Metaversalize

www.meta.reapress.com

Metaverse. Vol. 2, No. 1 (2025) 21–30.

Paper Type: Original Article

Energy-Efficient Communication in Edge Computing IoT Networks

Akhil Pandey*

School of Computer Science Engineering, KIIT (Deemed to Be) University, Bhubaneswar – 751024, Odisha, India; 2229006@kiit.ac.in.

Citation:

Received: 12 May 2024 *Revised: 7 September 2024* *Accepted: 1 December 2024*	Pandey, A. (2025). Energy-efficient communication in edge computing IoT networks. *Metaversalize*, 2(1), 21-30.

Abstract

The rapid growth of Internet of Things (IoT) devices necessitates the development of energy-efficient communication strategies, particularly in edge computing environments where resource limitations are a primary concern. This paper presents a novel framework to enhance energy efficiency in edge computing IoT networks by integrating adaptive routing algorithms and data compression techniques. Our approach minimizes energy consumption while maintaining optimal data transmission rates and low latency. Extensive simulations and practical implementations demonstrate that our framework achieves up to 30% reduction in energy usage compared to traditional methods without compromising communication reliability. We also discuss the trade-offs between energy efficiency and network performance, providing valuable insights for various IoT applications. This research contributes to advancing sustainable IoT solutions, paving the way for more efficient edge computing systems.

Keywords: Energy efficiency, Edge computing, Internet of things, Resource optimization, Latency reduction.

1 | Introduction

The rapid expansion of Internet of Things (IoT) devices has revolutionized industries by facilitating enhanced data collection and automation. As billions of devices generate continuous data streams, the demand for efficient communication strategies becomes critical, particularly in edge computing environments where data processing is closer to the source [1]. Edge computing significantly reduces latency and alleviates bandwidth constraints, yet it also introduces challenges, particularly regarding energy consumption, a key concern for battery-operated devices [2–4].

In this context, energy efficiency plays a vital role in ensuring the longevity and reliability of IoT networks. Conventional communication protocols often prioritize throughput and reliability over energy conservation

Corresponding Author: 2229006@kiit.ac.in
https://doi.org/10.22105/metaverse.v2i1.47

metrics, leading to increased power usage and reduced device lifespans. This inefficiency can be particularly detrimental in applications involving remote sensors and devices deployed in inaccessible locations.

2 | Background on Energy-Efficient Communication in Edge Computing IoT Networks

The IoT has evolved from basic interconnected devices to sophisticated networks supporting various applications, from smart cities and healthcare to industrial automation and transportation. However, IoT networks encounter challenges such as limited device battery life, latency sensitivity, and computational requirements. The introduction of edge computing addresses some of these issues by moving computation closer to data sources, thereby reducing latency and alleviating the burden on centralized cloud infrastructures. Yet, the energy efficiency challenge remains crucial, especially as IoT devices are typically energy-constrained and rely on wireless communications, which can be energy-intensive [3].

Edge computing provides an environment where IoT devices offload data and computations to nearby edge servers, allowing faster data processing, reduced communication latency, and improved scalability [4]. However, this shift demands careful management of communication and computational processes to prevent excessive energy consumption, especially in multi-hop networks where devices communicate indirectly through other devices. To enable sustainable IoT networks, strategies are required to minimize energy usage at both device and network levels, particularly in data transmission and processing tasks.

3 | Algorithms for Energy-Efficient Communication

Energy-efficient communication in IoT and edge computing networks often relies on specialized algorithms that minimize energy usage by optimizing data transmission, processing, and network management. These algorithms are crucial in extending the battery life of IoT devices, improving network sustainability, and ensuring reliable communication even in energy-constrained environments. Here are some key algorithms and approaches commonly employed for energy-efficient communication:

3.1 | Clustering and Hierarchical Routing Algorithms

I. Low-Energy Adaptive Clustering Hierarchy （LEACH) [5] is a popular algorithm in Wireless Sensor Networks (WSNs) that forms clusters of nearby devices. Each cluster elects a leader (cluster head) responsible for collecting and aggregating data from member nodes before sending it to the base station. This reduces the number of direct transmissions to the base station, significantly lowering energy consumption.

