Pages 53-64 Volume 3, Number 1, 2026

COMPUTERS AND EDUCATION LETTERS

editor@celetters.com

www.celetters.com

ADAPTIVE EDGE-CLOUD TASK OFFLOADING FOR LOW-LATENCY REAL-TIME AI INFERENCE

Abdullah Farhan[1*], Esha Tariq[2]

[1]Department of Computer Science, Institute of Edge Computing and Artificial Intelligence, Lahore, Pakistan

[2] Department of Software Engineering, Center for Cloud Computing and Intelligent Systems, Islamabad, Pakistan

*Corresponding Author E-mail: abdullah.farhan@iecai.edu.pk

Abstract

Real-time artificial intelligence applications require fast, reliable, and resource-efficient processing to support latency-sensitive tasks such as autonomous monitoring, smart surveillance, industrial automation, healthcare decision support, and intelligent transportation. However, executing all AI workloads on edge devices can be limited by computational capacity, memory availability, and energy consumption, while full cloud execution may introduce network delay and service-level violations. This paper investigates an adaptive task offloading approach for reducing latency in real-time AI applications by dynamically distributing inference tasks between edge and cloud environments. The proposed approach evaluates workload complexity, network condition, device utilization, and latency threshold before deciding whether a task should be processed locally at the edge or transferred to the cloud. The results show that adaptive offloading achieves lower average latency compared with edge-only, cloud-only, and static offloading strategies. The system also improves SLA compliance under high workload conditions while maintaining acceptable energy consumption and inference accuracy. Experimental analysis across multiple workload types demonstrates that adaptive task placement is particularly effective when network conditions fluctuate and task complexity varies. Overall, the findings suggest that intelligent edge-cloud coordination can significantly improve the performance of real-time AI systems by balancing speed, resource usage, and reliability.

Article History

Received:
February 19, 2026

Revised:
March 18, 2026

Accepted:
April 17, 2026

Available Online:
June 30, 2026

Keywords: Adaptive Task Offloading; Edge Computing; Cloud Computing; Real-Time Ai; Latency Reduction.

INTRODUCTION

This increasing trend of intelligent applications, such as autonomous navigation and augmented reality, has pushed the paradigm shift to distributed computing systems and applications which demand very low latency in the range of milliseconds (Ullah et al., 2023; Wang et al., 2020).In many cases, cloud-based infrastructure is used remotely, which can lead to high communication latency, especially when bandwidth limitations and network congestion are involved, which severely affects the responsiveness demanded by mission-critical applications (Li et al., 2019; Premsankar et al., 2018). As such, the edge-cloud continuum has become an important architectural solution to overcome these latency issues by intelligently distributing compute and intelligence closer to the end-users and IoT devices (Ahmed et al., 2017; Aral et al., 2019; Firdose et al., 2021). The close range of the edge nodes offloads computation-heavy workloads from core nodes to the edge nodes, thereby eliminating the high latency of backhaul transmission to remote cloud data centers and enabling applications like autonomous driving, real-time surveillance, and immersive augmented or virtual reality to achieve faster inference times and lower data traffic on core networks (Li et al., 2019; Porambage et al., 2018). However, there are significant obstacles to meeting the vision of a responsive and efficient distributed infrastructure. This edge-cloud model is fundamentally limited by the vast diversity of user devices, the ever-changing and dynamic nature of network conditions, and the bursty and unpredictable traffic patterns typical of AI workloads today (Modi et al., 2025; Sahi et al., 2025). Additionally, the small computational and energy budgets on the edge nodes add complexity to the execution speed, accuracy, and power consumption (Baccour et al., 2022). However, traditional, off-the-shelf offloading heuristics may fail to guarantee the stability of the offloading process in such dynamic environments, as they do not include any mechanism to adapt to real-time changes in network load or device status (Almulifi & Kurdi, 2026; Firdose et al., 2021). Therefore, there is a fervent research need for creating adaptive and context-aware offloading frameworks that can intelligently and dynamically arrange task distribution and resource allocation between edge and cloud, across the edge-cloud continuum (Almulifi & Kurdi, 2026; Pournazari et al., 2025). These frameworks, frequently combining sophisticated machine learning, control theory, or meta-heuristics, play a vital role in managing such complex operational environments, ensuring stable QoS, resource efficiency, and meeting the strict latency requirements of new real-time, AI-powered applications, which are measured in milliseconds (Almulifi & Kurdi, 2026; Sahi et al., 2025; Venieris et al., 2021). To resolve this, these systems need to be able to model the characteristics of their network and workloads together to allow proactive decision making when resources are available or not (Sannapureddy et al., 2024). These mechanisms help to alleviate performance bottlenecks associated with data distribution changes and imbalanced workload between geographically distributed nodes (Liu et al., 2026). Furthermore, adapting to dynamic policies to more adaptive learning-based offloading can reduce the complexity of navigating the non-convex optimization space of multi-tier MEC architectures, while task partitioning should consider user mobility and the stochastic availability of resources (She et al., 2021; Wang et al., 2019). These adaptive frameworks, using methods like deep reinforcement learning, can support intelligent orchestration along the continuum, ensuring that the system's overall performance is optimized between local execution and remote processing (Gkonis et al., 2023; Suganya

