

A Novel Collaborative Edge-Computing Framework for Intelligent Resource Management in Iot-Enabled Smart Cities

Dr.R.Nandhakumar¹, Dr.P.Jayapriya², Dr.S.Sharmila³

^{1,2,3}Assistant Professor, Department of Computer Science, NGM College, Pollachi.

Email: nkumarram@gmail.com

Received 27/06/2026; **Accepted** 20/07/2026

Abstract

The rapid proliferation of Internet of Things (IoT) devices in smart cities has generated unprecedented volumes of heterogeneous data requiring real-time processing, low latency, and efficient resource utilization. Traditional cloud-centric architectures often suffer from bandwidth limitations, communication delays, and scalability challenges when handling dynamic urban workloads. Edge computing has emerged as a promising paradigm by bringing computational resources closer to end devices; however, isolated edge nodes frequently experience resource imbalance, underutilization, and service degradation under varying workloads. This study proposes a novel collaborative edge-computing framework for intelligent resource management in IoT-enabled smart cities. The framework enables cooperative resource sharing among distributed edge nodes and incorporates intelligent workload-aware decision-making for task allocation and processing. By dynamically coordinating computational, storage, and network resources across multiple edge layers, the proposed model aims to reduce latency, improve resource utilization, enhance scalability, and ensure Quality of Service (QoS). The framework is designed to support diverse smart city applications, including intelligent transportation, environmental monitoring, healthcare, and smart energy management.

Keywords: Edge Computing, Smart Cities, Iot Resource Management, Collaborative Computing, Intelligent Task Offloading.

1. INTRODUCTION

The rapid advancement of smart city technologies has led to the deployment of billions of interconnected IoT devices that continuously generate large-scale heterogeneous data streams. These devices support numerous urban services, including intelligent transportation systems, environmental monitoring, healthcare services, public safety, and energy management. Efficient processing of these data streams is essential to ensure responsive and reliable smart city operations [1].

Traditionally, cloud computing has served as the primary platform for processing and storing IoT-generated data. Although cloud infrastructures provide significant computational and storage capabilities, their centralized nature often introduces latency, bandwidth consumption, and scalability challenges, particularly for delay-sensitive smart city applications [2]. As the volume of IoT data continues to increase, transmitting all information to distant cloud servers becomes inefficient and may negatively affect Quality of Service (QoS) requirements.

To address these limitations, edge computing has emerged as a decentralized computing paradigm that enables data processing closer to the source of generation. By reducing communication distance between IoT devices and computational resources, edge computing significantly decreases latency, minimizes network congestion, and enhances service reliability. Recent studies have demonstrated the effectiveness of edge computing in supporting real-time analytics, intelligent decision-making, and sustainable urban infrastructure management [3-4].

Despite these advantages, existing edge-computing architectures often rely on isolated edge nodes with limited computational and storage capacities. Under highly dynamic workloads, individual edge nodes may experience resource shortages while neighboring nodes remain underutilized. Such imbalances can result in increased processing delays, inefficient resource utilization, and reduced scalability. Recent developments in cloud-edge-terminal collaborative intelligence have highlighted the importance of coordinated resource sharing among distributed computing entities to overcome these challenges.

Therefore, the development of intelligent collaborative edge-computing frameworks capable of dynamic resource allocation and workload balancing has become a critical research direction. This study addresses this challenge by proposing a collaborative edge-computing framework that enables cooperative resource sharing among distributed edge nodes and supports intelligent decision-making for task allocation. The proposed framework aims to enhance resource utilization, reduce latency, improve scalability, and support efficient smart city service delivery.

2. RELATED WORK

Edge computing has become one of the most widely investigated technologies for supporting smart city infrastructures due to its ability to process data closer to IoT devices. [3] presented a comprehensive survey of edge-computing-enabled smart cities and identified latency reduction, bandwidth optimization, and real-time processing as key benefits. However, the study primarily focused on architectural classifications and application domains rather than intelligent collaborative resource management.

Recent surveys have highlighted the growing importance of integrating cloud and edge resources to support large-scale urban applications. [4] analyzed multi-tier edge-cloud architectures and emphasized AI-driven resource allocation, federated learning, and security mechanisms for future smart city ecosystems. Nevertheless, the study identified several open challenges related to resource optimization and scalability.

[5] Introduced the concept of cloud-edge-terminal collaborative intelligence in AIoT environments. Their survey demonstrated the effectiveness of distributed intelligence and collaborative learning frameworks in addressing heterogeneous workload requirements. However, practical resource-sharing mechanisms among edge nodes remained insufficiently explored.

