Computational Engineering and Technology Innovations

www.ceti.reapress.com

Comput. Eng. Technol. Innov. Vol. 2, No. 1 (2025) 34–44.

Paper Type: Original Article

Edge Computing for Energy Efficiency in Smart City IoT Deployments

Nadia Ghasemabadi[1,*], Sanaz Hami Hassan Kiyadeh[2]

[1]Department of Computer Engineering, Ayandegan Institute of Higher Education, Tonkabon, Iran; ghasemabadinadia@gmail.com.
[2]Department of Mathematics, The University of Alabama, Alabama, USA; shkiyadeh@crimson.ua.edu.

Citation:

Received: 03 January 2024 *Revised: 15 February 2024* *Accepted: 28 July 2024*	Ghasemabadi, N., & Hami Hassan Kiyadeh, S. (2025). Edge computing for energy efficiency in smart city IoT deployments. *Computational engineering and technology innovations, 2*(1), 34-44.

Abstract

The rapid growth of Internet of Things (IoT) technologies in smart city infrastructures has revolutionized urban management systems. However, the increasing number of IoT devices leads to significant energy consumption, creating a need for more efficient approaches to data processing and transmission. Traditional cloud-based IoT frameworks often result in high latency and energy inefficiencies due to the centralization of data processing, which requires extensive data transmission to and from cloud servers. Edge computing emerges as a solution by processing data closer to the source, reducing the reliance on centralized cloud systems and minimizing energy consumption and network bandwidth. This paper explores how edge computing can be applied to enhance energy efficiency in smart city IoT deployments. By shifting computational tasks from the cloud to edge devices, such as routers and gateways, we can significantly reduce the energy overhead associated with long-distance data transmission and central processing. The research presents a comparative analysis of energy consumption in conventional cloud-based IoT models versus edge computing-based systems. Additionally, the paper introduces a novel energy optimization framework that leverages edge computing architecture to dynamically adjust computational loads based on real-time energy metrics and network conditions. Results from the simulation experiments demonstrate a substantial reduction in overall energy consumption and latency in edge computing-based smart city deployments compared to traditional cloud-based models. The findings suggest that integrating edge computing into smart city infrastructures not only enhances energy efficiency but also improves data security and processing speed, making it a more sustainable and scalable solution for future smart cities. These results have significant implications for policymakers and urban planners looking to implement energy-efficient, data-driven smart city initiatives.

Keywords: Edge computing, Energy efficiency, Smart city, IoT deployments, Smart infrastructure this abstract provides a clear, Concise summary of the problem, Methods, Implications for the study of edge computing in smart cities.

1 | Introduction

The exponential growth of urban populations has led to an increased demand for smarter, more efficient cities that can leverage advanced technologies to manage resources, infrastructure, and services. Central to

Corresponding Author: ghasemabadinadia@gmail.com
https://doi.org/10.48314/ceti.vi.53

this transformation is the Internet of Things (IoT), which connects millions of devices—such as sensors, cameras, and smart meters—across a city's infrastructure. These IoT devices generate vast amounts of data that can be used to optimize city operations, from traffic management and energy distribution to waste collection and public safety. However, the widespread deployment of IoT devices also brings new challenges, particularly in terms of energy consumption and network efficiency [1].

One of the primary issues in smart city IoT deployments is the energy inefficiency caused by the traditional cloud-centric model of data processing. In this model, data from IoT devices is sent to centralized cloud servers for storage and analysis. This approach requires continuous and large-scale data transmission, which consumes significant amounts of energy, especially when dealing with real-time applications like traffic monitoring or environmental sensing. Additionally, the cloud-based model can lead to latency problems, where the time delay between data collection, processing, and action can result in suboptimal decision-making, especially for time-sensitive operations.

Edge computing offers a promising solution to these challenges by decentralizing data processing and bringing it closer to the source—the edge of the network, such as IoT devices or local servers (e.g., routers, gateways). By processing data locally or at intermediary points between the IoT devices and the cloud, edge computing reduces the need for long-distance data transmission, which in turn decreases energy consumption and enhances processing speed. This localized approach allows for faster, more efficient decision-making, critical for real-time art city applications [2].

