Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 2
Article Received: 10 March 2026 **Revised**: 16 May 2026 **Accepted**: 10 June 2026

A Structured Comparative Analysis of Cloud, Fog, and IoT Computing with Performance Metric Classification

S. Natarajan[1], G. Anandharaj[2],
[1]Research Scholar, Department of Computer Science, Sri Vasavi College (Govt Aided), Erode 638-316, India
Email: sendmail2natarajan@gmail.com , ORCID: 0009-0008-4342-7801
[2]Assistant Professor, Department of Computer Science, Sri Vasavi College (Govt Aided), Erode 638-316, India
Email: anandcssvc@gmail.com , ORCID: 0000-0002-2209-0460

Abstract: Rapid increase in Internet of Things (IoT) devices has created a need for fast and efficient data processing. Cloud computing has been widely used because high computation and storage are provided, but delay is introduced due to centralized data handling. Fog computing has been proposed to reduce this delay by moving processing closer to data sources.

In this paper, a comparative study of Cloud, Fog, and Internet of Things (IoT) systems is presented. Differences in architecture, processing capability, latency, and resource management are examined in a clear manner. A layered structure is described to show how IoT devices, fog nodes, and cloud systems work together for data processing. Important performance metrics such as latency, energy consumption, resource utilization, and Quality of Service are also discussed for system evaluation.

It is observed that IoT devices mainly generate real-time data with limited resources, cloud systems handle large-scale processing with higher delay, and fog computing provides intermediate processing with reduced latency. Integration of these paradigms is necessary to support real-time and scalable applications such as smart cities, healthcare monitoring, and industrial systems.

Keywords: Cloud Computing, Fog Computing, Internet of Things, Distributed Systems, Latency, Resource Management

1. INTRODUCTION

Rapid growth of connected devices has been observed in recent years. Large volumes of data are generated continuously by sensors, mobile devices, and smart systems. The Internet of Things (IoT) enables devices to collect and share data in real time. However, IoT devices are designed with limited processing power and storage capacity. Therefore, external systems are required for data processing and analysis [25].

Cloud computing has been widely adopted to provide large-scale storage and high computational capability. Data collected from IoT devices is transmitted to centralized cloud data centers for processing. Although this approach supports heavy computation, higher latency is introduced because of long communication paths. This delay becomes critical in applications that require immediate response [25].

Fog computing has been introduced to reduce this limitation. In fog computing, processing resources are placed closer to users and devices. Data can be partially processed at intermediate nodes before being sent to the cloud. This approach reduces latency, network traffic, and response time [16], [22]. A clear understanding of how Cloud, Fog, and IoT differ is required for efficient system design. Therefore, a detailed comparison is presented in this paper.

2. RELATED WORK

Recent studies in fog computing have focused on improving task scheduling, latency reduction, and resource management in IoT-based systems. Various optimization and learning-based methods have been applied to enhance system performance.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 2
Article Received: 10 March 2026 **Revised**: 16 May 2026 **Accepted**: 10 June 2026

In the work by Alsadie and Alsulami, an energy-efficient task scheduling approach based on Grey Wolf Optimization was proposed. Energy consumption was reduced by optimizing task allocation in fog environments. However, real-time latency constraints were not deeply analyzed [1].

In another study, Alsamarai introduced an improved scheduling algorithm for fog–cloud systems using optimization techniques. Performance and cost were balanced during task execution. Still, dynamic workload conditions were not fully considered [2].

Deep reinforcement learning-based scheduling methods were presented by Choppara and co-authors. Intelligent task allocation was achieved by learning system behavior over time. Although improved performance was observed, higher computational complexity was introduced [3], [4].

A multi-objective scheduling framework was developed by Deepakraj and colleagues to optimize resource allocation in fog-assisted IoT networks. Multiple parameters such as latency and resource utilization were considered together. However, scalability under large-scale environments remained a challenge [5].

Habtie and Teshome proposed a scheduling algorithm focused on reducing response time in fog computing systems. Faster task execution was achieved, but energy efficiency was not addressed [6].

