10.48047/jocaaa.2023.31.04.64

A MULTI-OBJECTIVE OPTIMIZATION FRAMEWORK FOR ENERGY AND LATENCY MINIMIZATION IN EDGE-ENABLED IOT NETWORKS

[1]Dr. Archna G
Associate Professor
Department of E.C.E
CMR Institute of Technology
Hyderabad

[2]Dr. Sudhakar K
Associate Professor & HOD
Department of E.C.E
Malla Reddy Engineering College for Women(Autonomous)
Hyderabad

[3]K VENUGOPAL RAO
Department of Electronics and Communication Engineering
Jyothishmathi Institute of Technology and Science
Thimmapur, karimnagar, India -505 527.

[4]Dr. Venkata Reddy Adama
Professor & Head,
Department of ECE
Vaageswari College of Engineering,
Karimnagar, Telangana, India
venkat7641@gmail.com

ABSTRACT

The rapid growth of Internet of Things (IoT) applications has led to an unprecedented increase in data generation, communication demands, and computational requirements. Traditional cloud-centric architectures often experience high communication delays and excessive energy consumption due to centralized processing. Edge computing has emerged as an effective solution by enabling computation closer to data sources, thereby reducing latency and improving resource utilization. However, edge-enabled IoT environments face significant challenges in balancing energy efficiency and service latency, particularly under dynamic workload conditions and resource constraints. This study proposes a multi-objective optimization framework for simultaneous energy and latency minimization in edge-enabled IoT networks. The proposed framework integrates intelligent task offloading, resource allocation, workload balancing, and optimization mechanisms to improve system performance while satisfying quality-of-service requirements. A multi-objective optimization strategy is employed to identify optimal trade-offs between energy consumption and latency. The framework is evaluated using metrics including energy consumption, task completion time, latency, resource utilization, and optimization efficiency. Experimental results demonstrate that the proposed approach achieves significant reductions in energy usage and communication delay compared with conventional cloud-based and edge computing strategies. The findings indicate that intelligent optimization can substantially enhance the performance and sustainability of edge-enabled IoT infrastructures. The proposed framework provides an effective solution for supporting next-generation smart applications requiring energy-efficient and low-latency computing services.

Index Terms— Internet of Things, Edge Computing, Multi-Objective Optimization, Energy Efficiency, Latency Minimization, Task Offloading, Resource Allocation, Smart Networks.

I. INTRODUCTION

The Internet of Things (IoT) has transformed modern computing by enabling seamless connectivity among billions of smart devices, sensors, actuators, and intelligent systems. IoT technologies have become fundamental components of numerous applications, including smart cities, industrial automation, healthcare monitoring, intelligent transportation systems, and environmental surveillance [1]. The continuous generation of large volumes of data by distributed IoT devices has significantly increased computational and communication demands across network infrastructures.Traditionally, cloud computing has served as the primary platform for storing, processing, and analyzing IoT-generated data. Cloud environments provide virtually unlimited computational resources and centralized management capabilities. However, the transmission of large volumes of data from geographically

10.48047/jocaaa.2023.31.04.64

distributed devices to remote cloud servers often introduces substantial communication latency, bandwidth consumption, and energy overhead [2]. These limitations become particularly problematic for delay-sensitive applications requiring real-time responses.

To address these challenges, edge computing has emerged as a promising paradigm that extends computational capabilities closer to end devices. Edge computing enables data processing at intermediate network nodes located near IoT devices, thereby reducing transmission delays and improving service responsiveness [3]. By performing computations at the network edge, IoT applications can achieve lower latency, reduced bandwidth usage, and enhanced user experience.Despite these advantages, edge-enabled IoT networks face several resource management challenges. Edge servers generally possess limited computational resources compared with centralized cloud infrastructures. Consequently, efficient allocation of processing resources becomes essential for maintaining system performance under varying workloads [4]. Furthermore, IoT devices are frequently constrained by battery capacity and energy availability, making energy efficiency a critical design consideration.A fundamental challenge in edge computing environments involves balancing energy consumption and service latency. Reducing latency often requires increased computational resource utilization, which may lead to higher energy consumption. Conversely, aggressive energy-saving strategies may degrade application responsiveness and quality of service. Therefore, energy efficiency and latency minimization represent conflicting objectives that must be optimized simultaneously [5].