II. Threshold-sensitive Energy-Efficient Network（TEEN) is used for time-sensitive applications where energy is saved by setting thresholds. Sensors only transmit data if values surpass a specified threshold, reducing redundant data transmission.

III. Power-Efficient Gathering in Sensor Information System（PEGASIS) organizes [6] nodes into chains rather than clusters, where each node communicates only with a close neighbor. Forwarding data along the chain to a central base station reduces the overall communication cost.

3.2 | Adaptive Duty Cycling Algorithms

I. Sensor Medium Access Control (S-MAC) is [7] a duty-cycling MAC protocol where nodes periodically switch between sleep and active states. By reducing active time, S-MAC conserves energy while maintaining data exchange capabilities.

II. Timeout-MAC (T-MAC) [8] improves upon S-MAC by dynamically adjusting duty cycles based on network traffic. Nodes stay active only when there is data to transmit, which can significantly reduce energy waste during low-traffic periods.

3.3|Data Compression and Aggregation Algorithms

I. Compressive Sensing (CS) reduces [9] the number of samples needed for accurate data reconstruction. This approach reduces the amount of transmitted data, lowering energy consumption while still capturing essential information.

II. In-network data aggregation combine data from multiple sources to eliminate redundancy, reducing the total amount of data sent [10]. Algorithms like Data Fusion or Aggregation Trees are common in IoT networks for minimizing energy by preprocessing data at the edge nodes.

3.4|Energy-Efficient Routing Algorithms

I. Energy-Aware Geographic Routing (EAGR) uses [11] location-based information to select routes that minimize energy consumption. By balancing loads and avoiding low-battery nodes, these protocols prolong the network lifetime.

II. Multipath routing protocols such as directed diffusion, energy-aware ad hoc on-demand distance vector use multiple routes for data transmission, which balances energy consumption across the network. This method prevents energy depletion in certain nodes and distributes the energy load [12].

III. Energy harvesting adaptive routing take energy harvesting capabilities into account, routing data through nodes with available harvested energy to maximize network longevity [13].

3.5|Machine Learning-Based Energy Optimization

I. Reinforcement Learning (RL)-based algorithms optimize communication decisions dynamically. For instance, RL can be applied to optimize data offloading decisions and balance computation between edge devices and servers to save energy [14].

II. Predictive wake-up scheduling mainly supervised learning models, predict when devices must transmit data. Devices can then be scheduled to "wake up" only when needed, minimizing energy used in idle states [15].

III. Deep Q-Networks (DQN) manage energy resources by dynamically adjusting routing, offloading, or communication schedules. For example, a DQN can be trained to find optimal offloading paths for data-intensive tasks, balancing energy efficiency with latency constraints [16].

4|Case Studies

4.1|Smart Agriculture in Rural IoT Networks

In a smart agriculture IoT deployment, sensor nodes are placed across large fields to monitor soil moisture, temperature, and humidity to optimize crop yield [17]. These IoT devices are generally battery-powered and located in remote areas where maintenance is infrequent, making energy-efficient communication crucial. To conserve energy, a combination of data aggregation and duty-cycling algorithms was implemented. By using in-network data aggregation techniques, redundant data was minimized before transmission to edge nodes. An adaptive duty-cycling protocol was also used, where devices alternated between active and sleep modes based on environmental conditions and data significance thresholds. The energy-efficient communication protocols extended battery life by up to 30% across the network. Furthermore, using edge computing nodes enabled local data processing, reducing the frequency of data transmissions to a central server and conserving bandwidth and energy. Data aggregation and adaptive duty cycling can significantly reduce the energy load on rural IoT networks, where power sources are limited and maintenance access is challenging.

4.2|Real-Time Health Monitoring System

A health monitoring system using wearable IoT devices was developed to monitor vital signs such as heart rate, oxygen levels, and body temperature in real-time for patients in a clinical setting [18]. These wearables required energy-efficient algorithms to maintain continuous operation without frequent charging. Machine learning-based energy optimization algorithms were implemented to predict low-activity periods and schedule device “wake-up” times accordingly. RL algorithms were also used to balance the trade-off between energy consumption and data accuracy, ensuring that vital signs were only transmitted when critical changes were detected. Predictive wake-up scheduling reduced energy usage by approximately 25%, while RL-based optimization extended device operating time by selectively transmitting important data. This approach allowed wearable devices to run longer, which is particularly valuable for overnight patient monitoring. Machine learning-based optimization, including predictive wake-up scheduling and selective data transmission, can greatly enhance battery life in real-time health monitoring systems by focusing only on transmitting critical data.