Energy-efficient scheduling using multi-objective optimization was also explored, where both time and cost constraints were considered. Improved balance between performance and operational cost was achieved, but system adaptability remained limited [7].

A location-aware scheduling framework was proposed to improve task execution efficiency in fog–cloud environments. Resource allocation was optimized based on node location. However, real-time adaptability was not fully addressed [8].

From the above studies, it is observed that existing research mainly focuses on task scheduling and optimization techniques. However, a clear comparative analysis of Cloud, Fog, and IoT based on architecture, processing capability, latency, and resource management is not sufficiently addressed. In addition, performance metrics are often studied individually rather than in an integrated manner.

From the above discussion, it is observed that most existing studies focus on specific optimization techniques, while a unified comparative framework and structured metric classification are not clearly addressed. This limitation motivates the proposed approach in this paper. To address the identified limitations in existing studies, a structured framework and task allocation strategy are proposed in this work.

3. PROPOSED FRAMEWORK AND NOVEL CONTRIBUTIONS

A structured classification of performance metrics is proposed to improve the evaluation of Cloud, Fog, and IoT systems. In existing studies, metrics are often analyzed individually, and a unified grouping is not clearly defined [25]. To address this limitation, performance metrics are organized into three main levels based on system functionality.

At the first level, system-level metrics are defined. These metrics include latency, response time, and makespan. System performance is measured based on execution efficiency and delay handling [6], [7].

At the second level, resource-level metrics are considered. These include CPU utilization, memory utilization, bandwidth usage, and energy consumption. Resource efficiency is evaluated by measuring how effectively system components are used [13], [16].

At the third level**,** application-level metrics are introduced. These include Quality of Service (QoS), reliability, availability, and packet loss rate. Application performance is measured based on service quality and user experience [19], [23], [25].

This classification provides a clear structure for evaluating distributed computing systems. Improved comparison can be achieved by grouping related metrics instead of analyzing them separately.

3.1. Proposed Task Allocation Strategy

A simple task allocation strategy is proposed for efficient utilization of IoT, Fog, and Cloud resources. In many existing studies, task scheduling focuses on optimization techniques such as metaheuristic and learning-based methods [1], [3], [9], but clear rule-based decision logic is not explicitly defined.

In the proposed approach, task allocation is decided based on system requirements.

If a task requires real-time response and low latency, processing is assigned to fog nodes [6], [7]. If a task requires high computational power and large storage, it is assigned to cloud systems [2], [8]. If a task involves data sensing and immediate interaction, it is handled by IoT devices [25].

This rule-based strategy enables efficient distribution of workload across different layers. Delay-sensitive tasks are processed closer to the source, while computation-intensive tasks are handled by centralized systems. This improves overall system performance and reduces unnecessary communication overhead.

3.2. Scenario-Based Comparative Analysis

A scenario-based comparison is introduced to provide practical insights into system behavior. In many comparative studies, general differences are described without application context [25].

In smart healthcare systems, IoT devices are used to collect patient data such as heart rate and temperature. Fog nodes process this data locally to generate immediate alerts, while cloud systems are used for long-term analysis [3], [11]. In this case, fog computing is preferred for real-time decision making.

In smart city traffic systems, large volumes of data are generated continuously. Fog nodes handle local traffic control, while cloud systems perform large-scale data analysis [6], [24]. A hybrid model is required for efficient operation.

In industrial IoT environments, machines generate continuous sensor data. Fault detection is performed at fog nodes to reduce delay, while cloud systems are used for predictive maintenance [5], [20]. This improves system reliability and reduces downtime.

This scenario-based analysis provides a clearer understanding of how each computing paradigm is used in real-world applications.

4. OVERVIEW OF COMPUTING PARADIGMS

4.1. Internet of Things (IoT)

IoT refers to a network of interconnected devices such as sensors, actuators, cameras, and smart objects. These devices collect data from the environment and transmit it to other systems. Data generation is continuous and often time-sensitive. However, IoT devices have limited computational capability and energy resources. As a result, most processing tasks are offloaded to fog or cloud systems [25].