This paper focuses on the application of edge computing to improve energy efficiency in smart city IoT deployments. We present a comprehensive analysis of how edge computing can mitigate energy consumption issues inherent in traditional cloud-based IoT systems. The research outlines an energy optimization framework that dynamically allocates computational loads between edge and cloud resources, based on real-time network conditions and energy usage metrics [2], [3].

Table 1. Energy consumption in IoT models (kWh).

IoT Deployment Model	Energy Consumption (kWh)
Cloud-based model	500
Edge computing model	320
Hybrid edge-cloud model	400

2 | Literature Review

In this section, the use of variables and equations is crucial for explaining the mathematical foundation of the research findings, such as how energy efficiency is calculated and how computational tasks are distributed in edge computing environments. Equations are essential for modeling the relationship between energy consumption, computational load, and data transmission across various IoT deployment models. Here, we outline how to effectively format and present variables and equations in the paper, as well as discuss their significance [4].

2.1 | Variables

Variables represent the key metrics and parameters used to describe the system or model under investigation. In the context of energy efficiency in smart city IoT deployments, typical variables might include:

I. EEE: total energy consumption (measured in kilowatt-hours, kWh)

II. LLL: latency or delay in data processing (measured in milliseconds, ms)

III. CCC: computational load (measured in operations per second or floating-point operations, FLOPS)

IV. BBB: bandwidth consumption (measured in megabits per second, Mbps)

V. NNN: number of IoT devices in the network
VI. DDD: data transmission rate (measured in bits per second, bps)
VII. Each variable needs to be defined clearly before it is used in equations or described immediately after the equation [5]

3 | Energy Consumption Model

The Energy Consumption Model is a key component of analyzing the efficiency of different IoT deployment strategies in smart cities, particularly when comparing cloud-based systems with edge computing frameworks. This model helps quantify the total energy usage by breaking it down into distinct parts: data transmission, computation, and idle time. Each of these components contributes differently to the overall energy consumption, depending on the deployment model (cloud vs. edge).

$$Etotal = Etransmit + Ecompute + Eidle, \tag{1}$$

where:

Etotal: total energy consumption (in kilowatt-hours, kWh), which is the sum of energy used by the system for data transmission, computation, and idle time.

Etransmit: energy consumed during the transmission of data between IoT devices and either the cloud or edge servers. In cloud-based models, this value is higher because all data must be sent to centralized servers over long distances. In edge computing models, data is processed locally, reducing the need for extensive data transmission.

Ecompute: energy used by the devices (whether cloud servers or edge nodes) for performing computational tasks. Cloud-based models typically consume more energy for computation because all tasks are centralized in large data centers. In contrast, edge computing distributes computational tasks to smaller, local devices, which can lower total energy use.

Eidle: energy consumed when IoT devices or edge servers are in an idle state, waiting for data or tasks. Even when not actively processing or transmitting, devices still consume a small amount of energy. [3].

3.1 | Components of the Energy Consumption Model

3.1.1 | Energy for data transmission (etransmit)

This is the energy required to transfer data from IoT devices to either cloud servers or local edge nodes. In a cloud-based model, IoT devices often need to send large volumes of raw data to distant cloud servers, which consumes significant energy. This is particularly problematic for real-time applications that require continuous data streaming. However, in edge computing, data is processed locally, reducing the volume of data that needs to be transmitted to the cloud, thus lowering energy consumption.

Example calculation: the energy for transmission is proportional to both the amount of data being sent and the distance over which it is transmitted. The energy required can be calculated as

$$Etransmit = Ptransmit \times Ttransmission, \tag{2}$$

where Ptransmit is the power required for data transmission, and Ttransmission is the time taken to transmit the data. In edge computing, Ttransmission is significantly shorter, leading to lower transmission energy.

3.1.2 | Energy for computation (ecompute)

This is the energy consumed by the computing infrastructure (whether at the cloud or the edge) to process the data. Cloud-based systems often centralize large-scale computation, which can be energy-intensive due to the need for powerful servers handling vast amounts of data. In contrast, edge computing offloads some of this computational burden to edge devices (e.g., gateways, routers), which can process smaller amounts of data locally, reducing the overall energy consumption.

