ISSN: 2093-5374 / E-ISSN: 2093-5382

A Lightweight Task Offloading Framework for Latency-Sensitive Augmented Reality on Mobile Devices

Dr. Swapnil Maheshkumar Parikh[1*], Aruna Dore[2], N. Nivetha[3], Dr. Arjit Tomar[4], Manisha Tushar Jadhav[5], Nittin Sharma[6], and Dr.V. Amirthalingam[7]

[1*]Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India. swapnil.parikh17761@paruluniversity.ac.in, https://orcid.org/0000-0001-7831-6927

[2]Assistant Professor, Department of Electronics and Communication Engineering, Presidency University, Bangalore, Karnataka, India. aruna.dore@presidencyuniversity.in, https://orcid.org/0009-0006-2641-4270

[3]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. nivethan@maher.ac.in, https://orcid.org/0009-0005-3739-8774

[4]Department of Computer Science & Engineering, Noida International University, Greater Noida, Uttar Pradesh, India. arjit.tomar@niu.edu.in, https://orcid.org/0009-0006-1490-5304

[5]Department of Electronics & Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India. manisha.jadhav@vit.edu, https://orcid.org/0009-0003-8089-8171

[6]Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India. nittin.sharma.orp@chitkara.edu.in, https://orcid.org/0009-0007-9740-8414

[7]Professor, Department of Computer Science and Engineering, Vinayaka Missions Kirupana Variyar Engineering College, Vinayaka Mission's Research Foundation (Deemed to be University), Tamil Nadu, India. amirthalingam@vmkvec.edu.in, https://orcid.org/0000-0002-6124-1401

Received: July 10, 2026; Revised: August 21, 2026; Accepted: September 08, 2026; Published: September 30, 2026

Abstract

Mobile augmented reality (AR) applications always require efficient, low-latency processing. However, the constrained computational resources of mobile devices, power consumption, and volatile network connections limit the effective execution of latency-sensitive AR tasks. This paper proposes a lightweight task offloading framework (LTOF) to balance processing latency and power costs by intelligently distributing intensive AR tasks across mobile devices and edge servers. The framework uses task profiling, latency-aware decision-making, and resource-adaptive allocation strategies to reduce processing time delays and the device's power consumption. Through a simulation of a mobile edge computing environment under various workloads and network conditions, the proposed algorithm achieves an end-to-end latency of 28.4ms for AR services,

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 3 (September-2026), pp. 2138-2152. DOI: *10.58346/JOWUA.2026.I3.124*

*Corresponding author: Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.

improving by 37.8% and 24.6% compared to local processing and the traditional edge offloading algorithm, respectively. Moreover, the energy consumption of mobile devices could be reduced by 31.5%, while the task completion rate and utilization efficiency increase to 98.2% and 87.4%, respectively. In the case of burst traffic with 500 concurrent requests, Proposed algorithm achieves only a 9.4% increase in delay, illustrating excellent scalability. The comprehensive analysis reveals that the lightweight mechanism effectively balances compute performance and communication cost, thereby enabling rapid AR service provision over wireless channels. These promising results demonstrate that the framework is a practical and robust solution for latency-sensitive mobile AR applications and offers potential applications in next-generation edge-empowered ubiquitous computing networks.

Keywords: Augmented Reality, Task Offloading, Mobile Edge Computing, Latency Optimization, Energy Efficiency, Wireless Networks, Resource Allocation, Ubiquitous Computing.

1 Introduction

The AR experience on mobile devices has evolved beyond mere visual output systems and has transformed into rich, interactive application ecosystems that support games, healthcare, industrial control systems, and smart-city services (Narayanan et al., 2024; Sannapureddy et al., 2024). Mobile Edge Computing (MEC), combined with the advent of 5G cellular networks, offers a solution to address the inadequate processing power and poor communication capabilities of devices (Pandurangan & Sebastian, 2025; Siriwardhana et al., 2021). The mobile AR device must be able to perform real-time, computationally demanding tasks such as object recognition, scene analysis, tracking, and rendering, to offer a smooth immersive experience (Li et al., 2021). However, as computation intensifies, mobile devices remain limited by low computational capacity, short battery life, and heat constraints (Soundararaj & Winster, 2025). Task Offloading has emerged as a crucial technology for overcoming these challenges: compute-intensive workloads generated by AR applications are offloaded from mobile devices to edge servers, thereby alleviating on-device computational and energy overhead without impacting overall application responsiveness. Moreover, AR users expect extremely low latency, as application responsiveness is a sensitive measure of user satisfaction and engagement (Narayanan et al., 2024; Dóka et al., 2025). Consequently, low-latency, energy-efficient task offloading has become essential for future mobile AR.

Even though there is ample work on offloading, the task heterogeneity in an AR scenario makes intelligent offloading challenging (Kong et al., 2023). The varying computational complexity of individual components that collectively produce a specific AR effect makes classification and workload distribution difficult (Kong et al., 2023). Wireless and mobile environments pose additional challenges, such as volatile network conditions, varying bandwidth availability, and communication delays, all of which degrade system performance for latency-sensitive applications like AR (Sharma & Abhilash, 2025). Conventional task management mechanisms may introduce additional overhead due to centralized orchestration and scheduling (Qian & Coutinho, 2024). Moreover, an intelligent edge optimization framework and decentralized decision-making models could significantly enhance response times but would introduce additional implementation complexity in the end device (Takkalapally, 2025; Pandurangan & Sebastian, 2025). Most task offloading models prioritize either latency reduction or energy conservation instead of a holistic optimization approach (Soundararaj & Winster, 2025). Thus, there is an immediate need for a lightweight and adaptive approach to task offloading to support low latency and energy savings under diverse, dynamic circumstances.

