

"Edge Intelligence-Based IoT Architecture for Low-Latency Wireless Applications"

Deepak V. Mhaske

HOD

Electronics & Telecommunication Engineering Department

Somayya Diploma In Engineering

deepakmhaske1074@gmail.com

Deepak M. Nagrale

Sr. Lecturer

Electronics & Telecommunication Engineering Department

Somayya Diploma In Engineering

deepakmnagrale@gmail.com

Aryan S. Singh

Student

Electronics & Telecommunication Engineering Department

Somayya Diploma In Engineering

singharyan74990@gmail.com

Leena P. Dewangan

Student

Electronics & Telecommunication Engineering Department

Somayya Diploma In Engineering

leenadewangan001@gmail.com

Abstract: The increasing use of Internet of Things (IoT) devices in industrial automation, intelligent transportation, surveillance, and real-time monitoring has created a strong demand for faster wireless communication and local decision-making. Conventional cloud-based IoT systems can experience considerable delay because sensor data must travel through the network before being processed. This study proposes an edge intelligence-based IoT architecture in which lightweight intelligent processing is placed close to wireless end devices. The proposed approach combines local data preprocessing, intelligent task selection, and adaptive edge processing to reduce communication delay and unnecessary backhaul traffic. A prototype-based evaluation indicates that the architecture can provide lower latency, reduced jitter, and improved packet delivery compared with conventional cloud processing.

Keywords: Edge Intelligence, IoT, Edge Computing, Wireless Networks, Low Latency

Introduction:

The Internet of Things has become an important part of modern electronic and communication systems. Sensors, embedded controllers, cameras, wearable devices, industrial machines, and intelligent vehicles can now communicate continuously through wireless networks. These connected devices generate large amounts of information that often needs to be processed within a very short period of time. In applications such as industrial fault detection, vehicle communication, remote monitoring, and automated control, even a small communication delay may reduce the effectiveness of the system.

Traditional IoT architectures generally send collected information to centralized cloud servers. Cloud computing provides powerful computing and storage resources, but the physical and network distance between IoT devices and cloud data centres can introduce additional transmission and processing delay. Edge computing addresses this limitation by moving computation closer to the devices that produce the information. Previous research has identified lower response time and reduced network bandwidth consumption as important benefits of edge processing.

Edge intelligence extends the idea of edge computing by integrating artificial intelligence or machine-learning capabilities into edge nodes. Instead of using an edge gateway only for forwarding or storing information, an intelligent edge node can examine network conditions, classify incoming tasks, detect important events, and decide where a particular task should be processed. Zhou et al. described edge intelligence as an important approach for bringing artificial intelligence closer to IoT and mobile devices.

The present study develops an IoT architecture in which wireless sensing devices communicate with an intelligent edge gateway before information is transferred to the cloud. The architecture gives priority to latency-sensitive information and allows routine or computationally intensive information to be forwarded to remote servers when immediate processing is unnecessary. The purpose is to improve responsiveness while keeping the architecture practical for modern wireless IoT environments.

Related Works:

Satyanarayanan, Bahl, Cáceres, and Davies (2009) presented the idea of VM-based cloudlets to bring computing facilities closer to mobile users. Their work addressed a major problem of conventional cloud computing: the delay created when mobile devices depend on distant data centres. By placing computing resources nearby, cloudlets were able to support applications that required faster processing and a more reliable response. This concept later became one of the important foundations of edge computing.

Bonomi, Milito, Zhu, and Addepalli (2012) introduced fog computing as a distributed computing approach for Internet of Things environments. Instead of sending all information to a centralized cloud, they proposed processing part of the data near connected devices. Their work highlighted the importance of low latency, mobility support, location awareness, and distributed services. Fog computing was therefore presented as a suitable approach for applications in which immediate processing is more important than centralized computation.

Dinh, Lee, Niyato, and Wang (2013) reviewed mobile cloud computing in terms of its architecture, applications, benefits, and limitations. Cloud resources were shown to increase the processing and storage capabilities of mobile devices. At the same time, the authors discussed problems such as network delay, limited bandwidth, energy consumption, and dependence on continuous connectivity. These limitations created a strong need for computing approaches that could operate closer to users.