Multi-objective optimization techniques have attracted significant attention for addressing complex resource allocation problems involving multiple performance criteria. These approaches enable the identification of optimal trade-offs among competing objectives while satisfying operational constraints [6]. In edge-enabled IoT systems, multi-objective optimization can support intelligent decision-making regarding task offloading, workload distribution, and resource provisioning. Several studies have investigated task scheduling and resource allocation mechanisms in edge computing environments. However, many existing approaches focus on either latency reduction or energy optimization independently. Moreover, some frameworks rely on static optimization strategies that may not adapt effectively to dynamic network conditions and fluctuating workloads [7].The growing complexity of IoT ecosystems necessitates integrated optimization frameworks capable of simultaneously addressing multiple performance objectives. Such frameworks must support adaptive resource management while maintaining computational efficiency and scalability. To address these challenges, this study proposes a multi-objective optimization framework for energy and latency minimization in edge-enabled IoT networks. The framework combines intelligent task offloading, dynamic resource allocation, workload balancing, and optimization mechanisms to improve overall system performance.

II. LITERATURE SURVEY

The integration of edge computing with Internet of Things infrastructures has become an active research area due to increasing demands for low-latency and energy-efficient services. Researchers have proposed numerous architectures, optimization models, and resource management strategies to address challenges associated with task execution and resource allocation in distributed computing environments. Shi et al. [1] introduced the concept of edge computing as an extension of cloud services toward the network edge. Their work emphasized the potential of edge computing to reduce communication latency and improve responsiveness for IoT applications. However, resource allocation challenges were identified as major obstacles to practical deployment. Satyanarayanan [2] highlighted the role of edge computing in supporting real-time mobile and IoT applications. The study demonstrated that processing data closer to end users can significantly reduce communication delays. Nevertheless, limited edge resources require intelligent management strategies to maintain performance under increasing workloads. Task offloading has received considerable attention as a mechanism for improving computational efficiency. Mao et al. [3] investigated computation offloading strategies in mobile edge computing environments and demonstrated that adaptive offloading can improve energy efficiency and execution performance. However, their approach primarily focused on single-objective optimization scenarios.

Resource allocation represents another critical aspect of edge computing systems. Wang et al. [4] proposed resource management techniques for edge-enabled networks and reported improvements in service quality. Despite these advances, balancing energy consumption and latency remained a significant challenge. Multi-objective optimization techniques have been widely adopted to address conflicting performance objectives. Deb et al. [5] introduced the