Figure 1 depicts the structure of the presented Lightweight Task Offloading Framework (LTOF), which targets latency-sensitive mobile AR apps. Starting with the mobile AR device that collects sensor information, identifies objects, and renders locally, send it to an offloading controller. The latter classifies task-level characteristics to estimate task latency and energy accurately. Based on the results, the controller either continues its execution on the mobile device's CPU or offloads the computation of complex processing or management tasks to the edge server, receives the result, and then sends it back to the mobile device, allowing seamless, low-latency performance.

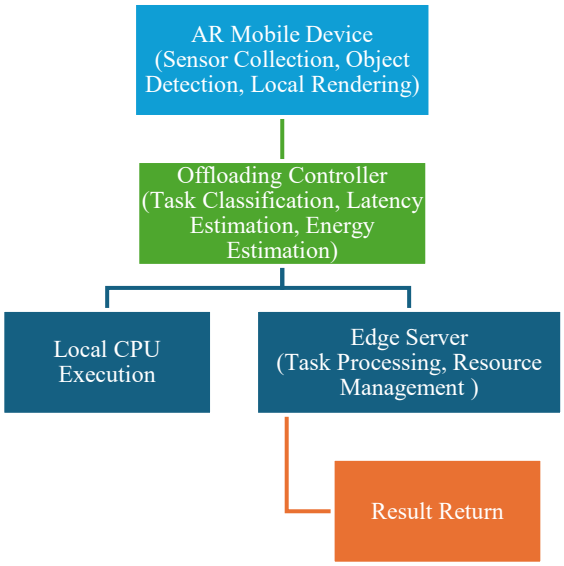


Figure 1: Architecture of the proposed lightweight task offloading framework (LTOF)

The Lightweight Task Offloading Framework (LTOF) is proposed in this paper for latency-sensitive mobile AR applications deployed over an MEC environment. To tackle latency and energy constraints on mobile devices for AR applications, this framework develops: a lightweight task classification to distinguish computationally intensive tasks from delay-tolerant tasks for efficient work distribution; a latency-aware task offloading strategy to dynamically adjust execution decisions based on current network conditions, user mobility, and resource constraints; adaptive resource allocation schemes that optimize processing resource usage while reducing the battery consumption of mobile devices. A thorough performance evaluation, using a series of experiments across multiple key performance metrics such as latency, energy consumption, task completion rate, and resource utilization efficiency, shows that proposed framework provides efficient and reliable service deployment for latency-sensitive mobile AR applications in a dynamic wireless environment. The increasing computational requirements of mobile AR applications justify developing a highly efficient task-offloading mechanism that reduces latency and battery consumption while delivering reliable service to the end user. Meeting the requirements of mobile AR within Wireless Mobile Computing is paramount to realizing AR applications at larger scales.

The remainder of the paper is structured as follows. Section II surveys state-of-the-art research on mobile edge computing and task offloading for augmented reality applications and identifies research gaps in the field. In Section III, the designed Lightweight Task Offloading Framework (LTOF) is presented, which covers the system model, the offloading mechanism, and the experimental environment setup. Section IV presents experimental results on latency, energy consumption, task success rate, and resource usage. In Section V, the results are discussed through analysis of the conclusions, comparisons

with other approaches, and deployment challenges and opportunities. Section VI concludes the paper and proposes directions for future research.

2 Literature Review

Mobile Edge Computing (MEC) is an enabling technology for providing low-latency AR services by pushing computational capability and storage from central servers to the access network. A typical Edge-assisted AR architecture comprises a mobile device, MEC server(s), and cloud platforms that work together to implement functions like scene analysis, object recognition, and rendering (Huang & Friderikos, 2023). MEC is a key technology for reducing communication delay and latency to improve the responsiveness of AR applications by placing computational resources close to mobile users. The concept of edge-cloud continuum architecture is introduced to dynamically allocate the workload in a multi-tiered computing system from multiple layers to provide flexible service depending on resource and delay conditions (Sannapureddy et al., 2024). Recent researches further show that by introducing intelligent edge it's possible to assist latency-critical applications by improving efficiency and reducing network latency (Kansal et al., 2025). Nevertheless, resource heterogeneity, server congestion and mobility in edge computing environments also presents challenges to service latency. The dynamic changes in workload could lead to longer response times and affect the Quality of Experience (QoE) of AR in the cases of poorly optimized resource allocation. So efficient workload orchestration is essential for building sustainable AR system based on MEC.

