VOL. 12. NO. 2 NOVEMBER 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i2.8325

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

COMPARATIVE LATENCY ANALYSIS OF EDGE AND CLOUD ARCHITECTURES FOR IOT-BASED WATERLOGGING DETECTION

Deosa Putra Caniago[1*], Hetty Rohayani[2], Nova Agustina[3], Arip Kristiyanto[4], Najirah Umar[5], Muhammad Dzikri Al Farizi[1], Yopy Mardiansyah[1]

Institut Teknologi Batam, Batam, Indonesia[1]
https://iteba.ac.id[1]
deosa@iteba.ac.id*, 2422044@iteba.ac.id, yopy@iteba.ac.id

Universitas Muhammadiyah Jambi, Jambi, Indonesia[2]
https://umjambi.ac.id/[2]
hettyrohayani@gmail.com

Universitas Teknologi Bandung, Bandung, Indonesia[3]
https://utb-univ.id/[3]
nova@utb-univ.ac.id

Universitas Pamulang, Serang, Indonesia[4]
https://unpam.ac.id/[4]
dosen10027@unpam.ac.id

Universitas Handayani Makassar, Makassar, Indonesia[5]
https://handayani.ac.id/[5]
najirah@handayani.ac.id

The creation is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License

Submitted: 26 February 2026, **Revised:** 10 November 2026, **Accepted:** 16 November 2026, **Published:** 20 November 2026.

Abstract— *Urban waterlogging detection systems increasingly rely on Internet of Things (IoT) technology for real-time environmental monitoring. However, most existing implementations adopt cloud-centric architectures in which threshold evaluation and alert triggering are executed on remote servers, potentially introducing significant response latency. This study aims to evaluate whether relocating computational logic from the cloud layer to the edge device can significantly reduce end-to-end response time in a waterlogging detection framework. A comparative experimental design was conducted using identical hardware configurations under two architectural scenarios: cloud-centric and edge-based processing. Latency was measured as the time interval between threshold detection and actuator activation across 50 repeated trials for each scenario. Statistical analysis was performed using mean, standard deviation, and independent sample t-test at a significance level of 0.05. The results indicate that the cloud-centric architecture produced an average latency of 7,925 ms with a standard deviation of 710.2 ms, whereas the edge-based architecture achieved an average latency of 1,745 ms with a standard deviation of 195.8 ms. The latency reduction reached approximately 77.98%, and the difference was statistically significant ($p < 0.05$). These findings demonstrate that edge-based processing substantially improves both responsiveness and latency stability in time-sensitive IoT warning systems. The study contributes empirical evidence supporting latency-aware architectural design for distributed environmental monitoring applications.*

Keywords: *Cloud architecture, edge computing, IoT latency, latency analysis, waterlogging detection.*

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

VOL. 12. NO. 2 NOVEMBER 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i2.8325

INTRODUCTION

Urban waterlogging and localized flooding have become recurring hydrometeorological challenges in regions characterized by high rainfall intensity and rapid land-use transformation. Increased precipitation variability and extreme rainfall events significantly contribute to surface runoff accumulation in low-elevation urban areas[1], [2]. The reduction of natural infiltration zones due to urban expansion further intensifies runoff concentration and overloads drainage systems [3], [4]. Conventional structural mitigation approaches such as drainage channel expansion and flood control infrastructure require substantial financial investment and long implementation periods, limiting their applicability at localized scales [5].

As an alternative, infiltration-based ecological techniques such as biopore infiltration systems have been widely adopted to enhance soil permeability and reduce surface runoff [6], [7]. Biopores improve vertical water absorption and increase soil porosity through biological activity within the infiltration cavity [8], [9]. Empirical studies have reported that biopore systems can reduce surface water stagnation and improve infiltration performance in residential environments [10], [11]. However, conventional biopore implementations operate passively without real-time monitoring capability. The absence of monitoring mechanisms restricts early detection of soil saturation conditions, clogging events, or reduced infiltration capacity, ultimately limiting system reliability in preventing localized waterlogging [12], [13].

The advancement of Internet of Things (IoT) technology enables distributed environmental sensing systems capable of transmitting real-time measurements [14], [15]. Several studies have implemented IoT-based monitoring platforms integrating soil moisture sensors, ultrasonic level sensors, and wireless communication modules for environmental observation [16], [17]. These systems commonly transmit sensor data to cloud platforms for centralized processing and visualization [18], [19].

In cloud-centric IoT architectures, computational logic, threshold evaluation, and alert triggering mechanisms are executed on remote servers. While this approach simplifies data aggregation and system scalability, it introduces dependency on network connectivity and remote processing availability [20], [21]. End-to-end response time in such architectures consists of sensor acquisition delay, uplink transmission delay, cloud processing delay, and downlink response delay [22], [23]. For latency-sensitive applications such as early warning systems, cumulative communication delay may significantly reduce responsiveness and compromise system effectiveness [24], [25]. Network instability, bandwidth limitations, and gateway congestion further contribute to latency variability and unpredictable system behavior [26], [27].

From a distributed computing perspective, responsiveness is a fundamental performance indicator in real-time systems. Architectural decisions regarding computation placement directly influence latency behavior and system reliability. Centralized processing models prioritize scalability but may sacrifice response time due to multi-hop transmission overhead. Conversely, decentralized or distributed processing reduces network dependency by relocating time-critical logic closer to the data source.

