Faculty of Natural and Applied Sciences Journal of Computing and Applications
Print ISSN: 3026-8133
www.fnasjournals.com
Volume 2; Issue 4; December 2025; Page No. 95-104.
DOI: https://doi.org/10.63561/jca.v2i4.1079

Hybrid Edge–Cloud Collaboration for Addressing Bandwidth Limitations and Latency in IoT Applications

***[1]Franklin, M., & [2]Obenobe, I.O.**
[1]Department of Computer Engineering, Rivers State University, Port Harcourt, Rivers State
[2]Department of Computer Science, College of Education, MOSOGAR,2in affiliation with Delta State University, Abraka, Delta State

***Corresponding author email**: mission.franklin@ust.edu.ng

Abstract
The exponential growth of Internet of Things (IoT) ecosystems has intensified the demand for processing frameworks that can manage vast and continuous data flows with low latency and high efficiency. Conventional cloud-centric architectures, while offering scalable storage and computational power, are hindered by inherent limitations such as network congestion, high latency, and bandwidth bottlenecks. These constraints pose significant challenges for real-time and mission-critical IoT applications, including healthcare monitoring, autonomous vehicles, and industrial automation, where timely decision-making is paramount. To address these issues, Edge–Cloud collaboration has emerged as a hybridized computing paradigm that integrates the complementary strengths of edge and cloud layers. In this model, edge nodes perform immediate, latency-sensitive computations near the data source, ensuring rapid responsiveness, while more complex, resource-intensive tasks are delegated to the cloud for large-scale analysis, storage, and global optimization. This collaborative approach not only reduces the volume of raw data transmitted across networks, thereby alleviating bandwidth constraints, but also enhances scalability, reliability, and adaptability of IoT infrastructures. Despite its advantages, challenges such as dynamic task partitioning, interoperability, energy efficiency, and data security remain central to its successful implementation. This paper critically analyses the architecture, benefits, and limitations of Edge–Cloud collaboration, with a specific focus on its effectiveness in mitigating latency and bandwidth constraints. It also examines the applicability of this hybrid computing model across various IoT domains where distributed processing has become essential. The paper concludes with key insights and offers practical recommendations for practitioners seeking to optimize Edge–Cloud deployments.

Keywords: Edge–Cloud Collaboration, Internet of Things (IoT), Latency Reduction, Bandwidth Optimization, Hybrid Computing Architecture

Introduction

The rapid proliferation of the Internet of Things (IoT) has created a hyperconnected digital ecosystem where billions of devices continuously collect, transmit, and process data. Projections suggest that by 2030, over 25 billion IoT devices will be in active use, producing massive volumes of heterogeneous data that require efficient processing frameworks (Statista, 2024). Traditional cloud computing has played a central role in addressing these demands, offering virtually unlimited processing power and storage. However, the reliance on centralized data centers creates performance bottlenecks: Latency, bandwidth congestion, and transmission overheads, which significantly hinder the effectiveness of real-time IoT applications such as autonomous driving, telemedicine, smart grid control, and industrial automation (Daruvuri & Patibandla, 2023; Liu et al., 2023). In contexts where milliseconds determine system reliability and safety, cloud-only solutions are inadequate, making the problem of latency and bandwidth constraints in IoT both critical and urgent.

A promising solution lies in edge computing, which shifts computational tasks from distant centralized servers to processing units closer to the data source. Devices such as fog nodes, micro data centers, and even smart routers can process and analyse data locally, thereby reducing latency and alleviating bandwidth usage. Despite these advantages, edge nodes are resource-constrained, with limited power, storage, and processing capacity (Zhang et al., 2020). These limitations make it infeasible for edge computing alone to handle the complex data analytics, machine learning tasks, and large-scale optimization required in diverse IoT applications. Consequently, researchers and practitioners are increasingly advocating for Edge–Cloud collaboration. In this model, edge nodes perform immediate latency-sensitive processing while computationally intensive and large-scale operations are delegated to the cloud. This division of labour ensures both responsiveness and computational depth, effectively bridging the gap between edge autonomy and cloud scalability. The idea of collaborative architectures between edge and cloud has been widely studied in recent years. Recent studies have shown that collaborative edge–cloud architectures significantly improve task-offloading performance by enabling dynamic workload allocation across local and remote nodes. Liu et al. (2023) demonstrated how deep-reinforcement-learning–based scheduling enhances offloading efficiency, while Zhang et al. (2024) and Yu et al. (2023) showed that partitioning computational tasks, especially latency-sensitive and AI inference workloads, between edge and cloud resources reduces delay and optimizes system utilization.