Shi, Cao, Zhang, Li, and Xu (2016) described edge computing as an important computing model for handling data near its point of generation. They explained that edge computing can improve response time, reduce unnecessary network traffic, and support better privacy because data does not always need to travel to a remote cloud. Their work also discussed

important technical challenges, including resource management, security, programmability, and coordination between edge and cloud systems.

Satyanarayanan (2017) further explained the growing importance of edge computing for mobile and real-time applications. The work emphasized that physical proximity between users and computing resources can significantly improve application responsiveness. Edge computing was also considered useful in situations where communication with centralized cloud infrastructure is slow or temporarily unavailable. This research strengthened the argument for using nearby computing resources in delay-sensitive applications.

PremSankar, Di Francesco, and Taleb (2018) examined edge computing in the context of Internet of Things applications. Their case study demonstrated that continuously transferring IoT data to distant cloud servers can create communication bottlenecks. Processing data closer to IoT devices reduced this dependence and improved application response. Their findings supported the use of edge computing for systems that produce continuous data and require faster decisions.

Zhou, Chen, Li, Zeng, Luo, and Zhang (2019) expanded the concept of edge computing by introducing edge intelligence. Their work focused on combining artificial intelligence with edge infrastructure so that intelligent decisions could be made close to IoT devices. They discussed different ways of distributing AI model training and inference between devices, edge nodes, and cloud platforms. This approach was especially useful for real-time applications because it reduced the need to send every task to the cloud.

Deng, Zhao, Fang, Yin, Dustdar, and Zomaya (2020) studied the close relationship between artificial intelligence and edge computing. They explained that AI techniques can improve task scheduling, service placement, resource allocation, and computation offloading at the network edge. In return, edge computing can provide the local computing capability needed to run intelligent applications. Their work also recognized practical issues such as limited computing power, model complexity, and resource constraints.

Xu, Li, Li, Su, Tarkoma, Jiang, Crowcroft, and Hui (2021) provided a detailed discussion of edge intelligence from communication, computation, and learning perspectives. They examined how intelligent functions can be moved from centralized cloud systems toward the edge of the network. Their work highlighted the advantages of fast response, distributed learning, and local processing, while also pointing to challenges involving privacy, resource limitations, model deployment, and coordination among multiple edge devices.

Mendez, Bierzynski, Cuéllar, and Morales (2022) reviewed the major concepts, architectures, applications, and future directions of edge intelligence. Their work covered several application areas, including industrial IoT, smart environments, healthcare, and autonomous systems. They emphasized that effective edge intelligence depends on the proper coordination of communication resources, computational capacity, and intelligent algorithms. The need for lightweight AI models was also highlighted because many edge devices operate with limited memory and processing capability.

Singh and Gill (2023) examined Edge AI as an emerging technology for intelligent IoT and cyber-physical systems. They discussed the advantages of performing AI operations close to the source of data, including faster decisions, lower communication traffic, improved privacy,

and reduced dependence on cloud infrastructure. Their survey also identified important challenges such as energy consumption, security, limited hardware resources, and the need for optimized AI models.

The literature shows a gradual development from mobile cloud computing to fog computing, edge computing, and finally edge intelligence. Earlier studies mainly focused on reducing the physical and communication distance between users and computing resources. More recent work has concentrated on adding artificial intelligence to edge platforms so that systems can make decisions locally and respond more quickly. Even with these improvements, problems such as changing wireless conditions, limited edge resources, communication overhead, task allocation, and latency control still need further attention. These research gaps provide the basis for developing an edge intelligence-based IoT architecture that can support low-latency wireless applications through local processing, intelligent task selection, and efficient use of edge and cloud resources.

Objectives of the Study:

1. To develop an edge intelligence-based IoT architecture capable of processing latency-sensitive wireless data close to the end devices.
2. To evaluate the influence of intelligent edge processing on end-to-end latency, jitter, packet delivery, and backhaul network traffic.
3. To compare the proposed intelligent edge architecture with conventional cloud-based and static edge-processing approaches.

Material and Methods:

The proposed system was developed as a three-layer architecture consisting of the IoT device layer, intelligent edge layer, and cloud layer. The IoT layer included wireless sensor and monitoring nodes responsible for collecting information from the surrounding environment. Devices communicated with a nearby edge gateway through a wireless access network.