4.3 | Smart City Traffic Management

A smart city traffic management system installed an IoT network of cameras, sensors, and edge nodes at intersections to monitor traffic flow and pedestrian activity [19]. The goal was to optimize traffic signals based on real-time data, reduce congestion, and improve safety, all while maintaining energy efficiency. Energy-aware geographic routing and task-offloading algorithms were applied. Geographic routing algorithms efficiently relay data from the sensors to edge servers, optimizing energy usage based on device locations. Additionally, task offloading was applied to edge devices with ample power supply, ensuring that data processing tasks were handled locally whenever possible. The traffic management system improved energy efficiency by over 35%, as data processing was conducted at nearby edge servers rather than sent to a central server. This saved energy and decreased latency, allowing for near-instantaneous traffic signal adjustments. For smart city applications, energy-aware routing and task offloading algorithms effectively balance real-time responsiveness with energy efficiency, especially in densely networked environments.

4.4 | Industrial IoT for Predictive Maintenance

An industrial IoT deployment involves sensors embedded in factory machinery, monitoring parameters like temperature, vibration, and pressure to predict potential equipment failures [20]. The system aimed to ensure continuous operation by predicting maintenance needs while minimizing energy consumption. Multi-Objective Optimization (MOO) algorithms were used to balance energy efficiency, latency, and bandwidth. Additionally, CS was employed to capture and transmit only essential data, reducing transmission loads on the network. The MOO algorithm allowed the system to adjust its data collection and transmission parameters dynamically, reducing energy consumption by approximately 40% compared to a baseline system. CS reduced data volume, leading to further energy savings and decreased unnecessary network congestion. MOO and CS can be very effective for industrial IoT, especially when continuous monitoring is required, but network resources are limited.

4.5 | Energy-Efficient Environmental Monitoring in Remote Areas

In a remote environmental monitoring project, IoT sensors were deployed to track air quality, temperature, and wildlife activity [21]. Limited access to power sources and harsh environmental conditions made energy-efficient communication a top priority. Cluster-based algorithms like LEACH and energy-harvesting adaptive routing were implemented. Data was aggregated at cluster heads before transmission to edge servers by clustering devices. Energy-harvesting techniques (e.g., solar) were also used, where nodes prioritized using solar-powered devices for data transmission when possible. The combination of LEACH clustering and adaptive routing reduced network energy consumption by about 50%. Energy harvesting also extended operational time, allowing the network to function with minimal human intervention.

5 | Challenges and Future Directions

Energy-efficient communication in edge computing IoT networks faces several challenges that limit the full realization of sustainable, scalable, and high-performing IoT applications. Additionally, as technology advances and the volume of IoT devices increases, new research directions and technologies are emerging to address these challenges.

I. Limited power sources for IoT devices: IoT devices, especially in remote or mobile environments, often rely on batteries with limited capacity, constraining the energy available for data processing and communication. Replacing or recharging batteries in large-scale IoT deployments, such as smart cities or environmental monitoring, can be costly and impractical. Energy harvesting methods (e.g., solar or ambient energy) offer solutions but are unreliable in every environment and often don't provide consistent power.

II. Network scalability and heterogeneity: IoT networks are increasingly composed of heterogeneous devices with varying energy capabilities, data requirements, and communication standards. Ensuring efficient communication across this diverse network without draining power on less capable devices is complex. Furthermore, as networks scale, the volume of data traffic can overwhelm network resources, complicating efforts to maintain energy efficiency.

III. Latency and real-time processing requirements: many IoT applications, like healthcare monitoring or autonomous vehicles, demand real-time processing and low-latency communication. Balancing these demands with energy efficiency can be difficult, especially when latency-sensitive data cannot be delayed or offloaded to conserve energy. Achieving low-latency processing while managing power consumption remains a critical issue in edge computing-based IoT networks.