Despite these advances, significant challenges remain in realizing seamless Edge–Cloud integration. Issues such as dynamic task allocation, data synchronization, interoperability across heterogeneous devices, and ensuring data privacy and security continue to hinder large-scale deployment. For example, Daruvuri and Patibandla (2023) noted that task scheduling in hybrid environments must balance energy efficiency, latency requirements, and computational workloads, which requires sophisticated optimization techniques. Similarly, security concerns are heightened in the collaborative settings, where sensitive data must be secured both locally and in transit to the cloud (Liu et al., 2023). These challenges underscore the importance of further research into optimization strategies, secure communication protocols, and intelligent workload distribution mechanisms to maximize the effectiveness of Edge–Cloud systems.

Recent literature demonstrates significant progress in simulating and optimizing hybrid edge–cloud environments, often leveraging frameworks such as iFogSim for modeling. Abbas et al. (2018) reviewed mobile edge computing paradigms and underscored the potential of edge–cloud cooperation, a view reinforced by Satyanarayanan (2017), while also drawing attention to the persistent trade-offs among latency, energy consumption, and bandwidth in distributed systems. Building on this foundation, other recent studies have examined offloading and task-partitioning strategies in IoT-based edge–cloud systems, highlighting how adaptive workload distribution and resource management can improve performance and energy efficiency under specific conditions (Chen et al., 2022; Yu et al., 2023; Zhang et al., 2024). Their classification of decision-making engines and optimization techniques highlights the complexity of balancing computation between edge and cloud layers. More recently, Ullah et al. (2023) proposed a Double Deep Q Network (DDQN)–based dynamic offloading model that improved resource utilization, minimized task rejections, and reduced latency compared to heuristic-based solutions. Complementing this, Zhang et al. (2024) examined dependent-service scenarios in mobile edge computing, where layered service dependencies were jointly optimized to balance latency and energy efficiency through integrated scheduling and resource allocation strategies.
Energy efficiency has increasingly become a central theme in edge–cloud integration research. For instance, Alharbi et al. (2023) designed an energy-aware and secure offloading framework for multi-tier edge–cloud architectures, identifying dynamic energy constraints as a major challenge while advocating for the integration of security-aware decision mechanisms. Along similar lines, Birhanie et al. (2024) proposed a base-station-orchestrated task offloading mechanism for MEC–IoT networks, demonstrating improvements in task completion, latency, and energy management through intelligent scheduling. Puligundla et al. (2024) advanced this trajectory by employing deep reinforcement learning to optimize resource allocation and content distribution, thereby striking a balance among latency, energy efficiency, Quality of Service (QoS), and adaptability in edge–cloud IoT environments.

Other innovative directions include the use of unmanned aerial vehicles (UAVs) to support edge computing in vehicular networks. Bi et al. (2024), for example, introduced a dueling DQN–based approach that combines computing resource allocation with UAV position optimization, enhancing system performance in dynamic mobile environments. Reinforcement learning continues to play a pivotal role in decision-making frameworks. Studies in recent time have shown that reinforcement learning and deep-learning–based methods can effectively manage dynamic offloading and resource allocation in edge–cloud ecosystems (Mao et al., 2017; Chen et al., 2022).

Recent research has reinforced the view that collaborative edge–cloud architectures can improve energy efficiency in IoT environments while sustaining acceptable levels of service quality. Yuan et al. (2023) demonstrated this through their ELECT framework, which integrates intelligent node management with deep-reinforcement-learning–based scheduling to minimize energy consumption in device–edge–cloud pipelines, without compromising data-processing performance in remote IoT applications. Likewise, Ferreira et al. (2023) showed that optimizing node selection in a layered edge–fog–cloud architecture reduces network-wide energy use, while still satisfying required QoS thresholds. Collectively, these studies illustrate that coordinated workload distribution across multi-tier IoT infrastructures, supported by adaptive resource allocation and intelligent offloading, offers a viable pathway toward energy-aware IoT systems that maintain reliable service delivery.

Another stream of research has focused on specific application domains. In healthcare, hybrid computing architectures have been shown to improve the reliability of wearable health monitoring systems by ensuring real-time responsiveness at the edge while enabling predictive analytics in the cloud (Luo et al., 2022). In industrial automation, Chen et al. (2023) demonstrated how Edge–Cloud collaboration reduces downtime in smart manufacturing environments by balancing control tasks between local edge servers and cloud-based predictive maintenance platforms. These studies collectively reinforce the argument that Edge–Cloud collaboration offers a comprehensive solution to the limitations of standalone cloud or edge infrastructures. These show that cutting-edge conceptual paradigms are beginning to emerge that extend beyond conventional architectures. Hossain et al. (2024) introduced the notion of Quantum Edge–Cloud Computing, exploring its potential to address challenges of latency, scalability, and security in IoT systems through quantum-enabled augmentation. Collectively, these contributions reflect a rapidly evolving research frontier in which edge–cloud collaboration is being advanced through intelligent algorithms, multi-tier architectures, and even quantum-inspired designs, pointing toward a future of more adaptive, efficient, and resilient IoT infrastructures.