Edge Computing has emerged as a distributed computing paradigm that relocates processing logic from centralized cloud servers to edge devices near the sensing layer [28], [29]. By executing threshold evaluation and decision-making mechanisms directly on embedded devices, communication overhead and network round-trip delay can be minimized [30]. In latency-sensitive IoT scenarios, such as environmental hazard detection and emergency alert systems, edge-based processing is theoretically capable of reducing response time and improving temporal stability compared to cloud-centric models [31].

In long-range environmental monitoring systems, communication technologies such as LoRa and LoRaWAN are widely adopted due to their low power consumption and extended coverage capability. However, although LoRa provides reliable long-distance communication, it may introduce additional transmission delay depending on spreading factor and network configuration [32]. Therefore, architectural optimization becomes increasingly important when long-range communication protocols are integrated into latency-sensitive IoT frameworks.

Despite the theoretical advantages of edge-based architectures, empirical quantitative comparisons between cloud and edge processing in environmental monitoring systems remain limited. Existing IoT-enabled biopore systems primarily emphasize deployment feasibility and monitoring functionality rather than rigorous architectural performance evaluation. The absence of latency decomposition modeling, repeated experimental measurement, and statistical validation creates a research gap in determining the measurable

CC BY NC

performance improvement achieved by migrating threshold processing from the cloud layer to the edge layer.

Therefore, this study investigates whether relocating threshold evaluation and alert triggering mechanisms from a cloud server to an edge device significantly reduces end-to-end system latency and improves response stability in an IoT-based waterlogging detection framework. The proposed architecture maintains the cloud layer for data logging and visualization, while critical decision logic is executed locally on the edge device. Specifically, this research aims to:

1. formulate a latency decomposition model distinguishing communication delay and local processing delay in cloud and edge architectures;
2. experimentally measure and compare average response time and latency stability under repeated controlled testing; and
3. statistically evaluate whether the observed latency reduction achieved through edge-based processing is significant.

By providing empirical architectural performance analysis supported by statistical validation, this study contributes to latency-aware IoT system design strategies and supports the development of reliable distributed environmental warning systems, particularly in long-range communication environments integrating LoRa-based networks.

The novelty of this study lies in three aspects: (1) the proposed latency decomposition model distinguishing communication and processing delays in cloud and edge architectures, (2) the integration of LoRa-based environmental sensing within a smart biopore monitoring framework, and (3) the statistical validation of architectural performance through repeated experimental measurements and inferential analysis.

MATERIALS AND METHODS

Research Design

This study employs an experimental quantitative design to compare end-to-end response latency between two computational architectures in an IoT-based waterlogging detection system:

1. Cloud-Centric Architecture
2. Edge-Based Architecture

The primary dependent variable is response latency (milliseconds), defined as the time interval between threshold detection at the sensor node and activation of the local alert actuator (buzzer/LED).

The independent variable is the computational architecture model (Cloud vs Edge). Control variables include:

a. Identical hardware configuration
b. Identical sensor types
c. Identical threshold values
d. Identical communication protocol

Each experimental scenario was executed in repeated measurements (N = 50 iterations) to obtain statistically reliable latency data.

System Architecture Configuration

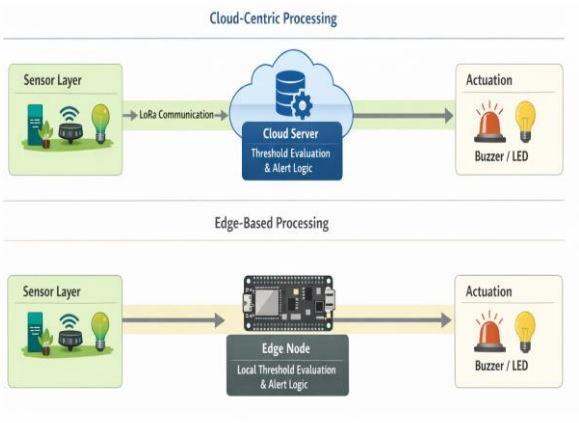
The experimental system consists of three main layers:

1. Sensing Layer
2. Processing Layer
3. Communication and Visualization Layer

The architectural distinction between the two scenarios lies in the location of threshold evaluation and alert triggering logic.

In the Cloud-Centric model, sensor data are transmitted to a cloud server where threshold processing is executed. The alert command is then transmitted back to the device. In the Edge-Based model, threshold evaluation and alert triggering are executed locally on the microcontroller, while the cloud layer is used only for data logging and visualization.

Figure 1 illustrates the comparative architectural configuration between cloud-centric and edge-based processing.



Source: (Research Results, 2026)

Figure 1. Comparative Architecture of Cloud-Centric and Edge-Based IoT Processing

Figure 1 illustrates the comparative architecture between cloud-centric and edge-based IoT processing models used in this study. In the cloud-centric configuration, sensor data from the sensing layer are transmitted via LoRa communication to a remote cloud server where threshold evaluation

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

VOL. 12. NO. 2 NOVEMBER 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i2.8325

and alert logic are executed before a response signal is sent back to activate the actuator. This architecture introduces uplink, server processing, and downlink delays as components of total response latency. In contrast, the edge-based configuration relocates threshold evaluation and alert logic directly to the edge node (ESP32), enabling local decision-making without network round-trip dependency for critical alert activation. The cloud layer in this model is utilized solely for data logging and visualization. The structural difference between these two processing flows forms the basis for the latency decomposition and comparative performance analysis conducted in this research.

