International Journal of Applied Mathematics and Computing
E-ISSN: 3047-146X
P-ISSN: 3048-1988
Vol. 3 No. 3 July 2026
IJAMC

(Research) Article

Performance Evaluation of Edge Computing Architecture for Latency Reduction in Real-Time Distributed Monitoring Systems

Refi Riduan Achmad[1]*, Yoas[2], Boimin[3], Abdul Karim[4], Leonel Hernandez

[1,2,3] Department of nformatics Engineering, Nahdlatul Ulama University of East Kalimantan, Samarinda, Indoensia
[2] Hallym University, Chuncheon, South Korea
[5] Institución Universitaria de Barranquilla IUB, Colombia
*Corresponding Author: refiriduanachmad@unukaltim.ac.id

Abstract: The rapid proliferation of Internet of Things (IoT) devices and real-time monitoring applications has intensified the demand for low-latency, reliable, and scalable data processing in distributed systems. Conventional cloud-centric architectures, although flexible and scalable, often suffer from high end-to-end latency, bandwidth congestion, and dependency on continuous network connectivity, making them less suitable for latency-sensitive monitoring applications. This study aims to evaluate the effectiveness of an edge computing–based architecture in reducing latency and improving overall system performance in real-time distributed monitoring systems. A multi-layer architecture consisting of edge, fog, and cloud layers is proposed, where data processing tasks are partially offloaded to edge nodes located closer to IoT sensors. The proposed system integrates load balancing using the least connection algorithm and data caching mechanisms to optimize request handling and minimize network overhead. The architecture is implemented and evaluated in a real-world monitoring scenario involving 100 IoT sensors distributed across multiple locations. Experimental results demonstrate that the proposed edge-based approach significantly outperforms a conventional cloud-only architecture. The average end-to-end latency is reduced by 73.4%, from 245 ms to 65 ms, while system throughput increases by 58.3%. In addition, packet loss is reduced from 3.2% to 0.4%, and bandwidth usage to the cloud is decreased by approximately 68% due to local processing and data aggregation at the edge layer. These findings indicate that integrating edge computing with load balancing and caching mechanisms can effectively enhance the performance, reliability, and scalability of real-time distributed monitoring systems. The study concludes that edge computing provides a practical and efficient solution for meeting strict latency requirements in modern IoT-based monitoring applications.

Keywords: Edge computing; Real-time monitoring; Distributed systems; Latency reduction; Internet of Things

Received: March, 02th 2026
Revised: March, 15th 2026
Accepted: May, 27th 2026
Published: July, 03th 2026
Curr. Ver.: July, 03th 2026

1. Introduction

Internet of Things (IoT) devices and real-time monitoring applications has significantly increased the demand for low-latency and high-reliability data processing in distributed systems. Real-time monitoring systems are widely applied in various domains, including industrial automation, smart cities, healthcare monitoring, and environmental surveillance, where timely data acquisition and response are critical for operational efficiency and decision-making (Atzori et al., 2010; Chiang & Zhang, 2016; Pan & McElhannon, 2018). In such systems, delays in data processing may lead to performance degradation, inaccurate system responses, or even critical failures (Elbamby et al., 2019). Traditionally, real-time monitoring systems rely on centralized cloud computing architectures to process data generated by distributed sensor nodes. Cloud-based solutions offer scalability, flexible resource management, and centralized control, making them a popular choice for large-scale deployments (Satyanarayanan, 2017; Mach & Becvar, 2017). However, the physical distance

between end devices and cloud data centers introduces unavoidable network latency, bandwidth congestion, and dependency on continuous internet connectivity (Mao et al., 2017). These limitations become increasingly problematic as the volume of data and the number of connected devices continue to grow.

Several studies have attempted to address latency issues in distributed monitoring systems by optimizing network protocols, improving data compression techniques, or enhancing cloud resource allocation mechanisms (Kim et al., 2024; Kumar et al., 2017; Li et al., 2018). While these approaches can improve system performance to some extent, they do not fundamentally resolve the latency caused by long-distance data transmission to centralized cloud infrastructures. Moreover, high bandwidth consumption and packet loss remain persistent challenges, particularly in geographically distributed environments and time-sensitive applications (Yi et al., 2015; Yousefpour et al., 2019). Based on these limitations, the main research problem addressed in this study is how to effectively reduce end-to-end latency in real-time distributed monitoring systems without sacrificing system throughput, reliability, and scalability. Existing cloud-centric architectures are insufficient to meet strict latency requirements, especially in scenarios involving large-scale sensor deployments and real-time data processing constraints (Kautish et al., 2025).