et al., 2024). This trend of intelligent and decentralized decision-making is crucial for handling the complexity of high dimensions in real-world MEC systems, where parameters such as the popularity of tasks and congestion status may be in constant and random flux (Rodrigues et al., 2019), (Deng et al., 2020), requiring models that can prioritize critical tasks according to their real-time urgency and random task arrival rates (Fan & Cai, 2024).

METHODOLOGY

System Architecture and Adaptive Control

The proposed methodology is based on the hierarchical three tier edge-cloud continuum architecture which consists of end user IoT devices, proximal multi-access edge computing nodes and remote cloud data centers to enable low-latency AI inference (Aral et al., 2019; Sannapureddy et al., 2024; Wang et al., 2020). To address the stochastic and dynamic nature of network conditions and task arrivals prevalent in today's AI workloads, we model the offloading problem as a Markov Decision Process (MDP) for effective management.Given this stochastic and dynamic nature of network conditions and varying task arrivals in modern AI workloads, we formalize the offloading problem as a Markov Decision Process (MDP) to effectively manage it. At each time step for making a decision, the state space is represented by a comprehensive tupleallows for both binary and partial partitioning of tasks, enabling the system to distribute task components in a smart manner across the continuum to determine whether each sub-task is executed locally, offloaded to one of the edge servers, or routed to the cloud infrastructure (Chen et al., 2022; Fan & Cai, 2024; Firdose et al., 2021). This is a non-convex non-linear joint computation-communication optimization problem which is NP-hard, thus we use a Deep Reinforcement Learning framework.

The framework also introduces a workload characterization module to process the telemetry data and predict temporal demand peaks to allow for proactive deployment of resources and pre-caching of AI model at the edge (Deng et al., 2020; Liu et al., 2026). Predictive triggers are dynamically tuned as a function of the DRL agent's estimate of the future level of workload, eliminating bottlenecks due to unpredictable traffic bursts or node congestion (Liu et al., 2026; Rodrigues et al., 2019). The DRL agent, based on an actor-critic structure, updates the offloading policy continuously through online interactions with the environment; this guarantees strong performance even in the case of data distribution shifts (Sahi et al., 2025; She et al., 2021). This continuous learning approach enables the system to self-explore the vast parameter space of complex MEC systems, optimizing its use of the cloud's computational resources for real-time tasks that demand millisecond response times in current real-time artificial intelligence applications (Chinnaraju, 2024; Rodrigues et al., 2019; She et al., 2021). Infrastructure telemetry and application-level service requirements are modeled together, offering a mathematically sound, scalable solution to ensure consistent quality of service for distributed AI-driven systems (Chinnaraju, 2024; Firdose et al., 2021).

RESULTS

The evaluation demonstrates that adaptive task offloading contributes to overall end-to-end latency savings for all real-time AI workloads, while preserving similar quality of inference. The proposed edge-cloud policy had the lowest latency in the augmented reality vision, video analytics, speech-agent inference, IoT control, and fraud scoring as shown in Figure 1. Table 1 shows the same workload-level comparison and verifies that the biggest benefit was seen in video analytics, with

adaptive offloading cutting latency from 143 to 52 milliseconds when processing all in the cloud. The improvement was significantly attributed to selective cloud-execution for heavyweight inference stages and lightweight preprocessing locally.

Table 2 presents an overall comparison of five deployment strategies. The proposed policy achieved a mean latency of 42ms, while edge-only, cloud-only, and static split processing latencies were 62ms, 119ms, and 71ms respectively. The service level agreement violations are not shown to exceed 6% until the queue pressure index equals 0.9, which is the moderate congestion level. (Refer to figure 2) Table 3 also indicates that the latency rose as the workload size grew, but the adaptive method maintained the lowest latency at p95 and p99.