Hardware Components

The experimental hardware configuration is identical for both scenarios to ensure fair comparison. Table 1 presents the hardware specifications used in the experiment.

Table 1. Hardware Components and Functional Roles

No	Component	Specification	Functional Role
1	Microcontroller	ESP32-WROOM-32	Edge processing unit
2	Soil Moisture Sensor	Capacitive Sensor	Soil saturation detection
3	Ultrasonic Sensor	HC-SR04	Water level detection
4	Communication Module	LoRa Ra-02	Data transmission
5	Actuator	Buzzer & LED	Local alert indicator

Source: (Research Results, 2026)

The ESP32 microcontroller executes firmware developed in Arduino IDE. In the cloud-centric scenario, threshold processing is performed by the cloud platform. In the edge-based scenario, the ESP32 executes threshold logic locally.

Latency Measurement Model

The latency decomposition model adopted in this study follows the end-to-end delay concept commonly used in IoT performance evaluation studies [23], [27]. Response latency is defined as:

$$T_{response} = T_{alert} - T_{threshold} \quad (1)$$

where:

- $T_{threshold}$= timestamp when sensor reading exceeds predefined threshold
- T_{alert}= timestamp when actuator is activated

Cloud-Centric Latency Model

$$T_{cloud} = T_{sensor} + T_{uplink} + T_{cloud_process} + T_{downlink} \quad (2)$$

Edge-Based Latency Model

$$T_{edge} = T_{sensor} + T_{local_process} \quad (3)$$

The latency decomposition allows isolation of communication-induced delay from local processing delay.

Experimental Procedure

The experimental procedure was conducted under controlled laboratory conditions.

Step 1: Threshold Simulation

Soil moisture and ultrasonic levels were artificially adjusted to exceed predefined critical values:

a. Soil Moisture ≥ 85%
b. Water Capacity ≤ 20%

Step 2: Timestamp Recording

A high-resolution internal timer (millisecond resolution) embedded within the ESP32 firmware recorded the threshold detection time and actuator activation time.

Step 3: Iterative Testing

Each architectural scenario was tested 50 times (N = 50).
Network conditions were maintained constant during experimentation.

Step 4: Data Logging

Latency values were logged and exported for statistical analysis.

Table 2. Experimental Environment Configuration

Parameter	Value
Communication Protocol	LoRaWAN
Spreading Factor	SF7
Bandwidth	125 kHz
Coding Rate	4/5
Node–Gateway Distance	50 m
Number of Nodes	1
Gateway Type	LoRaWAN Gateway
Internet Access	Fiber Optic
Cloud Platform	Blynk Cloud
Packet Size	32 bytes

Source: (Research Results, 2026)

CC BY NC

The experimental configuration was designed to isolate the effect of computational placement on response latency by maintaining consistent communication conditions across the cloud-centric and edge-based architectures. The experiments were conducted using a single IoT node under controlled laboratory conditions, with a fixed node-to-gateway distance of 50 meters and LoRa parameters set to SF7, a bandwidth of 125 kHz, and a coding rate of 4/5. These controlled settings enabled a consistent comparison of the two architectures; however, they limit the generalizability of the findings to large-scale deployments and dynamically varying network conditions. Furthermore, the effects of alternative LoRa configurations and communication distances on latency were not evaluated, and power consumption was not directly measured. Consequently, the latency improvement observed in this study should not be interpreted as evidence of improved energy efficiency. Future research should investigate multi-node deployments, dynamic network conditions, varying LoRa parameters and communication distances, and direct energy consumption measurements to provide a more comprehensive assessment of the performance, scalability, and energy efficiency of edge-based and cloud-centric IoT architectures.

Statistical Analysis

Latency data were analyzed using:

1. Mean Response Time (μ)
2. Standard Deviation (σ)

Mean latency is calculated as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} T_i \quad (4)$$

Standard deviation is calculated as:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i - \mu)^2} \quad (5)$$

Statistical analysis follows conventional independent sample t-test procedures widely applied in comparative system performance evaluation [29]. To evaluate whether the difference between cloud and edge latency is statistically significant, an independent sample t-test is applied:

$$t = \frac{\mu_{cloud} - \mu_{edge}}{\sqrt{\frac{\sigma^2_{cloud}}{N} + \frac{\sigma^2_{edge}}{N}}} \quad (6)$$

The significance level is set at α = 0.05.