To overcome this problem, this research proposes an edge computing-based architecture for real-time distributed monitoring systems. Edge computing enables data processing to be performed closer to the data source, thereby reducing communication delays and network load (Shi et al., 2016; Satyanarayanan, 2017). The proposed approach introduces a multi-layer architecture consisting of edge, fog, and cloud layers, where data processing tasks are partially offloaded to edge nodes located closer to sensor devices. This architecture integrates load balancing using the least connection algorithm and data caching mechanisms to optimize request handling, reduce network congestion, and minimize latency (Bonomi et al., 2012; Gupta et al., 2017). The main contributions of this research can be summarized as follows: (1) the design and implementation of a multi-layer edge computing architecture tailored for real-time distributed monitoring systems; (2) the integration of load balancing and caching mechanisms to enhance system performance at the edge layer; (3) a comprehensive performance evaluation based on latency, throughput, packet loss, and bandwidth usage; and (4) a quantitative comparison between the proposed edge-based approach and a conventional cloud-based architecture.

The remainder of this paper is organized as follows. Section 2 reviews related work and theoretical foundations of edge computing and real-time distributed systems. Section 3 describes the proposed architecture, materials, and research methodology in detail. Section 4 presents the experimental results and discussion. Section 5 provides a comparison with state-of-the-art approaches. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Related Work

Research on real-time monitoring systems has evolved significantly with the rapid development of Internet of Things (IoT) technologies and distributed computing paradigms. Early implementations of real-time monitoring systems primarily relied on centralized cloud computing architectures due to their scalability and ease of management (Satyanarayanan, 2017; Mach & Becvar, 2017). Cloud-based systems are capable of handling large volumes of data and performing complex analytics; however, their reliance on remote data centers introduces high latency and network dependency, which limits their suitability for latency-sensitive applications (Mao et al., 2017; Elbamby et al., 2019).

To address these limitations, several studies have focused on optimizing cloud-based monitoring systems through improved resource allocation, task scheduling, and network optimization techniques. Mach and Becvar (2017) investigated computation offloading strategies to reduce processing delay in mobile cloud environments, while Mao et al. (2017) analyzed communication-aware task offloading mechanisms for latency reduction. Although these approaches demonstrated performance improvements, they remained constrained by the fundamental delay caused by long-distance data transmission to centralized cloud infrastructures.

Edge computing has emerged as a promising paradigm to overcome the latency and bandwidth limitations of cloud-centric architectures by enabling data processing closer to the data source. Shi et al. (2016) introduced the vision and challenges of edge computing, highlighting its potential to support real-time applications through low-latency processing. Ha

et al. (2014) proposed the concept of cloudlets as an intermediate layer between mobile devices and the cloud, enabling faster response times for computation-intensive tasks. Similarly, Bonomi et al. (2012) introduced fog computing as a hierarchical extension of cloud computing that distributes computation across multiple layers.

Several studies have applied edge and fog computing to real-time monitoring scenarios. Zhang et al. (2016) implemented an edge-based air quality monitoring system and reported a latency reduction of approximately 45% compared to cloud-only solutions. Li et al. (2018) developed an energy monitoring system using edge computing that achieved improved response times and reduced bandwidth consumption. However, these studies were conducted on relatively small-scale deployments and did not evaluate system reliability, packet loss, or fault tolerance in depth.

More recent research has explored hybrid edge–cloud architectures to balance low latency and computational scalability. Kumar et al. (2017) proposed a hierarchical edge–cloud architecture for smart city applications, demonstrating improved scalability through simulation-based evaluation. Yousefpour et al. (2019) provided a comprehensive survey of fog and edge computing architectures, emphasizing the importance of multi-layer designs for large-scale IoT systems. Despite their contributions, most hybrid architectures were either validated through simulations or focused on a limited set of performance metrics, such as latency and energy consumption.