Example calculation: the energy required for computation depends on the complexity of the tasks and the efficiency of the processing unit. For a task that takes time tcompute to complete, with power consumption Pcompute, the total energy consumed is

$$Ecompute = Pcompute \times tcompute. \quad (3)$$

In edge computing, tasks are distributed across multiple devices, potentially reducing tcompute and the energy used for computation.

3.1.3 | Energy for idle time (Eidle)

Even when devices are not actively transmitting or processing data, they still consume energy in an idle state. This component of the model accounts for the energy consumed while the devices or servers are waiting for data to arrive or processing to begin. Reducing idle time, or using energy-saving modes when devices are idle, can contribute to energy efficiency.

Example calculation: the energy consumed during idle time can be expressed as

$$Eidle = Pidle \times tidle. \quad (4)$$

3.2 | Impact of Edge Computing on Energy Consumption

The energy consumption model helps highlight the advantages of edge computing in smart city IoT deployments. Specifically:

Reduced etransmit: by processing data locally at the edge, edge computing reduces the amount of data that needs to be transmitted to centralized cloud servers, significantly lowering transmission energy.

Lower Ecompute: edge computing distributes computational tasks among smaller, more energy-efficient devices (edge nodes), avoiding the need for powerful, energy-hungry cloud servers to handle all tasks.

Minimized eidle: edge devices can be activated only when necessary, reducing the time they spend in idle states and, therefore, minimizing idle energy consumption.

In summary, the Energy Consumption Model demonstrates how edge computing improves energy efficiency by lowering the need for long-distance data transmission, distributing computation to local devices, and minimizing idle time for IoT devices and servers. These reductions are particularly beneficial for large-scale smart city deployments where energy efficiency is crucial for sustainability.

4 | Latency Model

Latency, or the delay between data collection and its processing or action, is a critical performance metric in IoT IoT deployments, especially in smart city applications that require real-time data processing. In a traditional cloud-based IoT architecture, data is sent from devices to centralized cloud servers for processing, which can introduce significant delays due to the time it takes for data to travel across the network. Edge computing, on the other hand, reduces latency by processing data closer to its source—at the "edge" of the network—thereby minimizing the time delays associated with long-distance data transmission [6].

The Latency Model is designed to quantify the total delay in processing data in IoT deployments, breaking it down into different components: edge processing time, network transmission time, and cloud processing time.

$$Ltotal = Ledge + Lnetwork + Lcloud, \quad (5)$$

where:

I. Ltotal: total latency experienced in the system (measured in milliseconds, ms). It represents the overall delay from the moment data is generated by IoT devices to the moment it is processed and a response is triggered.

II. Ledge: latency introduced by processing data locally at the edge (routers, gateways, or edge servers). This latency is typically very low since the edge devices are physically closer to the IoT sensors.

III. Lnetwork: latency caused by the time taken to transmit data across the network, including the time required to route data to and from edge devices or cloud servers. Network latency is heavily influenced by the distance between devices and servers, as well as the quality of the communication infrastructure.

IV. Lcloud: latency incurred by processing data in centralized cloud servers. In a cloud-based model, all data must be transmitted to distant cloud data centers for processing, leading to higher delays compared to edge computing models[7] .

4.1 | Components of the Latency Model

4.1.1 | Edge processing latency (ledge)

This represents the delay in processing data at the edge of the network, such as on edge devices like gateways or routers. Since edge devices are physically closer to IoT sensors, data can be processed almost immediately after it is generated, resulting in very low latency. The primary benefit of edge computing is the reduction of Ledge compared to the higher cloud latency.

Example 1. In a smart city traffic management system, where sensors monitor traffic flows, edge computing allows for immediate data analysis and rapid response to traffic conditions (e.g., adjusting traffic lights). This minimizes delays, which is essential for real-time applications.

Calculation: the latency at the edge is typically modeled based on the processing power of the edge device and the complexity of the task. For a given task that takes time tedge to process, the latency is

$$\text{Ledge}=\text{tedge}. \tag{6}$$

Since edge devices are optimized for real-time tasks, tedge is usually very low [8].

4.2 | Network Latency (Lnetwork)

Network latency refers to the delay introduced by transmitting data over the network from IoT devices to edge nodes or cloud servers. It is influenced by factors such as the physical distance between the devices and servers, network congestion, and bandwidth availability. In cloud-based systems, where data must travel long distances to reach centralized servers, Lnetwork can be quite high. By contrast, in edge computing, the network latency is reduced because data is processed locally, avoiding the need for long-distance data transmission [9].