Evaluation Metrics

The system performance is evaluated based on:

1. Average Response Time Reduction
2. Latency Stability (Standard Deviation Comparison)
3. Statistical Significance (p-value)

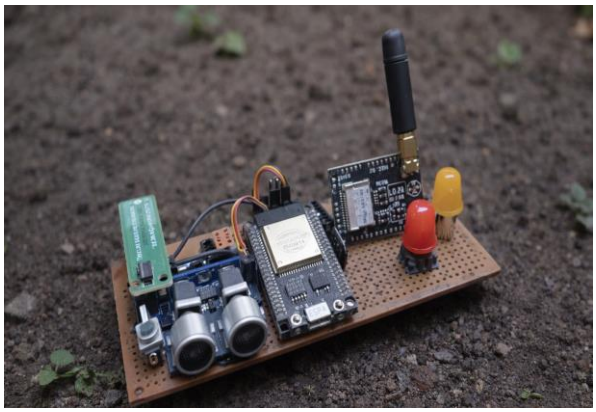
The percentage of latency reduction is calculated as:

$$Efficiency(\%) = \frac{\mu_{cloud} - \mu_{edge}}{\mu_{cloud}} \times 100 \quad (7)$$

RESULTS AND DISCUSSION

System Realization and Functional Verification

Before latency evaluation, functional testing was conducted to ensure that both architectural scenarios operated correctly under identical hardware configurations. The physical implementation of the IoT node is shown in **Figure 2**.



Source: (Research Results, 2026)

Figure 2. Physical Implementation of the IoT Edge Node

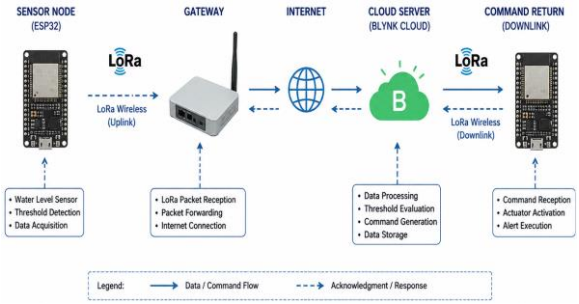
The hardware implementation integrates ESP32, soil moisture sensor, ultrasonic sensor, LoRa module, buzzer, and LED actuator. Functional testing confirmed:

a. Sensor readings were stable across repeated measurements.
b. Threshold logic was correctly executed in both scenarios.
c. Actuator activation corresponded precisely to threshold detection events.

This validation ensured that latency differences observed later were caused by architectural variation rather than hardware malfunction.

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

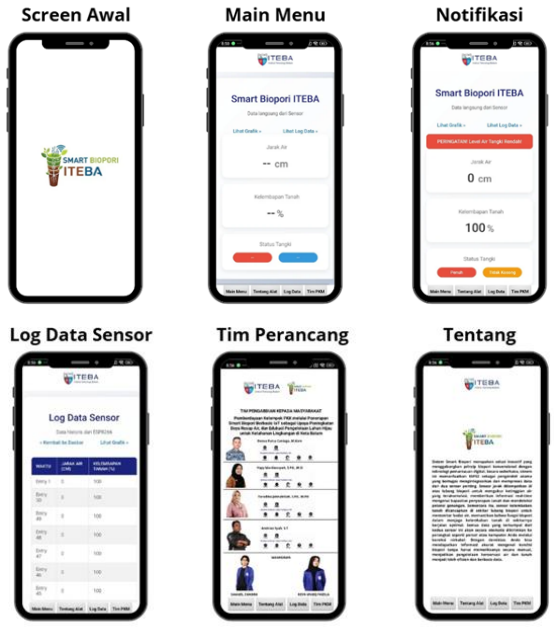
VOL. 12. NO. 2 NOVEMBER 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i2.8325



Source: (Research Results, 2026)
Figure 3. Cloud-Based Experimental Configuration

Cloud Dashboard Monitoring Layer

Although the focus of this study is architectural latency comparison, the cloud layer was maintained for logging and visualization purposes. Figure 4 shows the monitoring dashboard used during experimentation.



Source: (Research Results, 2026)
Figure 4. Cloud-Based Monitoring Dashboard

The dashboard confirms that data transmission was successfully maintained in both scenarios. However, in the cloud-centric configuration, threshold evaluation was performed on the cloud server, introducing additional network round-trip delay before actuator activation. This distinction is critical in understanding the latency difference measured in the next subsection.

Detailed Latency Measurement Results

The raw latency measurements were recorded for 50 iterations under each architectural configuration. Descriptive statistical results are presented in **Table 3**.

Table 3. Latency Measurement Results (N = 50)

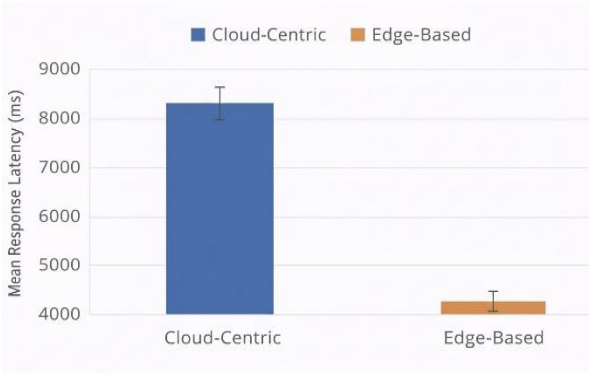
Metric	Cloud-Centric	Edge-Based
Maximum (ms)	9,870	2,150
Minimum (ms)	6,520	1,480
Mean (μ) (ms)	7,925	1,745
Standard Deviation (σ) (ms)	710.2	195.8

Source: (Research Results, 2026)

The maximum latency observed in the cloud-centric architecture reached 9,870 ms, whereas the maximum latency in the edge-based architecture was only 2,150 ms. The minimum latency values also demonstrate a substantial separation between architectures. The 6-second gap between average response times confirms that network-induced delay dominates total response time in cloud-based processing.