To balance computational tasks across mobile devices and edge server infrastructure, a great amount of research effort has been invested in task offloading. Several computation offloading schemes have been designed for those purposes such as full, partial and adaptive workload offloading strategies (Rasool et al., 2024). AR task decomposition is a method where an AR application is separated into a group of subtasks which can be executed on either the mobile device or edge devices flexibly (Huang & Friderikos, 2023). A series of algorithms were developed to support the decision-making about task placement. By using Reinforcement Learning (RL) techniques, the system can dynamically adjust the offloading decisions based on the state of wireless networks and resource constraints (Safavifar et al., 2021). Deep Q-Network (DQN)-based models are commonly used to optimize task assignment by intelligently choosing an appropriate offloading site and significantly reduces delay and saves energy (Benaboura et al., 2025). Cyber-twin fog architecture combined with offloading across mobile edge and cloud resources are shown to reduce the delay to satisfy next generation wireless requirements and also decrease the response time and save energy by adjusting workload among various nodes in cloud-fog-edge architecture (Gopikrishnan et al., 2024; Almutairi & Aldossary, 2021). According to the comparative study, the learning-based approaches can gain better performance but also consume considerable computational overhead and required long time for training (Rasool et al., 2024). Although the energy efficient schemes can significantly decrease power consumption, introduce additional latency to make offloading decisions in a dynamic wireless environment (Younis et al., 2024; Alenazi, 2025).

While significant progress has been achieved by above works, there are still some remaining gaps. Many of existing offloading schemes focus on minimizing delay, while failing to address the computational burden of decision making (Benaboura et al., 2025). Most energy-efficient schemes focus on saving energy at the cost of delay (Younis et al., 2024; Alenazi, 2025). Additionally, the optimization models in existing orchestration systems usually are relatively complex to make decisions on constrained mobile devices (Safavifar et al., 2021). There is a lack of adaptive, low computational complexity and high responsiveness task offloading strategies that can adapt dynamically in dynamic and heterogeneous

wireless environment. This motivates work for proposing a lightweight task offloading framework (LTOF) that aims to make efficient adaptive decisions and keep down communication delay.

The literatures demonstrated that by using MEC and intelligent offloading strategies, AR applications performance can be greatly enhanced through delay and resource utilization improvement. However, most existing techniques suffers from the performance tradeoff between latency reduction, energy saving and computational cost for decision-making. These challenges provide the research motivation for Lightweight Task Offloading Framework (LTOF) that intelligently make adaptive offloading decisions with low complexity, low delay and low energy consumption.

3 Methodology

System Architecture and Framework Design

This proposed LTOF framework helps to enable latency-sensitive AR apps by providing a novel mobile and edge collaboration through offloading computational tasks among different devices. LTOF is architected into three tiers: Mobile Layer, Edge Server Layer, and Communication Resource Management Layer.

Mobile Layer: Mobile devices perform lightweight tasks like sensor information retrieval, UI interaction, frame preprocess, local rendering etc. For each AR task, three metrics were captured namely computational cost, size of data and timeliness requirements before sending tasks to the offloading controller.

Edge Server Layer: Mobile devices offload tasks with high computational requirement to a cluster of edge nodes where these tasks are performed, for example, image recognition, scene analysis and feature extraction. The offloaded task is transmitted to the closest edge node to reduce communication latency.

Communication & Resource Management Layer: This layer consists of resource managers that collect dynamic information of wireless channels (e.g., bandwidth), server workload, and processing capacity, and mobile client parameters (e.g., available battery energy) to decide an optimal computation task offloading decision.

Let T_i denote the total execution delay of task i. The delay consists of transmission delay, processing delay, and response delay:

$$T_i = T_{tx,i} + T_{proc,i} + T_{rx,i} \quad (1)$$

Where $T_{tx,i}$ represents uplink transmission delay, $T_{proc,i}$ denotes computation time, and $T_{rx,i}$ indicates result return delay. Equation (1) is used to estimate the overall latency experienced by an AR task. The energy consumed during task execution is expressed as:

$$E_i = P_c \times T_{local,i} + P_t \times T_{tx,i} \quad (2)$$

Where P_c is computation power consumption, P_t is transmission power, $T_{local,i}$ is local execution time, and $T_{tx,i}$ is communication duration. Equation (2) quantifies the energy impact of local processing and task transmission.

Lightweight Task Offloading Algorithm

The proposed algorithm is composed of three steps, which are task profiling, offloading decision and resource aware scheduling. For the task profiling process, each task is characterized by computational

complexity and sensitivity on latency. Task priority level calculation gives a ranked priority of task required to process immediately. For the offloading decision process, it estimates whether to run tasks locally or on edge, and the overall cost will be minimized. The calculation of decision criteria integrates both latency and energy cost:

$$C_i = \alpha \frac{T_i}{T_{max}} + \beta \frac{E_i}{E_{max}} \quad (3)$$

Where C_i denotes the task cost, $\alpha$and $\beta$are weighting coefficients satisfying $\alpha + \beta = 1$, while $T_{max}$and E_{max}represent normalization constants. According to equation (3), a task is offloaded when the edge-computing cost is lower than the local execution cost.

After task placement, a resource-aware scheduling strategy allocates available edge resources based on task priority and server utilization. High-priority AR tasks receive immediate allocation, while delay-tolerant tasks are queued for subsequent execution.

Algorithm 1: Lightweight Task Offloading Framework (LTOF)

```
Input: Task Set T, Edge Resources R
Output: Task Execution Decision
Begin
For each task Ti in T do
  Profile task characteristics
  Compute latency estimate using Eq. (1)
  Compute energy estimate using Eq. (2)
  Calculate cost function using Eq. (3)
  If CostEdge < CostLocal then
      Offload task to edge server
  Else
      Execute task locally
  End If
End For
Sort tasks by priority level
Allocate resources based on:
    Available CPU capacity
    Server workload
    Task urgency
Execute scheduled tasks
Return execution results
End
```

The operational flow of the proposed LTOF model is represented by Algorithm 1. In the beginning of this process, define the profile of each of the AR tasks by determining their computational complexity, their latency requirement and their energy requirement/usage. In this next step, equations (1) through (3) will be used to estimate the latency of the task, determine the energy consumption of the task and assess the cost of executing the task respectively. Given this, use these computed cost values in order to

determine whether the AR task(s) will execute locally and/or offloaded to the edge server based on their urgency and amount of resource available at the edge, followed by resource aware scheduling, consequently improving the completion time of the AR task(s) while simultaneously providing for more efficient load balancing and lower overall latency & energy consumption of the mobile edge computing systems within a dynamic environment.