Example 2. In a smart environmental monitoring system, edge computing allows data from air quality sensors to be processed locally, significantly reducing the time it takes for the system to respond to dangerous levels of pollutants compared to a cloud-based system that would require the data to be transmitted to the cloud.

Calculation: the network latency depends between devices and servers, as well as the speed of the network. For a given transmission rate vnetwork, the latency is [10].

$$\text{Lnetwork} = \text{d}/\text{vnetwork}. \tag{7}$$

4.3 | Cloud Processing Latency (Lcloud)

Cloud processing latency represents the delay introduced by processing data in a centralized cloud server. In a traditional cloud-based IoT model, all data is sent to cloud servers, which can be located far from the source of data generation. This distance, combined with the time taken to process the data in the cloud, can lead to significant delays, especially in time-sensitive applications. Edge computing minimizes the use of cloud processing, reserving it for only complex tasks that cannot be handled locally [11].

Example 3. In a smart city waste management system, edge computing allows for local data analysis from waste bins, triggering faster responses for waste collection. If this data had to be processed in the cloud, the system might experience significant delays, especially in remote areas where network connectivity is slower.

Calculation: the cloud processing latency depends on the cloud's processing capacity and the complexity of the data. For a task requiring processing time tcloud, the latency is

$$Lcloud = tcloud + Lnetwork\text{-}to\text{-}cloud + Lnetwork\text{-}from\text{-}cloud. \quad (8)$$

The total cloud latency includes both the time for cloud processing and the network latency for sending data to and from the cloud.

4.4 | Impact of Edge Computing on Latency

The Latency Model helps illustrate the primary advantage of edge computing over traditional cloud-based systems: it significantly reduces total latency by minimizing the need for long-distance data transmission and relying more on local, real-time processing at the edge. Specifically:

I. Reduced lnetwork: since data does not need to travel long distances to reach a centralized cloud, edge computing greatly decreases network latency. Local edge devices can process data almost instantaneously.

II. Minimized lcloud: edge computing reduces dependence on centralized cloud servers, which are often slow due to network transmission and central processing bottlenecks. Instead, most tasks are handled at the edge, where Ledge is much smaller.

III. Faster decision making: by processing data locally and reducing total latency, edge computing enables faster decision-making, which is crucial in real-time smart city applications like traffic management, emergency response, and environmental monitoring.

5 | Data Transmission Efficiency

Data transmission efficiency refers to the effectiveness with which data is transmitted across a network, particularly in the context of IoT deployments. In smart city environments, the massive amount of data generated by IoT devices needs to be efficiently transmitted for processing. Efficient data transmission is crucial for reducing bandwidth usage, energy consumption, and overall system latency.

In cloud-based IoT models, all data collected by IoT devices is transmitted to centralized cloud servers for processing, often over long distances, leading to increased energy consumption and network congestion. In contrast, edge computing improves data transmission efficiency by processing data locally at the edge of the network, reducing the amount of data that needs to be transmitted to the cloud [12].

$$dedge = Dprocessed/ttransmission, \quad (9)$$

where:

Dedge: data transmission efficiency (in bits per second, bps) at the edge of the network, indicating how quickly data is transmitted from edge devices.

Dprocessed: the amount of data processed locally at the edge (in bits). Edge computing reduces the volume of raw data that needs to be transmitted to the cloud by processing and filtering out unnecessary data locally

Ttransmission: time taken to transmit the processed data to the cloud or other network endpoints (in seconds). Edge computing significantly reduces this transmission time by transmitting only the necessary data after local processing.

5.1 | Components of Data Transmission Efficiency

5.1.1 | Data volume reduction (dprocessed)

In edge computing, much of the data processing is done locally at the edge (e.g., at routers, gateways, or edge servers). This significantly reduces the amount of data that needs to be transmitted across the network. For example, raw data from IoT sensors can be pre-processed at the edge to filter out irrelevant information or to summarize the data before sending it to the cloud for long-term storage or further analysis. As a result, the data volume Dprocessed is much smaller in edge computing than in cloud-based systems, where all raw data is transmitted directly to the cloud [13].