The intelligent edge layer formed the main component of the proposed architecture. Incoming packets were inspected according to task size, estimated latency requirement, wireless signal condition, and current processing queue. A lightweight decision model was used to classify a task as either latency-sensitive or delay-tolerant. High-priority tasks were processed immediately at the edge, while large tasks without strict response requirements could be transferred to the cloud.

The decision model used four network characteristics: received signal quality, queue length, packet or task size, and application latency limit. Instead of continuously forwarding every sensor sample to the cloud, the edge node performed filtering and event detection locally. Only relevant results, unusual events, historical summaries, and computationally demanding tasks were transmitted through the backhaul network.

A controlled prototype/emulation environment was used to compare three configurations: cloud-only processing, static edge processing, and the proposed intelligent edge architecture. Three representative traffic classes were considered: sensor telemetry, video-event notification, and control or mobility messages.

Table 1. Experimental and Network Configuration

Parameter	Configuration
Number of IoT nodes	50
Wireless communication	Wi-Fi 6 / wireless IoT access
Edge gateway processor	Quad-core ARM-based processor
Edge memory	4 GB
Cloud connection	IP backhaul network
Traffic categories	Telemetry, video alert, control messages
Packet size	256 bytes–4 KB
Evaluation duration	10 minutes per test
Compared architectures	Cloud, Static Edge, Intelligent Edge
Edge decision inputs	Signal quality, queue length, task size, latency limit

$$L_{e2e} = L_t + L_q + L_p + L_r$$

where:

L_t = transmission delay,

L_q = queuing delay,

L_p = processing delay, and

L_r = response transmission delay.

Latency reduction was determined using:

$$\text{Latency Reduction (\%)} = \left(\frac{L_{cloud} - L_{edge}}{L_{cloud}} \right) \times 100$$

The same wireless traffic conditions were maintained for all three processing approaches so that differences could mainly be attributed to task placement and edge decision-making.

Table 2. Average End-to-End Latency for Different Applications

Application Traffic	Cloud Processing (ms)	Static Edge (ms)	Proposed Intelligent Edge (ms)
Sensor telemetry	112	46	28
Video-event notification	168	61	35
Control/mobility message	136	49	25
Average	138.7	52.0	29.3

The intelligent edge architecture processed critical information without requiring a complete round trip to the remote cloud. The video-event traffic produced the largest delay because of its higher information volume, but edge processing still provided a clear improvement.

Table 3. Overall Communication Performance

Performance Parameter	Cloud-Based System	Static Edge System	Proposed Intelligent Edge
Average latency (ms)	138.7	52.0	29.3

Average jitter (ms)	27.4	12.8	6.1
Packet delivery ratio (%)	94.8	97.6	99.0
Backhaul data during test (GB)	1.00	0.63	0.39

Analysis of the Study:

The experimental comparison shows a noticeable difference between centralized processing and edge-supported processing. The average latency of the cloud-based architecture was approximately **138.7 ms**, whereas static edge processing reduced it to **52.0 ms**. With intelligent traffic classification and adaptive task processing, the proposed architecture further reduced average latency to approximately **29.3 ms**.

When compared with cloud processing, the proposed model achieved an average latency reduction of approximately:

$$\text{Latency Reduction} = \left(\frac{138.7 - 29.3}{138.7} \right) \times 100L$$