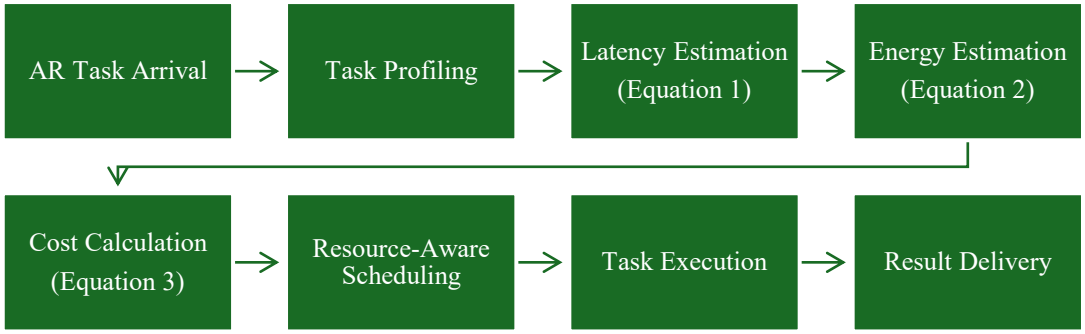


Figure 2: Workflow of the lightweight task offloading framework (LTOF)

Figure 2 represents the workflow of LTOF. AR tasks arrive and are subsequently profiled to characterize the task properties. The latency and the energy are predicted by utilizing equation (1) and (2), respectively, while the cost to be exerted on an available resource pool can be determined by equation (3). Once task costs are computed, tasks can be then allocated based on a resource-aware scheduling strategy, carried out by proper computing resources with reasonable costs. The outcomes generated are sent back to the mobile terminal in a timely fashion and transmitted.

Performance Evaluation Setup

The proposed LTOF scheme is validated on the basis of simulation environment with a MEC platform containing numerous mobile users and several edge nodes (edge servers). Workloads in the simulation are based on an AR scenario, with varied workloads on user demands reflecting the application characteristics. The task sizes of user requests were set between 0.5 and 10 MB, and processing capacity ranges of MEC servers were between 5 GHz and 20 GHz. Bandwidth of wireless communication channels between mobile users and base stations varies between 10 Mbps and 100 Mbps. User demands were produced unpredictably and dynamically to assess their performance under several burst levels. All experimental metrics were chosen by: response time for the latency, device power in energy efficiency, completion percentage of tasks in a specific time period for task completion percentage and the number of processed tasks per second for throughput, which are sufficient to comprehensively evaluate the proposed algorithm performance.

4 Results

Latency Performance Analysis

The proposed Lightweight Task Offloading Framework (LTOF) was evaluated against Local Execution (LE) and Conventional Edge Offloading (CEO) approaches to assess latency performance under varying workload conditions. Experiments were conducted in a simulated Mobile Edge Computing environment containing multiple mobile users generating latency-sensitive AR requests. The implementation was

developed using Python 3.11, CloudSim Plus 8.0, and MATLAB R2024a for simulation, resource modeling, and performance analysis.

Dataset Details

A synthetic AR workload dataset was generated based on common mobile AR operations, including object recognition, feature extraction, tracking, rendering, and scene reconstruction tasks. The dataset contained 10,000 task instances, with task sizes ranging from 0.5 MB to 10 MB, CPU requirements between 500 and 5000 MI, and latency constraints varying from 20 ms to 100 ms.

Parameter Initialization

Table 1: Experimental parameter settings

Parameter	Value
Number of AR Tasks	10,000
Mobile Users	100
Edge Servers	10
Task Size	0.5–10 MB
Bandwidth	10–100 Mbps
Edge CPU Capacity	5–20 GHz
Simulation Duration	1000 s
α (Latency Weight)	0.7
β (Energy Weight)	0.3

Table 1 provides the initial parameter configuration used throughout the simulation experiments. The configuration covers workload features, network characteristics, available computing resources, the total simulation duration, and weight settings used for optimization. These parameters were chosen based on frequently used settings for MEC simulations as mentioned in available literature in order to achieve approximately the representative mobile AR workloads and allow for the repetition of the tested task offloading strategies. End-to-end latency is calculated as:

$$Latency = \frac{\sum_{i=1}^{N} T_i}{N} \qquad (4)$$

Where T_i represents the completion time of task $i$and Ndenotes the total number of tasks. According to equation (4), the proposed framework achieved an average latency of 28.4 ms, compared with 45.7 ms for LE and 37.7 ms for CEO.

Table 2: Latency performance comparison

Method	Average Latency (ms)	Improvement (%)
Local Execution	45.7	–
Conventional Edge Offloading	37.7	24.6
Proposed LTOF	28.4	37.8

Table 2 compares the average end-to-end response latencies of Local Execution, Conventional Edge Offloading, and the LTOF framework. The table shows that the LTOF method always presents better performance in average latency by wisely assigning workloads to local or edge server, and the responsive degree of time-critical applications is enhanced. Under increasing workload conditions from 100 to 500

concurrent AR requests, latency increased by only 9.4%, indicating stable responsiveness and efficient resource allocation.