Iteration-Level Latency Comparison

To visualize latency distribution across all iterations, the comparative latency performance is illustrated in **Figure 5**.



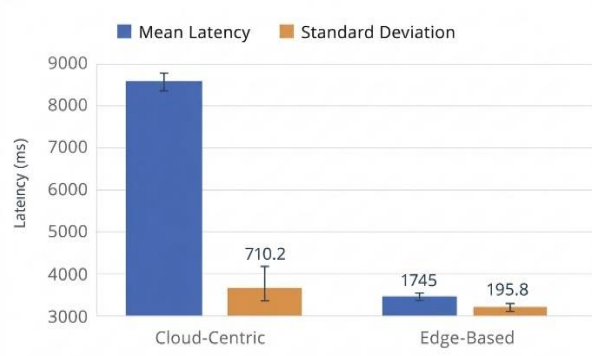
Source: (Research Results, 2026)
Figure 5. Latency Comparison Across 50 Iterations

Figure 5 shows consistent separation between cloud-centric and edge-based latency values across all experimental repetitions. No overlap was observed between average latency ranges of both architectures.
This consistency strengthens the reliability of the findings and indicates that latency reduction is structurally caused rather than incidentally observed.

Mean and Variability Visualization

Figure 6 provides a visual comparison of mean latency and standard deviation between architectures.

CC BY NC

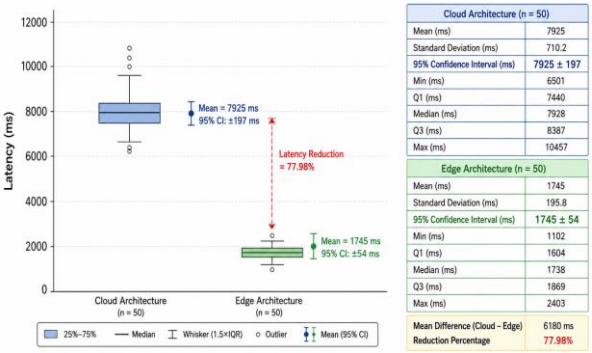


Source: (Research Results, 2026)
Figure 6. Mean Latency and Standard Deviation Comparison

The significantly shorter error bar in the edge-based configuration indicates improved stability. In cloud-centric processing, variability is influenced by:

a. LoRa transmission delay fluctuation
b. Internet gateway processing
c. Server-side execution variability

In contrast, the edge-based system eliminates uplink and downlink delay for alert activation, reducing external dependency.



Source: (Research Results, 2026)
Figure 7. Latency Distribution Boxplot with 95% Confidence Interval

Figure 7 provides a more detailed statistical visualization of latency behavior across both architectural configurations. The boxplot reveals that the cloud-centric architecture exhibits a wider interquartile range (IQR) and a larger number of extreme observations compared to the edge-based architecture. This indicates greater latency variability caused by network transmission, gateway processing, and cloud execution delays. The 95% confidence interval further confirms the robustness of the observed difference. The cloud architecture produced a mean latency of 7,925 ± 197 ms, whereas the edge architecture achieved a mean latency of 1,745 ± 54 ms. The absence of confidence interval overlap suggests a substantial separation between both architectures and supports the statistical significance observed in the hypothesis testing stage. The narrower distribution and shorter whisker range observed in the edge-based configuration indicate improved response consistency, which is particularly important for time-sensitive environmental monitoring applications.

Statistical Validation of Latency Difference

The independent sample t-test confirms that the observed difference is statistically significant ($p < 0.05$). The computed t-value indicates that the difference between mean latencies is not attributable to random variation. This statistical validation is essential for Sinta 2-level rigor because it transforms the result from descriptive comparison into inferential confirmation.

Table 4. Statistical Validation Results

Metric	Value
t-value	57.81
df	98
p-value	<0.001
Cohen's d	8.12

Source: (Research Results, 2026)

Latency Decomposition Interpretation

Based on the latency model introduced earlier:

$$T_{cloud} = T_{sensor} + T_{uplink} + T_{cloud_process} + T_{downlink}$$

$$T_{edge} = T_{sensor} + T_{local_process}$$

The experimental difference of approximately 6,180 ms can be interpreted as the cumulative impact of:

a. Uplink transmission delay
b. Remote server processing delay
c. Downlink response propagation

The elimination of these components in the edge-based architecture directly explains both lower mean latency and reduced variability.

Efficiency Analysis

Using the efficiency equation:

$$Efficiency(\%) = \frac{7{,}925 - 1{,}745}{7{,}925} \times 100$$

The result of 77.98% latency reduction demonstrates that more than three-quarters of the response delay in cloud-centric architecture

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

VOL. 12. NO. 2 NOVEMBER 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i2.8325

originates from communication and remote processing overhead.
This magnitude of reduction is significant for time-sensitive IoT applications.

