

# Performance Evaluation of a Self-Hosted Edge-IoT Data Pipeline for Hydroponic pH and EC Monitoring

Josh Delon<sup>1\*</sup>, Samuel P.<sup>1</sup>, Albert Sagala<sup>1</sup>

<sup>1</sup>Electrical Engineering, Faculty of Informatics and Electrical Engineering,  
Institut Teknologi Del, Toba, North Sumatera, 22381, Indonesia

\*Corresponding Author: [joshdelon@del.ac.id](mailto:joshdelon@del.ac.id)

## Article Info

Received: 29 July 2026

Revised: 2 September 2026

Accepted: 4 September 2026

Available online: 30 September 2026

## Keywords

Edge computing, electrical conductivity,  
hydroponics, InfluxDB, Internet of  
Things, MQTT

## Abstract

Many hydroponic Internet-of-Things (IoT) prototypes lack empirical validation regarding data completeness, traceability, and workload scalability across local pipelines. This study presents a rigorous performance evaluation of a self-hosted edge-IoT data pipeline where an ESP32 microcontroller publishes pH and electrical conductivity (EC) measurements via Mosquitto MQTT to Telegraf, InfluxDB, and Grafana on a Raspberry Pi 4. The evaluation benchmarks sensor accuracy across 80 calibration samples, temporal continuity over 48 hours (2,881 one-minute records), high-rate ingestion across three 30-minute trials, and multi-node scalability via simulated stress testing. Experimental results show high sensor accuracy, with pH mean absolute percentage errors (MAPE) of 0.775% (pH 4.00) and 0.386% (pH 7.00), and EC MAPE of 2.173% (1,413  $\mu\text{S}\cdot\text{cm}^{-1}$ ) and 0.986% (12,880  $\mu\text{S}\cdot\text{cm}^{-1}$ ). The 48-hour continuous run preserved all 2,881 expected records with zero loss or timestamp inversion, while high-rate ingestion (approx. 5.3 s intervals) achieved 99.80% storage completeness (1,018 of 1,020 records). Furthermore, workload stress testing demonstrated that the Raspberry Pi 4 edge stack accommodates up to 50 concurrent simulated nodes publishing at 1 Hz with under 33% CPU load and a 99.85% delivery success rate. These findings confirm the feasibility of self-hosted edge pipelines for robust, auditable precision hydroponic monitoring without reliance on external cloud services.

## 1. Introduction

Hydroponic cultivation relies on a recirculating aqueous nutrient solution rather than soil to supply essential minerals directly to plant root zones [1]. In closed hydroponic loops, solution pH dictates nutrient solubility and bioavailability, while electrical conductivity (EC) serves as an established proxy for the total concentration of dissolved ionic salts [2], [3]. Maintaining these physicochemical parameters within strict species-dependent tolerance windows is paramount for preventing nutrient deficiencies, toxicities, and irreversible osmotic stress [4]. Consequently, real-time sensing of pH and EC can reveal subtle environmental shifts and solution dynamics that periodic, manual grab sampling inevitably overlooks [5], [6]. However, sensor measurements possess agronomic value only if the underlying data architecture can capture, transport, ingest, persist, query, and visualize them without compromising sequence integrity, record completeness, or temporal consistency [7].

Agricultural Internet-of-Things (Ag-IoT) paradigms have emerged as central drivers of smart greenhouse management, yet field implementations must overcome significant hurdles regarding network reliability, heterogeneous protocol interoperability, power constraints, and data provenance [8], [9]. While cloud-centric hydroponic architectures offer remote visualization and external analytical capabilities, their heavy reliance on continuous wide-area network (WAN) connectivity introduces severe operational vulnerabilities in rural and agricultural regions where uplink bandwidth is frequently intermittent, expensive, or insecure [2], [10].

Edge computing offers a compelling architectural paradigm by migrating message brokering, parsing, storage, and visualization directly onto localized single-board compute hardware situated within the greenhouse perimeter [11], [12]. This edge-native approach guarantees low-latency responsiveness, data privacy, and autonomous operational continuity during wide-area internet outages [13].

**Research Gap:** Despite the expanding literature on Raspberry-Pi-based hydroponic prototypes, most existing contributions prioritize end-user software features—such as user interface widgets, automation rule logic, or machine learning forecasts—while treating data transport and storage as trivial "black-box" utilities [6], [14], [15]. A visually coherent dashboard graph does not prove that an edge pipeline preserves sampling regularity, avoids silently dropping packets, or remains stable under scaling workloads [10], [16]. Few studies provide an explicit, traceable evidence chain establishing sensor calibration agreement, long-term sequence continuity, database record completeness under stress, and end-to-end broker-to-dashboard traceability on resource-constrained hardware [17], [18].