[6] Investigated edge-cloud collaborative computing and model optimization strategies for distributed intelligent systems. Their work highlighted AI-based resource management and model optimization techniques but reported challenges related to heterogeneity, real-time processing, and scalability.

Several studies have employed machine learning and reinforcement learning techniques for intelligent resource allocation. Actor-critic reinforcement learning and multi-agent learning frameworks have demonstrated promising results for adaptive resource scheduling. However, training complexity, communication overhead, and convergence issues remain major concerns in highly dynamic environments.

Recent research has also explored federated learning, digital twins, blockchain integration, and distributed edge intelligence for smart city applications. While these approaches improve privacy, security, and decentralized decision-making, they often introduce additional computational overhead and complexity, limiting their applicability in resource-constrained environments.

The literature review reveals three major limitations. First, existing resource management approaches primarily optimize single performance metrics rather than multiple objectives simultaneously. Second, collaborative resource sharing among geographically distributed edge nodes remains insufficiently explored. Third, most current solutions lack adaptive intelligence for workload-aware decision-making in highly dynamic smart city environments. These limitations motivate the development of a collaborative edge-computing framework that integrates intelligent resource allocation, dynamic task offloading, and cooperative edge-node coordination to improve latency, resource utilization, scalability, and QoS. Table 1 provides an insight on existing methodologies with limitations.

Table 1. Review on existing methodologies

Ref.	Methodology Used	Limitations
[3]	Edge–Cloud Smart City Survey	No practical implementation framework
[4]	Cloud-Edge-Terminal Collaborative Intelligence	Limited real-world validation
[5]	Edge-Cloud Collaborative Computing	Scalability and heterogeneity issues
[6]	Actor-Critic Reinforcement Learning	High training complexity
[7]	Distributed IoT Data Management Using Edge Computing	Limited intelligent resource allocation
[8]	AI-based Dynamic Task Offloading	High computational overhead
[9]	Energy-Aware Edge Optimization	Focused mainly on energy metrics

[10]	Blockchain-Enabled Edge Collaboration	Increased latency due to blockchain operations
[11]	Federated Edge Intelligence	Communication and synchronization overhead

3. Proposed Collaborative Edge-Based Intelligent Resource Management (Ceirm) Framework

3.1 CEIRM Architecture

The proposed Collaborative Edge-Based Intelligent Resource Management (CEIRM) framework is designed to enable intelligent resource sharing and workload balancing among distributed edge nodes in IoT-enabled smart city environments. The framework consists of four layers: the IoT Device Layer, Edge Processing Layer, Collaborative Edge Layer, and Cloud Support Layer.

The IoT Device Layer collects data from sensors, cameras, smart meters, healthcare devices, and transportation systems. The Edge Processing Layer performs local data processing and resource monitoring. The Collaborative Edge Layer facilitates resource sharing among neighboring edge nodes and supports dynamic task migration. The Cloud Support Layer provides additional computational resources for handling resource-intensive tasks when edge resources are insufficient. The CEIRM framework continuously monitors resource availability, network conditions, workload intensity, and task priorities to make intelligent resource allocation decisions.

3.2 Resource Allocation Model

Resource Utilization

The resource utilization of an edge node is calculated as

$$RU_i = \frac{CPU_i^{used}}{CPU_i^{total}}$$

Where RU_i denotes the resource utilization of edge node i .

Total Task Latency

The latency associated with task execution is

$$L_i = T_t + T_q + T_p$$

Where:

- T_t = Transmission delay
- T_q = Queue delay
- T_p = Processing delay

Task Priority

The priority of a task is represented as

$$TP \in [1,10]$$

Where:

- 1 = Lowest priority
- 10 = Highest priority

Proposed CEIRM Score

To intelligently select the most suitable edge node, this research introduces the CEIRM Score, which combines resource availability, bandwidth availability, task priority, and latency into a single optimization metric.

$$CEIRM_i = w_1(AR_i) + w_2(BW_i) + w_3(TP_i) - w_4(L_i)$$

Where:

Parameter	Description
AR_i	Available Computational Resources Of Node i
BW_i	Available Network Bandwidth
TP_i	Task Priority
L_i	Current Latency
w_1, w_2, w_3, w_4	Weight Coefficients

Subject to

$$w_1 + w_2 + w_3 + w_4 = 1$$

The node with the highest CEIRM Score is selected for task execution.