4.2. Cloud Computing

Cloud computing provides centralized infrastructure for data storage and processing. Large data centers are used to handle heavy computational workloads. Services such as data analytics, machine learning, and long-term storage are supported in the cloud. However, data must travel long distances from IoT devices to cloud servers. This increases latency and network usage, which can affect system performance in real-time applications [25].

4.3. Fog Computing

Fog computing acts as an intermediate layer between IoT devices and cloud systems. It consists of distributed nodes such as gateways, routers, and local servers. Data processing is performed closer to the source, which reduces delay and improves response time. Fog computing also supports local decision making and reduces dependency on centralized systems [16], [22]

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 2
Article Received: 10 March 2026 **Revised**: 16 May 2026 **Accepted**: 10 June 2026

5. INTEGRATED SYSTEM ARCHITECTURE

A hierarchical architecture is used to combine IoT, Fog, and Cloud systems. The architecture consists of three main layers.

The edge layer includes IoT devices that generate and collect data. Basic processing and sensing operations are performed at this level. The fog layer is placed above the edge and consists of intermediate nodes that perform local processing and filtering. The cloud layer is located at the top and handles large-scale processing and storage.

This layered approach reduces communication delay and distributes workload efficiently. Real-time tasks can be handled at the edge or fog layer, while complex tasks can be processed in the cloud [16], [25].

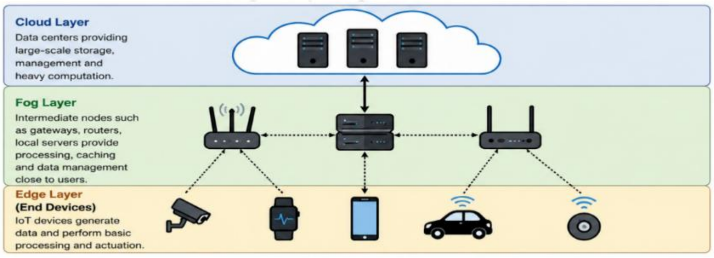


Figure 1: IoT–Fog–Cloud Architecture

Figure 1 illustrates the integrated architecture of Internet of Things (IoT), Fog, and Cloud computing systems. A hierarchical structure is shown with three layers: edge layer, fog layer, and cloud layer.

At the bottom, the edge layer consists of IoT devices such as sensors, cameras, and smart objects. Data is generated continuously at this layer. Basic sensing and initial processing operations are performed before data is transmitted.

The fog layer is positioned between the edge and cloud layers. It includes intermediate nodes such as gateways, routers, and local servers. Data is partially processed, filtered, and analyzed at this layer. Delay is reduced because processing is performed closer to the data source. Real-time decisions can be supported at this level.

At the top, the cloud layer provides centralized data storage and high computational capability. Large-scale data processing, analytics, and long-term storage are handled in this layer.

Data flow is shown from IoT devices to fog nodes and then to cloud systems. Time-sensitive tasks are processed at the edge or fog layer, while computation-intensive tasks are forwarded to the cloud. This layered approach improves system efficiency, reduces latency, and supports scalable application design.

It is observed that an integrated approach combining IoT, fog, and cloud layers provides better performance compared to using a single paradigm. Efficient task allocation plays a key role in achieving optimal system behaviour.

6. COMPARATIVE ANALYSIS

A comparison of Cloud, Fog, and IoT is presented in Table I. Each paradigm is evaluated based on its role, location, latency, storage, and processing capability.

Table 1: Cloud vs Fog Computing vs Edge Computing

Aspect	Cloud Computing	Fog Computing	Edge Computing
Distance to user	Far (via backbone networks)	Near (access/metro networks)	Very near (on-device or one hop)
Latency	High (tens to hundreds of ms)	Low (few to tens of ms)	Ultra-low (sub ms to few ms)
Storage capacity	Very high (large data centers)	Medium (local servers/clusters)	Very low (device memory only)
Control	Centralized management	Distributed coordination	Local device-level control
Typical functions	Heavy analytics, backup, training	Pre-processing, aggregation, caching	Sensing, actuation, basic inference