IV. Dynamic and unpredictable network conditions: IoT networks often operate in environments where connectivity, data traffic, and energy availability can be highly variable. Dynamic factors, such as network congestion, device mobility, or fluctuating environmental conditions, impact energy efficiency. Algorithms that perform well in stable conditions may struggle with these variables, requiring adaptive solutions that can respond to changing conditions.

V. Security and privacy concerns: IoT networks and edge devices often process sensitive data, and adding security protocols typically increases energy consumption. Lightweight security algorithms help reduce this overhead but may not be robust enough for applications requiring high security, like healthcare or financial transactions. Ensuring secure communication without significantly compromising energy efficiency is a continuous challenge.

VI. Data processing and storage constraints: edge devices have limited processing power and storage, constraining their ability to perform energy-intensive tasks. Efficient algorithms are required to optimize storage and manage computational loads while conserving energy, particularly in applications requiring frequent data collection and processing, like environmental monitoring or industrial IoT.

The future directions of energy-efficient communication in edge computing IoT networks are as follows:

I. Advanced energy harvesting techniques: future IoT networks could rely more on advanced energy harvesting, such as wireless energy transfer, thermoelectric energy conversion, or advanced photovoltaic cells, to provide consistent and sustainable power sources. Integrating multi-source energy harvesting (e.g., combining solar and kinetic) into IoT devices could significantly improve energy availability, reducing the need for frequent battery replacements.

II. AI-driven energy management: ML and AI hold promise for adaptive energy management by predicting network traffic patterns, dynamically adjusting device duty cycles, and optimizing data offloading based on historical and real-time data. For example, RL algorithms could enable IoT networks to self-optimize by learning energy-efficient communication patterns that balance power and performance.

III. Low-power edge AI models: developing lightweight AI models that can run on resource-constrained edge devices is a priority for future IoT systems. Techniques such as model pruning, quantization, and edge AI frameworks allow ML models to run efficiently on low-power devices, enabling real-time decision-making without frequent communication with the cloud. These edge AI models can help devices process data locally, reducing energy consumption from data transmission.

IV. Cross-layer and cross-domain optimization: future networks could benefit from cross-layer optimization, where different layers (e.g., physical, data link, network) communicate to optimize energy use holistically. This approach allows simultaneous adjustments across multiple protocol layers, improving energy efficiency without sacrificing performance. Cross-domain optimization, where IoT and edge computing systems work with cloud resources, can also help achieve a balanced distribution of computational and communication tasks.

V. Dynamic clustering and hierarchical network structures: dynamic clustering, where IoT devices form clusters based on their proximity, energy level, or task requirements, can improve energy efficiency. By forming temporary clusters or hierarchical structures, devices can collaborate on data aggregation and processing tasks, reducing the need for continuous communication. Future research may focus on adaptive clustering algorithms that respond to changing network and environmental conditions.

VI. Integration of blockchain for secure and energy-efficient IoT: blockchain has potential in IoT networks for secure data management and communication, but it typically consumes a lot of power. Future efforts could explore lightweight blockchain protocols or consensus mechanisms specifically tailored for energy-constrained IoT devices, ensuring security without overloading the devices' power resources.

VII. Green IoT frameworks and sustainable design: with a growing focus on sustainability, green IoT frameworks are emerging to guide the design of eco-friendly IoT systems. These frameworks emphasize efficient resource usage, low-power hardware, recyclable materials, and sustainable data transmission methods. Sustainable IoT design aims to reduce the environmental impact of large-scale deployments, which is increasingly relevant as IoT continues to expand.

6 | Role of Edge AI in Energy Efficiency

Edge Artificial Intelligence (Edge AI) [22] implemented at the network edge, is pivotal in enhancing energy efficiency in IoT networks. By enabling localized data processing, decision-making, and optimization at edge nodes, Edge AI minimizes the need for constant data transmission to central servers, which conserves energy, reduces latency, and supports real-time applications. Here's a breakdown of the specific roles and benefits of Edge AI in driving energy-efficient IoT systems.

6.1 | Localized Data Processing and Reduced Transmission Needs

By Processing data locally on edge devices, Edge AI significantly reduces the volume of data that needs to be sent to cloud servers. For example, a camera in a smart city environment might use edge AI to process video data and send only relevant traffic patterns or incidents to the central server, minimizing energy used in data transmission. Edge AI can filter irrelevant or redundant data, send only the most meaningful data, or aggregate information across multiple devices. This reduces the energy demand from transmitting large data volumes and is particularly valuable in applications with high data output, like industrial IoT and real-time monitoring.