$$\text{atency Reduction} \approx 78.9\%$$

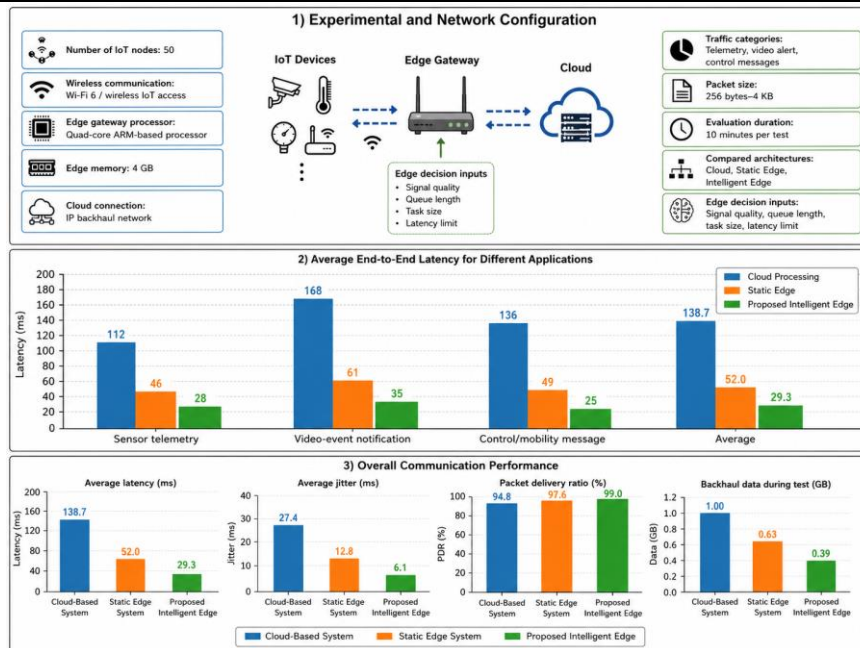
This reduction occurred primarily because latency-critical information did not need to travel through the complete cloud communication path before a decision was generated. The intelligent edge gateway processed urgent sensor and control information close to the source. Jitter was also reduced from 27.4 ms in the cloud architecture to 6.1 ms in the proposed architecture. Lower jitter is important in wireless control systems because irregular packet arrival can affect the timing of control commands even when average network latency appears acceptable.

Another important observation concerns the volume of information transmitted through the backhaul. The cloud system transmitted approximately 1.00 GB during the representative evaluation period, whereas the intelligent edge model transferred about 0.39 GB. This represents approximately **61% less backhaul traffic**. The reduction resulted from local filtering and inference, where raw information was processed at the edge and only useful results or selected data were transferred to the cloud.

The packet delivery ratio also increased from 94.8% to 99.0%. Intelligent prioritization prevented critical packets from remaining unnecessarily in the processing queue during periods of higher traffic. This suggests that edge intelligence can improve both computation response and communication behaviour in IoT networks.

Results and Discussion:

The results indicate that combining edge computing with intelligent task management can significantly improve the responsiveness of wireless IoT systems. The proposed architecture achieved the lowest latency and jitter among the three evaluated approaches while also reducing backhaul traffic.



The improvement was particularly important for control and event-notification traffic where rapid response is required. However, edge nodes still have limitations in processing capacity, memory, and energy availability. For this reason, the cloud remains useful for long-term storage, complex model training, and large-scale analytics. A cooperative edge–cloud design is therefore more practical than completely replacing cloud computing.

Conclusion:

This study presented an edge intelligence-based IoT architecture designed for low-latency wireless applications. The proposed system moves time-sensitive processing closer to IoT devices while retaining cloud infrastructure for tasks that do not require immediate execution. The evaluation showed reductions in end-to-end latency, jitter, and backhaul traffic together with an improvement in packet delivery performance. The results suggest that intelligent edge processing can provide a practical foundation for industrial IoT, intelligent transportation, wireless monitoring, automation, and other real-time communication applications. Future work may extend the architecture through federated learning, mobility-aware edge selection, 5G/6G integration, and more advanced energy-efficient edge AI models.

References:

1. Satyanarayanan, M., Bahl, P., Cáceres, R., & Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4), 14–23. <https://doi.org/10.1109/MPRV.2009.82>
2. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 13–16. <https://doi.org/10.1145/2342509.2342513>

3. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 13(18), 1587–1611. <https://doi.org/10.1002/wcm.1203>
4. 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>
5. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. <https://doi.org/10.1109/MC.2017.9>
6. PremSankar, G., Di Francesco, M., & 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>
7. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762. <https://doi.org/10.1109/JPROC.2019.2918951>
8. Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469. <https://doi.org/10.1109/JIOT.2020.2984887>
9. Xu, D., Li, T., Li, Y., Su, X., Tarkoma, S., Jiang, T., Crowcroft, J., & Hui, P. (2021). Edge intelligence: Empowering intelligence to the edge of network. *Proceedings of the IEEE*, 109(11), 1778–1837. <https://doi.org/10.1109/JPROC.2021.3119950>
10. Mendez, J., Bierzynski, K., Cuéllar, M. P., & Morales, D. P. (2022). Edge intelligence: Concepts, architectures, applications, and future directions. *ACM Transactions on Embedded Computing Systems*, 21(5), Article 48, 1–41. <https://doi.org/10.1145/3486674>
11. Singh, R., & Gill, S. S. (2023). Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 3, 71–92. <https://doi.org/10.1016/j.iotcps.2023.02.004>