server multi-user multi-task computation offloading for mobile edge computing networks. Sensors, 19(6), 1446. https://doi.org/10.3390/s19061446

Huang, L., Bi, S., & Zhang, Y. J. (2020). Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Transactions on Mobile Computing, 19(11), 2581–2593. https://doi.org/10.1109/TMC.2019.2928811

Jeong, S., Simeone, O., & Kang, J. (2018). Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. IEEE Transactions on Vehicular Technology, 67(3), 2049–2063. https://doi.org/10.1109/TVT.2017.2706308

Jain, A. & Raja, R. (2023). Automated Novel Heterogeneous Meditation Tradition Classification via Optimized Chi-Squared 1DCNN Method. Journal of Information Technology Management, 15(Special Issue: Intelligent and Security for Communication, Computing Application (ISCCA-2022)), 1-22. doi: 10.22059/jitm.2023.95223

Kashyap, R., Jain, A., Ahamad, S., Soni, A.S. & Behar, N. (2025). Generative AI-Driven Hyper Personalized Wearable Healthcare Devices: A New Paradigm for Adaptive Health Monitoring. Journal of Information Technology Management, 17(Special Issue on SI: Intelligent Security and Management), 130-154. doi: 10.22059/jitm.2025.102925

Liang, R. (Qiyu3816). (2024). MSMU-CO: A dataset generation tool based on MCMF for multi-server multi-user computation offloading [Computer software]. GitHub. https://github.com/qiyu3816/MSMU-CO

Liang, R., Yang, B., Chen, P., Cao, X., Yu, Z., Debbah, M., Poor, H. V., & Yuen, C. (2024). A dataset for multi-server multi-user computation offloading [Data set]. IEEE Dataport. https://doi.org/10.21227/386p-7h83

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera Y. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International

Conference on Artificial Intelligence and Statistics (AISTATS 2017) (pp. 1273–1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html

Patnaik, A., Mallik, B. & Vamsi Krishna, M. (2021). Cluster Node Migration Oriented Holistic Trust Management Protocol for Ubiquitous and Pervasive IoT Network. Journal of Information Technology Management, 13(1), 100-118. doi: 10.22059/jitm.2021.80027

Wang, Z., Zhao, W., Hu, P., Zhang, X., Liu, L., Fang, C., & Sun, Y. (2024). UAV-assisted mobile edge computing: Dynamic trajectory design and resource allocation. Sensors, 24(12), 3948. https://doi.org/10.3390/s24123948

Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951

Bibliographic information of this paper for citing:

Ahamed, Suhel; Dewangan, Amit Kumar; Jain, Abhishek; Behar, Nishant & Shrivastava, Pankaj Shankar (2026). A Novel Edge AI Framework for Selective Subtask Offloading in UAV-Assisted Multiuser Multiserver Mobile Edge Computing. *Journal of Information Technology Management*, 18 (4), 44-58. https://doi.org/10.22059/jitm.2026.108914