A comprehensive review of Edge–Cloud collaboration is timely and essential for both academia and industry. By consolidating insights from recent literature and analysing their implications for practical deployments, this work contributes to advancing the discourse on next-generation IoT infrastructures. It provides a pathway toward designing intelligent, adaptive, and secure hybrid frameworks capable of overcoming existing constraints while preparing for the exponential growth of connected devices.

Statement of the Problem

As IoT systems continue to generate massive volumes of real-time data, traditional cloud-centric architectures are increasingly unable to meet stringent latency and bandwidth requirements. High network congestion, delays in data transmission, and dependence on centralized processing limit the responsiveness and reliability of critical IoT applications such as healthcare monitoring, autonomous systems, and industrial automation. Although Edge–Cloud collaboration offers a hybrid approach to address these constraints, its practical implementation is challenged by architectural complexity, resource management issues, and inconsistent performance across diverse IoT environments. This gap necessitates a systematic examination of how Edge–Cloud integration can be optimized to effectively mitigate latency and bandwidth limitations while supporting scalable and dependable IoT operations.

Aim and Objectives

The purpose of this study is to analyses the Edge–Cloud collaboration mitigates latency and bandwidth limitations and assess its effectiveness across key IoT domains, with the following objectives:

(i) Examine core Edge–Cloud architectural models;
(ii) Evaluate its impact on latency, bandwidth use, and real-time performance;
(iii) Identify key limitations and challenges of Edge–Cloud integration;
(iv) Assess its applicability and benefits in major IoT domains.

Materials and Methods

This study adopts a simulation-based approach using **iFogSim2** to evaluate the Hybridized Edge–Cloud collaboration in IoT environments. The system architecture comprises three layers: IoT devices (sensors), edge nodes (fog servers), and cloud servers.

Design Architecture of the Hybridized Edge–Cloud Collaboration
The hybridization of the Edge–Cloud collaborative architecture consists of: (i) *Device Layer*: IoT sensors and actuators that collect and transmit data; (ii) *Edge Layer*: Local gateways, fog nodes, or edge servers that process time-sensitive tasks; (iii) *Cloud Layer*: Remote data centres for resource-intensive computation, global analytics, and model training.

The Data Flow: The process of data flow from data sources to computational processing engines in the cloud is indicated to include: (i) Raw sensor data is pre-processed at the edge, as the data is collected from the device layer and transmitted to the edge layer; (ii) Time-critical decisions are handled locally at the edge to reduce latency, and reduce the consumption of bandwidth; aggregated or historical data is sent to the cloud for long-term storage or model refinement. This is because such data transmission and processing take time and that would increase latency and use up bandwidth that would be used for efficient running of the system and give timely response.
Deployment Scenarios: Three deployment scenarios were simulated, (i) **Cloud-Only, where** all the data was processed in the cloud, (ii) **Edge-Only**, where all the data was processed at edge nodes, and (iii) Hybrid **Edge–Cloud Collaborative, where a** real-time task is handled at the edge, and complex tasks sent to the cloud, depending on some parameters of efficiency.
The simulations focused on a smart healthcare use case, where patient data- heart rate is continuously monitored. Performance was assessed using: **Latency, Bandwidth usage, Task completion rate, Energy consumption.** A rule-based offloading strategy directed latency-sensitive tasks to the edge and compute-heavy tasks to the cloud. Each scenario was run 10 times with varying device counts (10–100).
Tools used in the Experimental Design:
(i) **iFogSim/iFogSim2** – IoT–Edge–Cloud simulation,
(ii) **Java JDK + IDE** – model development;
(iii) **Python/MATLAB** – data analysis & visualization,
(iv) **Optimization tools** –MATLAB,
(v) **Heuristic algorithms** – GA, PSO, NSGA-II.

In a simulation-based study of hybrid edge–cloud collaboration, **iFogSim /iFogSim2** serves as the primary tool for modeling IoT devices, fog nodes, and cloud servers, enabling controlled experimentation with latency and energy trade-offs. The simulator is typically developed and customized using **Java JDK and an IDE-** Eclipse, while output data is analysed and visualized with **Python (NumPy, Pandas, Matplotlib) and MATLAB** for clearer interpretation of system behaviour. To optimize task placement and scheduling, **optimization frameworks** like MATLAB's Optimization Toolbox are employed, combined with **heuristic or metaheuristic algorithms** such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), or NSGA-II to balance multiple objectives. Together, these tools provided the adaptable environment for simulating, analysing, and improving hybrid edge–cloud strategies in IoT environments.