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 2
Article Received: 10 March 2026 **Revised**: 16 May 2026 **Accepted**: 10 June 2026

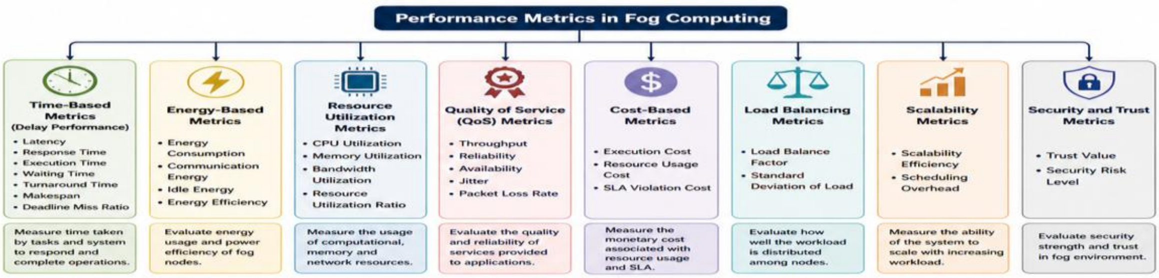
The comparison in Table I shows that cloud computing provides high computational power and storage but introduces higher latency due to centralized processing. Fog computing reduces latency by processing data closer to users, while maintaining moderate resource capacity. Edge computing provides the lowest latency by processing data directly at devices, but it is limited by low storage and processing capability. Therefore, fog computing acts as an intermediate layer that balances performance and delay.

Cloud computing provides high computational power and storage capacity. However, latency is high due to centralized processing. Fog computing provides moderate resources and operates closer to users, which reduces latency. IoT devices operate at the edge and provide real-time data generation with minimal delay but limited processing capability.

These paradigms are not independent. Instead, they work together to form a complete system. IoT devices generate data, fog nodes process data locally, and cloud systems perform large-scale analysis.

7. PERFORMANCE METRICS IN FOG COMPUTING

Table II: Classification of Performance Metrics of Fog Computing



Performance evaluation is required in fog computing to measure system efficiency under different workloads. Fog environments consist of distributed and resource-constrained nodes, so multiple evaluation metrics are used to analyze system behavior. These metrics support comparison of scheduling algorithms and system designs [7], [16], [25].

Table III: Performance Metrics of Fog Computing

Category	Metrics	Description	Purpose / Importance
Time-Based Metrics (Delay Performance)	• Latency • Response Time • Execution Time • Waiting Time • Turnaround Time • Makespan • Deadline Miss Ratio	Time for data travel from IoT device to fog node and back. Total time from task submission to completion. Actual time required to execute a task on a fog node. Time a task waits in the queue before execution. Time from task submission to final completion. Total time to complete all tasks in the system. Percentage of tasks not completed within deadlines.	Measures communication delay. Evaluates overall system response. Measures computation efficiency. Indicates queue congestion. Measures overall task completion. Used to evaluate scheduling efficiency. Indicates real-time performance.
Energy-Based Metrics	• Energy Consumption • Communication Energy • Idle Energy • Energy Efficiency	Total energy used for computation and communication. Energy consumed during data transmission. Energy consumed when a node is idle. Number of tasks completed per unit of energy.	Evaluates overall energy usage. Measures communication cost. Indicates power wasted in idle state. Measures energy effectiveness.
Resource Utilization Metrics	• CPU Utilization • Memory Utilization • Bandwidth Utilization • Resource Utilization Ratio	Percentage of CPU used during execution. Amount of memory used by tasks. Amount of network bandwidth used. Ratio of used resources to total available resources.	Measures processor usage. Measures memory usage. Measures network usage. Indicates efficient resource usage.
Quality of Service (QoS) Metrics	• Throughput • Reliability • Availability • Jitter • Packet Loss Rate	Number of tasks completed per unit time. Ability of the system to perform without failure. Percentage of time the system is operational. Variation in delay in packet or data delivery. Percentage of data packets lost during transmission.	Indicates system performance. Indicates dependability of the system. Indicates service continuity. Important for real-time applications. Indicates communication quality.
Cost-Based Metrics	• Execution Cost • Resource Usage Cost • SLA Violation Cost	Cost of executing tasks on fog or cloud nodes. Cost based on CPU, memory and bandwidth usage. Penalty cost for not meeting SLA requirements.	Measures execution expense. Evaluates resource related cost. Indicates service satisfaction level.
Load Balancing Metrics	• Load Balance Factor • Standard Deviation of Load	Measures how evenly workload is distributed. Variation in workload among different nodes.	Indicates load distribution quality. Lower value indicates better balance.
Scalability Metrics	• Scalability Efficiency • Scheduling Overhead	Performance change as the number of tasks increases. Extra time required to manage and allocate tasks.	Indicates ability to handle large scale. Lower overhead improves scalability.
Security and Trust Metrics	• Trust Value • Security Risk Level	Reliability level of fog nodes or services. Probability of attack or vulnerability.	Indicates node/service trustworthiness. Indicates security level of the system.