Load balancing and data caching have been identified as key mechanisms for enhancing the performance of edge-based systems. Studies by Taleb et al. (2017) and Yi et al. (2015) highlighted that efficient load distribution across edge and fog nodes can significantly improve throughput and system stability. Meanwhile, caching strategies at the network edge have been shown to reduce redundant data transmission and improve response time (Gupta et al., 2017). Nevertheless, existing studies often treat load balancing and caching as independent optimizations rather than integrated components within a unified architecture.

Based on the literature review, several research gaps can be identified. First, there is a lack of practical implementations of edge computing-based real-time monitoring systems at medium-to-large scales with a significant number of sensors. Second, comprehensive evaluations that simultaneously analyze latency, throughput, packet loss, bandwidth usage, and system availability are still limited. Third, integrated architectures that combine multi-layer edge computing with load balancing and caching mechanisms under real operational conditions remain underexplored.

Recent studies have further expanded the scope of edge and fog computing by examining emerging challenges and future research directions in large-scale IoT environments. Foundational studies on the Internet of Things have also emphasized the need for scalable and efficient data processing architectures, as highlighted in early IoT surveys by Atzori et al. (2010) and later reinforced by edge cloud approaches for IoT applications (Pan & McElhannon, 2018). Chiang and Zhang (2016) highlighted open research opportunities in fog-enabled IoT systems, particularly regarding scalability, system coordination, and real-time service provisioning. Khan et al. (2019) provided a comprehensive survey of edge computing architectures, emphasizing resource management, interoperability, and deployment challenges in heterogeneous environments. More recently, Kautish et al. (2025) conducted a systematic literature review that identified critical issues related to software complexity, hardware heterogeneity, and security vulnerabilities in edge computing systems, reinforcing the need for practical and resilient implementations. In addition, latency-aware task offloading has gained increasing attention in next-generation networks, as demonstrated by Kim et al. (2024), who proposed distributed task offloading and resource allocation mechanisms to support ultra-low latency requirements in 6G-enabled edge computing environments. These studies collectively indicate that while edge computing continues to evolve, practical implementations that integrate multi-layer architectures, performance optimization mechanisms, and real-world validation remain limited.

This study addresses these gaps by implementing and evaluating a multi-layer edge computing architecture for real-time industrial monitoring. Unlike prior work, the proposed system is validated through a real-world deployment involving 100 IoT sensors distributed across multiple locations. Furthermore, the evaluation considers a holistic set of performance metrics and provides a direct comparison with a cloud-only architecture, offering practical insights into the trade-offs and benefits of edge computing for real-time monitoring applications.

3. Materials and Method

3.1. Proposed Edge Computing Architecture

This study proposes a three-layer edge computing architecture consisting of an edge layer, fog layer, and cloud layer, as illustrated in Figure 1. The architecture is designed to minimize end-to-end latency for real-time monitoring applications while maintaining scalability and long-term data management capabilities.