To resolve this empirical gap, this study formulates a structured performance evaluation of a self-hosted edge-IoT data pipeline for hydroponic pH and EC monitoring. Specifically, the principal contributions of this research are:

- A modular, three-layer self-hosted architecture that isolates perception, edge brokering, time-series storage, and visualization entirely on a local Raspberry Pi 4 platform.
- An end-to-end verification framework that connects measurable empirical evidence to four core claims: sensor agreement against analytical standards, 48-hour sampling sequence continuity, high-frequency storage completeness, and broker-to-dashboard value matching.
- A comprehensive system-level scalability and workload analysis demonstrating that the edge stack maintains sub-35% CPU overhead and over 99.8% message delivery under high-frequency multi-node telemetry.

## 2. Related Work

The integration of embedded microcontrollers with IoT middleware for smart hydroponic monitoring has been extensively explored in recent literature. Wan et al. [11] developed a modular edge-computing architecture for aquaponics, demonstrating the benefits of localized edge processing in reducing communication traffic. Lukito et al. [12] implemented a Raspberry Pi edge gateway to track water quality parameters in hydroponics, utilizing edge nodes for preliminary data filtering prior to cloud synchronization. Abu Sneineh and Shabaneh [14] designed an ESP32-based hydroponic monitoring device interfaced with cloud servers, validating basic telemetry across low-cost sensors. Kushawaha et al. [15] presented an automated compact urban hydroponics system focusing on consumer-grade ease of use and low-cost sensor integration. Naa [17] documented an ESP32 and Raspberry Pi arrangement utilizing MQTT and Node-RED for greenhouse data routing. Stevens et al. [19] investigated optical and spectroscopic nutrient monitoring in micro indoor hydroponic farms, underscoring the critical necessity of high data reliability in automated closed-loop dosing. Sulaiman [20] implemented an IoT monitoring system for basic hydroponic environmental tracking.

While these works successfully demonstrate functional prototypes, their evaluations rarely probe the underlying data pipeline's persistence guarantees or audit missing records under stress. In contrast, this paper isolates the localized data pipeline—consisting of Eclipse Mosquitto, Telegraf, InfluxDB, and Grafana—and subjects it to strict quantitative audits of record counts, time-series interval distributions, and broker-to-dashboard consistency. Table 1 positions this work relative to representative systems in the recent literature.

**Table 1** Positioning Against Representative Systems

| Work                        | Compute Platform       | Data Architecture            | Key Focus / Evaluation Scope                                  |
|-----------------------------|------------------------|------------------------------|---------------------------------------------------------------|
| Wan et al. [11]             | Edge Nodes             | Edge + Cloud Hybrid          | Modular aquaponic communication performance                   |
| Lukito et al. [12]          | Raspberry Pi           | Edge + Cloud Filtering       | Water-quality control and cloud monitoring                    |
| Abu Sneineh & Shabaneh [14] | ESP32                  | IoT Cloud Interface          | Hydroponic sensing and remote dashboard                       |
| Kushawaha et al. [15]       | ESP32 / Arduino        | Local Display / IoT          | Compact small-scale hydroponic automation                     |
| Stevens et al. [19]         | Microcontroller        | IoT Cloud Logging            | Spectroscopic nutrient monitoring in smart hydroponics        |
| Sulaiman [20]               | Microcontroller        | IoT Broker                   | Basic hydroponic parameter tracking                           |
| This work                   | ESP32 + Raspberry Pi 4 | Fully Self-Hosted Edge Stack | Multi-node continuity, storage completeness, and traceability |

## 3. Problem Definition and Verification Logic

In precision agriculture, dashboard displays can mask data loss if queries perform silent interpolation, downsampling, or if missing timestamps are obscured by chart smoothing [13], [21]. To rigorously evaluate system reliability, the localized data pipeline is modeled as a sequential information path:

$$P : S \rightarrow M \rightarrow T \rightarrow I \rightarrow G \quad (1)$$

where  $S$  represents the ESP32 sensor publisher,  $M$  denotes the Mosquitto MQTT broker,  $T$  is the Telegraf metrics collector,  $I$  is the InfluxDB time-series database, and  $G$  is the Grafana visualization engine [10], [13]. A visually plausible dashboard can still conceal an acquisition error, a missing message, a failed write, or an ambiguous timestamp. Evaluation must therefore connect each claim to observable empirical evidence.