Comparison with Related IoT Systems

Previous IoT monitoring systems primarily emphasized functional deployment and cloud visualization. However, architectural latency evaluation was rarely quantified.
Unlike prior systems, this study provides:

a. Controlled repeated experimentation
b. Latency modeling
c. Statistical significance testing
d. Stability evaluation using standard deviation

The findings demonstrate that computational architecture selection directly influences responsiveness.

Implications for IoT Architecture Design

The results suggest that for latency-sensitive systems:

a. Edge computing should handle threshold logic.
b. Cloud systems should manage non-critical functions.
c. Hybrid architectures provide optimal balance.

This architectural separation ensures responsiveness without sacrificing scalability and centralized monitoring capability.
Despite its latency advantages, edge-based processing presents several limitations. Edge devices typically possess limited computational resources, memory capacity, and energy availability compared with cloud servers. Complex analytical tasks and large-scale data aggregation may therefore remain more suitable for cloud-based infrastructures. Consequently, hybrid architectures combining edge responsiveness and cloud scalability represent a more balanced solution for practical deployments.

CONCLUSION

This study conducted a comparative latency analysis between cloud-centric and edge-based IoT architectures in a waterlogging detection framework. The experimental results demonstrate that relocating threshold evaluation and alert triggering mechanisms from the cloud layer to the edge device significantly reduces end-to-end response latency. The average response time decreased from 7,925 ms in the cloud-centric architecture to 1,745 ms in the edge-based configuration, representing a latency reduction of approximately 77.98%. In addition to reducing average response time, the edge-based architecture also improved latency stability, as indicated by the substantial decrease in standard deviation from 710.2 ms to 195.8 ms. This improvement confirms that network transmission delay is the dominant contributor to response variability in cloud-centric processing. The independent sample t-test further validated that the observed latency difference is statistically significant, indicating that the performance improvement is structurally caused by architectural modification rather than random fluctuation.

The findings confirm that computational architecture plays a decisive role in determining the responsiveness of latency-sensitive IoT systems. By eliminating network round-trip dependency for time-critical decision logic, edge-based processing enhances both response speed and temporal consistency. The results support the adoption of hybrid IoT architectures in which edge devices handle real-time processing tasks, while cloud platforms are reserved for data storage, visualization, and long-term analytics. Although the experimental evaluation was conducted using a single node under controlled laboratory conditions, a fixed node-to-gateway distance, and a predefined LoRa configuration, the findings provide empirical evidence of the latency benefits of edge-based processing.

Nevertheless, the generalizability of these results remains limited by the experimental scope, particularly regarding multi-node deployments, dynamically varying network conditions, and alternative LoRa communication parameters, including spreading factor, bandwidth, coding rate, and communication distance. Furthermore, since power consumption and energy usage were not directly measured, the observed latency improvement should not be interpreted as evidence of improved energy efficiency. Future research should extend the evaluation to multi-node deployments, dynamic network conditions, diverse LoRa configurations, and direct power and energy measurements to provide a more comprehensive understanding of the performance, scalability, and energy implications of edge-based and cloud-centric IoT architectures.

DECLARATION OF AI USE

The authors declare that ChatGPT, developed by OpenAI, was used solely to assist with language refinement and grammar checking of the manuscript. The tool was not used in the preparation of the research content, data analysis,

CC BY NC

interpretation of results, or scientific conclusions. The authors reviewed and approved the final statement and take full responsibility for the entire manuscript.

AUTHOR CONTRIBUTIONS

Author	Contribution Roles
Deosa Putra Caniago	Conceptualization, Methodology, Formal Analysis, Project Administration, Supervision, Writing – Original Draft, Writing – Review & Editing. Developed the research concept, designed the methodology, analyzed the results, supervised the research, and prepared the manuscript.
Hetty Rohayani	Validation, Supervision, Writing – Review & Editing. Reviewed the research approach, validated the findings, and provided academic feedback.
Nova Agustina	Formal Analysis, Validation, Writing – Review & Editing. Reviewed the data analysis, interpretation, and scientific findings.
Arip Kristiyanto	Methodology, Validation, Writing – Review & Editing. Reviewed the experimental design, methodology, and research procedures.
Najirah Umar	Investigation, Validation, Writing – Review & Editing. Reviewed the research implementation, experimental procedures, and findings.
Muhammad Dzikri Al Farizi	Investigation, Data Curation, Software, Visualization. Assisted with IoT prototype implementation, experimental testing, data collection, and visualization.
Yopy Mardiansyah	Investigation, Resources, Software, Writing – Review & Editing. Supported equipment preparation, system implementation, technical testing, and manuscript review.

ACKNOWLEDGMENTS

The authors would like to acknowledge the technical and institutional support provided by Mechatronics Laboratory of ITEBA during the implementation and testing of this research. The authors also appreciate the assistance of all individuals who contributed to the experimental activities.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest related to the research, writing, or publication of this article.

REFERENCE

[1] R. Ramadhan, M. Marzuki, W. Suryanto, S. Sholihun, H. Yusnaini, and R. Muharsyah, "Rainfall variability in Indonesia new capital associated with the Madden-Julian Oscillation and its contribution to flood events," *Quaternary Science Advances*, vol. 13, Art. no. 100163, 2024, doi: 10.1016/j.qsa.2024.100163.