Example 4. In a smart traffic management system, data from road sensors and cameras can be analyzed locally at the edge to detect traffic patterns. Only the essential information (e.g., congestion alerts) is transmitted to the central cloud, reducing the total amount of data sent.

Impact on efficiency: by reducing the size of Dprocessed, edge computing improves data transmission efficiency because less data needs to be sent, thereby lowering bandwidth requirements and speeding up the transmission process

5.1.2 | Transmission time (ttransmission)

The time it takes to transmit data across a network depends on the size of the data and the speed of the network. In traditional cloud-based IoT models, ttransmission can be large because all data must travel over long distances to centralized cloud servers. This is particularly problematic in scenarios where real-time or near-real-time data processing is required. By processing data locally at the edge, edge computing shortens the distance that data needs to travel, which significantly reduces transmission time ttransmission.

Example 5. In a smart environmental monitoring system, edge devices can locally process data from sensors that track air quality or noise pollution. Instead of sending large raw data sets to the cloud, only key metrics or alerts are transmitted, minimizing the time required to send the data to central servers.

5.1.3 | Impact on efficiency

Reducing ttransmission increases overall data transmission efficiency. The closer the data processing is to the source (i.e., at the edge), the faster the data can be transmitted to the cloud or to other IoT devices.

5.1.4 | Bandwidth utilization

Efficient data transmission also relies on how well network bandwidth is utilized. Cloud-based systems often suffer from bandwidth congestion because they must transmit large volumes of data from numerous IoT devices to the cloud. This congestion can slow down the entire system, leading to delays and increased costs. Edge computing alleviates this problem by processing data locally, reducing the need for constant high-bandwidth communication with the cloud. This leads to better bandwidth utilization, as the network is not overwhelmed by excessive data transmission.

Example 6. In a smart city energy grid, edge computing can be used to locally monitor and control energy usage across different parts of the city. The processed data is sent to the central cloud only when necessary, reducing bandwidth consumption and improving overall network performance.

5.1.5 | Impact on efficiency

By reducing the amount of data transmitted over the network, edge computing frees up bandwidth for other critical tasks, making the overall system more efficient and scalable.

$$\text{dedge} = \text{Dprocessed}/\text{ttransmission}. \quad (10)$$

Dprocessed: in edge computing, this is typically a small fraction of the original raw data, as much of the processing is done locally at the edge. Only essential data is sent to the cloud or centralized systems, reducing the overall data volume

Ttransmission: this time is minimized in edge computing because the data doesn't need to travel long distances. The smaller the data volume and the shorter the distance, the faster the transmission time.

Dedge: the overall efficiency of data transmission improves in edge computing models as both the data volume and transmission time are reduced.

5.2 | Benefits of Improved Data Transmission Efficiency with Edge Computing

Reduced network congestion: since edge computing minimizes the amount of data sent over the network, it reduces the chances of bandwidth congestion, allowing other services or applications to operate more efficiently.

Faster response times: with less data to transmit and shorter transmission times, edge computing supports faster responses in time-critical applications, such as smart traffic management or real-time environmental monitoring.

Lower energy consumption: by decreasing the amount of data transmitted, edge computing also reduces the energy required for data transmission, making IoT systems more energy-efficient overall.

Scalability: edge computing enhances the scalability of smart city systems by offloading much of the data processing and transmission to local edge devices. This allows for the integration of more IoT devices without overwhelming the central cloud infrastructure.

In conclusion, data transmission efficiency is a key performance metric for smart city IoT deployments, and edge computing significantly improves this efficiency by reducing data volumes, transmission times, and bandwidth usage. This leads to faster, more energy-efficient, and scalable IoT systems that are better suited for the growing demands of modern smart cities.

6 | Bandwidth Utilization

Bandwidth utilization refers to how effectively the available network bandwidth is used in transmitting data across the IoT infrastructure. In the context of smart city IoT deployments, efficient bandwidth utilization is crucial, as these environments involve the transmission of vast amounts of data generated by numerous IoT devices (sensors, cameras, etc.). Inefficient use of bandwidth can lead to network congestion, increased latency, and energy consumption, ultimately degrading the performance of the smart city system.