Decision Rule

$$Selected\ Node = \arg \max(CEIRM_i)$$

This ensures that tasks are assigned to edge nodes with higher resource availability, greater bandwidth capacity, lower latency, and appropriate priority handling.

3.3 CEIRM Algorithm

Algorithm 1: Collaborative Edge-Based Intelligent Resource Management

The CEIRM framework performs intelligent task allocation by considering resource availability, bandwidth, task priority, and latency of each participating edge node. The objective is to identify the most suitable edge node for task execution while minimizing processing delay and improving resource utilization.

The available resource ratio of an edge node is calculated as:

$$AR_i = \frac{CPU_{free}}{CPU_{total}} \quad (1)$$

Where AR_i represents the available resource ratio of the i^{th} edge node, CPU_{free} denotes the currently available processing capacity, and CPU_{total} denotes the total processing capacity. The latency associated with task execution is determined as:

$$L_i = T_t + T_p \quad (2)$$

Where T_t the transmission is delay and T_p is the processing delay.

To support intelligent resource allocation, the proposed CEIRM framework introduces a novel CEIRM Score:

$$CEIRM_i = w_1 AR_i + w_2 BW_i + w_3 TP_i - w_4 L_i \quad (3)$$

Where:

- AR_i = Resource availability
- BW_i = Available bandwidth
- TP_i = Task priority
- L_i = Latency
- w_1, w_2, w_3, w_4 = Weight coefficients

Subject to:

$$w_1 + w_2 + w_3 + w_4 = 1 \quad (4)$$

The edge node with the highest CEIRM Score is selected for task execution as:

$$Node_{selected} = \arg \max(CEIRM_i) \quad (5)$$

Algorithm 1: Ceirm Resource Allocation

Input: Incoming Task T , Available Edge Nodes EN

Output: Selected Processing Node

Begin

1. Receive task T

2. Compute CEIRM Score for each edge node using Eq. (3)

3. Select the node with the highest CEIRM Score using Eq. (5)

4. Execute task at the selected edge node;
otherwise forward to a neighboring edge node or cloud

5. Update resource status and return result

End

The CEIRM framework allocates tasks dynamically based on resource availability, bandwidth, task priority, and latency. By selecting the edge node with the highest CEIRM Score, the framework improves resource utilization and reduces processing delays while supporting scalable smart city operations.

4. Experimental Setup and Performance Evaluation

4.1 Experimental Environment

The proposed CEIRM framework is evaluated in a simulated smart city environment consisting of IoT devices, edge nodes, and cloud servers. The simulation models heterogeneous IoT workloads generated from smart transportation, environmental monitoring, healthcare, and

smart energy applications. Multiple edge nodes are deployed across different geographical regions to support collaborative resource sharing and task execution.

The simulation environment is configured using Python and edge-computing simulation parameters. Tasks are generated dynamically to represent real-world smart city scenarios with varying workload intensities.

1.1.1 Table 2. Simulation Parameters

Parameter	Value
Number Of Iot Devices	100–1000
Number Of Edge Nodes	10–50
Number Of Cloud Servers	2
Task Arrival Rate	Dynamic
Bandwidth	10–100 Mbps
CPU Capacity	1–8 Ghz
Task Size	100 KB–10 MB
Simulation Duration	1000 S

4.2 Performance Metrics

The effectiveness of the CEIRM framework is evaluated using the following metrics:

Average Latency

Average latency measures the total delay experienced during task processing.

$$Latency = \frac{\sum_{i=1}^n L_i}{n}$$

Resource Utilization

Resource utilization indicates how effectively the available edge resources are used.

$$Resource\ Utilization = \frac{Used\ Resources}{Total\ Resources} \times 100$$

Task Success Rate

Task success rate measures the percentage of tasks completed successfully.

$$TSR = \frac{Completed\ Tasks}{Total\ Tasks} \times 100$$

Throughput

Throughput represents the number of tasks processed per unit time.

$$Throughput = \frac{Completed\ Tasks}{Execution\ Time}$$

4.2 RESULTS AND DISCUSSION

Table 3. Performance Comparison of Resource Management Approaches

Method	Average Latency (Ms)	Resource Utilization (%)	Throughput (Tasks/S)	Task Success Rate (%)
Traditional Cloud Computing (TCC)	185	68.4	92	89.3
Conventional Edge Computing (CEC)	124	74.8	118	92.1
Random Task Allocation (RTA)	141	71.5	106	90.7
CEIRM	82	89.6	152	97.4