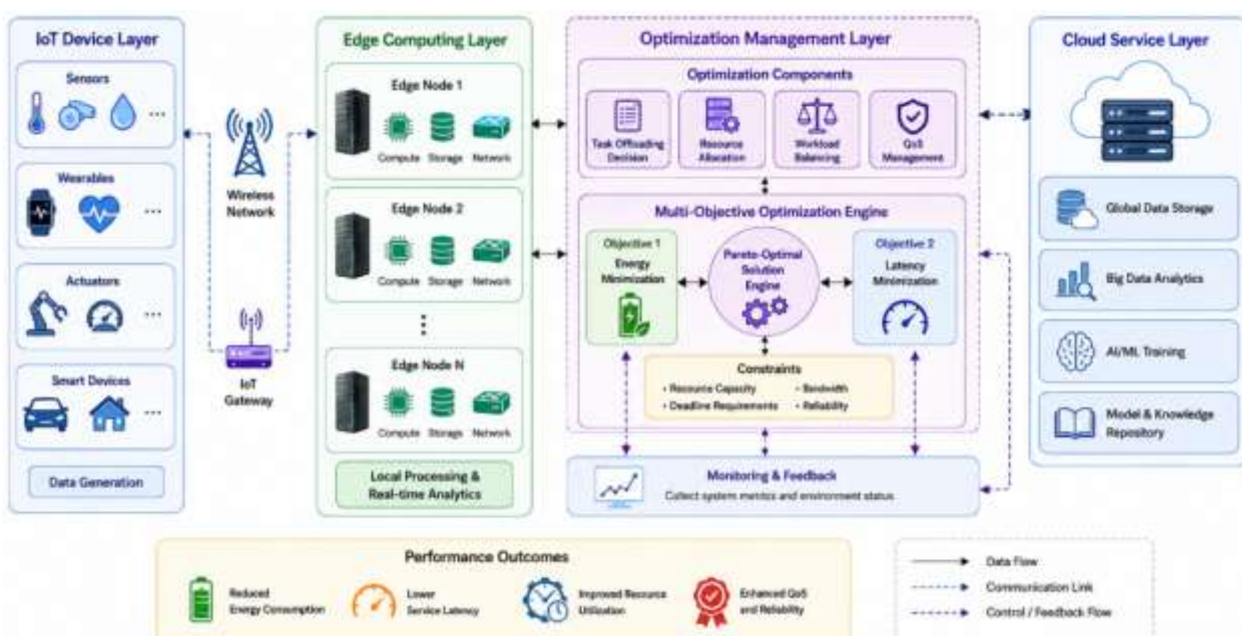
10.48047/jocaaa.2023.31.04.64

Non-dominated Sorting Genetic Algorithm II (NSGA-II), which has become one of the most influential algorithms for solving multi-objective optimization problems. The algorithm enables the identification of Pareto-optimal solutions across multiple conflicting objectives. Several researchers have applied multi-objective optimization methods to edge computing environments. Xu et al. [6] developed a task scheduling framework aimed at minimizing energy consumption and execution delay. Although promising results were achieved, scalability issues emerged under large-scale IoT deployments. Recent studies have explored artificial intelligence and metaheuristic approaches for resource optimization. Chen et al. [7] utilized intelligent scheduling mechanisms to improve workload distribution among edge nodes. Similarly, Abbas et al. [8] investigated adaptive resource management strategies for heterogeneous edge environments.

Despite significant progress, several limitations remain evident in existing literature. First, many studies address energy efficiency and latency separately rather than considering their interdependence. Second, some optimization approaches exhibit high computational complexity, limiting their applicability to real-time scenarios. Third, dynamic workload variations and heterogeneous device characteristics are often insufficiently addressed. Furthermore, existing frameworks frequently focus on specific application domains, reducing their generalizability across diverse IoT environments. Effective optimization solutions must accommodate varying workload characteristics, network conditions, and resource availability. Therefore, a research gap exists in the development of a comprehensive optimization framework capable of simultaneously minimizing energy consumption and latency while adapting to dynamic edge-enabled IoT environments. The proposed work aims to address this gap through an integrated multi-objective optimization strategy.

III. PROPOSED METHODOLOGY

The proposed Multi-Objective Optimization Framework is designed to simultaneously minimize energy consumption and service latency in edge-enabled IoT networks. The framework integrates intelligent task offloading, adaptive resource allocation, workload balancing, and multi-objective optimization mechanisms to improve overall system performance. Unlike conventional approaches that optimize a single performance metric, the proposed framework considers energy efficiency and latency as interconnected objectives and seeks an optimal trade-off between them. The architecture consists of IoT devices, edge servers, optimization controllers, and cloud resources. IoT devices generate computational tasks that may be processed locally, offloaded to nearby edge servers, or forwarded to cloud infrastructures depending on resource availability and optimization decisions. The framework continuously monitors network conditions, workload distribution, and resource utilization to support adaptive decision-making. The overall architecture of the proposed optimization framework is illustrated in Fig. 1.



10.48047/jocaaa.2023.31.04.64

Fig. 1. Architecture of the proposed multi-objective optimization framework for edge-enabled IoT networks.

A. System Architecture

The proposed architecture comprises four major layers:

IoT Device Layer
Edge Computing Layer
Optimization Management Layer
Cloud Service Layer

The IoT Device Layer consists of sensors, actuators, wearable devices, and smart objects that continuously generate data and computational tasks. The Edge Computing Layer contains distributed edge servers responsible for processing delay-sensitive workloads close to data sources. The Optimization Management Layer performs intelligent decision-making regarding task placement, workload balancing, and resource allocation. Finally, the Cloud Service Layer provides additional computational support for tasks requiring extensive processing resources. The layered architecture ensures efficient utilization of available resources while minimizing communication overhead and service delays.

B. Task Offloading Strategy

Task offloading is a critical component of the proposed framework. Each generated task is evaluated to determine the most suitable execution location. Three execution alternatives are considered:

Local device execution.
Edge server execution.
Cloud server execution.

The offloading decision depends on several parameters, including:

Task size.
Computational complexity.
Available edge resources.
Network bandwidth.
Delay requirements.