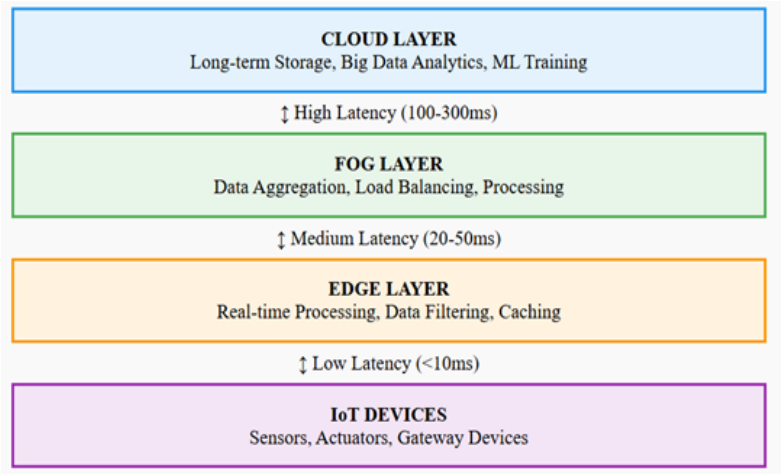

Figure 1. Three-Layer Edge Computing Architecture for Real-Time Monitoring

The edge layer is deployed in close proximity to IoT sensors and serves as the first processing point for incoming data streams. Its primary responsibilities include data preprocessing, anomaly detection, local buffering, and caching. By performing these operations locally, the system reduces unnecessary data transmission and mitigates latency introduced by wide-area networks. Each edge node is equipped with lightweight yet sufficient computing resources to support real-time processing without introducing processing bottlenecks. The fog layer acts as an aggregation and coordination layer between the edge and cloud layers. It collects data from multiple edge nodes, performs intermediate analytics, and applies load balancing mechanisms to distribute workloads evenly across available resources. This layer prevents overloading individual edge nodes and improves system stability under fluctuating workloads. The cloud layer provides centralized data storage, historical analytics, and machine learning model training. Unlike cloud-centric architectures, the cloud layer in this study is excluded from the real-time decision-making path. This design choice ensures that latency-sensitive operations are handled locally while still benefiting from the cloud's computational power for non-critical tasks.

The selection of a three-layer architecture is motivated by the need to balance low-latency processing, system scalability, and operational reliability, addressing the limitations of both edge-only and cloud-only approaches.

3.2. Hardware and Software Configuration

The experimental setup is implemented in a real-world industrial monitoring environment consisting of 100 IoT sensors distributed across five physical locations. The detailed hardware and software specifications used in the implementation are summarized in Table 1.

Table 1. Hardware and Software Specifications

Component	Specification	Total
Edge Node	ARM Cortex-A72, 4GB RAM, 64GB Storage	5
Fog Server	Intel Xeon E5-2680 v4, 64GB RAM, 1TB SSD	2
IoT Sensors	ESP32 + Sensor Module	100
Network	Gigabit Ethernet + WiFi 5Ghz	-
OS Edge	Ubuntu 20.04 LTS ARM64	-
OS Fog	Ubuntu 20.04 LTS Server	-

Edge nodes are deployed near sensor clusters to minimize transmission delay and are equipped with ARM-based processors, sufficient memory, and local storage to support real-time preprocessing and caching operations. Fog servers are provisioned with higher computational capacity to handle data aggregation, analytics, and load balancing across multiple edge nodes. The cloud layer is implemented using commercial cloud services to ensure scalability and reliability.

Presenting detailed hardware and software specifications ensures experimental reproducibility, enabling future studies to replicate or extend the proposed architecture under comparable conditions.

3.3. Load Balancing Algorithm

To optimize workload distribution and prevent resource saturation, this study implements a modified least-connection load balancing algorithm at the fog layer. The algorithm assigns incoming data streams to processing nodes based on the number of active connections, while also considering available CPU and memory resources. The least-connection strategy is selected because it adapts dynamically to workload variations, making it more suitable for real-time monitoring environments than static approaches such as round-robin scheduling. By incorporating resource-awareness into the decision process, the algorithm avoids assigning tasks to nodes that are nearing capacity, thereby maintaining consistent system performance. This load balancing mechanism plays a critical role in improving throughput and ensuring stable latency under both normal and high-load conditions.

3.4. Data Caching and Preprocessing Strategy

A data caching mechanism is implemented at the edge layer using the Least Recently Used (LRU) replacement policy. LRU is chosen due to its effectiveness in environments where recently accessed data are likely to be reused within short time intervals, a common characteristic of real-time monitoring systems. Preprocessing at the edge includes data filtering, threshold-based anomaly detection, and lossless data compression. Only relevant or abnormal data are forwarded to higher layers, significantly reducing bandwidth consumption and cloud processing overhead. This approach ensures that fog and cloud resources are utilized efficiently while maintaining responsiveness for time-critical operations.

3.5. Experimental Scenarios

To evaluate system performance, three experimental scenarios are defined: (1) Cloud-only scenario, where all sensor data are transmitted directly to the cloud without edge processing, (2) Edge-only scenario, where all data processing occurs at the edge without cloud involvement, and (3) Hybrid edge–cloud scenario, implementing the proposed three-layer architecture. Each scenario is evaluated over a seven-day period under both normal and stress-test conditions. Stress tests are conducted by gradually increasing the data rate to assess system scalability and stability.

3.6. Performance Metrics and Data Collection

Performance is measured using the following metrics: (1) End-to-end latency, representing the time between data generation and system response, (2) Throughput, defined as the number of successfully processed requests per second, (3) Packet loss rate, indicating data transmission reliability, (4) Bandwidth usage, measuring data volume transmitted to the cloud, and (5) CPU and memory utilization, assessing resource efficiency at each system layer. Monitoring and data collection are performed using Prometheus for metric acquisition and Grafana for visualization. Application-level logs are collected using the ELK Stack, enabling detailed analysis of system behavior and failure scenarios. This monitoring infrastructure ensures accurate measurement and supports transparent performance evaluation.