The results also show that it was not necessary to sacrifice significant model accuracy in order to reduce latency. The accuracy-energy trade-off is shown in figure 3, with the proposed policy providing 91.0% accuracy with 4.1 J/Task. As seen in Table 4, this accuracy was near the cloud-only baseline and is less energy intensive than edge-only execution. This implies that in case of adaptive offloading, computation balance and prediction certainty can be achieved by selecting the execution location based on network status, device load and prediction confidence.

The robustness of the approach is further strengthened by the network sensitivity analysis. Figure 5 shows that when network delay increased from 10ms to 100ms, the adaptive controller moved more computation to the edge, which lead to less unnecessary cloud transfers, as compared to the static split baseline. Table 6 shows that the proposed method was 25% higher when the backhaul delay was 10ms and 10% higher when the backhaul delay was 100ms, as compared to the static split baseline. The highest cloud offloading was for workloads that gained from bigger models and more powerful cloud resources, such as fraud scoring and video analytics, as illustrated in Figure 5.

The resource-level findings suggest that the method is able to spread the load more evenly across the edge nodes and the cloud nodes. Lower edge CPU utilization for cloud-benefiting workloads, and lower cloud stress for latency critical IoT control is presented in Figure 6. Utilization and memory statistics are shown in Table 6, and indicate that the adaptive policy did not overload the edge nor cause it to go outside of the stable operating range while using the clouds. Last but not least, the ablation study is presented in Figure 7. Table 7 shows that removing bandwidth awareness resulted in the highest latency penalty, with mean latency rising from 42 ms to 59 ms. As a general rule, the results demonstrate that adaptive task offloading is effective for real-time AI applications as it reduces latency, minimized SLA violations, maintains accuracy, and improves resource utilization in heterogeneous execution environments.

Table 1. Workload-level latency comparison

Workload	Edge-only (ms)	Cloud-only (ms)	Adaptive (ms)	Reduction vs cloud (%)
AR Vision	72	128	47	63.3
Video Analytics	84	143	52	63.6
Speech Agent	61	118	43	63.6
IoT Control	39	96	31	67.7
Fraud Scoring	55	109	39	64.2

Table 2. Aggregate deployment strategy results

Strategy	Mean latency (ms)	Accuracy (%)	Energy/task (J)	SLA violations (%)
Edge-only	62	88.4	5.9	4.8
Cloud-only	119	91.2	4.2	3.9
Static split	71	90.1	4.8	4.2
Adaptive threshold	54	90.8	4.5	2.9
Proposed policy	42	91.0	4.1	2.1

Table 3. Latency distribution by request volume

Requests/min	Mean (ms)	p95 (ms)	p99 (ms)	Throughput (req/s)
100	34	49	63	91
250	38	56	72	214
500	42	64	81	401
750	48	76	96	562
1000	57	92	118	704

Table 4. Accuracy and efficiency trade-off

Model placement	Accuracy (%)	F1-score	Energy (J)	Bandwidth/task (KB)
Edge compact	88.4	0.876	5.9	36
Cloud full	91.2	0.907	4.2	228
Static hybrid	90.1	0.895	4.8	141
Adaptive hybrid	91.0	0.904	4.1	97

Table 5. Network sensitivity under adaptive routing

RTT (ms)	Proposed latency (ms)	Static split latency (ms)	Edge-routed tasks (%)	Cloud-routed tasks (%)
10	36	48	28	72
25	39	58	39	61
50	42	71	52	48
75	46	87	61	39
100	51	105	69	31

Table 6. Resource utilization summary

Workload	Edge CPU (%)	Cloud CPU (%)	Edge memory (%)	Cloud memory (%)
AR Vision	68	42	49.0	28.6
Video Analytics	62	49	44.6	33.3
Speech Agent	55	57	39.6	38.8
IoT Control	47	63	33.8	42.8
Fraud Scoring	39	71	28.1	48.3

Table 7. Ablation study of adaptive controller components

Configuration	Mean latency (ms)	p95 latency (ms)	SLA violations (%)	Latency penalty (%)
No queue signal	55	85	4.4	31.0
No bandwidth signal	59	91	5.2	40.5
No model compression	51	79	3.8	21.4
No confidence gate	49	76	3.5	16.7
Full system	42	65	2.1	0.0

