Global Research Journal in Engineering, Science, Technology and Management (GRJESTM)
Volume 01, Issue 02 (Mar-Apr) 2025, pp. 65-69
GRJESTM :: An odaswa™ Journal
ISSN : 3107-3611 (Online)

RESEARCH ARTICLE

# Optimizing Edge Computing for IoT Applications: A Lightweight Framework for Real-Time Data Processing

**Sourav Patra[1]*, Ipsita Das[2], and Manas Ranjan Sahoo[3]**

**Abstract.** The increasing deployment of Internet of Things (IoT) devices has led to a surge in real-time data generation, challenging traditional cloud-centric processing models due to latency, bandwidth, and privacy constraints. Edge computing addresses these limitations by bringing computation closer to data sources. However, the constrained resources at edge nodes demand lightweight and efficient processing frameworks. This paper proposes a modular, lightweight edge computing framework optimized for real-time IoT data processing. The framework comprises data acquisition, an adaptive scheduling engine, a real-time processing core, and a communication interface designed for minimal resource consumption. Performance evaluations conducted in a simulated smart agriculture environment demonstrate significant improvements in latency, throughput, and energy efficiency over cloud-only architectures. The results indicate that the proposed approach is scalable, adaptable, and suitable for latency-critical IoT applications.

**Keywords:** Edge computing, IoT, real-time data processing, lightweight framework, adaptive scheduling, latency reduction, smart agriculture.

1* Department of Computer Science, Gandhi Institute for Technological Advancement, Bhubaneswar.
2. Department of Electronics and Communication, C.V. Raman Global University, Bhubaneswar
3. Department of Information Technology, Silicon Institute of Technology, Bhubaneswar


Published By **odaswa**. Published at Chennai, Tamil Nadu, India

## 1. Introduction

The rapid proliferation of Internet of Things (IoT) devices across industries—ranging from healthcare and agriculture to manufacturing and smart cities—has led to exponential data generation at the network's edge. Traditional cloud-based models are increasingly inadequate for real-time processing needs due to bandwidth limitations, high latency, and privacy concerns [1][2]. Edge computing emerges as a compelling paradigm that brings computation and data storage closer to the data source, thereby enhancing responsiveness and efficiency [3]. However, the limited computational and storage capacities of edge nodes pose challenges to processing complex data streams efficiently [4].

As IoT applications become more latency-sensitive, such as in autonomous vehicles, remote surgery, and industrial control systems, the demand for instant analytics has grown. In these environments, delays of even a few milliseconds can lead to system failures or safety hazards [5][6]. Relying solely on cloud infrastructure not only introduces unacceptable latency but also increases dependence on network availability [7]. This necessitates a new computing architecture that enables devices to act on data at the point of collection, making real-time edge processing not just advantageous but critical [8].

Moreover, many IoT deployments operate in resource-constrained environments with intermittent connectivity, such as rural farms or remote monitoring stations [9]. For such contexts, it is vital to develop lightweight, scalable, and energy-efficient frameworks that can operate autonomously at the edge with minimal cloud dependency [10]. This paper addresses this need by presenting a novel edge computing framework that combines real-time responsiveness with low resource overhead, suitable for diverse and demanding IoT scenarios.

The primary contributions of this study include the design and implementation of a modular lightweight edge framework, empirical evaluation in a real-world-inspired setup, and comparative analysis against cloud-centric architectures. The results highlight how even resource-limited edge devices can efficiently process streaming data when equipped with optimized software modules and scheduling intelligence [11][12].

## 2. Related Works

Edge computing has evolved as a response to the limitations of centralized cloud architectures in handling IoT data. Several studies have explored distributed edge-to-cloud architectures that aim to reduce latency and increase fault tolerance. Chiang and Zhang [13] introduced fog computing as a complementary layer between cloud and edge, highlighting improvements in network efficiency. In another study, Varghese et al. [14] discussed the benefits of edge-first strategies in managing real-time data flows across smart environments.

To enable real-time inference on edge devices, lightweight frameworks such as NanoEdge AI and Edge Impulse have emerged [15][16]. These frameworks focus on energy-efficient deployment of models in constrained environments but often trade off between model accuracy and inference time. Jeong et al. [17] investigated dynamic resource allocation techniques that address these trade-offs in edge nodes. Meanwhile, Zhou et al. [18] proposed load balancing strategies that adjust based on device heterogeneity.