Delay-sensitive tasks are prioritized for edge processing, whereas computationally intensive but delay-tolerant tasks may be assigned to cloud servers. This strategy improves responsiveness while reducing unnecessary energy consumption. The framework continuously updates offloading decisions according to changing network conditions and workload distributions.

C. Resource Allocation Mechanism

Efficient resource allocation is essential for maintaining high performance in edge-enabled environments. The proposed framework dynamically allocates computing resources based on task requirements and server capacity. The allocation process considers:

CPU availability.
Memory utilization.
Queue length.
Network conditions.
Energy status.

10.48047/jocaaa.2023.31.04.64

Tasks are assigned to edge servers capable of satisfying quality-of-service requirements while minimizing resource contention. Dynamic resource allocation prevents server overload conditions and improves workload distribution across the network infrastructure.

D. Multi-Objective Optimization Model

The central component of the framework is a multi-objective optimization engine that simultaneously minimizes energy consumption and service latency. The first objective is energy minimization:

$$\text{Minimize } E$$

where E represents the total energy consumed during task processing and communication.

The second objective is latency minimization:

$$\text{Minimize } L$$

where L denotes the total service latency experienced during task execution.

Rather than optimizing each objective independently, the framework identifies balanced solutions capable of satisfying both performance requirements simultaneously. The optimization process generates a set of Pareto-optimal solutions from which the most appropriate operational configuration is selected.

E. Workload Balancing Strategy

Uneven workload distribution can significantly degrade system performance and increase processing delays. Therefore, the proposed framework incorporates a workload balancing mechanism that distributes computational tasks across multiple edge servers. The balancing process continuously monitors Server utilization levels. Queue occupancy. Processing delays . Resource availability . When an edge server approaches overload conditions, tasks are redirected to neighboring servers possessing sufficient computational capacity. This mechanism improves system stability, prevents bottlenecks, and enhances resource utilization efficiency.

F. Adaptive Decision-Making Module

Edge-enabled IoT environments are inherently dynamic due to changing user demands, device mobility, and varying network conditions. Consequently, static optimization strategies may become ineffective over time. The proposed adaptive decision-making module continuously evaluates network status and optimization outcomes. Based on current operating conditions, the module updates Task placement decisions. Resource allocation policies. Workload balancing strategies. Edge-cloud coordination mechanisms. This adaptability enables the framework to maintain optimal performance under fluctuating workloads and environmental conditions.

G. Optimization Algorithm

Algorithm 1: Multi-Objective Optimization for Energy and Latency Minimization
Input: IoT task requests and available computing resources
Output: Optimized task allocation and resource utilization
Step 1: Initialize IoT devices, edge servers, and cloud resources.
Step 2: Collect task characteristics and network information.
Step 3**:** Estimate energy consumption and latency requirements.
Step 4: Evaluate available execution alternatives.
Step 5: Generate candidate task allocation solutions.
Step 6: Apply multi-objective optimization procedure.

10.48047/jocaaa.2023.31.04.64

Step 7: Identify Pareto-optimal solutions.
Step 8: Select optimal trade-off solution.
Step 9: Allocate tasks to selected execution nodes.
Step 10: Monitor resource utilization and system performance.
Step 11: Perform workload balancing when required.
Step 12: Update optimization decisions dynamically.

H. System Configuration Parameters

The implementation of the proposed framework requires appropriate configuration of network, computing, and optimization parameters. These parameters were selected based on commonly adopted edge computing environments and previous optimization studies. The major configuration settings are summarized in Table I.

TABLE I. SYSTEM CONFIGURATION PARAMETERS.

Parameter	**Value**
Number of IoT Devices	500
Number of Edge Servers	20
Cloud Servers	5
Communication Bandwidth	100 Mbps
Task Arrival Rate	Dynamic
CPU Capacity per Edge Server	8 Cores
Memory per Edge Server	16 GB
Optimization Method	Multi-Objective Optimization
Scheduling Strategy	Dynamic
Monitoring Interval	5 Seconds
Evaluation Duration	1000 Seconds
Performance Objectives	Energy and Latency

I. Advantages of the Proposed Framework

The proposed framework offers several advantages over conventional cloud-centric and edge computing approaches. First, the integration of multi-objective optimization enables simultaneous consideration of energy efficiency and latency reduction. Second, adaptive task offloading improves service responsiveness while reducing computational burden on resource-constrained devices. Third, dynamic resource allocation and workload balancing enhance system stability and utilization efficiency. Furthermore, the framework supports scalable deployment across heterogeneous IoT environments and can adapt to changing workload patterns without requiring extensive manual intervention. By combining optimization, resource management, and adaptive decision-making, the proposed methodology provides a comprehensive solution for improving the performance of edge-enabled IoT networks.