[2] Q. Zhang, S. Zhou, J. Li, and Z. Xu, "International Journal of Disaster Risk Reduction Waterlogging susceptibility assessment in developed urban area using explainable machine learning methods with different negative sampling strategies," *Int. J. Disaster Risk Reduct.*, vol. 132, no. July 2025, p. 105978, 2026, doi: 10.1016/j.ijdrr.2025.105978.

[3] A. Rakhim, A. Mansida, F. Gaffar, and H. Bancong, "Journal of Hydrology : Regional Studies Enhancing hydrological resilience through infiltration wells and biopore holes on critical lands of sub-watershed areas," *J. Hydrol. Reg. Stud.*, vol. 64, no. February, p. 103261, 2026, doi: 10.1016/j.ejrh.2026.103261.

[4] H. Li, C. Yan, C. Zhou, and L. Zhou, "Building Virtual Drainage Systems Based on Open Road Data and Assessing Urban Flooding Risks," *Water*, vol. 18, no. 3, Art. no. 341, 2026, doi: 10.3390/w18030341.

[5] L. Luo, C. Cheng, W. Mobley, K. Lieberknecht, J. Kim, and F. Leite, "International Journal of Disaster Risk Reduction Optimizing fluvial flood mitigation strategies : A multi-objective approach for cost-effective and socially-aware infrastructure feasibility analysis," *Int. J. Disaster Risk Reduct.*, vol. 133, no. May 2025, p. 106004, 2026, doi: 10.1016/j.ijdrr.2026.106004.

[6] A. Sarminingsih, S. Sumiyati, H. Wandra, and B. P. Samadikun, "Analysis of the effect of permeability values on the distribution of heavy metals and organic content in soil around biopore infiltration holes," *IOP Conference Series: Earth and Environmental Science*, vol. 1556, no. 1, Art. no. 012043, 2025, doi: 10.1088/1755-1315/1556/1/012043.

[7] N. Naharuddin, S. D. Massiri, H. Pribadi, A. Maiwa, and M. Ihsan, "Evaluation of Rainwater Harvesting and Bio-pore Infiltration Holes for Flood Mitigation and Soil Conservation," *Journal of Civil and Hydraulic Engineering*, vol. 2, no. 3, pp. 142–

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

150, 2024, doi: 10.56578/jche020302.

[8] S. Qin *et al.*, "Geoderma Long-term no-tillage farming mitigates waterlogging during extreme rainfall events in black soil (Mollisols) region of Northeast China," *Geoderma*, vol. 462, no. July, p. 117542, 2025, doi: 10.1016/j.geoderma.2025.117542.

[9] N. Jarvis et al., "Interactions between soil structure dynamics, hydrological processes, and organic matter cycling: A new soil-crop model," *European Journal of Soil Science*, vol. 75, no. 2, Art. no. e13455, 2024, doi: 10.1111/ejss.13455.

[10] I. D. M. A. Karyawan, I. D. G. J. Negara, I. W. Yasa, M. Mahendra, and I. W. Joniarta, "Socialization of environmental road maintenance in settlements with the application of biopores to reduce water runoff to the road surface," Unram Journal of Community Service, vol. 5, no. 2, pp. 71–77, 2024, doi: 10.29303/ujcs.v5i2.610.

[11] A. Fitri, N. A. B. Ahmad, A. Zain, M. Mera, and D. Pratiwi, "Optimizing bio-pore infiltration systems for urban flood mitigation: A field study in Bandar Lampung, Indonesia," *E3S Web of Conferences*, vol. 677, Art. no. 06007, 2025, doi: 10.1051/e3sconf/202567706007.

[12] L. A. P. Barbosa, F. V. Lima, and H. H. Gerke, "Neutron imaging of exchange flow between biopore and matrix for Bt versus C horizons," *Vadose Zone Journal*, vol. 23, no. 4, Art. no. e20320, 2024, doi: 10.1002/vzj2.20320.

[13] N. R. Puspita and V. P. Prattyni, "Community Empowerment in Improving Environmental Drainage Infrastructure Through Active Participation and Appropriate Technology," *International Journal of Technology & Energy*, vol. 1, no. 3, pp. 123–130, 2025, doi: 10.71364/ijte.v1i3.17.

[14] D. P. Caniago, Jufri, M. & Masril, M.A., 2024. Sistem Pengawasan Berbasis IoT pada Robot Vision Untuk Peningkatan Keamanan Perimeter di Industri Batam. The Indonesian Journal of Computer Science, [e-journal] 13(6). Tersedia di: https://doi.org/10.33022/ijcs.v13i6.4476

[15] D. P. Caniago, Meinadia, M. Y., Mardiansyah, Y., Damanik, E. M., Hilmy, A., & Syah, "Penerapan sistem deteksi dini cuaca ekstrem dan gelombang rob berbasis IoT untuk pengurangan risiko bencana di pulau terluar Kota Batam." pp. 424–434, 2024. doi: https://doi.org/10.33373/jmb.v8i2.6886.

[16] Y. Mardiansyah, N. Ilmi, D. P. Caniago, M. A. Masril, R. E. Fahruddini, and H. Sumardi, "Application of smart indoor hydroponic technology to support food security," *Abdimas J. Pengabdi. Masy. Univ. Merdeka Malang*, vol. 8, no. 4, pp. 572–582, Nov. 2023, doi: 10.26905/abdimas.v8i4.11275.