Formulations

1. Local compute (Dynamic Voltage and Frequency Scaling) — optimal energy under a time budget
$$f_i^* = \frac{c_i}{t_i}\ ;\ T_i^{loc} = \frac{c_i}{f_i^*}\ ;\ E_i^{loc} = k(f_i^*)^2 c_i = k\,\frac{c_i^3}{t_i^2} \quad \text{.................................} (1)$$
2. Edge execution latency (node j)
$$T_{ij}^{edge} = t_i^{\uparrow} + \frac{c_i}{f_{ij}} + t_i^{\downarrow} \quad \text{...} (2)$$
3. **Cloud execution latency (via backhaul)**
$$T_{ik}^{cloud} = t_i^{\uparrow} + L_i^{bh} + \frac{c_i}{f_{ij}} + t_i^{\downarrow} \quad \text{...} (3)$$
4. **Minimum radio energy to send d bits in time t (use for uplink/downlink)**
$$E_{tx}(d,t) = \frac{N_0 W\,t}{g}\left(2^{\frac{d}{Wt}} - 1\right) \quad \text{...} (4)$$

5. **Per-task selection + bi-objective (latency–energy) scalarization**
Let
$$T_i = \min\left\{T_i^{loc}, \min_j T_{ij}^{edge}, \min_k T_{ik}^{cloud}\right\} \quad \text{.......................................} (5)$$

$$E_i = \min\left\{E_i^{loc}, E^{tx}\left(d_i^{\uparrow}, t_i^{\uparrow}\right), E^{tx}\left(d_i^{\downarrow}, t_i^{\downarrow}\right)\right\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(6)}$$

We then optimize:

$$\min \alpha \textstyle\sum_i \; T_i + (1-\alpha) \textstyle\sum_i \; E_i\,,\ \alpha \in [0,1] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(7)}$$

Explanation of the formulations

1. The formulation expressed in equation 1 shows the trade-off between time and energy when a device processes a task locally. A task needing c_i CPU cycles must finish within a time limit t_i, so the device sets its CPU frequency to $f_i^* = \frac{c_i}{t_i}$, just enough to meet the deadline. The resulting latency is exactly t_i, while the energy consumed is $k\frac{c_i^3}{t_i^2}$, where κ is a device constant. This means that longer deadlines allow lower CPU frequency and save energy, while tighter deadlines require higher frequency and sharply increase energy consumption.

2. The formulation expressed in equation 2 describes the latency when offloading a task to the edge server. The total delay is the sum of two parts: the transmission delay $\frac{d_i}{r_i}$, which is the time taken to send the task input of size d_i bits over a communication(wired or wireless) link with rate r_i , and the edge execution time $\frac{c_i}{f_e}$, , which is how long the edge server (with computing speed f_e) takes to process the task requiring c_i CPU cycles. In simpler terms, the device first spends time sending the data to the edge, then waits for the edge to compute the result. This formula highlights how offloading reduces the device's own burden, but performance still depends on wireless bandwidth and edge processing capacity.
3. The formulation expressed in equation 3 represents the total delay of processing a task in the cloud. It includes four parts: the time to upload data from the device ($t_i^{\uparrow}$), the backhaul delay to reach the cloud (L_i^{bh}), the execution time in the cloud ($\frac{c_i}{f_{ij}}$), and the time to send results back to the device ($t_i^{\downarrow}$). In short, cloud latency is often dominated by network transmission and backhaul delays**,** while the computation time is usually very small due to the cloud's high processing power.

4. The formulation expressed in equation 4 gives the minimum energy required to transmit **d** bits within time t over a wireless channel. Here, N_0 is the noise power spectral density, W is the bandwidth, and g is the channel gain. The term inside the brackets, $\left(2^{\frac{d}{Wt}} - 1\right)$,comes from the Shannon capacity formula, ensuring the transmission meets the rate requirement. In simple terms, this shows that sending more data in less time, or over a weaker channel, requires much higher energy.
5. The formulation expressed in equation 5. This formulation describes how each task decides the best place to run—locally, at the edge, or in the cloud. while balancing both latency and energy use**.** For a given task i, the latency T_i is chosen as the minimum among local, edge, or cloud execution times, and the energy E_i is chosen as the minimum among local computation energy and transmission energy (uplink/downlink). The optimization then combines the two objectives (i) total latency $\sum_i T_i$ and (ii) total energy $\sum_i E_i$ —into a single scalar function using a weight $\alpha \in [0,1]$**.** If α is closer to 1, the system prioritizes speed (low latency), while if α is closer to 0, it prioritizes energy efficiency. This provides a flexible way to trade off performance and battery consumption depending on application needs.