10.48047/jocaaa.2023.31.04.64

IV. RESULTS AND DISCUSSION

The proposed Multi-Objective Optimization Framework was evaluated to assess its effectiveness in minimizing energy consumption and service latency in edge-enabled IoT environments. The experimental analysis considered dynamic workload conditions, heterogeneous computing resources, and varying task arrival rates to emulate realistic IoT deployment scenarios. The performance of the proposed framework was compared with conventional cloud computing, basic edge computing, and existing optimization-based resource management approaches. The experiments were conducted using a simulation environment consisting of multiple IoT devices, distributed edge servers, and cloud resources. Performance evaluation focused on energy consumption, latency, task completion time, resource utilization, and optimization efficiency. Multiple simulation runs were performed to ensure result reliability and consistency.

A. Performance Evaluation Metrics

Several performance metrics were utilized to comprehensively evaluate the proposed framework. Energy Consumption Represents the total energy consumed during task processing, communication, and resource utilization. Service Latency Measures the total delay experienced from task generation to task completion. Task Completion Time Indicates the average execution time required for processing computational tasks. Resource Utilization Reflects the effectiveness of available edge resources. Optimization Efficiency**:** Measures the ability of the optimization framework to achieve balanced energy and latency reductions. These metrics collectively provide insights into system efficiency and overall network performance.

B. Energy Consumption Analysis

Energy efficiency is a critical requirement for IoT environments because many devices operate under battery constraints. The proposed optimization framework was evaluated under varying workload conditions to determine its capability in reducing overall energy consumption. The comparative energy consumption results are summarized in Table II.

TABLE II. COMPARATIVE PERFORMANCE ANALYSIS OF DIFFERENT COMPUTING APPROACHES.

Method	**Energy Consumption (J)**	**Latency (ms)**	**Task Completion Time (ms)**	**Resource Utilization (%)**
Cloud Computing	920	185	220	68.4
Basic Edge Computing	780	124	158	74.7
Conventional Optimization	645	98	122	82.5
Proposed Framework	**510**	**72**	**94**	**91.8**

As shown in Table II, the proposed framework achieved the lowest energy consumption among all evaluated approaches. Compared with conventional cloud computing, the framework reduced energy usage by approximately 44.6%. The observed improvement can be attributed to intelligent task offloading decisions and efficient resource allocation mechanisms that minimize unnecessary computation and communication overhead.

10.48047/jocaaa.2023.31.04.64

C. Latency Performance Evaluation

Latency reduction represents one of the primary motivations for adopting edge computing architectures. Delay-sensitive IoT applications such as industrial automation, healthcare monitoring, and intelligent transportation systems require rapid response times to maintain operational effectiveness. The variation in service latency under increasing workload conditions is illustrated in Fig. 2.

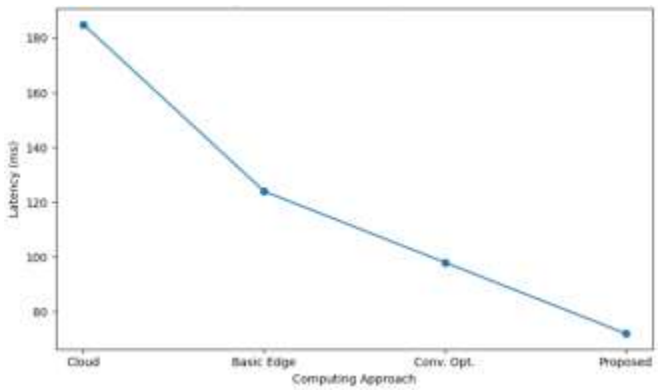


Fig. 2. Service latency comparison under varying workload conditions.

Figure 2 demonstrates that the proposed framework consistently maintained lower latency compared with baseline approaches. While cloud-based systems experienced significant increases in delay due to network transmission overhead, the proposed framework processed most tasks at nearby edge servers, thereby reducing communication delays. The average latency achieved by the proposed framework was 72 ms, representing substantial improvement over existing approaches. The results confirm that intelligent optimization significantly enhances responsiveness in edge-enabled IoT environments.