Table II presents a structured classification of performance metrics used in fog computing systems. It is observed that time-based and energy-based metrics are critical for real-time applications, while resource utilization and QoS metrics are important for system efficiency and service quality. Cost and load balancing metrics help in optimizing operational performance, and scalability and security metrics ensure system stability under dynamic conditions. Proper selection of these metrics is required for effective system evaluation.

7.1. Time-Based Metrics (Delay Performance)

Time-related metrics are critical for real-time applications.

Latency is defined as the time required for data transmission between IoT devices and fog nodes. Lower latency indicates faster system response [25]. Response time represents the total time from task submission to completion, including waiting, execution, and transmission delays [7].

Execution time measures the actual processing time of a task on a fog node, while waiting time represents the delay before execution begins. Turnaround time is measured from submission to completion of a task. Makespan represents the total time required to complete all tasks in the system. Deadline miss ratio indicates the percentage of tasks that fail to meet timing constraints. These metrics are widely used in scheduling studies [6], [7], [18].

7.2. Energy-Based Metrics

Energy consumption is an important factor in fog environments due to limited power availability.

Total energy consumption includes both computation and communication energy. Communication energy represents the energy used for data transfer between nodes, while idle energy indicates power consumption when nodes are not actively processing tasks. Energy efficiency is defined as the number of tasks completed per unit of energy. Energy-aware scheduling approaches aim to reduce power usage while maintaining performance [1], [13], [18].

7.3. Resource Utilization Metrics

Resource utilization metrics measure how effectively system resources are used.

CPU utilization indicates processor usage during execution. Memory utilization measures how much memory is consumed by tasks, and bandwidth utilization represents network usage. Resource utilization ratio is defined as the ratio of used resources to available resources. Efficient utilization improves system performance and reduces idle resources [16], [23].

7.4. Quality of Service (QoS) Metrics

QoS metrics are used to evaluate service quality in fog systems.

Throughput measures the number of tasks completed per unit time. Reliability represents system stability during operation, while availability indicates the percentage of uptime. Jitter measures variation in delay, and packet loss rate indicates data loss during transmission. These metrics ensure that application requirements are satisfied [19], [23], [25].

7.5. Cost-Based Metrics

Cost metrics are important in fog–cloud integrated systems.

Execution cost represents the cost of processing tasks on fog or cloud nodes. Resource usage cost depends on CPU, memory, and bandwidth consumption. SLA violation cost represents penalties when QoS requirements are not satisfied. Cost-aware scheduling aims to balance performance and operational cost [2], [18].

7.6. Load Balancing Metrics

Load balancing ensures equal distribution of tasks across nodes.

Load balance factor measures how evenly workload is distributed, while standard deviation of load indicates variation among nodes. Proper load balancing reduces bottlenecks and improves system efficiency [6], [21].

7.7. Scalability Metrics

Scalability metrics evaluate system performance under increasing workload.

Scalability efficiency measures performance variation as the number of tasks increases. Scheduling overhead represents additional time required for task management. Efficient systems maintain stable performance even with large-scale workloads [14], [24].