In terms of intelligent scheduling, Alqahtani et al. [19] developed context-driven orchestration algorithms for edge-assisted IoT, significantly reducing packet drop rates. Complementary work by Abbas et al. [20] provided a taxonomy of task offloading strategies and emphasized the role of AI in optimizing resource allocation across edge ecosystems.

Despite these advancements, few studies offer a unified framework that integrates real-time stream analytics, adaptive scheduling, and cross-layer communication under low resource budgets. For instance, Zhang et al. [21] focused solely on offloading, while Liang et al. [22] emphasized federated learning but did not address scheduling. This paper aims to bridge these gaps by proposing a lightweight yet comprehensive architecture tailored for edge-driven IoT applications.

## 3. Methods

### 3.1. Framework Architecture

To address the computational limitations of edge devices, we developed LEAF (Lightweight Edge Analytics Framework), a modular and adaptive edge computing solution. LEAF is designed to operate seamlessly on low-power edge devices, such as Raspberry Pi 4 Model B units, characterizing typical IoT deployment environments. The framework is comprised of four core functional modules deployed on edge nodes to handle the sensing, processing, scheduling, and communication of data.

### 3.2. Data Preprocessing and Analytics

LEAF begins with a Stream Preprocessing Module, tasked with filtering and cleaning raw sensor data, mitigating sensor noise, and performing lightweight feature extraction. By reducing data volume at this stage, subsequent processing becomes more efficient. The filtered data is then passed to the Analytics Engine, which is optimized for running lightweight machine learning models, primarily built using TensorFlow Lite. These models focus on real-time anomaly detection, event prediction, and localized decision-making.

### 3.3. Adaptive Scheduling and Offloading

Given the limited computational and energy resources at the edge, LEAF employs a Scheduler and Offloader module that operates based on real-time system metrics (e.g., CPU load, battery level, and network latency). This module employs dynamic rule-based algorithms to selectively process tasks locally or offload them to the fog or cloud, thus balancing latency and energy efficiency. Critical real-time tasks are always prioritized for local execution to minimize delay.

### 3.4. Communication Protocols

The Data Communication Layer ensures lightweight and reliable data exchange between edge, fog, and cloud platforms. LEAF leverages the MQTT protocol for secure, low-bandwidth communication; its lightweight nature suits IoT's resource-constrained environment and provides better reliability under intermittent network conditions.

### 3.5. Experimental Setup

To validate LEAF's performance, a testbed comprising three Raspberry Pi 4 edge devices was set up. Data streams emulating real-world environmental conditions (temperature, humidity, and motion) were generated. The performance of LEAF was compared across three architectures: a traditional cloud-centric approach, a hybrid edge-fog-cloud approach, and the proposed edge-centric LEAF system. Each architecture was evaluated based on latency, energy consumption, throughput, CPU utilization, and scalability under varying workloads.

## 4. Results and Discussion

### 4.1. Performance Metrics

The lightweight framework achieved a 58% reduction in average latency, improving response time from 1.2 seconds (cloud-only) to 0.5 seconds. Throughput increased by 35%, from 80 messages/second to 108, validating the efficiency of the edge processing engine.

### 4.2. System Utilization

Average CPU usage remained below 70%, and RAM usage did not exceed 60%, confirming the framework's suitability for resource-constrained edge devices. Energy usage per transaction was reduced by 24%, as fewer packets were transmitted to the cloud.

### 4.3. Scalability and Fault Tolerance

The Adaptive Task Scheduler dynamically balanced load across edge and fog layers as the number of devices increased. Even at 25 concurrent sensor inputs, latency remained under 1 second, indicating excellent scalability. The framework gracefully handled network interruptions by buffering data locally and syncing once connectivity was restored.

### 4.4. Comparative Evaluation

| Metric | Cloud-Only | Edge Framework |
|---|---|---|
| Latency (s) | 1.2 | 0.5 |
| Throughput (msg/s) | 80 | 108 |
| CPU Usage (%) | 90 | 68 |
| Energy Use (mWh) | 40 | 30.4 |

These results support the conclusion that decentralized, lightweight edge solutions are viable for latency-critical and bandwidth-limited environments. The modularity of the system also allows for customization per application domain.

## 5. Conclusion

This paper presented a lightweight, modular edge computing framework optimized for real-time data processing in IoT ecosystems. By integrating efficient data acquisition, adaptive scheduling, and lightweight analytics modules, the framework significantly reduces latency and resource consumption, even in resource-constrained environments. Its performance was validated in a smart agriculture scenario, where it outperformed traditional cloud-based models across key metrics.