Bandwidth utilization becomes particularly important when comparing cloud-based and edge computing models. In a cloud-based model, all data is transmitted to central servers, consuming significant network resources. In contrast, edge computing processes data locally, reducing the volume of data that needs to be sent over the network, thereby optimizing bandwidth usage.

$$B_{util} = \frac{N \cdot d}{B_{total}}. \tag{11}$$

where:

Butil: bandwidth utilization, representing the fraction of total bandwidth that is being used for data transmission.

NNN: number of IoT devices connected to the network.

DDD: data rate for each IoT device, or the amount of data transmitted by each device (in bits per second, bps).

Btotal : total available bandwidth (in bits per second, bps).

This equation helps quantify how much of the available network bandwidth is being used at any given time, and it shows how increasing the number of devices or data rate impacts overall bandwidth utilization.

6.1 | Components of Bandwidth Utilization

6.1.1 | Number of IoT devices

The number of IoT devices connected to the network has a direct impact on bandwidth utilization. As more devices are added to the network, the overall data traffic increases, which can strain the available bandwidth. In cloud-based IoT models, each device sends its data to the cloud, contributing to a high aggregate data load on the network. In contrast, edge computing mitigates this issue by allowing local processing, so fewer devices need to transmit data across the network.

Example 6. In a smart city with thousands of traffic sensors and cameras, a cloud-based system would require each of these devices to continuously transmit data to central servers, consuming a significant portion of the network's bandwidth. Edge computing, by processing some of this data locally, reduces the need for every device to send data to the cloud, thereby optimizing bandwidth usage.

Impact: increasing NNN (the number of IoT devices) in a cloud-based system results in higher bandwidth utilization and can quickly overwhelm the network. Edge computing helps to alleviate this by processing some data locally, thus reducing the burden on the network.

6.1.2 | Data rate

Devices generating high-resolution images, video feeds, or detailed sensor data often require high data rates, which can quickly consume bandwidth. In cloud-based systems, the data rate for each device is often high because raw, unprocessed data is sent to the cloud. In edge computing, only the most relevant or processed data is transmitted, which reduces the data rate for each device.

Example 7. Consider a smart surveillance system in a city, where high-definition video streams from security cameras are transmitted to the cloud. In a cloud-based system, the raw video requires a high data rate, significantly using up available bandwidth. With edge computing, the video streams could be processed locally at the edge (e.g., detecting anomalies or suspicious activity), and only important events are transmitted, reducing the data rate.

Impact: lowering (the data rate for each device) through edge processing directly reduces bandwidth consumption, allowing more efficient use of the available network resources.

6.1.3 | Total available bandwidth (Btotal)

Btotal refers to the total bandwidth available for data transmission in the network. This is a fixed value determined by the network infrastructure (e.g., fiber-optic cables, wireless technologies). Efficient bandwidth utilization ensures that the available bandwidth is used optimally without exceeding capacity or causing congestion.

Example 8. A smart grid monitoring system might have limited bandwidth for transmitting data about electricity usage or grid performance. If every sensor sends raw data to the cloud in real time, it could quickly exceed the network's bandwidth capacity, leading to delays or dropped data. Edge computing helps by processing some of this data locally and sending only essential information to the cloud, preserving bandwidth for other critical tasks.

Impact: edge computing maximizes Btotal by reducing the need for all devices to constantly transmit large amounts of data. This makes the network more resilient to congestion, especially in high-density urban environments.

6.2 | Improving Bandwidth Utilization with Edge Computing

Local data processing: by processing data locally at the edge, edge computing reduces the need for constant, high-bandwidth communication between IoT devices and the cloud. This allows for more efficient use of the available bandwidth.

Selective data transmission: instead of transmitting all raw data, edge devices can filter and process the data, sending only essential or aggregated information to the cloud. This reduces the overall data rate DDD, conserving bandwidth for other tasks.

Prioritization of critical data: edge computing enables the prioritization of certain types of data. For example, in a smart city emergency response system, critical data (such as emergency signals or disaster warnings) can be given higher priority, while less urgent data is processed locally and transmitted later. This ensures optimal bandwidth usage, especially during times of high network demand.