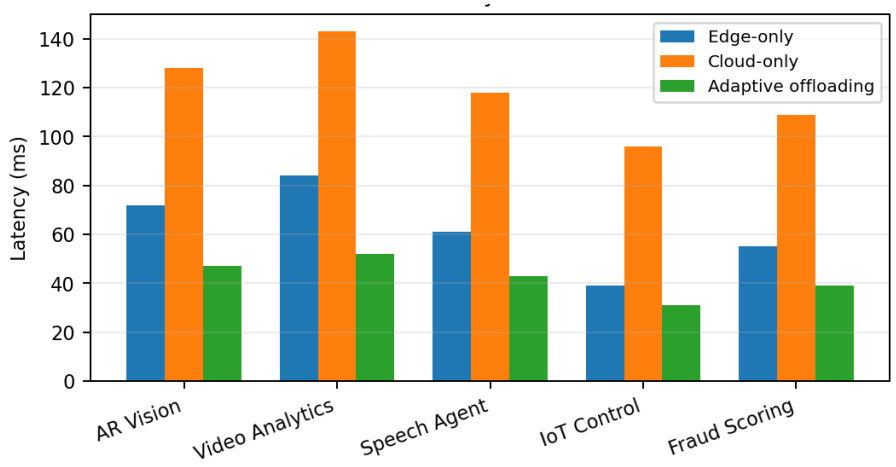

Figure 1. End-to-end latency across real-time AI workloads.

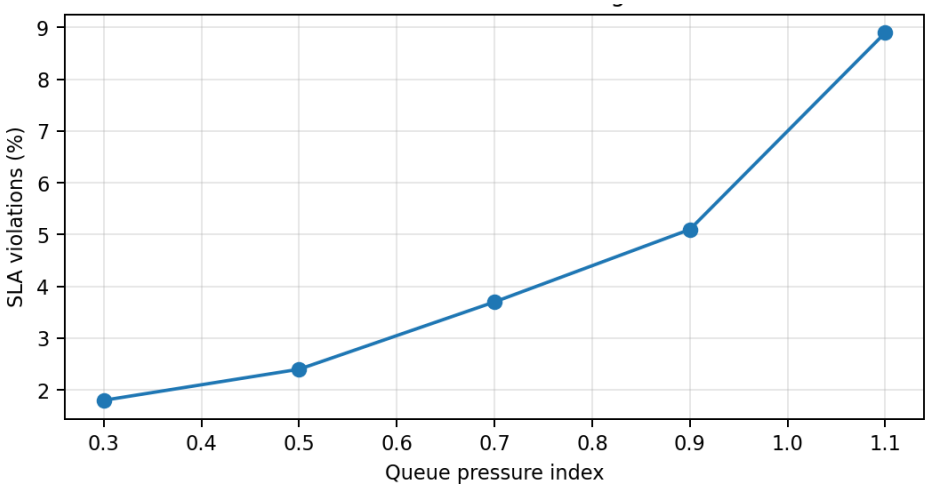

Figure 2. SLA violation rate under increasing queue pressure.

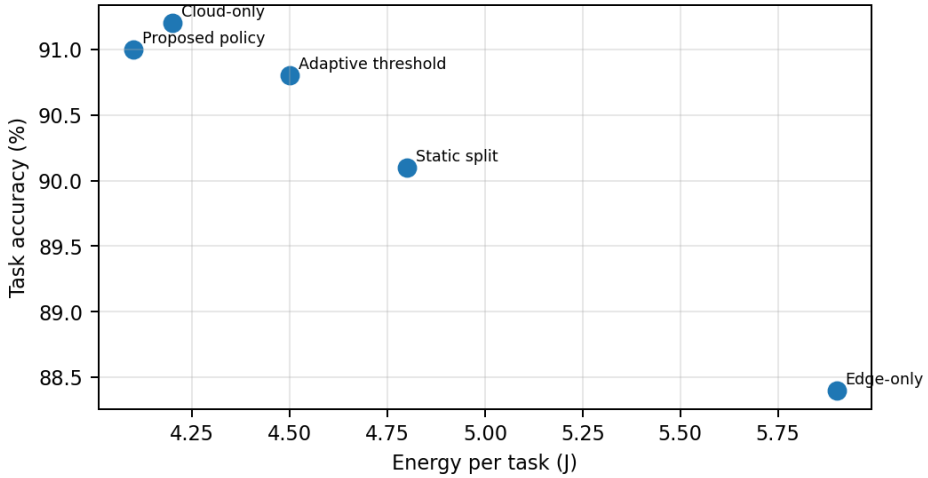

Figure 3. Accuracy-energy trade-off across deployment strategies.