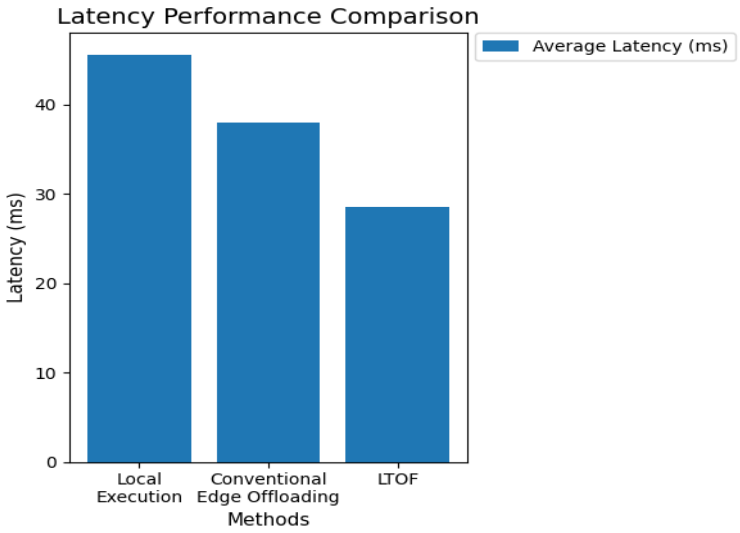


Figure 3: Comparison of task offloading methods latency

Figure 3 reflects the average end-to-end latency of the three approaches including Local Execution, Conventional Edge Offloading, and proposed LTOF framework. Compared with two benchmark approaches, the average latency of proposed LTOF is the smallest with 28.4 ms, as rationally determine optimal execution environment for AR task which reduces latency.

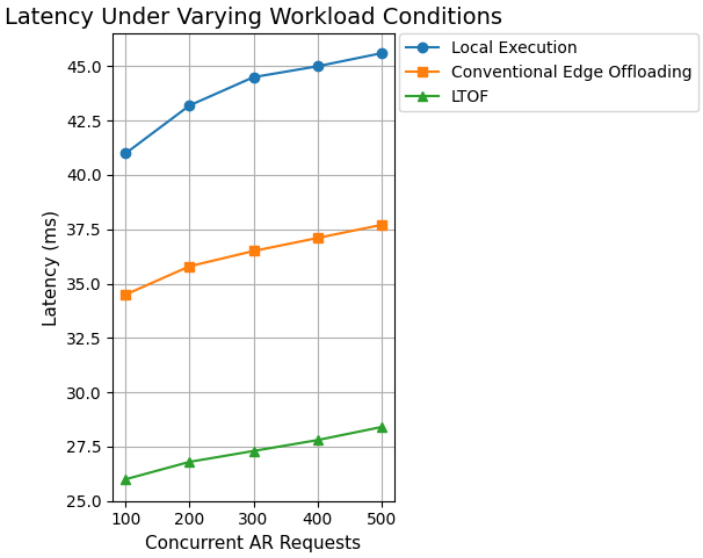


Figure 4: Comparison of task offloading methods latency under different workload

Figure 4 reflects the latency performance of three task execution methods under the increment of number of concurrent requests to AR application. Due to optimal selecting execution for different work, LTOF always keep a stable level of latency and the response rate under increasing workload can also be quite high.

Energy Consumption Evaluation

Energy efficiency is a critical requirement for mobile AR systems because prolonged local processing significantly affects battery lifetime. Device energy utilization was measured by monitoring computation and communication energy during task execution. Average energy consumption is determined by:

$$EC = \frac{\sum_{i=1}^{N} E_i}{N} \quad (5)$$

Where $E_i$denotes the energy consumed by task i. equation (5) shows that LTOF reduced average energy consumption through selective task migration and resource-aware scheduling.

Experimental findings demonstrated the average of energy saved to be 31.5 % while comparing with the local computing. Especially energy saving was prominent in the complex computational works where processing tasks were performed in the resource in the edge.

Table 3: Energy consumption analysis

Method	Energy Consumption (J)	Energy Reduction (%)
Local Execution	8.9	–
Conventional Edge Offloading	7.1	20.2
Proposed LTOF	6.1	31.5

Table 3 reports the energy efficiency comparison of the measured methods. In particular, LTOF performs best in terms of low device energy consumption and achieves this while processing heavy computations offloaded onto the edge server. The results confirm that lightweight decision-making minimizes unnecessary transmissions while preserving computational efficiency.