[17] D. P. Caniago and M. A. Masril, "Transformasi Pertanian Dalam Ruangan: Hidroponik Cerdas Berbasis IoT," *Indones. J. Comput. Sci.*, vol. 12, no. 5, pp. 2747–2755, Oct. 2023, doi: 10.33022/ijcs.v12i5.3408.

[18] C. Lee, J. Kim, H. Ko, and B. Yoo, "Internet of Things Addressing IoT storage constraints : A hybrid architecture for decentralized data storage and centralized management," *Internet of Things*, vol. 25, p. 101014, 2024, doi: 10.1016/j.iot.2023.101014.

[19] R. M. M. Salem, M. S. Saraya, and A. M. R. M. T. Ali-eldin, "An Industrial Cloud-Based IoT System for Real-Time Monitoring and Controlling of Wastewater," *IEEE Access*, vol. 10, pp. 6528–6540, 2022, doi: 10.1109/ACCESS.2022.3141977.

[20] H. Kuchuk and E. Malokhvii, "Integration of IoT with cloud, fog, and edge computing: A review," Advanced Information Systems, vol. 8, no. 2, pp. 65–75, 2024, doi: 10.20998/2522-9052.2024.2.08.

[21] M. Kumar, A. Kumar, S. Verma, P. Bhattacharya, D. Ghimire, S.-h. Kim, and A. S. M. S. Hosen, "Healthcare Internet of Things (H-IoT): Current trends, future prospects, applications, challenges, and security issues," Electronics, vol. 12, no. 9, 2023, doi: 10.3390/electronics12092050.

[22] J. Li, Y. Chen, Y. Chen, W. Zhang, and Z. Liu, "Results in Engineering A smart energy IoT model based on the Itsuku PoW technology ☆," *Results Eng.*, vol. 18, no. April, p. 101147, 2023, doi: 10.1016/j.rineng.2023.101147.

[23] M. Beshley, N. Kryvinska, H. Beshley, O. Kochan, and L. Barolli, "Measuring End-to-End Delay in Low Energy SDN IoT Platform," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 19–41, 2022, doi: 10.32604/cmc.2022.018579.

[24] H. Zhang, R. Zhang, and J. Sun, "Developing real-time IoT-based public safety alert and emergency response systems," *Scientific Reports*, vol. 15, Art. no. 29056, 2025, doi: 10.1038/s41598-025-13465-7.

[25] G. Premsankar, "Energy-Efficient Service Placement for Latency-Sensitive Applications in Edge Computing," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17926–17937, 2022, doi: 10.1109/JIOT.2022.3162581.

[26] A. P., H. S. Vimala, and S. J., "Comprehensive review on congestion detection, alleviation,

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.

and control for IoT networks," *Journal of Network and Computer Applications*, vol. 221, Art. no. 103749, 2024, doi: 10.1016/j.jnca.2023.103749.

[27] A. Kaushik and H. S. Al-Raweshidy, "A hybrid latency- and power-aware approach for beyond fifth-generation Internet-of-Things edge systems," IEEE Access, vol. 10, pp. 87974–87990, 2022, doi: 10.1109/ACCESS.2022.3200035.

[28] D. P. Caniago, L. Hernando, R. D. Permatasari, M. A. Bora, A. A. Dermawan, Alhamidi, A. Syah, and S. T. Chandra, "Swarm Robot Communication Using ESP-NOW Mesh Protocol for Multi-Agent Coordination in Indoor Navigation," *Journal Européen des Systèmes Automatisés*, vol. 59, no. 5, pp. 1263–1274, 2026, doi: 10.18280/jesa.590507.

[29] J. de Winter, "Neither the t-test nor Welch's test: A case for robust alternatives in two-sample comparisons with non-ideal data," *Quality & Quantity*, vol. 60, pp. 18673–18697, 2026, doi: 10.1007/s11135-026-02909-5.

[30] J. Karim, O. F. Goni, M. Ahsan, J. Haider, and M. Kowalski, "Enhancing agriculture through real - time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad - CAM," *Sci. Rep.*, pp. 1–23, 2024, doi: 10.1038/s41598-024-66989-9.

[31] D. P. Caniago, Y. Mardiansyah, F. Jabnabillah, S. Candra, R. U. Faizila, A. Syah, and M. A. S. Aritonang, "Pemberdayaan kelompok PKK dalam penerapan smart biopori IoT untuk peningkatan daya resap air dan ketahanan lingkungan di Batam," Minda Baharu, vol. 9, no. 2, pp. 316–328, Dec. 2025, doi: 10.33373/jmb.v9i2.8382.

[32] M. Abrar Masril, D. P. Caniago, and S. A. Wibowo, "Development of SERA (Smart Ergonomic Rescue Apparel) with Biometric Sensors and LoRa for the Early Detection of Hypothermia in the Waters of the Riau Islands," *J. Eur. des Syst. Autom.*, vol. 58, no. 8, pp. 1639–1650, 2025, doi: 10.18280/jesa.580809.

Accredited Rank 2 (SINTA 2) based on Decree No. 225/E/KPT/2022 issued by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, dated December 7, 2022. Published by LPPM Universitas Nusa Mandiri.