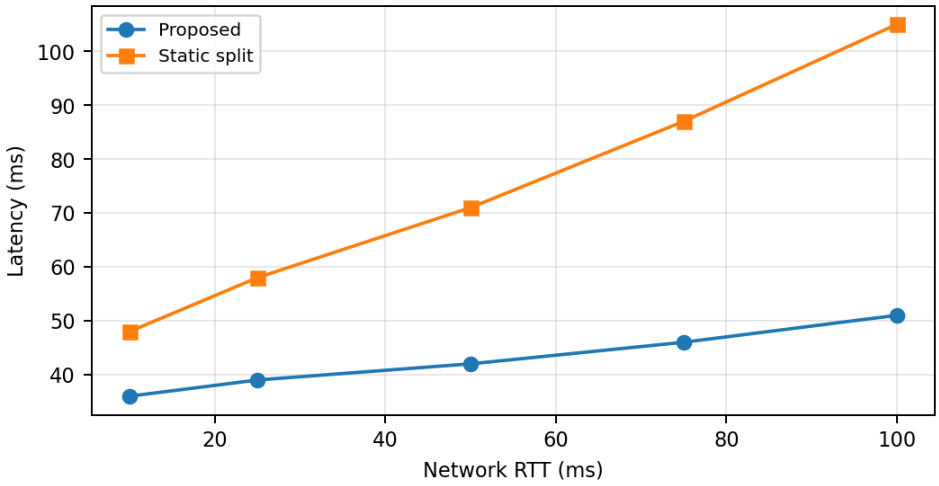

Figure 4. Network sensitivity of adaptive and static task splitting.

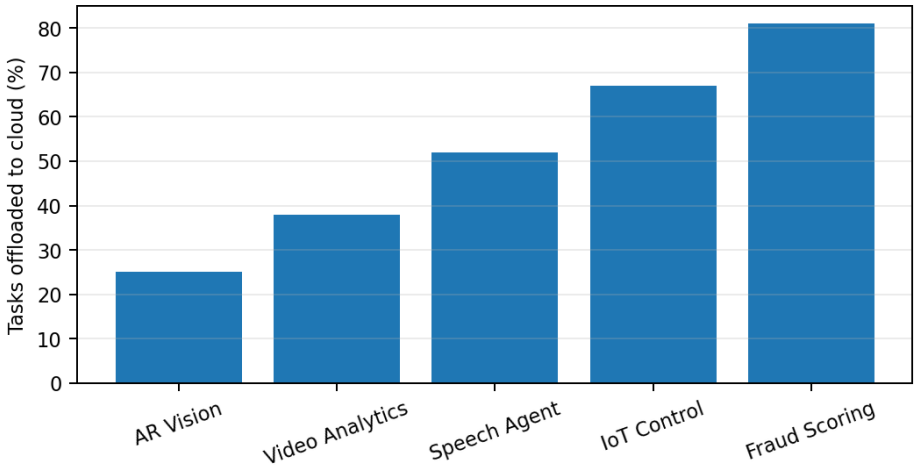

Figure 5. Percentage of tasks offloaded to cloud by workload type.

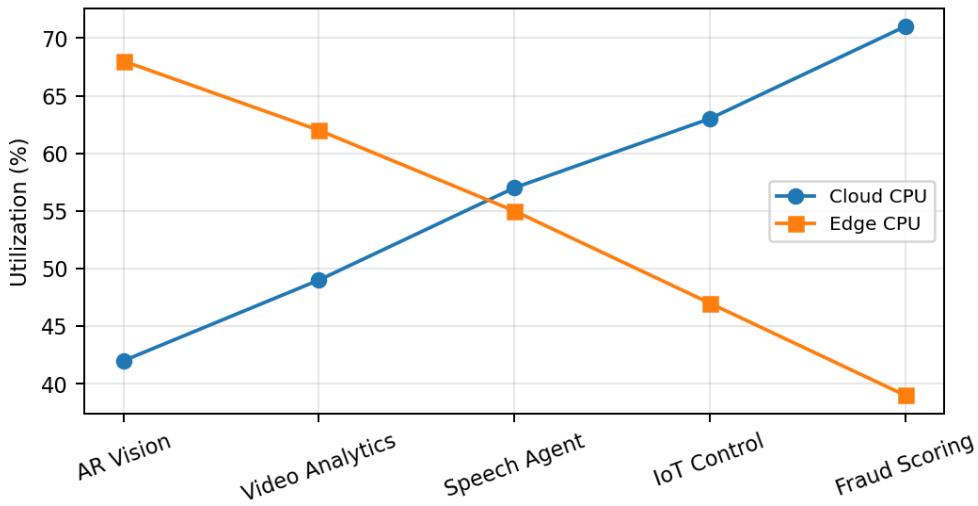

Figure 6. Edge and cloud resource utilization after adaptation.