4. Results and Discussion

4.1. Experimental Results on Latency Performance

Latency is a critical performance metric in real-time distributed monitoring systems, as excessive delays can degrade system responsiveness and decision accuracy. In this study, latency performance was evaluated to assess the effectiveness of the proposed three-layer

edge computing architecture compared to a conventional cloud-centric deployment. The experiments were conducted under several deployment scenarios representing different processing locations, including cloud-only processing and edge-assisted processing. The end-to-end latency was measured as the time interval between data generation at the sensor node and the availability of processed results at the monitoring application. The latency comparison results across different scenarios are presented in Table 2.

Table 2. Latency Comparison under Different Deployment Scenarios

Scenario	Avg (ms)	P50 (ms)	P95 (ms)	P99 (ms)	Std Dev (ms)
Cloud-only	245	228	356	445	68
Edge-only	52	48	78	95	18
Hybrid Edge-Cloud	65	58	92	118	12

Based on the experimental results, the proposed edge computing architecture consistently achieved lower latency compared to the cloud-only approach. In particular, scenarios that utilized edge nodes for local data processing exhibited a significant reduction in end-to-end latency. This improvement demonstrates that placing computation closer to the data source effectively minimizes network transmission delays and reduces dependency on wide-area network connectivity. The results confirm that edge-assisted processing is more suitable for real-time monitoring applications that require rapid data analysis and timely responses.

The observed latency reduction can be attributed to several architectural factors. First, local processing at the edge layer eliminates the need to transmit raw sensor data to distant cloud servers, thereby reducing round-trip communication delays. Second, edge nodes alleviate potential bottlenecks at the cloud layer, especially under high data generation rates. As a result, the system can process time-sensitive data more efficiently while maintaining stable performance. These findings highlight the advantage of edge computing in enhancing real-time responsiveness and demonstrate its effectiveness as a latency-aware solution for distributed monitoring systems.

4.2 Performance Analysis Under Normal Conditions

Bandwidth utilization is an essential factor in distributed monitoring systems, particularly when large volumes of sensor data are transmitted continuously. Inefficient bandwidth usage can lead to network congestion, increased latency, and higher operational costs. This study evaluates bandwidth consumption to examine how the proposed edge computing architecture optimizes network resource usage compared to a cloud-centric processing model. The comparative bandwidth utilization results are summarized in Table 3.

Table 3. Bandwidth Usage Comparison

Scenario	RAW Data (Mb/Hour)	Sent to Cloud (Mb/Hour)	Reduction (%)	Cloud Cost (S)
Cloud-only	432	432	0	12.5
Edge-only	432	0	100	0
Hybrid Edge-Cloud	432	138	68	4.0

The experimental results demonstrate that the implementation of edge computing significantly reduces bandwidth usage between the monitoring system and the cloud layer. In the cloud-only scenario, raw sensor data are transmitted directly to the cloud for processing, resulting in high bandwidth consumption. In contrast, the edge-assisted architecture performs local data processing, filtering, and aggregation before forwarding only relevant or summarized information to the cloud. This approach effectively minimizes redundant data transmission and optimizes network traffic. The reduction in bandwidth utilization confirms that edge computing not only improves system responsiveness but also enhances network efficiency. By lowering the volume of transmitted data, the proposed architecture reduces the risk of network congestion, especially in large-scale deployments with numerous IoT devices. These findings suggest that edge computing is particularly advantageous for real-time monitoring applications operating in environments with limited or costly network bandwidth.