6.2 | Intelligent Duty Cycling and Adaptive Wake-Up Mechanisms

Edge AI models can predict low-activity periods and automatically adjust device duty cycles, allowing devices to enter low-power sleep modes when not actively collecting critical data. For example, AI models may predict that environmental sensors in a smart agriculture setup need only activate at specific times of day, saving energy during low-impact periods. With event-based AI, devices activate only in response to specific triggers, such as a change in sensor readings or the presence of a human in a surveillance area. This approach reduces the need for constant data transmission, saving significant energy, especially in areas with unpredictable activity patterns.

6.3 | Real-Time Decision-Making and Localized Response

Edge AI enables real-time processing, enabling IoT devices to make instant decisions without consulting a central server. This is especially valuable in energy-critical applications such as healthcare, where wearable devices might monitor heart rate and initiate alerts directly from the edge without the power and time expense of cloud communications. Edge AI can dynamically assess whether a task should be performed locally or offloaded to another edge or cloud server based on current network conditions and energy costs. Machine learning algorithms, such as RL, can continuously learn optimal offloading policies that balance energy usage and performance.

6.4 | Energy-Aware Resource Allocation

Edge AI can allocate resources (like processing power and bandwidth) based on real-time energy availability and network demands. For instance, in a low-power situation, an edge device may prioritize essential tasks and defer less critical ones to preserve energy, all guided by AI-driven policies. AI-driven scheduling algorithms at the edge can prioritize tasks to reduce energy-intensive operations. For example, in industrial IoT, certain sensors can delay non-urgent readings, conserving energy without compromising essential monitoring quality.

6.5 | Predictive Maintenance and Proactive Management

Edge AI algorithms can predict potential device failures by analyzing sensor data patterns. This proactive approach to maintenance reduces the energy cost of continuous monitoring. It enables predictive actions, such as replacing batteries or recalibrating sensors before they fail, thus optimizing energy usage and extending device lifetime. Edge AI can maintain energy efficiency across entire networks by monitoring energy levels and adapting operations based on device conditions. This proactive management is especially beneficial for networks with heterogeneous devices, where devices of different energy capacities must coexist and operate efficiently.

7 | Data Offloading Strategies for Edge Computing

Data offloading in edge computing is transferring data or computational tasks from IoT devices or edge nodes to nearby edge servers, fog nodes, or cloud servers to optimize resource usage, enhance performance, and reduce energy consumption [23]. Offloading decisions depend on factors such as network conditions, latency requirements, and energy constraints, making efficient strategies that balance these factors essential. Here's an overview of the various data offloading strategies and their role in improving edge computing IoT networks' performance and energy efficiency.

7.1 | Partial vs. Full Offloading

In full offloading, all processing tasks are transferred from the IoT device to an edge or cloud server. This strategy is ideal when the device has limited processing power, needs to conserve energy, or requires processing capabilities that exceed local resources. However, full offloading can increase network traffic and may not be suitable for latency-sensitive applications [24]. In Partial offloading, only selected task parts are offloaded, while some computations remain on the device. This is often achieved through partitioning algorithms, which divide tasks based on energy requirements, available resources, or latency needs. Partial offloading provides a balance by reducing device energy consumption without incurring excessive network delays [25].

7.2 | Opportunistic Offloading

opportunistic offloading considers current network conditions, such as bandwidth availability and network congestion, to decide the optimal times for offloading. For instance, offloading may be delayed if the network is busy, saving power and reducing the risk of data transmission failure [26]. In Device-to-Device (D2D) offloading, tasks are offloaded to nearby IoT devices with spare capacity rather than a central server. This approach helps in scenarios where cloud connectivity is limited or energy conservation by avoiding long-distance data transmission is needed [27].