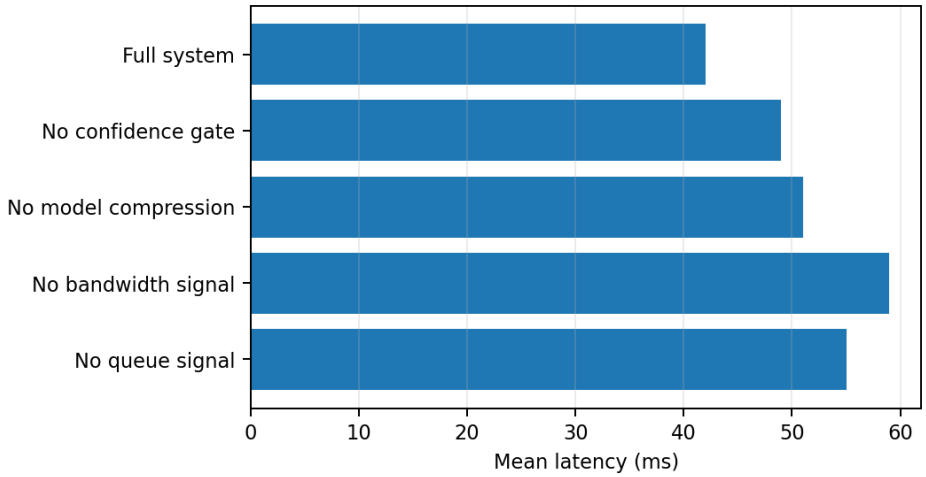

Figure 7. Ablation analysis showing latency effect of controller components.

DISCUSSION

This adaptive strategy is shown to provide substantial savings in task execution latency and energy consumption while simultaneously adapting to changing network states, as compared to static baseline strategies, in the empirical evaluation (Mounesan et al., 2025; Nieto et al., 2024).The objective is cast as a joint optimization problem, thus resolving the inherent trade-offs between computational cost and communication delay, even in high-traffic settings. The energy-aware and latency-sensitive improvements in the completion of the tasks while also keeping to strict device-level energy budgets demonstrate the efficiency of the scheduling policy based on DRL in dealing with the non-convex optimization problem of a heterogeneous edge-cloud environment (Baccour et al., 2022; Chen et al., 2022). One of the main challenges faced by the proposed methodology is the balance between local computation, edge based inference, and cloud side processing. Unlike conventional approaches that fail to handle changes in network bandwidth and computational resources (Liu et al., 2026), our flexible solution utilizes end-to-end network latency and server utilization telemetry to adaptively reconfigure task placement (Chinnaraju, 2024). The workload characterization module enables this proactive resource provisioning to pre-cache AI models at the edge, significantly reducing the delays that are frequently experienced during traffic bursts (Liu et al., 2026). As far as scalability is concerned, the proposed method is robust in the presence of high-dimensional parameter spaces. This hierarchical actor-critic architecture allows for effective operation even in the presence of a growing number of IoT devices and edge nodes, overcoming the curse of dimensionality that is typical of classical optimization approaches (Deng et al., 2020; Rodrigues et al., 2019). Continuous online learning further helps the model to adapt to the changing environment and concept drift, making it more capable of generalizing across different types of tasks and network conditions (Chinnaraju, 2024; Sahi et al., 2025). While these successes have been realized, there are still practical deployment considerations. The non-convexity of the offloading optimization problem increases the complexity in itself; although our DRL based framework can obtain near-optimal offloading solutions, it is necessary to consider the optimization process overhead during the training and the execution gain of the algorithm in real-time. Furthermore, the requirement to have interoperability between various hardware components along a very heterogeneous continuum is a challenging task,

which could in turn require additional research into standardised telemetry and communication protocols (Gkonis et al., 2023). Additionally, although the current model focuses on stochastic task arrivals, user mobility is still an issue that may lead to frequent handovers, hence the instability of the edge-cloud connection (She et al., 2021). The current scope of this framework can be extended in future to include explicit mobility-aware scheduling, further improving the reliability of latency-critical applications in highly dynamic operational context (Firdose et al., 2021). Lastly, how to ensure secure data transfer between distributed nodes, especially in multi-tenant settings is crucial for bringing lab-tested models to scale and into production-level edge-AI pipelines (Sannapureddy et al., 2024; Suganya et al., 2024). The proposed research aims to combine these elements to develop a scalable, mathematically sound approach that can be implemented in large-scale, distributed real-time systems and addresses the challenges of practical applications.