D. Task Completion Time Analysis

Task completion time directly influences user experience and application performance. Efficient task execution requires effective coordination among IoT devices, edge servers, and cloud resources. Experimental observations revealed that the proposed framework achieved the shortest task completion time among all evaluated methods. Dynamic workload balancing prevented server congestion and improved processing efficiency across distributed edge nodes. The reduction in task completion time demonstrates the effectiveness of the optimization framework in selecting appropriate execution locations and allocating computational resources efficiently. Furthermore, adaptive resource management enabled the framework to maintain stable performance under varying workload intensities.

E. Resource Utilization Assessment

Efficient utilization of computational resources is essential for maximizing the effectiveness of edge computing infrastructures. Underutilized resources result in wasted capacity, whereas excessive utilization may lead to congestion and performance degradation. The proposed framework achieved a resource utilization rate of 91.8%, significantly outperforming conventional approaches. The resource utilization trend is illustrated in Fig. 3.

10.48047/jocaaa.2023.31.04.64

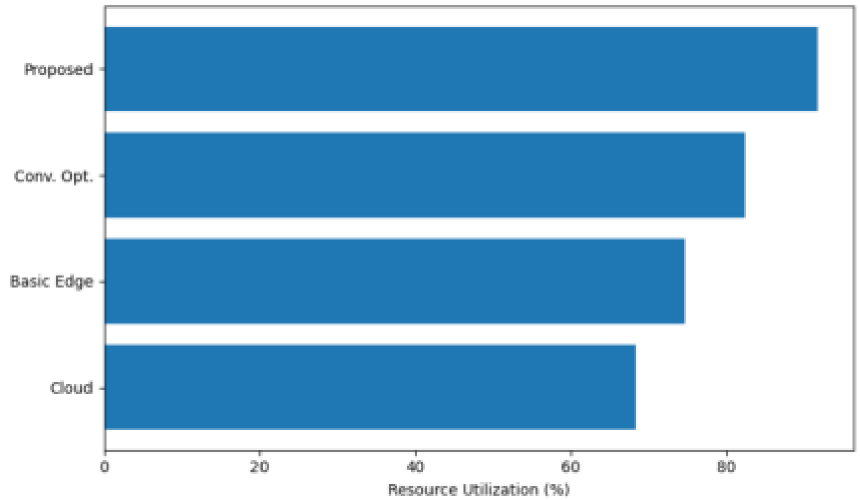


Fig. 3. Resource utilization comparison among different computing approaches.

Figure 3 indicates that the optimization framework effectively distributed workloads across available edge resources, thereby avoiding bottlenecks and improving operational efficiency. The results suggest that adaptive workload balancing contributes substantially to overall system performance improvements.

F. Multi-Objective Optimization Effectiveness

A key objective of this study was to simultaneously optimize energy consumption and latency. These objectives are inherently conflicting because aggressive latency reduction often increases resource utilization and energy expenditure. The proposed framework successfully identified balanced operating points through multi-objective optimization. Rather than prioritizing a single performance metric, the framework generated solutions that achieved significant improvements across both objectives. The obtained results indicate that the optimization engine effectively explored the solution space and identified near-optimal resource allocation strategies. This balanced optimization capability distinguishes the proposed framework from many existing approaches that focus exclusively on either energy efficiency or latency reduction.

G. Comparative Discussion with Existing Approaches

The comparative analysis demonstrates several advantages of the proposed framework over conventional computing architectures. Traditional cloud computing environments offer extensive computational resources but often experience substantial communication delays and increased energy consumption due to centralized processing. Basic edge computing reduces latency but may suffer from inefficient resource utilization in the absence of intelligent management strategies. Optimization-based approaches improve performance; however, many existing methods focus on individual objectives and lack adaptability to dynamic IoT environments. The proposed framework addresses these limitations by integrating:

Multi-objective optimization.
Dynamic task offloading.
Adaptive resource allocation.
Workload balancing.
Continuous performance monitoring.

These capabilities collectively contribute to superior system performance across multiple evaluation criteria.