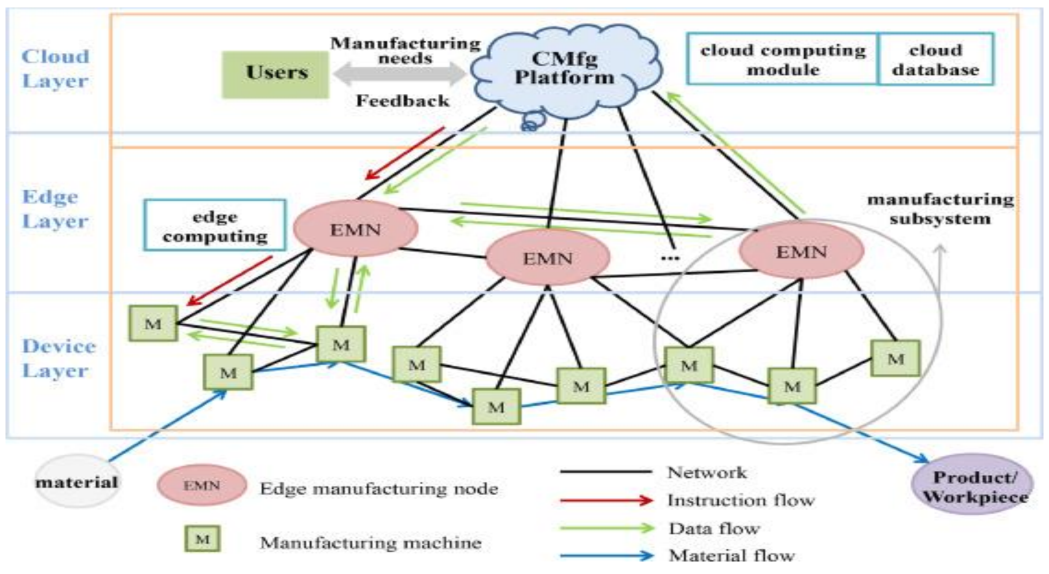

Figure 1: Edge-Cloud collaborative Architecture (source: Yang et al., 2022)

Results

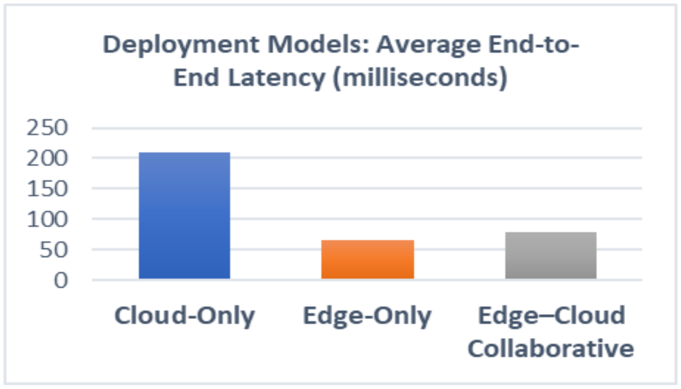

Figure 2: Average End-to-End Latency (milliseconds)

The average End-to-End Latency of the deployment models clearly shown in Figure 2, that cloud-only, edge-only and edge-cloud collaboration with latencies of 210 ms, 65ms, and 78 ms respectively.

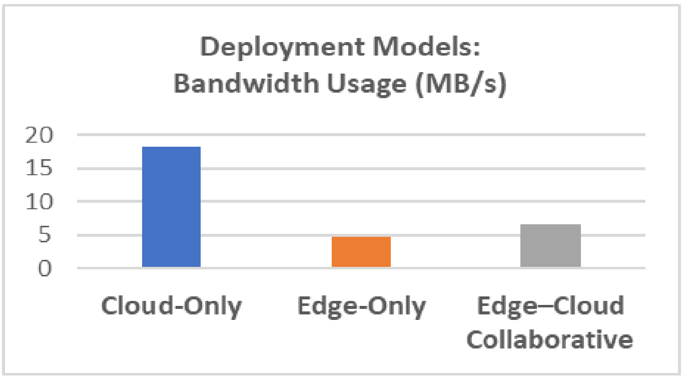

Figure 3: bandwidth Consumption (MB/s)

The Bandwidth consumption (MB/s) of the deployment models shown in Figure 3, that cloud-only, edge-only and edge-cloud collaboration with latencies of 18.2, 4.8, and 6.5 respectively.