7.3 | Latency-Aware Offloading

Proximity-based offloading, offloades tasks to the nearest available edge nodes. This approach benefits applications requiring immediate feedback, such as real-time video analytics or autonomous vehicle systems. By leveraging edge nodes close to the data source, latency is minimized, which is especially valuable for IoT applications with stringent delay constraints [28]. Hierarchical offloading in large IoT networks, can involve multiple layers, such as local edge devices, regional servers, and cloud centers. Each layer handles tasks based on their latency and processing requirements. For instance, low-latency tasks are offloaded to edge devices, while high-processing but delay-tolerant tasks go to the cloud. This layered approach optimizes latency and energy use across the network [29].

7.4 | Task-Based and Priority-Based Offloading

In Task-based offloading, tasks are classified based on processing needs, data size, and criticality. Resource-intensive tasks may be offloaded to nearby edge nodes or the cloud, while simpler tasks are handled locally. This strategy is often used in smart city environments, where tasks like real-time surveillance and traffic management have unique processing needs [30]. Decisions in priority-based offloading are made based on task priority, where critical tasks are prioritized for offloading to meet deadlines. Low-priority tasks may be delayed or processed locally when resources are limited. This approach works well in scenarios where certain tasks, like emergency alerts in healthcare, must take precedence over routine data collection [30].

7.5 | Energy-Aware Offloading

Energy-aware offloading algorithms consider the power levels of IoT devices and adjust offloading decisions to conserve battery life. For example, tasks may be offloaded more frequently when a device's battery is low, preserving energy for essential operations. Certain applications can tolerate minor delays, allowing for energy-efficient offloading strategies prioritizing low energy usage over speed. For instance, in environmental monitoring, data transmission may be delayed without significant impact, allowing devices to conserve power by avoiding frequent transmissions [31].

8 | Conclusion

In conclusion, energy-efficient communication in edge computing IoT networks is critical for enabling sustainable, scalable, and high-performing IoT applications. By addressing the energy demands of data processing, transmission, and device operations, edge computing, and advanced offloading strategies help conserve energy, reduce latency, and optimize resource allocation across diverse IoT networks. The integration of Edge AI has further enhanced energy efficiency by enabling real-time decision-making, adaptive task management, and localized data processing, reducing the dependency on cloud resources.

Effective data offloading strategies — including partial, opportunistic, and context-aware offloading — play a vital role in managing energy use by intelligently distributing computational loads based on device capabilities, network conditions, and application requirements. These strategies ensure that only essential data is offloaded, balancing the need for performance and responsiveness with energy constraints and supporting applications ranging from smart cities and healthcare to industrial IoT.

However, challenges remain around network heterogeneity, dynamic conditions, and device limitations. Future advancements in AI-driven energy management, federated learning, and adaptive multi-tier offloading architectures are poised to address these issues, paving the way for resilient, energy-efficient IoT networks. As technology advances, innovations like 6G networks, green IoT frameworks, and emerging computational models like quantum and neuromorphic computing offer promising pathways toward achieving sustainable, efficient, and intelligent edge-based IoT systems.

References

[1] Shobanadevi, A., Maragathm, G., Gangadharan, S. M. P., Soni, M., Kumar, R., Tran, T. A., & Singh, B. K. (2022). Internet of things-based data hiding scheme for wireless communication. *Wireless communications and mobile computing*, *2022*(1), 6997190. https://doi.org/10.1155/2022/6997190

[2] Xie, Z., Song, X., Cao, J., & Xu, S. (2022). Energy efficiency task scheduling for battery level-aware mobile edge computing in heterogeneous networks. *ETRI journal*, *44*(5), 746–758. https://doi.org/10.4218/etrij.2021-0312

[3] Mohapatra, H., Kolhar, M., & Dalai, A. K. (2024). Efficient energy management by using sjf scheduling in wireless sensor network. *International conference on advances in distributed computing and machine learning* (pp. 211–221). Springer. https://doi.org/10.1007/978-981-97-1841-2_15

[4] Mohapatra, H., Rath, A. K., Lenka, R. K., Nayak, R. K., & Tripathy, R. (2024). Topological localization approach for efficient energy management of WSN. *Evolutionary intelligence*, *17*(2), 717–727. https://doi.org/10.1007/s12065-021-00611-z

[5] Shao, S., Zhang, Q., Guo, S., Sun, L., Qiu, X., & Meng, L. (2022). Intelligent farm meets edge computing: energy-efficient solar insecticidal lamp management. *IEEE systems journal*, *16*(3), 3668–3678. https://doi.org/10.1109/JSYST.2022.3174925