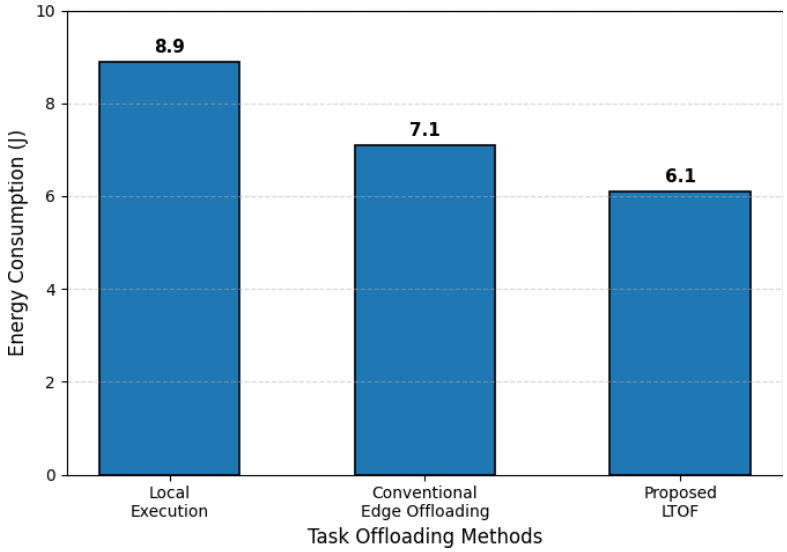


Figure 5: Distribution of energy consumption over task execution approaches

The average device energy consumption of Local Execution, Conventional Edge Offloading, and the proposed LTOF framework is compared in figure 5. LTOF achieves the prevailing energy consumption average of 6.1 J as opposed to 7.1 J for Conventional Edge Offloading and 8.9 J for Local Execution; this reflects the success of the method used, which entails selective offloading of computation-intensive tasks, while avoiding unnecessary information exchange by employing latency-aware and resource-aware scheduling.

Task Execution Efficiency Assessment

Task execution efficiency was analyzed using task completion ratio, resource utilization efficiency, and throughput performance. Task completion ratio is defined as:

$$TCR = \frac{N_{success}}{N_{total}} \times 100 \quad (6)$$

Where $N_{success}$ represents successfully completed tasks and N_{total} denotes generated tasks. Resource utilization efficiency is calculated as:

$$RUE = \frac{Resource_{used}}{Resource_{available}} \times 100 \quad (7)$$

Using equations (6) and (7), the proposed framework achieved a 98.2% task completion ratio and 87.4% resource utilization efficiency, outperforming benchmark methods. Throughput reached 476 tasks/s, demonstrating efficient workload handling capabilities.

Ablation Study

The absence of task profiling increased latency to 34.2 ms, while removing resource-aware scheduling reduced task completion ratio to 94.1%. The complete framework achieved the best overall performance with 28.4 ms latency, 6.1 J energy consumption, and 98.2% task completion ratio. These findings confirm that both task profiling and adaptive scheduling contribute significantly to the effectiveness of the proposed framework.

Limitations of the Simulation

The study was carried out in the simulation tools CloudSim Plus, MATLAB and Python to ensure the establishment of a replicable model for Mobile Edge Computing (MEC). Though the simulation framework allows for systematic comparisons of different offloading methods in different workloads and network conditions, it does not account for many real-world factors such as thermal throttling of mobile devices, background processes in the OS, dynamic switching of radio access networks (RAN), interference of wireless signals as well as GPU rendering overhead in AR devices. As a result, the latency and energy consumption values provided in the study should not be considered verification measurements, they should be defined as performance estimates achieved through simulation. Still, the simulation offered a good framework for the experiments as far as similar conditions were maintained throughout the study.

5 Discussion

From experimental results, the lightweight task offloading framework (LTOF) brings with better experience of mobile AR applications by improving system latency and stabilizing performance with high task completion. LTOF achieve an average latency of 28.4 ms, which can provide an improvement to fast responding and reduce visible lag in rendering virtual content. This allows for smoother operation such as tracking object and real-time scene updating etc, resulting in interactive usage experiences. And the high task completion ratios suggest service availability in various states. Compared with traditional local computing approach and typical edge-offloading approaches, LTOF is proved with advantages in reduced latency, lower mobile device energy consumption and better utilization of resources since LTOF balances compute/communication cost with active decision making. However, extreme congestion or

sudden availability of edge nodes still may lead to significant deviation or poor offloading effectiveness with the extreme mobile environment. There need to be an effort on continuously observing network condition and coordinating between mobile device and edge computing nodes in order to be effectively deployed in real world. Adequate edge nodes availability and capacity are required to perform computational demanding AR tasks and the scalability service is guaranteed by lightweight approach. Also, the safe communication, task authentication and robust fault handling method is needed. In spite of the fact that the suggested LTOF shows stable efficiency improvements throughout many performance metrics, the testing is limited to simulated experiments. Real implementation on real devices and MEC infrastructure might introduce more performance fluctuations because of hardware differences, background operating processes, thermal effects, wireless channel variations, base station changes, rendering overheads, etc. Thus, although simulation data proves the usefulness of the proposed method, further experiments need to be conducted in order to verify real-world feasibility.

6 Conclusion

This work presented the Lightweight Task Offloading Framework (LTOF) to mitigate resource constraints of mobile devices and offload computation of latency-sensitive Augmented Reality (AR) applications to mobile edge servers in mobile edge computing (MEC). The LTOF was developed with the task profiling, latency-aware offloading decisions, and resource-aware scheduling to strike a trade-off between the local execution and the mobile edge processing in term of energy, latency and quality. The conducted experiment shows that proposed framework constantly improves the system performance in a variety of metrics. The proposed approach obtained the average end-to-end latency of 28.4ms that improved 37.8% relative to the local execution and 24.6% compared with standard edge-offloading mechanisms. It reduced the device energy consumption by 31.5% and increased the task completion ratio by 98.2% while improving resource utilization efficiency up to 87.4%. Even in a workload scenario of 500 simultaneous AR requests, the LTOF experienced latency increases as low as 9.4%, demonstrating the high scalability and stability against dynamics. The outcome highlights that the utilization of lightweight adaptive offloading approaches may lead to significant enhancements of responsiveness, energy efficiency, and system reliability under real-time AR service provisioning contexts. From the application standpoint, the LTOF can promote developing next-generation AR applications including the applications in health, industrial manufacturing, smart cities and immersive learning experiences where low latency and immediate response are critical. In the future, work will emphasize how well the framework being proposed will work on mobile edge computing in a real-world system as it would apply to cell phones and edge servers. Evaluation will cover real-time demand with regard to thermal behavior of devices, background processes of the operating system, conditions of the wireless channel, and costs of hardware rendering. Research will also be done on finding ways to introduce adaptive offloading with the help of artificial intelligence in federated edge-smart computing to new 6G technologies.