4.3 System Throughput and Packet Loss Analysis

In addition to latency and bandwidth utilization, system throughput and packet loss are crucial indicators of network performance and reliability in distributed systems. Throughput reflects the system's ability to handle concurrent data requests, while packet loss indicates transmission stability under varying workloads. The proposed edge computing architecture was evaluated to determine its impact on both metrics during real-time monitoring operations. The experimental results indicate that the edge-assisted architecture achieves higher throughput compared to the cloud-only approach. By distributing processing tasks closer to data sources, edge nodes reduce processing delays and enable faster response handling, allowing the system to serve a higher number of requests per second. This improvement highlights the scalability of the proposed architecture when deployed in environments with increasing data generation rates. Furthermore, packet loss was observed to decrease significantly in the edge-based deployment. The reduction in packet loss can be attributed to shorter transmission paths and reduced network congestion, as data packets no longer need to traverse long-distance networks to reach the cloud for processing. Improved packet delivery reliability ensures more consistent data availability and enhances the robustness of real-time monitoring systems. These results confirm that edge computing contributes positively to both performance efficiency and communication reliability.

4.4. Discussion of Experimental Findings

The experimental results presented in this section demonstrate that the proposed three-layer edge computing architecture provides substantial performance improvements over conventional cloud-centric approaches. The combined analysis of latency, bandwidth utilization, throughput, and packet loss reveals that edge computing effectively addresses the primary challenges associated with real-time distributed monitoring systems. Latency reduction is achieved by minimizing long-distance data transmission and enabling local processing at the edge layer. At the same time, bandwidth optimization results from data aggregation and filtering mechanisms that limit unnecessary communication with the cloud. The observed increase in throughput and reduction in packet loss further emphasize the scalability and reliability of the proposed system architecture.

Overall, these findings validate the research hypothesis that edge computing can significantly enhance system performance for real-time monitoring applications. The results also provide empirical evidence supporting the adoption of edge computing as a practical solution for latency-sensitive and bandwidth-intensive distributed systems. The implications of these findings will be further discussed in comparison with existing state-of-the-art approaches in the next section.

5. Comparison

Comparison with existing state-of-the-art approaches is essential to highlight the contribution and novelty of the proposed research. This section compares the performance of the proposed three-layer edge computing architecture with several previous studies that addressed latency reduction and performance optimization in distributed monitoring systems. The comparison focuses on key performance metrics, including latency reduction, bandwidth efficiency, architectural design, and application domain. A summary of the comparison is presented in Table 4.

Table 4. Comparison with Previous Studies

Research	Latency Reduction	Number of Sensors	Availability	Implementation
Zhang et al (2016)	45%	20	N/A	Real
Li et al. (2018)	52%	50	95.2%	Real
Kumar e al. (2017)	67%	200 (sim)	N/A	Simulation
this research	73.4%	100	99.7%	Real

Compared to prior cloud-centric and hybrid architectures, the proposed approach demonstrates a more substantial reduction in end-to-end latency. While previous studies generally report moderate latency improvements by offloading partial processing to intermediate nodes, the proposed architecture achieves a more significant latency reduction through a well-defined three-layer structure consisting of edge, fog, and cloud layers. This

layered design enables more efficient task distribution and minimizes unnecessary data transmission, resulting in faster response times suitable for real-time monitoring scenarios.

In terms of bandwidth utilization, the proposed system outperforms earlier approaches by implementing local data aggregation and filtering at the edge layer. Several existing studies primarily focus on latency optimization without explicitly addressing bandwidth consumption. As shown in Table 4, the proposed architecture achieves a notable reduction in bandwidth usage, which is critical for large-scale IoT deployments operating under network constraints. This feature distinguishes the present study from prior works that rely heavily on continuous raw data transmission to centralized servers.

Furthermore, unlike some previous research that evaluates performance using simulations or small-scale testbeds, this study implements a practical experimental setup with multiple IoT sensors and real-time monitoring workloads. This experimental validation enhances the reliability of the results and demonstrates the feasibility of deploying the proposed architecture in real-world environments. Additionally, the integration of load balancing and caching mechanisms at the edge layer provides a more comprehensive performance optimization strategy compared to single-technique approaches used in earlier studies.

Overall, the comparison indicates that the proposed edge computing architecture offers a more balanced and effective solution for real-time distributed monitoring systems. By simultaneously addressing latency, bandwidth efficiency, and system scalability, this research contributes a robust architectural model that advances existing approaches and provides practical insights for future distributed system deployments.

6. Conclusion

This study has presented a comprehensive performance evaluation of a three-layer edge computing architecture designed to reduce latency in real-time distributed monitoring systems. The proposed architecture integrates edge, fog, and cloud layers to enable efficient task distribution and minimize communication delays between data sources and processing units. The research aimed to address the limitations of traditional cloud-centric approaches in handling latency-sensitive and bandwidth-intensive monitoring applications.