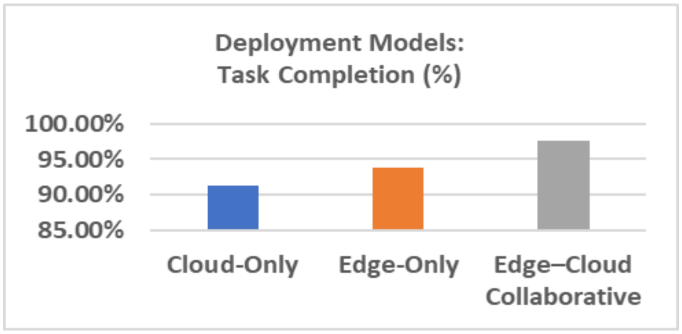

Figure 4: Task Completion Rate (%)

The Task Completion Rate (%) of the deployment models shown in Figure 4, that cloud-only, edge-only and edge-cloud collaboration with latencies of 91.2, 93.8, and 97.6 respectively

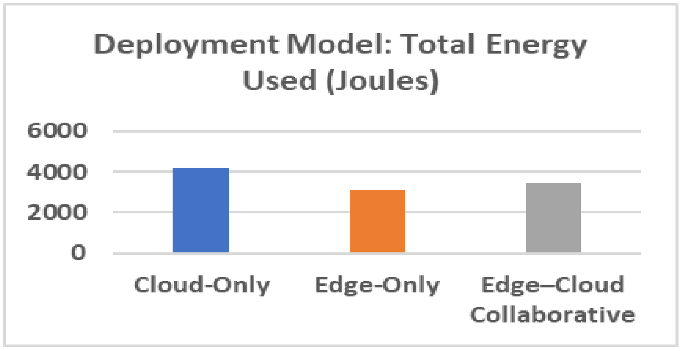

Figure 5. Energy Consumption (Joules)

The Total Energy Consumption/Usage (KJ) of the deployment models shown in figure 5, that cloud-only, edge-only and edge-cloud collaboration with latencies of 4.2. 3.1, and 3.4 respectively.

Discussion

Results were averaged and analysed to compare performance. The simulation using iFogSim and EdgeCloudSim, which evaluated three deployment models; Cloud-Only**,** Edge-Only**, and** Edge–Cloud Collaborative using a smart healthcare use case (heart rate monitoring). Performance metrics were compared across the models.
The result of the average end-to-end latency of the three scenarios, with figure 2 showing the graphical output in a bar chart. Cloud-only, Edge-only and Edge-Cloud collaboration are 210ms, 65ms, and 78ms respectively. The Edge–Cloud model significantly reduced latency by approx. 63% compared to Cloud-Only, and achieved near-Edge performance while leveraging cloud computation. While Figure 3 shows the result of the Bandwidth Consumption three scenarios of the deployment model, showing the graphical output. Cloud-only, Edge-only and Edge-Cloud collaboration Bandwidth consumption in MB/s are is 18.2, 4.8, 6.5 respectively**.** Edge–Cloud collaboration reduced bandwidth by approximately 64% compared to Cloud-Only, due to local pre-processing and selective data transmission.
Whereas the result of the Task completion rate of the different deployment models, with the graphical output. Cloud-only, Edge-only and Edge-Cloud collaboration Task completion rate (%) are 91.2, 93.8, 97.6 **respectively.** Collaborative processing improved system responsiveness and resilience under network stress, leading to the highest task completion rate. While considering Energy consumption of the three deployment models, our result shows the graphical output in a bar chart. Cloud-only, Edge-only and Edge-Cloud collaboration, the Energy consumption rates (kilojoules) are 4.2, 3.1, 3.4 respectively**.** Edge–Cloud collaboration offers a balanced energy profile, saving approx. 20% energy compared to cloud-only while providing better performance.

The outcome of this model of collaboration has key benefits, which includes Latency **r**eduction by over 60% compared to Cloud-Only, Bandwidth **r**eduction by over 60% via local processing**,** Task Reliability improvement by completion under variable loads, Energy Efficiency balanced energy savings with performance. The outcome has evidence that the collaboration model has the potentials to mitigate Latency and Bandwidth Challenges**. (i)** *Latency Reduction:* Local Processing **r**educes round-trip time for control signals. This is achieved by **c**aching and inference at the Edge, where deep learning inference tasks can run on edge accelerators, cutting delay. (ii) ***Bandwidth** Optimization:* Optimization is achieved through data filtering and aggregation**.** This process **r**educes data size before transmission to the cloud. The compression algorithms were **a**pplied at edge nodes to optimize bandwidth usage, offloading **t**asks. The **d**ynamic offloading balances loads across edge and cloud based on network conditions.