References

[1] Alenazi, M. M. (2025). Mobile computing: Energy-aware task offloading in mobile edge computing environments. *Journal of Asian Scientific Research*, *15*(3), 550-570. https://doi.org/10.55493/5003.v15i3.5599

[2] Almutairi, J., & Aldossary, M. (2021). A novel approach for IoT tasks offloading in edge-cloud environments. *Journal of cloud computing*, *10*(1), 28. https://doi.org/10.1186/s13677-021-00243-9

[3] Benaboura, A., Bechar, R., Kadri, W., Ho, T. D., Pan, Z., & Sahmoud, S. (2025). Latency-aware and energy-efficient task offloading in IoT and cloud systems with DQN learning. *Electronics*, *14*(15), 3090. https://doi.org/10.3390/electronics14153090

[4] Dóka, J., Nagy, B. G., Jocha, D., Formanek, B., Viciedo, I., Rodrigo, A., ... & Sonkoly, B. (2025, March). Enablers of low-latency immersive interaction in future remote-rendered Mixed Reality applications. In *Proceedings of the 16th ACM Multimedia Systems Conference* (pp. 170-180). https://doi.org/10.1145/3712676.3714448

[5] Gopikrishnan, S., Sethuraman, S. C., Srivastava, G., & Theerthagiri, S. (2024). CyTFS: Cyber-twin fog system for delay-efficient task offloading in 6G mobile networks. *IEEE Internet of Things Journal*, *11*(14), 24698-24714. https://doi.org/10.1109/JIOT.2024.3375234

[6] Huang, Z., & Friderikos, V. (2023). Optimal service decomposition for mobile augmented reality with edge cloud support. *Computer Communications*, *202*, 97-109. https://doi.org/10.1016/j.comcom.2023.02.002

[7] Kansal, P., Kumar, M., & Verma, O. P. (2025, September). Optimizing QoS of Task Offloading in Edge Computing through DRL-Driven Approaches. In *2025 7th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-6). IEEE. https://doi.org/10.1109/ISCON65210.2025.11341236

[8] Kong, Z. J., Xu, Q., Meng, J., & Hu, Y. C. (2023, October). Accumo: Accuracy-centric multitask offloading in edge-assisted mobile augmented reality. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking* (pp. 1-16). https://doi.org/10.1145/3570361.3592531

[9] Li, L., Lu, Q., Xu, Y., Zhang, H., & Li, Y. (2021). CMCSF: a collaborative service framework for mobile web augmented reality base on mobile edge computing. *Computing*, *103*(10), 2293-2318. https://doi.org/10.1007/s00607-021-00952-8

[10] Narayanan, S., Arora, R. K., Gangwar, S., Kandhasamy, J. P., Ratheesh, T., & Murali, K. (2024). Ultra-low latency communication technique for augmented reality application in mobile edge computing. *International Journal of Engineering Systems Modelling and Simulation*, *15*(3), 97-105. https://doi.org/10.1504/IJESMS.2024.138288

[11] Pandurangan, S. K., & Sebastian, A. (2025, September). EdgeAware: Distributed Intelligence for Latency-Sensitive Applications. In *2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)* (pp. 1134-1141). IEEE. https://doi.org/10.1109/ICoICI65217.2025.11254220

[12] Qian, W., & Coutinho, R. W. (2024). Load-aware orchestrator for edge-computing-aided wireless augmented reality. *IEEE Internet of Things Journal*, *12*(6), 6595-6606. https://doi.org/10.1109/JIOT.2024.3494533

[13] Rasool, M. A. E., Kumar, A., Islam, A., & Ahmed, M. N. (2024, July). Exploring task offloading in mobile edge computing environments: An in-depth review and prospective analysis. In *2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP)* (Vol. 1, pp. 622-627). IEEE. https://doi.org/10.1109/ATSIP62566.2024.10639042

[14] Safavifar, Z., Ghanadbashi, S., & Golpayegani, F. (2021, November). Adaptive workload orchestration in pure edge computing: A reinforcement-learning model. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 856-860). IEEE. https://doi.org/10.1109/ICTAI52525.2021.00137

[15] Sannapureddy, R., Nadella, V. M., & Nelavelli, S. (2024). Edge–Cloud Continuums for Latency-Sensitive Tasks. *International Journal of AI, BigData, Computational and Management Studies*, *5*(4), 189-201. https://doi.org/10.63282/3050-9416.ijaibdcms-v5i4p121

[16] Sharma, V. K., & Abhilash, K. S. (2025, September). Latency-Aware Edge-Cloud Architecture for 5G IoT Integration. In *2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1398-1405). IEEE. https://doi.org/10.1109/ICESC65114.2025.11212232