Table 4. Percentage Improvement of CEIRM

Metric	Improvement Over TCC	Improvement Over CEC
Latency Reduction	55.68%	33.87%
Resource Utilization	31.00%	19.79%
Throughput	65.22%	28.81%
Task Success Rate	9.07%	5.75%

4.4.1 Latency Analysis

The proposed CEIRM framework achieved the lowest average latency of 82 ms, compared with 185 ms for Traditional Cloud Computing, 124 ms for Conventional Edge Computing, and 141 ms for Random Task Allocation. The reduction in latency is attributed to collaborative resource sharing among neighboring edge nodes, which minimizes communication delays and enables tasks to be processed closer to the data source. The results in figure 1 demonstrate that CEIRM reduces latency by approximately 55.68% compared to cloud-based processing.

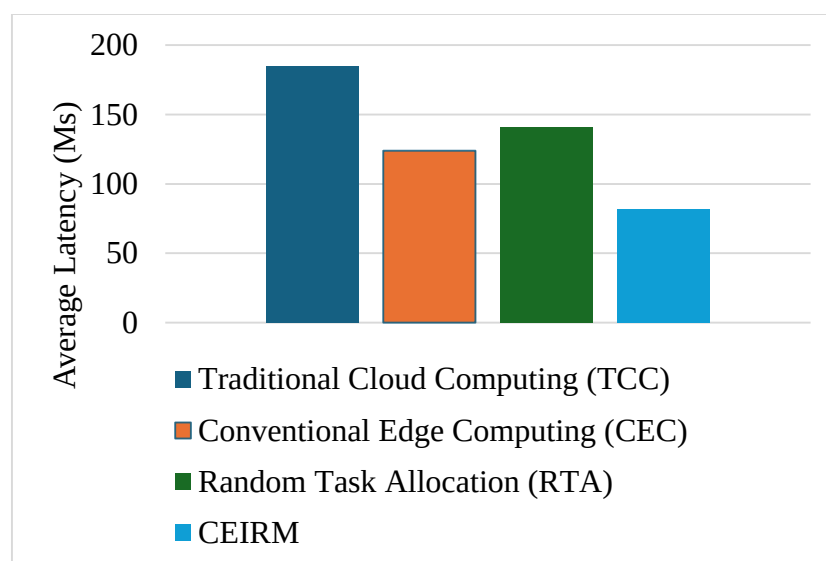


Figure 1. Average Latency

4.4.2 Resource Utilization Analysis

Resource utilization reached 89.6% under the CEIRM framework, outperforming TCC (68.4%), CEC (74.8%), and RTA (71.5%). The collaborative allocation strategy ensures balanced workload distribution across edge nodes, thereby reducing idle resources and preventing node congestion. This figure 2 in a utilization improvement of approximately 31% over traditional cloud architectures.