[6] Raeisi-Varzaneh, M., Dakkak, O., Habbal, A., & Kim, B. S. (2023). Resource scheduling in edge computing: architecture, taxonomy, open issues and future research directions. *IEEE access*, *11*, 25329–25350. https://doi.org/10.1109/ACCESS.2023.3256522

[7] Ezenugu, I. A. (2020). Low-Energy Adaptive Clustering Hierarchy (Leach) Algorithm for selection of cluster head in wireless sensor nodes. *Journal of multidisciplinary engineering science and technology (JMEST)*, *7*(1), 2458–9403. www.jmest.org

[8] Chinnaiah, K., & Rao, K. N. (2023). Swarm energy efficient power efficient gathering in sensor information systems protocol in wireless sensor networks. *Indonesian journal of electrical engineering and computer science*, *31*(1), 459–469. https://doi.org/10.11591/ijeecs.v31.i1.pp459-469

[9] Teshome, D., Yakob, D., & Tsegaye, E. (2021). Performance analysis and enhancement of contention-based sensors medium access control protocol in wireless sensor networks. *International journal of computer networks and applications (IJCNA)*, *8*(5), 465–476. https://doi.org/10.22247/ijcna/2021/209981

[10] Dhaka, R. S., Khurana, M., & Lamba, S. (2024). Comparison of sensor-mac (s-mac), timeout-mac (t-mac), berkeley-mac (b-mac), and modified ml-mac (m2l-mac) for wireless sensor networks. *Lecture notes in networks and systems* (Vol. 1001 LNNS, pp. 454–471). Springer. https://doi.org/10.1007/978-3-031-60935-0_41

[11] Li, L., Fang, Y., Liu, L., Peng, H., Kurths, J., & Yang, Y. (2020). Overview of compressed sensing: sensing model, reconstruction algorithm, and its applications. *Applied sciences*, *10*(17), 5909. https://doi.org/10.3390/app10175909

[12] Feng, A., Dong, D., Lei, F., Ma, J., Yu, E., & Wang, R. (2023). In-network aggregation for data center networks: A survey. *Computer communications*, *198*, 63–76. https://doi.org/10.1016/j.comcom.2022.11.004

[13] Sangaiah, A. K., Rostami, A. S., Hosseinabadi, A. A. R., Shareh, M. B., Javadpour, A., Bargh, S. H., & Hassan, M. M. (2021). Energy-aware geographic routing for real-time workforce monitoring in industrial informatics. *IEEE internet of things journal*, *8*(12), 9753–9762. https://doi.org/10.1109/JIOT.2021.3056419

[14] Shovon, I. I., & Shin, S. (2022). Survey on multi-path routing protocols of underwater wireless sensor networks: advancement and applications. *Electronics*, *11*(21), 3467. https://doi.org/10.3390/electronics11213467

[15] Han, B., Ran, F., Li, J., Yan, L., Shen, H., & Li, A. (2022). A novel adaptive cluster based routing protocol for energy-harvesting wireless sensor networks. *Sensors*, *22*(4), 1564. https://doi.org/10.3390/s22041564

[16] Saikia, P., Singh, K., Huang, W. J., & Duong, T. Q. (2024). Hybrid deep reinforcement learning for enhancing localization and communication efficiency in RIS-aided cooperative ISAC systems. *IEEE internet of things journal*, *11*(18), 29494–29510. https://doi.org/10.1109/JIOT.2024.3411158

[17] Mohammed, A. S., Saddi, V. R., Gopal, S. K., Jiwani, N., & Logeshwaran, J. (2024). Dynamic scheduling algorithms for serverless computing solutions in the cloud. *2024 international conference on e-mobility, power control and smart systems (ICEMPS)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICEMPS60684.2024.10559356

[18] Nakabi, T. A., & Toivanen, P. (2021). Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable energy, grids and networks*, *25*, 100413. https://doi.org/10.1016/j.segan.2020.100413

[19] Hemalatha, K., Likitha, L. G., Sireesha, N., & Usharani, A. (2023). An IoT based interface for the smart watering of plants using soil moisture sensor. *IRO journal on sustainable wireless systems*, *5*(2), 175–182. https://doi.org/10.36548/jsws.2023.2.008