Applicable Use Cases: Applicable areas of edge–cloud collaboration span several domains: in *smart healthcare*, wearable devices process vital signs locally and send anomalies to the cloud for expert analysis; in *autonomous vehicles*, onboard edge units manage real-time navigation while the cloud aggregates data for broader traffic pattern analysis; in *smart agriculture*, sensors monitor soil and climate conditions, with local edge nodes triggering irrigation decisions and the cloud analyzing long-term seasonal trends; and in *manufacturing systems*, sensors track data from assembly plants, where local edge nodes enable rapid response actions while the cloud identifies trends in defective situations.

Challenges and Open Issues: Despite the tremendous benefits of the collaboration model in improving latency and bandwidth utilization for source data processing, several challenges and open issues remain. Security and privacy pose significant concerns, as data handled across multiple layers is vulnerable to breaches, making secure protocols and encryption indispensable. Task scheduling complexity also arises from the difficulty of determining where and when tasks should be executed between edge and cloud layers. Furthermore, resource constraints persist because edge nodes possess limited computational and energy capacities, necessitating efficient allocation mechanisms. Finally, the absence of standardization and interoperability, particularly in terms of uniform APIs and protocols, continues to hinder seamless collaboration across heterogeneous systems.

Conclusion

The hybrid Edge–Cloud collaboration emerges as a promising computing paradigm that bridges the limitations of traditional cloud-centric IoT architectures by combining the responsiveness of edge computing with the scalability of the cloud. The simulation results presented in this study demonstrate its effectiveness: reducing average latency by more than 60%, lowering bandwidth consumption by nearly two-thirds, improving task completion rates under network stress, and achieving balanced energy savings compared to cloud-only models. These findings show that collaborative processing not only enhances system responsiveness but also ensures reliability and efficiency in handling dynamic IoT workloads. The strength of this model lies in its ability to adapt across diverse domains. In healthcare, edge-enabled wearables provide immediate monitoring while the cloud supports long-term analytics; in autonomous vehicles, edge units deliver real-time navigation while the cloud aggregates fleet-wide insights; in agriculture, edge nodes manage irrigation decisions in real time while the cloud analyses seasonal and regional patterns; and in manufacturing, rapid fault detection at the edge complements cloud-driven predictive maintenance. Together, these instances illustrate the versatility and impact of Edge–Cloud integration in enabling smarter, faster, and more resilient applications.

Nevertheless, several open issues remain central to its widespread adoption. Security and privacy require urgent attention as data flows across multiple layers and networks, demanding robust encryption and authentication protocols. Task scheduling continues to be a complex problem, requiring intelligent offloading strategies that dynamically balance edge and cloud workloads. Resource constraints at edge nodes also call for efficient allocation methods and lightweight computation techniques. Furthermore, the lack of standardization and interoperability across heterogeneous systems presents a significant barrier to seamless deployment. Overall, this study underscores that Edge–Cloud collaboration is not only technically feasible but also practically necessary for the next generation of IoT systems. By addressing the outstanding challenges through continued research, algorithmic innovation, and standardization efforts, this model has the potential to become the foundational architecture for real-time, bandwidth-efficient, and secure IoT infrastructures that can transform critical sectors such as healthcare, transportation, agriculture, and industry.

Recommendations

Based on our findings the following recommendations are made:

a) Adopt adaptive Edge–Cloud architectures that dynamically allocate tasks between edge nodes and cloud servers based on latency sensitivity and bandwidth availability.

b) Implement efficient resource orchestration frameworks to improve load balancing, scalability, and energy efficiency across distributed IoT systems.

c) Enhance security at both the edge and cloud layers, using lightweight encryption, secure communication protocols, and continuous authentication to protect distributed processing environments.
d) Use edge-based pre-processing and data compression to minimize bandwidth consumption and reduce the volume of data transmitted to the cloud.
e) Promote interoperability standards to ensure seamless integration of heterogeneous edge devices, platforms, and communication protocols.
f) Conduct domain-specific performance evaluations before deployment, as IoT applications have varying real-time and reliability requirements.
g) Invest in AI-driven optimization tools, such as predictive analytics for traffic management and automated task offloading, to improve overall system responsiveness and reliability.