7.8. Security and Trust Metrics

Security metrics are important in fog environments where sensitive data is processed.

Trust value represents the reliability of nodes, while security risk level indicates the probability of attacks or vulnerabilities. Secure scheduling approaches aim to protect data and ensure system integrity [22].

8. APPLICATIONS

Cloud, fog, and IoT systems are widely used in various applications. In smart city systems, IoT devices collect data such as traffic and environmental conditions. Fog nodes process this data locally to provide quick responses, while cloud systems perform large-scale analysis.

In healthcare systems, IoT sensors monitor patient conditions such as heart rate and temperature. Fog computing enables real-time alert generation, which improves patient safety. In industrial environments, IoT devices monitor machines, and fog nodes process data to detect faults. Cloud systems are used for long-term analysis and optimization [1], [7].

9. DISCUSSION

Each computing paradigm has its own strengths and limitations. IoT provides real-time data but lacks processing capability. Cloud computing provides high computational power but introduces delay. Fog computing provides a balance between these two by reducing latency and supporting distributed processing.

Efficient system design requires proper integration of all three paradigms. Task allocation should be based on latency requirements and resource availability. Real-time tasks should be processed at the edge or fog layer, while complex tasks should be handled by the cloud.

CONCLUSION

A comparative study of Cloud, Fog, and Internet of Things (IoT) computing paradigms has been presented. Key differences in architecture, processing capability, latency, and resource management have been analyzed. It has been observed that IoT devices are mainly used for data generation, while cloud systems provide large-scale processing and storage. However, higher delay is introduced in cloud-based processing due to long communication paths.

Fog computing has been identified as an effective intermediate solution that reduces latency by enabling processing closer to data sources. Improved response time and reduced network load can be achieved through fog-based systems. The integration of IoT, fog, and cloud has been shown to support efficient and scalable system design, especially for real-time applications.

Performance evaluation metrics such as latency, energy consumption, resource utilization, and Quality of Service have been discussed to measure system efficiency. Proper selection of these metrics is required for designing reliable fog computing environments.

Future improvements can be achieved by applying intelligent task scheduling algorithms and adaptive resource management techniques. Advanced approaches such as machine learning and optimization-based methods can enhance system performance under dynamic conditions. Continued research in this area can support the development of efficient and secure distributed computing systems. This study provides a foundation for designing efficient distributed systems by combining comparative analysis with structured performance evaluation.

REFERENCES

[1] D. Alsadie and M. Alsulami, "Modified grey wolf optimization for energy-efficient Internet of Things task scheduling in fog computing," Scientific Reports, vol. 15, Art. no. 14730, 2025.

[2] N. A. Alsamarai, "Improved performance and cost algorithm for scheduling IoT tasks in fog–cloud environment using Gray Wolf Optimization algorithm," Applied Sciences, vol. 14, no. 4, Art. no. 1670, 2024.

[3] P. Choppara and B. Lokesh, "Efficient task scheduling and load balancing in fog computing for healthcare through deep reinforcement learning," IEEE Access, vol. 13, pp. 26542–26563, 2025.

[4] P. Choppara and S. S. Mangalampalli, "An efficient deep reinforcement learning based task scheduler in cloud–fog environment," Cluster Computing, vol. 28, Art. no. 67, 2025.

[5] P. Choppara and S. S. Mangalampalli, "Resource adaptive automated task scheduling using deep deterministic policy gradient in fog computing," IEEE Access, vol. 13, pp. 25969–25994, 2025.

[6] D. Deepakraj, V. Saidulu, A. Hussain, S. Muruganandam, S. Kumar, and J. RajaSekhar, "Optimizing task scheduling and resource allocation using multi-criteria framework in fog-assisted IoT networks," International Journal of Computational and Experimental Science and Engineering, vol. 11, no. 2, pp. 3291–3300, 2025.

[7] A. B. Habtie and T. Teshome, "Task scheduling algorithm to reduce response time over fog computing environment," International Journal of Combinatorial Optimization Problems and Informatics, vol. 16, no. 3, pp. 391–404, 2025.