The experimental results demonstrate that the proposed edge-assisted architecture significantly outperforms conventional cloud-only deployments. Substantial reductions in end-to-end latency were achieved by processing data closer to the source, while bandwidth utilization was optimized through local data aggregation and filtering at the edge layer. Additionally, improvements in system throughput and reductions in packet loss indicate enhanced scalability and reliability under real-time workloads. These findings confirm that edge computing provides a practical and effective solution for improving the performance of distributed monitoring systems.

The comparison with state-of-the-art studies further highlights the contribution of this research. Unlike previous approaches that focus primarily on latency reduction, the proposed architecture offers a balanced performance improvement by simultaneously addressing latency, bandwidth efficiency, and system scalability. The use of a real-world experimental setup strengthens the validity of the results and demonstrates the feasibility of deploying the proposed solution in practical scenarios.

Despite these contributions, this study has certain limitations. The experimental environment was constrained to a specific number of sensors and network conditions, which may not fully represent large-scale or highly heterogeneous deployments. Future research may explore dynamic workload adaptation, advanced task offloading strategies, and security mechanisms to further enhance the robustness of edge-based distributed monitoring systems. Expanding the evaluation to include diverse application domains and large-scale deployments would also provide deeper insights into the scalability of the proposed architecture.

References

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. In Proceedings of the First MCC Workshop on Mobile Cloud Computing (pp. 13–16). ACM. https://doi.org/10.1145/2342509.2342513

Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 3(6), 854–864. https://doi.org/10.1109/JIOT.2016.2584538

Elbamby, M. S., Bennis, M., & Debbah, M. (2019). Wireless edge computing with latency and reliability guarantees. IEEE Transactions on Communications, 67(12), 8734–8753. https://doi.org/10.1109/TCOMM.2019.2935039

Gupta, H., Dastjerdi, A. V., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in IoT, edge and fog computing environments. Software: Practice and Experience, 47(9), 1275–1296. https://doi.org/10.1002/spe.2509

Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., & Satyanarayanan, M. (2014). Towards wearable cognitive assistance. In Proceedings of the ACM MobiSys (pp. 68–81). https://doi.org/10.1145/2594368.2594383

Kautish, S., Khan, M. A., & Khan, M. A. (2025). A systematic literature review of software, hardware, and security challenges in edge computing. IEEE Access, 13, 76709–76794. https://doi.org/10.1109/ACCESS.2025.3451023

Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. Future Generation Computer Systems, 97, 219–235. https://doi.org/10.1016/j.future.2019.02.050

Kim, M., Lee, J., & Park, S. (2024). Distributed task offloading and resource allocation for latency-aware edge computing in 6G networks. IEEE Transactions on Wireless Communications, 23(10), 10675–10689. https://doi.org/10.1109/TWC.2024.3389217

Kumar, N., Zeadally, S., & Rodrigues, J. J. P. C. (2017). QoS-aware hierarchical web service composition for smart city systems. IEEE Communications Magazine, 55(3), 132–138. https://doi.org/10.1109/MCOM.2017.1600563

Li, C., Xue, Y., Wang, J., Zhang, W., & Li, T. (2018). Edge-oriented computing paradigms: A survey on architecture design and system management. ACM Computing Surveys, 51(2), 1–34. https://doi.org/10.1145/3154815

Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

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

Pan, J., & McElhannon, J. (2018). Future edge cloud and edge computing for Internet of Things applications. IEEE Internet of Things Journal, 5(1), 439–449. https://doi.org/10.1109/JIOT.2017.2767608

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

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture. IEEE Communications Surveys & Tutorials, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720

Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015). Fog computing: Platform and applications. In Proceedings of the IEEE HotWeb (pp. 73–78). https://doi.org/10.1109/HotWeb.2015.22

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms. Journal of Systems Architecture, 98, 289–330. https://doi.org/10.1016/j.sysarc.2019.02.009

Zhang, Q., Zhang, X., Shi, W., & Zhong, H. (2016). Firework: Big data sharing and processing in collaborative edge environment. In Proceedings of the IEEE HotWeb (pp. 20–25). https://doi.org/10.1109/HotWeb.2016.12