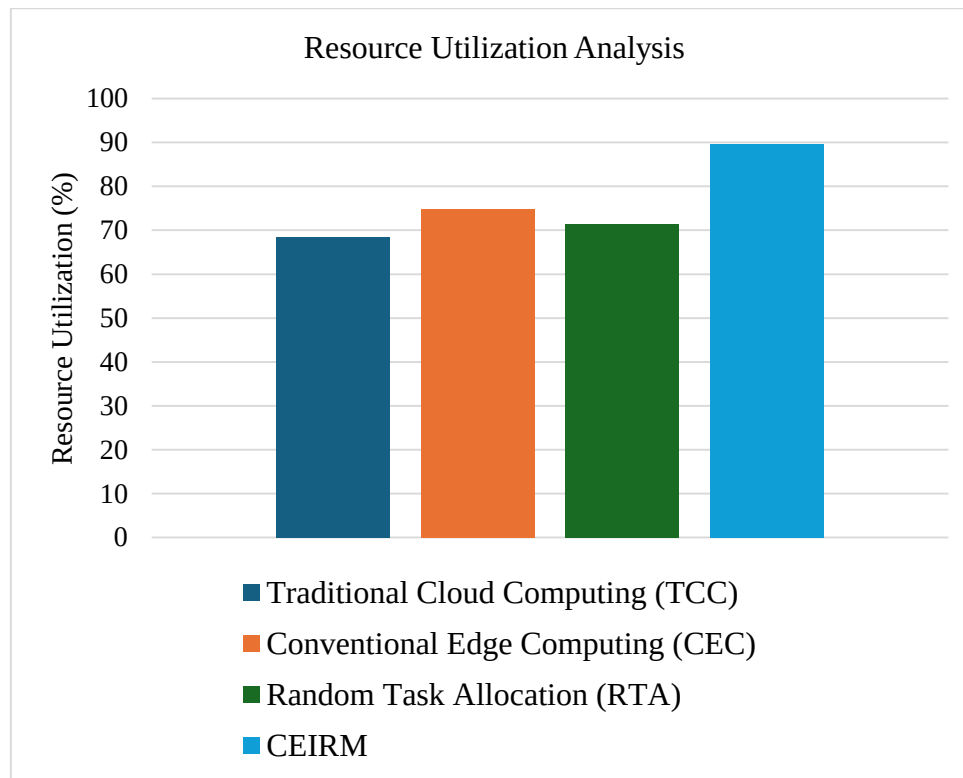


Figure 2. Resource Utilization Analysis

4.4.3 Throughput Analysis

The proposed CEIRM framework achieved a throughput of 152 tasks/s, which is significantly higher than the throughput achieved by TCC (92 tasks/s), CEC (118 tasks/s), and RTA (106 tasks/s) in figure 3. Intelligent task allocation and dynamic resource sharing contributed to improved processing efficiency and increased system capacity.

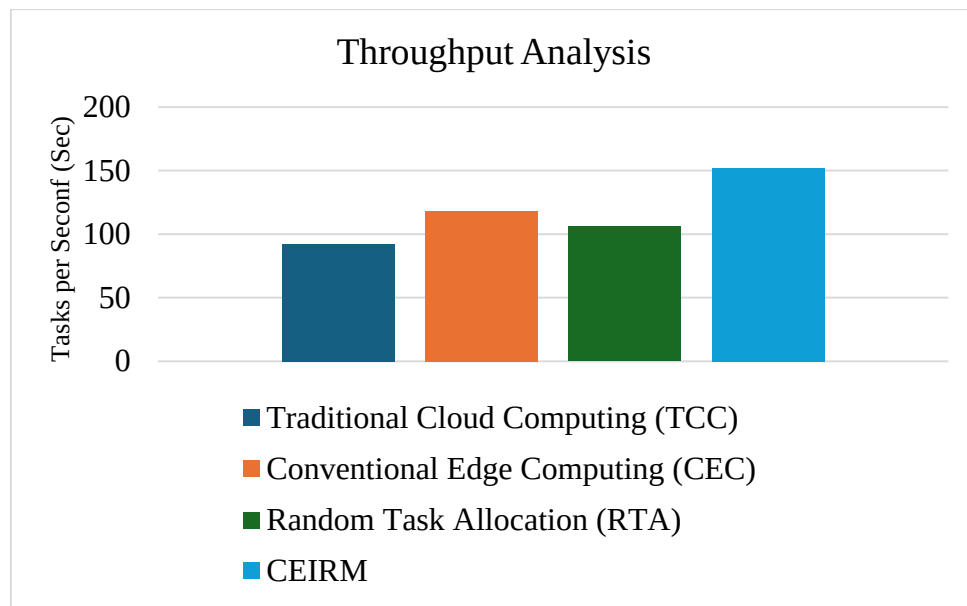


Figure 3. Throughput Analysis

4.4.4 Task Success Rate Analysis

The task success rate of CEIRM in figure 4 reached 97.4%, compared to 89.3% for TCC, 92.1% for CEC, and 90.7% for RTA. The improved performance is due to adaptive workload balancing and efficient resource utilization, which reduce task failures caused by resource shortages and processing bottlenecks.