[8] N. G. Hammouda, M. Shalaby, R. H. C. Alfilh, and N. S. Singh, "A quantum-driven multi-objective scheduler for scalable task orchestration in fog-based cyber-physical-social systems," Scientific Reports, vol. 16, Art. no. 6874, 2026.

[9] A. Heirati, M. Fartash, and M. Khalily-Dermany, "Optimized task scheduling in fog-cloud computing using hybrid deep learning and metaheuristic algorithms," Neural Processing Letters, vol. 58, Art. no. 5, 2026.

[10] Z. Huang, "Task scheduling in fog computing systems using deep learning," Egyptian Informatics Journal, vol. 31, Art. no. 100757, 2025.

[11] P. Jenifer and J. A. J. Sujana, "Quality of experience-aware application deployment in fog computing environments using machine learning," PeerJ Computer Science, vol. 11, Art. no. e3143, 2025.

[12] W. Jin and A. Rezaeipanah, "Dynamic task allocation in fog computing using enhanced fuzzy logic approaches," Scientific Reports, vol. 15, Art. no. 18513, 2025.

[13] S. K. Medishetti and G. R. Karri, "BSHOA: Energy efficient task scheduling in cloud–fog environment," International Journal of Computer Network and Information Security, vol. 16, no. 4, pp. 88–101, 2024.

[14] A. Mindil, A. Y. Hamed, M. R. Hassan, and M. Kh. Elnahary, "A novel approach for dynamic task scheduling for IoT in fog–cloud environment," Scientific Reports, vol. 16, Art. no. 5501, 2026.

[15] M. Sehrawat, H. Sehrawat, and V. Siwach, "A greedy constraints-based fog computing model to optimize task scheduling," International Journal of Engineering Trends and Technology, vol. 73, no. 5, pp. 359–368, 2025.

[16] K. Natarajan, "Performance evaluation of fog computing for latency and energy efficiency in IoT applications," Journal of Computer and Communication Networks, vol. 1, pp. 22–32, 2025.

[17] H. M. Patel, K. J. Modi, and C. I. Patel, "Cloud brain fog scheduler: Machine learning-enhanced task allocation," SSRG International Journal of Electronics and Communication Engineering, vol. 11, no. 7, pp. 200–207, 2024.

[18] S. Ijaz, S. G. Ahmad, K. Ayyub, E. U. Munir, and N. Ramzan, "Energy-efficient time and cost constraint scheduling algorithm using improved multi-objective differential evolution in fog computing," The Journal of Supercomputing, vol. 81, Art. no. 116, 2025.

[19] M. Rani, K. Guleria, and S. N. Panda, "Improving QoS parameters in fog computing environment using QCSO framework," Communications on Applied Nonlinear Analysis, vol. 32, no. 3, pp. 197–210, 2025.

[20] M. Rani, K. Guleria, and S. N. Panda, "Enhancing performance and latency optimization in fog computing with a smart job scheduling approach," Statistics, Optimization and Information Computing, vol. 13, pp. 309–330, 2025.

[21] M. Saad, R. N. Enam, and R. Qureshi, "Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data applications," Frontiers in Big Data, vol. 7, Art. no. 1358486, 2024.

[22] V. Thangaraj and T. Raja Sree, "Mobility-aware secure computation offloading in blockchain-enabled fog computing environments," Journal of Cloud Computing, vol. 13, Art. no. 88, 2024.

[23] V. Usha and T. K. R. K. Rao, "Quality of service-driven optimized resource provisioning in fog computing," International Journal of Engineering Trends and Technology, vol. 73, no. 2, pp. 1–8, 2025.

[24] X. Yu, M. Zhu, X. Zhou, and L. Long, "A novel location-aware job scheduling framework for optimizing fog-cloud IoT systems," Journal of Cloud Computing, vol. 14, Art. no. 50, 2025.

[25] M. S. Aslanpour et al., "Performance evaluation metrics for cloud, fog and edge: A review, taxonomy, benchmarks and standards for future research," Internet of Things, vol. 10, Art. no. 100273, 2020.