References

Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1), 450–465. https://doi.org/10.1109/JIOT.2017.2750180

Alharbi, H. A., Aldossary, M., Almutairi, J., & Elgendy, I. A. (2023). Energy-aware and secure ask offloading for multi-tier edge-cloud computing systems. *Sensors, 23*(6), 3254. https://doi.org/10.3390/s23063254

Bi, X., & Zhao, L. (2024). Two-layer edge intelligence for task offloading and computing capacity allocation with UAV assistance in vehicular networks. *Sensors, 24*(6), Article 1863. https://doi.org/10.3390/s24061863

Birhanie, H. M., & Adem, M. O. (2024). Optimized task offloading strategy in IoT edge computing network. Journal of King Saud University – Computer and Information Sciences, 36(2), Article 101942. https://doi.org/10.1016/j.jksuci.2024.101942

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*, Article 86. https://doi.org/10.1186/s13677-022-00365-8

Chen, J., Leng, Y., & Huang, J. (2023). An intelligent approach of task offloading for dependent services in Mobile Edge Computing. *Journal of Cloud Computing: Advances, Systems and Applications, 12*(1), Article 107. https://doi.org/10.1186/s13677-023-00477-9

Chen, X., Zhou, Z., Guan, G., & Li, Y. (2022). *Deep reinforcement learning–based offloading and resource allocation in edge-cloud computing systems*. Journal of Network and Computer Applications, 221, 102996.

Daruvuri, R., & Patibandla, K. K. (2023). Enhancing data security and privacy in edge computing: A comprehensive review of key technologies and future directions. *International Journal of Research in Electronics and Computer Engineering, 11*(1), 77–88.

Ferreira, R., Ranaweera, C., Lee, K., & Schneider, J.-G. (2023). *Energy efficient node selection in edge-fog-cloud layered IoT architecture*. Sensors, 23(13), 6039. https://doi.org/10.3390/s23136039

Hossain, M. I., Sumon, S. A., Hasan, H. M., Akter, F., Badhon, M. B., & Islam, M. N. U. (2024). Quantum-Edge Cloud Computing: *A Future Paradigm for IoT Applications*. arXiv preprint arXiv:2405.04824. https://doi.org/10.48550/arXiv.2405.04824

Liu, Y., Zhou, Z., Chen, X., Zhang, J., & Xu, K. (2023). *Optimizing task offloading and resource allocation in edge–cloud networks: A deep reinforcement learning approach*. Journal of Cloud Computing, 12, 1–17. https://doi.org/10.1186/s13677-023-00461-3

Luo, Z., Wang, S., Chen, X., & Lu, Y. (2022). A deep-learning-based collaborative edge–cloud telemedicine system for retinopathy of prematurity screening. *Sensors, 23*(1), Article 276.

Mao, Y., Zhang, J., & Letaief, K. B. (2017). *Resource allocation for mobile edge computation offloading with reinforcement learning*. IEEE Transactions on Wireless Communications, 17(7), 4958–4972.

Puligundla, N., Gangappa, M., Rajasekar, M., Sunil Kumar, T., & Reddy, G. S. (2024). Resource allocation for content distribution in IoT edge cloud computing environments using deep reinforcement learning. *Journal of High Speed Networks*, 3, 409–426. https://doi.org/10.3233/JHS-230165

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9

Statista. (2024). Number of IoT connected devices worldwide 2019–2030. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Ullah, I., Lim, H.-K., Seok, Y.-J., & Han, Y.-H. (2023). Optimizing task offloading and resource allocation in edge-cloud networks: A DRL approach. *Journal of Cloud Computing: Advances, Systems and Applications, 12*, Article 112. https://doi.org/10.1186/s13677-023-00461-3

Yang, C., Wang, Y., Lan, S., Wang, L., Shen, W., Huang, G.Q.(2022). Cloud-edge-device collaboration mechanisms of deep learning models for smart robots in mass personalization. *Robotics and Computer-Integrated Manufacturing*, Volume 77,102351. ISSN 0736-5845. https://doi.org/10.1016/j.rcim.2022.102351

Yu, H., Chen, X., Xu, J., & Li, Z. (2023). *Joint DNN partitioning and task offloading in mobile edge computing via deep reinforcement learning*. Journal of Cloud Computing, 12, 1–21. https://doi.org/10.1186/s13677-023-00493-9

Yuan, J., Xiao, H., Shen, Z., Zhang, T., & Jin, J. (2023). *ELECT: Energy-efficient intelligent edge–cloud collaboration for remote IoT services*. Future Generation Computer Systems, 147, 179–194. https://doi.org/10.1016/j.future.2023.04.030

Zhang, X., Li, Y., & Wang, J. (2020). Research on new edge computing network architecture and task offloading strategy for Internet of Things. *Wireless Networks*, *30*, 3619–3631. https://doi.org/10.1007/s11276-020-02516-8

Zhang, W., Huang, L., Wang, H., & Ren, Y. (2024). *Task partition-based computation offloading and content caching for cloud–edge cooperation networks*. Symmetry, 16(7), 906. https://doi.org/10.3390/sym16070906