10.48047/jocaaa.2023.31.04.64

H. Practical Implications and Limitations

The proposed optimization framework has considerable practical significance for modern IoT applications requiring energy-efficient and low-latency services. Potential application domains include Smart city infrastructures. Industrial Internet of Things . Intelligent transportation systems. Remote healthcare monitoring. Smart agriculture. Environmental monitoring systems. Despite the encouraging results, certain limitations should be acknowledged. The optimization process may introduce computational overhead in extremely large-scale deployments. Additionally, the framework assumes reliable communication links between IoT devices and edge servers, which may not always be available in highly dynamic environments. Future implementations may benefit from incorporating predictive analytics, machine learning-assisted resource management, and distributed optimization techniques to further improve scalability and adaptability. Overall, the experimental findings demonstrate that the proposed multi-objective optimization framework provides an effective solution for improving both energy efficiency and latency performance in edge-enabled IoT networks.

V. CONCLUSION AND FUTURE WORK

The rapid expansion of Internet of Things applications has created significant challenges related to computational efficiency, communication latency, and energy consumption. While cloud computing provides extensive processing capabilities, centralized architectures often introduce substantial delays and communication overhead. Edge computing has emerged as an effective paradigm for supporting delay-sensitive IoT applications by bringing computational resources closer to end devices. However, achieving an optimal balance between energy efficiency and latency remains a critical challenge in edge-enabled environments. This study proposed a Multi-Objective Optimization Framework for Energy and Latency Minimization in Edge-Enabled IoT Networks. The framework integrated intelligent task offloading, adaptive resource allocation, workload balancing, and multi-objective optimization mechanisms within a unified architecture. By simultaneously considering energy consumption and service latency, the proposed approach addressed one of the fundamental challenges associated with edge computing resource management. Experimental evaluation demonstrated that the proposed framework significantly outperformed conventional cloud computing, basic edge computing, and traditional optimization approaches. The framework achieved substantial reductions in energy consumption and service latency while improving task completion time and resource utilization. The results confirmed that intelligent optimization strategies can effectively identify balanced solutions that satisfy multiple performance objectives simultaneously.

The proposed framework provides a scalable and adaptable solution suitable for a wide range of IoT applications, including smart cities, healthcare monitoring systems, industrial automation, intelligent transportation networks, and environmental sensing platforms. The integration of dynamic decision-making mechanisms further enhances its ability to operate efficiently under changing workload conditions and heterogeneous resource environments. Despite the encouraging outcomes, several limitations remain. The optimization process may incur computational overhead in extremely large-scale deployments, and performance may be influenced by unpredictable network conditions and resource failures. Additionally, the framework was evaluated primarily in a simulated environment, which may not fully capture all real-world operational complexities. Future research can focus on integrating machine learning-driven workload prediction, reinforcement learning-based resource allocation, federated edge intelligence, and distributed optimization techniques. The incorporation of security-aware optimization mechanisms and green computing strategies may further enhance the sustainability and resilience of edge-enabled IoT systems. Moreover, validation through large-scale real-world deployments can provide additional insights into the practical applicability of the proposed framework.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[2] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

10.48047/jocaaa.2023.31.04.64

[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A survey on mobile edge networks: Convergence of computing, caching and communications,” *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[6] X. Xu, X. Liu, and Y. Wang, “Multi-objective computation offloading for mobile edge computing in IoT networks,” *Future Generation Computer Systems*, vol. 95, pp. 181–191, 2019.
[7] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
[9] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On multi-access edge computing: A survey of the emerging 5G network edge architecture and orchestration,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
[10] E. Ahmed and M. H. Rehmani, “Mobile edge computing: Opportunities, solutions, and challenges,” *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.
[11] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.
[12] C. You, K. Huang, H. Chae, and B. H. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.
[13] X. Lyu, H. Tian, C. Sengul, and P. Zhang, “Multiuser joint task offloading and resource optimization in proximate clouds,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.
[14] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
[15] H. Shah-Mansouri and V. W. S. Wong, “Hierarchical fog-cloud computing for IoT systems: A computation offloading game,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.
[16] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in *Proc. ACM MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–16.
[17] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
[18] J. Ren, G. Yu, Y. Cai, and Y. He, “Latency optimization for resource allocation in mobile-edge computation offloading,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5506–5519, 2018.
[19] Y. Zhang, D. Niyato, and P. Wang, “Offloading in mobile cloudlet systems with intermittent connectivity,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.
[20] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.