[20] Abdel-Gawad, M., Usama, M., Hesham, H., Ibrahim, O., & Abdellatif, M. M. (2022). Remote healthcare monitoring using wearable iot devices and cloud services. *2022 5th conference on cloud and internet of things (CIOT)* (pp. 108–113). IEEE. https://doi.org/ 10.1109/CIoT53061.2022.9766591

[21] Shabbir, A., Cheema, A. N., Ullah, I., Almanjahie, I. M., & Alshahrani, F. (2024). Smart city traffic management: acoustic-based vehicle detection using stacking-based ensemble deep learning approach. *IEEE access*, *12*, 35947–35956. https://doi.org/10.1109/ACCESS.2024.3370867

[22] Hadi, R. H., Hady, H. N., Hasan, A. M., Al-Jodah, A., & Humaidi, A. J. (2023). Improved fault classification for predictive maintenance in industrial IoT based on AutoML: A case study of ball-bearing faults. *Processes*, *11*(5), 1507. https://doi.org/10.3390/pr11051507

[23] Okafor, N. U., Alghorani, Y., & Delaney, D. T. (2020). Improving data quality of low-cost IoT sensors in environmental monitoring networks using data fusion and machine learning approach. *ICT express*, *6*(3), 220–228. https://doi.org/10.1016/j.icte.2020.06.004

[24] Shi, Y., Yang, K., Jiang, T., Zhang, J., & Letaief, K. B. (2020). Communication-efficient edge AI: algorithms and systems. *IEEE communications surveys & tutorials*, *22*(4), 2167–2191. https://doi.org/10.1109/COMST.2020.3007787

[25] Mitsis, G., Tsiropoulou, E. E., & Papavassiliou, S. (2022). Price and risk awareness for data offloading decision-making in edge computing systems. *IEEE systems journal*, *16*(4), 6546–6557. https://doi.org/10.1109/JSYST.2022.3188997

[26] Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y., & Guo, L. (2022). Computation offloading in mobile edge computing networks: A survey. *Journal of network and computer applications*, *202*, 103366. https://doi.org/10.1016/j.jnca.2022.103366

[27] Zhang, S., Gu, H., Chi, K., Huang, L., Yu, K., & Mumtaz, S. (2022). DRL-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network. *IEEE transactions on wireless communications*, *21*(12), 10934–10948. https://doi.org/10.1109/TWC.2022.3188302

[28] Kyung, Y., Kim, E., & Song, T. (2024). Opportunistic offloading scheme for content delivery service using electro-mobility networks. *IET intelligent transport systems*, *18*(4), 591–598. https://doi.org/10.1049/itr2.12255

[29] Ko, H., & Pack, S. (2020). Distributed device-to-device offloading system: design and performance optimization. *IEEE transactions on mobile computing*, *20*(10), 2949–2960. https://doi.org/10.1109/TMC.2020.2994138

[30] Babaghayou, M., Chaib, N., Maglaras, L., Yigit, Y., Ferrag, M. A., & Marsh, C. (2024). Proximity-driven, load-balancing task offloading algorithm for enhanced performance in satellite-enabled mist computing. *Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, lnicst* (Vol. 527 LNICST, pp. 29–44). Springer. https://doi.org/10.1007/978-3-031-58053-6_3

[31] Guo, K., Sheng, M., Tang, J., Quek, T. Q. S., & Qiu, Z. (2020). Hierarchical offloading for delay-constrained applications in fog RAN. *IEEE transactions on vehicular technology*, *69*(4), 4257–4270. https://doi.org/10.1109/TVT.2020.2972734

[32] Zhang, S., Yi, N., & Ma, Y. (2024). A survey of computation offloading with task types. *IEEE transactions on intelligent transportation systems*, *25*(8), 8313–8333. https://doi.org/10.1109/TITS.2024.3410896

[33] Hao, Y., Cao, J., Wang, Q., & Ma, T. (2021). Energy-aware offloading based on priority in mobile cloud computing. *Sustainable computing: informatics and systems*, *31*, 100563. https://doi.org/10.1016/j.suscom.2021.100563