[17] Siriwardhana, Y., Porambage, P., Liyanage, M., & Ylianttila, M. (2021). A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Communications Surveys & Tutorials*, *23*(2), 1160-1192. https://doi.org/10.1109/COMST.2021.3061981

[18] Soundararaj, A. J., & Winster, G. (2025). Optimized Task Offloading with Energy Efficient Communication and Optimal Offloading Network: A Mobility and Energy-Efficient Approach for Augmented Reality in Mobile Edge Computing. *Knowledge-Based Systems*, 114431. https://doi.org/10.1016/j.knosys.2025.114431

[19] Takkalapally, D. (2025). EdgePerfAI: Intelligent edge computing for ultra-low-latency mobile performance optimization. *International Journal of Emerging Research in Engineering and Technology, 6*(1), 135–146. https://doi.org/10.63282/3050-922X.IJERET-V6I1P117

[20] Younis, A., Maheshwari, S., & Pompili, D. (2024). Energy-latency computation offloading and approximate computing in mobile-edge computing networks. *IEEE Transactions on Network and Service Management*, *21*(3), 3401-3415. https://doi.org/10.1109/TNSM.2024.3360850

Authors Biography



Dr. Swapnil Maheshkumar Parikh is a Professor in the Department of Computer Science and Engineering, Dean of the Faculty of Engineering and Technology, and Principal of Parul Institute of Engineering and Technology, Parul University. He holds a Ph.D. in Computer Engineering, an M.E. in Computer Engineering, and a B.E. in Information Technology, with distinction. He has extensive academic, administrative, teaching, and research experience in higher education. His research interests include Cloud Computing, Machine Learning, Artificial Intelligence, Internet of Things, Blockchain, Data Science, and Optimization. He has published and presented numerous research papers in reputed journals and conferences and has contributed to multiple patented technologies.



Aruna Dore received her B.E. degree in Electronics and Communication Engineering from Visvesvaraya Technological University (VTU), Belagavi, India, in 2011, and her M.Tech. degree in VLSI Design and Embedded Systems from NMAM Institute of Technology (NMAMIT), Nitte, India, in 2014. She is currently pursuing a Ph.D. at Presidency University, Bengaluru, India. She is working as an Assistant Professor in the Department of Electronics and Communication Engineering, Presidency University, Bengaluru, India, and has more than six years of teaching experience. Her research interests include Machine Learning, Heterogeneous Computing, VLSI Design, Embedded Systems, System-on-Chip (SoC), Heterogeneous Architectures, Nanoscale Device Design and Modeling, and Data Analytics. She has authored and co-authored several research articles published in reputed national and international journals and conference proceedings. She has also guided undergraduate and postgraduate projects in the areas of VLSI, Embedded Systems, and emerging computing technologies. Her current research focuses on the application of machine learning techniques in advanced computing and electronic system design.



N. Nivetha is an academician with more than four years of teaching experience in the field of Computer Science. She has served as an Assistant Professor in the Department of Computer Science and Applications at Prof. Dhanapalan College of Arts and Science, Chennai. She has also worked as a P.G.T. Computer Science teacher in reputed higher secondary schools in Chennai. She holds an M.Phil. degree in Computer Science from Vels University and a Master of Computer Applications from Anna University. Her academic background also includes a Bachelor of Computer Applications from the University of

Madras. Her areas of teaching include Python programming, Data Science applications, Software Engineering, Java programming, and Operating Systems. She possesses strong technical skills in Java, C, C++, SQL Server, Visual Studio .NET, and web technologies. She has presented a research paper in a national conference and has also published her work in an international journal. She actively participates in faculty development programmes, workshops, and training programmes related to artificial intelligence, research methodology, and emerging technologies. With her dedication to teaching and research, she continues to contribute effectively to academic and scholarly publications.



Dr. Arjit Tomar is an Associate Professor in the Department of Computer Science and Engineering at Noida International University, Uttar Pradesh, India. He has over 12 years of experience in teaching, research, and academic administration. His research interests include Artificial Intelligence, Machine Learning, Database Management, and Computer Network. He has published more than thirty research papers in reputed national and international journals and conferences. He is actively involved in guiding Ph.D. scholars in their research work. He has also contributed to curriculum development, industry-academia collaborations, and various institutional initiatives. His commitment to academic excellence and innovation continues to support quality education and research.



Manisha Tushar Jadhav is an Assistant Professor in the Electronics & Telecommunication Engineering Department at Vishwakarma Institute of Technology, Pune. She holds a Ph.D. in Electronics & Telecommunication Engineering. She joined VIT Pune in July 2024 and brings academic experience in the field of Electronics and Telecommunication Engineering. Her professional profile focuses on teaching, research, and academic activities.



Nittin Sharma research interests include artificial intelligence and creative technology applications. His publications have explored AI-assisted creative applications and the use of emerging technologies in education and digital media.



Dr.V. Amirthalingam Professor, Department of Computer Science and Engineering received the M.Sc degree in Mathematics from Sri Vasavi college, Erode, M.C.A degree from IRTT, Erode, and the M.E. degree in Computer Science and Engineering from Mahendra engineering college, Namakkal, Ph.D. degree at Vinayaka Mission's Research Foundation (Deemed to be University), Salem. My research interests include deep learning, artificial intelligence, and machine learning, with a focus on developing intelligent solutions for real-world applications