The proposed framework demonstrates that real-time intelligence can be effectively delivered at the edge without relying on high-end hardware or constant cloud access. Future work will explore integrating federated learning for collaborative model training across multiple edge nodes and expanding the system's application to urban mobility, smart homes, and disaster monitoring. Additionally, automated optimization of scheduling policies using reinforcement learning may further enhance the adaptability of the system.

## References


[1] A. Botta, W. Donato, V. Persico, A. Pescapé, "Integration of cloud computing and Internet of Things: A survey", Future Generation Computer Systems, Volume 56, 2016, pp. 684–700.

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems, Volume 29, Issue 7, 2013, pp. 1645–1660.

[3] G. Premsankar, M. Di Francesco, T. Taleb, "Edge computing for the Internet of Things: A case study", IEEE Internet of Things Journal, Volume 5, Issue 2, 2018, pp. 1275–1284.

[4] S. Li, L. D. Xu, S. Zhao, "The internet of things: A survey", Information Systems Frontiers, Volume 17, 2018, pp. 243–259.

[5] F. Liu, Y. Zhao, M. Xu, "Delay-sensitive applications in IoT: Challenges and issues", Journal of Sensor and Actuator Networks, Volume 6, Issue 4, 2017, Article 34.

[6] Y. He, S. Guo, J. Liu, "Task scheduling in real-time edge computing for autonomous vehicles", IEEE Transactions on Industrial Informatics, Volume 14, Issue 5, 2018, pp. 2292–2300.

[7] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, X. Yang, "A survey on the edge computing for the Internet of Things", IEEE Access, Volume 6, 2018, pp. 6900–6919.

[8] R. Roman, J. Lopez, M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges", Future Generation Computer Systems, Volume 78, 2018, pp. 680–698.

[9] M. Aazam, E. N. Huh, "Fog computing micro datacenter-based dynamic resource estimation and pricing model for IoT",

Proceedings of the ACM International Conference on Advanced Communication Technology, Volume 1, 2014, pp. 464–469.
[10] R. S. Sinha, Y. Wei, S. H. Hwang, "A survey on LPWAN technologies in IoT: Architecture, applications, and challenges", Journal of Sensor and Actuator Networks, Volume 6, Issue 2, 2017, Article 23.
[11] J. Tang, T. He, Y. Yin, "Lightweight stream analytics for resource-constrained edge devices", IEEE Access, Volume 7, 2019, pp. 110996–111010.
[12] J. Xu, Y. Ren, X. Lin, "Edge intelligence: The confluence of edge computing and artificial intelligence", IEEE Internet of Things Journal, Volume 7, Issue 8, 2020, pp. 7457–7469.
[13] M. Chiang, T. Zhang, "Fog and IoT: An overview of research opportunities", IEEE Internet of Things Journal, Volume 3, Issue 6, 2016, pp. 854–864.
[14] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D. S. Nikolopoulos, "Challenges and opportunities in edge computing", Proceedings of the IEEE International Conference on Smart Cloud, 2016, pp. 20–26.
[15] STMicroelectronics, "NanoEdge AI Studio", Available: https://www.st.com/en/development-tools/nanoedgeai.html.
[16] Edge Impulse, "Edge Impulse Platform", Available: https://www.edgeimpulse.com.
[17] H. Jeong, H. Kim, J. Kim, "Adaptive scheduling for edge AI inference in real-time systems", Sensors, Volume 20, Issue 14, 2020, Article 4041.
[18] Z. Zhou, Y. Liu, L. Zhang, "Energy-aware load balancing for edge-cloud systems using deep Q-learning", Future Internet, Volume 13, Issue 2, 2021, Article 39.
[19] A. Alqahtani, B. Shihada, M. S. Alouini, "Intelligent orchestration of edge computing resources for IoT", IEEE Internet of Things Journal, Volume 6, Issue 3, 2019, pp. 5674–5686.
[20] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, "Mobile edge computing: A survey", IEEE Internet of Things Journal, Volume 5, Issue 1, 2018, pp. 450–465.
[21] C. Zhang, J. Liu, Y. Zhao, "Task offloading with data dependency in mobile edge computing", China Communications, Volume 14, Issue 8, 2017, pp. 98–111.
[22] X. Liang, Y. Zhao, S. Shetty, D. Bowden, "Federated learning for IoT security: Communication-efficient and privacy-preserving models", IEEE Internet of Things Journal, Volume 7, Issue 10, 2020, pp. 8520–8533.