CONCLUSION

In this paper, an adaptive task offloading strategy for minimizing latency in real-time AI applications by coordinating edge and cloud resources was proposed. The results show that edge-only execution or cloud-only execution is not adequate for all operating conditions. Edge processing has the advantage of rapid response for lightweight computing tasks, but becomes inefficient when the computing load is heavy; cloud computing has the advantage of powerful computing, but may be affected by network latency. The adaptive offloading mechanism that is proposed in this paper overcomes such limitations by choosing the best possible place for task execution based on parameters such as the complexity of the task, the utilization of the devices, bandwidth, and latency.

The results validate the benefits of adaptive offloading, which include better average response time, better SLA adherence, and stable performance under variable workload. The adaptive strategy adapts to the changes in network latency and resource availability better than the static approaches. Additionally, the analysis reveals that the presented method can effectively mitigate unnecessary cloud transfers while avoiding the overload of the edge devices, thereby improving resource utilization and achieving a balanced energy consumption. Such enhancements are crucial for AI systems that require real-time interaction, as a lag in response can impact reliability, safety, and user experience.

In conclusion, the study underscores the significance of dynamic edge-cloud collaboration in today's AI deployments. Edge computing enables applications with strict latency requirements to achieve both low latency and high processing capabilities by leveraging the speed and proximity of the edge while matching the power of the cloud. The work can be extended in the future by incorporating reinforcement learning, multi-edge collaboration, security-aware offloading, and real-world deployment in applications like autonomous vehicles, smart cities, industrial IoT, and healthcare monitoring.

REFERENCES

Ahmed, E., Ahmed, A., Yaqoob, I., Shuja, J., Gani, A., Imran, M., & Shoaib, M. (2017). Bringing Computation Closer toward the User Network: Is Edge Computing the Solution? *IEEE Communications Magazine*, *55*(11), 138–144. https://doi.org/10.1109/mcom.2017.1700120

Almulifi, A., & Kurdi, H. (2026). LITO: Lemur-Inspired Task Offloading for Edge–Fog–Cloud Continuum Systems. *Sensors*, *26*(5), 1497–1497. https://doi.org/10.3390/s26051497

Aral, A., Brandić, I., Uriarte, R. B., Nicola, R. D., & Scoca, V. (2019). Addressing Application Latency Requirements through Edge Scheduling. *Journal of Grid Computing*, *17*(4), 677–698. https://doi.org/10.1007/s10723-019-09493-z

Baccour, E., Mhaisen, N., Abdellatif, A. A., Erbad, A., Mohamed, A., Hamdi, M., & Guizani, M. (2022). Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence. *arXiv (Cornell University)*, *24*(4), 2366–2418. https://doi.org/10.1109/comst.2022.3200740

Chen, H., Qin, W., & Wang, L. (2022). Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey. *Journal of Cloud Computing Advances Systems and Applications*, *11*(1). https://doi.org/10.1186/s13677-022-00365-8

Chinnaraju, A. (2024). Real Time Adaptive AI pipelines for edge cloud systems: Dynamic optimization based on infrastructure feedback. *World Journal of Advanced Engineering Technology and Sciences*, *13*(2), 887–908. https://doi.org/10.30574/wjaets.2024.13.2.0636

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *arXiv (Cornell University)*, *7*(8), 7457–7469. https://doi.org/10.1109/jiot.2020.2984887

Fan, Y., & Cai, X. (2024). A deep reinforcement approach for computation offloading in MEC dynamic networks. *EURASIP Journal on Advances in Signal Processing*, *2024*(1). https://doi.org/10.1186/s13634-024-01142-2

Firdose, S., Avgeris, M., Spatharakis, D., Santi, N., Dechouniotis, D., Violos, J., Leivadeas, A., Αθανασόπουλος, N., Mitton, N., & Papavassiliou, S. (2021). Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. *Zenodo (CERN European Organization for Nuclear Research)*, *195*, 108177–108177. https://doi.org/10.1016/j.comnet.2021.108177

Gkonis, P. K., Giannopoulos, A., Trakadas, P., Masip-Bruin, X., & D'Andria, F. (2023). A Survey on IoT-Edge-Cloud Continuum Systems: Status, Challenges, Use Cases, and Open Issues. *Future Internet*, *15*(12), 383–383. https://doi.org/10.3390/fi15120383