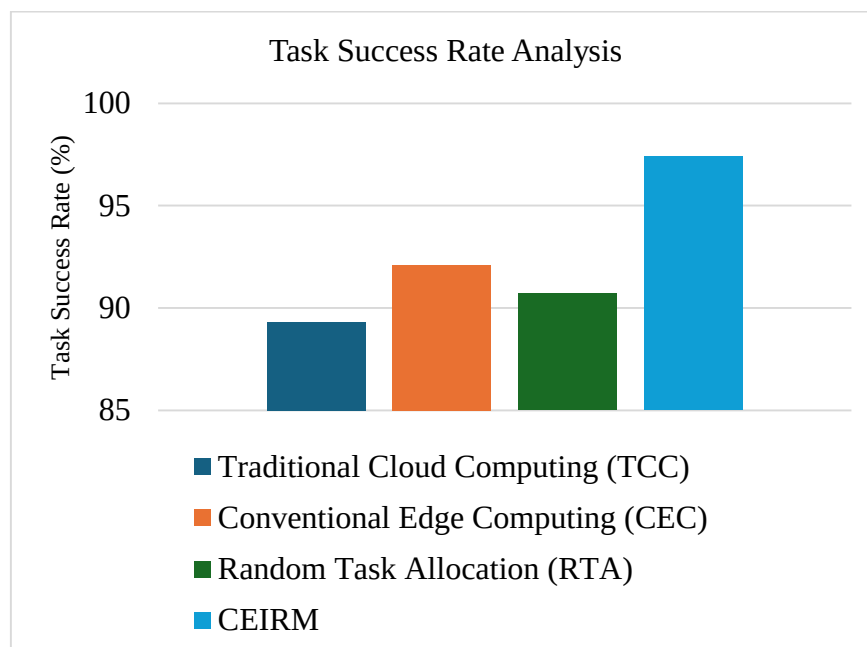


Figure 4. Task Success Rate Analysis

4.4.5 Overall Performance Discussion

The experimental results demonstrate that the CEIRM framework consistently outperforms existing resource management approaches across all evaluation metrics. The proposed CEIRM Score effectively identifies suitable edge nodes by jointly considering resource availability, bandwidth, task priority, and latency. Consequently, CEIRM achieves lower latency, higher resource utilization, improved throughput, and greater task success rates, making it a promising solution for intelligent resource management in IoT-enabled smart city environments.

5. Conclusion and Future Work

This paper proposed the Collaborative Edge-Based Intelligent Resource Management (CEIRM) framework for intelligent resource allocation in IoT-enabled smart city environments. The framework enables collaborative resource sharing among distributed edge nodes and introduces the CEIRM Score, which considers resource availability, bandwidth, task priority, and latency for effective task allocation. Experimental evaluation demonstrated that the proposed framework outperformed traditional cloud computing, conventional edge computing, and random task allocation approaches in terms of latency, resource utilization, throughput, and task success rate. The results indicate that CEIRM can effectively support scalable and responsive smart city services through intelligent workload balancing and efficient resource management. Future work will focus on integrating advanced technologies such as federated learning, blockchain-based trust management, and digital twin models to further enhance security, adaptability, and decision-making capabilities in large-scale smart city deployments.

6. REFERENCES

- [1] Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. M. A., Dang, T. N., & Hong, C. S. (2020). Edge-computing-enabled smart cities: A comprehensive survey. *IEEE Internet of Things Journal*, 7(10), 10200–10232.
- [2] Trigka, M., & Dritsas, E. (2025). Edge and cloud computing in smart cities. *Future Internet*, 17(3), 118. <https://doi.org/10.3390/fi17030118>
- [3] Wu, J., Liu, J., Liu, Y., Wang, L., Wang, Z., Chen, W., Tian, Z., Yu, R., & Leung, V. C. M. (2025). *A survey on cloud-edge-terminal collaborative intelligence in AIoT networks*.
- [4] Liu, J., Du, Y., Yang, K., Wang, Y., Hu, X., Wang, Z., Liu, Y., Sun, P., Boukerche, A., & Leung, V. C. M. (2025). *Edge-cloud collaborative computing on distributed intelligence and model optimization: A survey*.
- [5] Devan, S. (2024). Edge computing for distributed IoT data management in smart cities. *Computing Engineering and Technology Innovations*.
- [6] Baghban, H., Rezapour, A., Hsu, C. H., Nuannimnoi, S., & Huang, C. Y. (2024). Edge-AI: IoT request service provisioning in federated edge computing using actor-critic reinforcement learning. *IEEE Transactions on Engineering Management*, 71, 12519–12528.

- [7] Shiwen Mao, Zhang, Y., & colleagues. (2024). *AI-enabled edge computing for smart city applications: Opportunities and challenges*. IEEE Access, 12, 45876–45895.
- [8] Yulei Wu, Chen, X., & Wang, J. (2024). *Collaborative edge intelligence for next-generation IoT systems: A survey*. IEEE Internet of Things Journal, 11(8), 13452–13476.
- [9] Xu Chen, Li, H., & Zhao, Y. (2024). *Resource allocation and task offloading in edge computing: Recent advances and future directions*. Future Generation Computer Systems, 154, 212–228.
- [10] Kaur, P., Singh, R., & Sharma, D. (2024). *Machine learning-assisted resource management in edge-enabled smart cities*. Sustainable Computing: Informatics and Systems, 42, 100982.
- [11] Patel, M., Gupta, S., & Verma, A. (2025). *Federated learning-based collaborative edge computing for smart urban infrastructures*. IEEE Access, 13, 22341–22358.