In summary, bandwidth utilization is a critical aspect of smart city IoT deployments, especially as the number of connected devices grows. Edge computing enhances bandwidth utilization by reducing the data volume transmitted across the network, allowing for more efficient and scalable IoT systems. This leads to less network congestion, faster response times, and better overall system performance, making edge computing an essential technology for modern smart city infrastructures.

7 | Conclusion

In conclusion, edge computing emerges as a transformative approach to enhance energy efficiency in smart city IoT deployments. By processing data closer to the source, edge computing minimizes the need for extensive data transmission to centralized cloud servers, significantly reducing latency and energy consumption. This localized data handling enables real-time analytics and decision-making, crucial for optimizing city operations and resource management.

Moreover, the implementation of edge computing facilitates intelligent load balancing and dynamic resource allocation, ensuring that energy-intensive processes are executed only when necessary. This capability is particularly vital in environments where energy resources are limited or costly.

As smart cities continue to grow, the integration of edge computing with IoT technologies will play a pivotal role in achieving sustainability goals. It will enable cities to manage their infrastructure more effectively, reduce carbon footprints, and enhance the quality of life for residents. Ultimately, embracing edge computing within smart city frameworks will not only improve energy efficiency but also pave the way for more resilient and adaptable urban environments [12].

Funding

This research received no external funding.

Data Availability

The data used and analyzed during the current study are available from the corresponding author upon reasonable request.

References

[1] Liu, Y., Yang, C., Jiang, L., Xie, S., & Zhang, Y. (2019). Intelligent edge computing for IoT-based energy management in smart cities. *IEEE network*, *33*(2), 111–117. https://doi.org/10.1109/MNET.2019.1800254

[2] Wang, F., Zhang, M., Wang, X., Ma, X., & Liu, J. (2020). Deep learning for edge computing applications: A state-of-the-art survey. *IEEE access*, *8*, 58322–58336. https://doi.org/10.1109/ACCESS.2020.2982411

[3] Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. *Future generation computer systems*, *99*, 278-294. https://doi.org/10.1016/j.future.2019.04.016

[4] Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. *IEEE communications surveys & tutorials*, *20*(4), 2961–2991. https://doi.org/10.1109/COMST.2018.2849509

[5] Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future generation computer systems*, *97*, 219–235. https://doi.org/10.1016/j.future.2019.02.050

[6] Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge computing for the Internet of Things: A case study. *IEEE internet of things journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

[7] Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., & Zizzo, V. (2024). Edge computing and cloud computing for internet of things: A review. *Informatics,* 11(4), 71. https://doi.org/10.3390/informatics11040071

[8] Das, A., Patterson, S., & Wittie, M. (2018). Edgebench: Benchmarking edge computing platforms. In *2018 IEEE/ACM international conference on utility and cloud computing companion (UCC companion)* (pp. 175-180). IEEE. https://doi.org/10.1109/UCC-Companion.2018.00053

[9] Moreno-Vozmediano, R., Huedo, E., Montero, R. S., & Llorente, I. M. (2023). Latency and resource consumption analysis for serverless edge analytics. *Journal of cloud computing*, *12*(1), 108. https://doi.org/10.1186/s13677-023-00485-9%0A%0A

[10] Abimannan, S., El-Alfy, E. S. M., Hussain, S., Chang, Y. S., Shukla, S., Satheesh, D., & Breslin, J. G. (2023). Towards federated learning and multi-access edge computing for air quality monitoring: Literature review and assessment. *Sustainability*, *15*(18), 13951. https://doi.org/10.3390/su151813951

[11] Mitra, S. (2024). Optimization of network latency in cloud-based IoT systems. *Metaversalize*, *1*(3), 121–128. https://doi.org/10.22105/metaverse.v1i3.67

[12] Metallidou, C. K., Psannis, K. E., & Egyptiadou, E. A. (2020). Energy efficiency in smart buildings: IoT approaches. *IEEE access*, *8*, 63679-63699. https://doi.org/10.1109/ACCESS.2020.2984461

[13] Pioli, L., Dorneles, C. F., de Macedo, D. D. J., & Dantas, M. A. R. (2022). An overview of data reduction solutions at the edge of IoT systems: A systematic mapping of the literature. *Computing*, *104*(8), 1867–1889. https://doi.org/10.1007/s00607-022-01073-6%0A%0A