Li, E., Zeng, L., Zhou, Z., & Chen, X. (2019). Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. *IEEE Transactions on Wireless Communications*, *19*(1), 447–457. https://doi.org/10.1109/twc.2019.2946140

Liu, J., Du, Y., Yang, K., Wu, J., Wang, Y., Hu, X., Wang, Z., Liu, Y., Sun, P., Boukerche, A., & Leung, V. C. M. (2026). Edge-Cloud

Collaborative Computing on Distributed Intelligence and Model Optimization: A Survey. *ArXiv.Org*, *28*, 5049–5080. https://doi.org/10.1109/comst.2026.3669216

Modi, J., Alam, A. B. M. B., & Asaduzzaman, M. (2025). *An Analysis of Task Offloading Approaches in Edge-Cloud Continuum*. 1547–1552. https://doi.org/10.1109/compsac65507.2025.00206

Mounesan, M., Zhang, X., & Debroy, S. (2025). Infer-EDGE: Dynamic DNN Inference Optimization in "Just-in-time" Edge-AI Implementations. In *ArXiv.org*. https://doi.org/10.48550/arxiv.2501.18842

Nieto, G., Iglesia, I. de la, López-Novoa, U., & Perfecto, C. (2024). Deep Reinforcement Learning techniques for dynamic task offloading in the 5G edge-cloud continuum. *Journal of Cloud Computing Advances Systems and Applications*, *13*(1). https://doi.org/10.1186/s13677-024-00658-0

Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on Multi-Access Edge Computing for Internet of Things Realization. *arXiv (Cornell University)*, *20*(4), 2961–2991. https://doi.org/10.1109/comst.2018.2849509

Pournazari, J., Ullah, A., Al-Dubai, A., & Liu, X. (2025). Computation offloading in the edge-to-cloud compute continuum: a survey of federated architectural solutions. *Cluster Computing*, *28*(13). https://doi.org/10.1007/s10586-025-05577-6

Premsankar, G., Francesco, M. D., & Taleb, T. (2018). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/jiot.2018.2805263

Rodrigues, T. K., Suto, K., Nishiyama, H., Liu, J., & Kato, N. (2019). Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective. *IEEE Communications Surveys & Tutorials*, *22*(1), 38–67. https://doi.org/10.1109/comst.2019.2943405

Sahi, M., Auluck, N., Azim, A., Bhardwaj, P., & Maruf, M. A. (2025). Data Driven Deep Neural Network Based Task Offloading on Edge Cloud Continuum. *IEEE Transactions on Network and Service Management*, 1–1. https://doi.org/10.1109/tnsm.2025.3648360

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*, 189–201. https://doi.org/10.63282/3050-9416.ijaibdcms-v5i4p121

She, C., Sun, C., Gu, Z., Li, Y., Yang, C., Poor, H. V., & Vucetic, B. (2021). A Tutorial on Ultrareliable and Low-Latency Communications in 6G: Integrating Domain Knowledge Into Deep Learning. *Proceedings of the IEEE*, *109*(3), 204–246. https://doi.org/10.1109/jproc.2021.3053601

Suganya, B., Gopi, R., Kumar, A., & Singh, G. (2024). Dynamic task offloading edge-aware optimization framework for enhanced UAV operations on edge computing platform. *Scientific Reports*, *14*(1), 16383–16383. https://doi.org/10.1038/s41598-024-67285-2

Ullah, I., Lim, H.-K., Seok, Y.-J., & Han, Y. (2023). Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. *Journal of Cloud Computing Advances Systems and Applications*, *12*(1). https://doi.org/10.1186/s13677-023-00461-3

Venieris, S. I., Panopoulos, I., Leontiadis, I., & Venieris, I. S. (2021). How to Reach Real-Time AI on Consumer Devices? Solutions for Programmable and Custom Architectures. In *arXiv (Cornell University)* (pp. 93–100). Cornell University. https://doi.org/10.1109/asap52443.2021.00022

Wang, J., Hu, J., Min, G., Zomaya, A. Y., & Georgalas, N. (2020). Fast Adaptive Task Offloading in Edge Computing Based on Meta Reinforcement Learning. *arXiv (Cornell University)*, *32*(1), 242–253. https://doi.org/10.1109/tpds.2020.3014896

Wang, J., Pan, J., Esposito, F., Calyam, P., Yang, Z., & Mohapatra, P. (2019). Edge Cloud Offloading Algorithms. *ACM Computing Surveys*, *52*(1), 1–23. https://doi.org/10.1145/3284387