To avoid unsupported claims, four structured verification questions guide this evaluation. First, do the analog pH and EC probe outputs agree with standardized analytical reference solutions across repeated trials? Second, does the operational export over a 48-hour continuous deployment preserve the expected one-minute interval without temporal gaps or inversions? Third, when the publication frequency is increased substantially, does the local database retain the expected volume of transmitted records? Fourth, can individual telemetry payloads observed at the MQTT broker log be unambiguously matched with records extracted from Grafana, and what do the associated timestamps demonstrate? Table 2 formalizes the boundary between the empirical evidence gathered and the legitimate scientific claims supported by that evidence [21].

**Table 2** Evidence-to-claim boundary for system verification

| Verification Target              | Evidence Base                                 | Metric                                              | Supported Interpretation                           | Interpretation Not Supported                                    |
|----------------------------------|-----------------------------------------------|-----------------------------------------------------|----------------------------------------------------|-----------------------------------------------------------------|
| Sensor agreement                 | Repeated readings in four reference solutions | MAPE, RMSE, mean $\pm$ SD                           | Agreement at the tested reference points           | Full-range metrological accuracy across all conditions          |
| Monitoring continuity            | 48-h operational Grafana export               | Count, interval distribution, duplicate/order check | Presence of each expected one-minute position      | Physical correctness of every reading under flow variations     |
| Storage completeness             | Three 30-min high-rate trials                 | Stored/expected record ratio                        | Expected-record completeness for the tested trials | Exact packet-loss location or internal database engine accuracy |
| Broker-to-dashboard traceability | 43 matched pH observations                    | Value-match rate and timestamp-difference proxy     | Value correspondence within displayed time bins    | True end-to-end network latency without clock quantization      |

#### 4. System Architecture and Implementation

The implemented hydroponic system comprises three distinct architectural tiers: the Perception and Physical Layer, the Edge and Network Layer, and the Application and Access Layer, as delineated in Table 3 [16], [18].

**Table 3** Three-layer architecture of the implemented hydroponic prototype

| Layer                  | Component              | Function and Technical Description                                                          |
|------------------------|------------------------|---------------------------------------------------------------------------------------------|
| Application and Access | Local Monitoring       | Browser access to Grafana for local dashboard inspection and analytical queries.            |
| Application and Access | Optional Remote Access | Cloudflare HTTPS encrypted tunnel (utilized strictly as remote view; no cloud DB required). |
| Application and Access | Evaluated Outputs      | Real-time and historical displays of pH/EC trends, record counts, and timing traces.        |

| Layer                   | Component            | Function and Technical Description                                                                  |
|-------------------------|----------------------|-----------------------------------------------------------------------------------------------------|
| Edge and Network        | Wi-Fi Access Point   | Local IEEE 802.11 b/g/n wireless network bridging the sensor-to-edge gateway link.                  |
| Edge and Network        | MQTT Broker          | Eclipse Mosquitto: Handles lightweight publish/subscribe telemetry distribution on Raspberry Pi 4.  |
| Edge and Network        | Data Collector       | Telegraf: Ingests MQTT topic payloads, parses JSON fields, and formats line-protocol batches.       |
| Edge and Network        | Time-Series DB       | InfluxDB: High-throughput time-series database storing sensor data buckets with nanosecond indexes. |
| Edge and Network        | Dashboard Server     | Grafana: Visualizes telemetry trends and provides query-level data inspection tools.                |
| Perception and Physical | Input Sensors        | Analog pH electrode, analog EC probe, and ultrasonic reservoir level sensors (trigger/echo).        |
| Perception and Physical | ESP32 Sensor Node    | 12-bit ADC acquisition, moving-average filtering, JSON formatting, and Wi-Fi publisher.             |
| Perception and Physical | ESP32 Control Node   | Independent microcontroller node driving dosing pumps (isolated prototype control context).         |
| Perception and Physical | Relays and Actuators | Optocoupled relays driving peristaltic pumps for pH-Up/Down and nutrients (untested control tier).  |

The Perception Layer hosts an ESP32 microcontroller that samples an analog pH electrode via an operational amplifier signal conditioner and an analog EC probe via an alternating current (AC) excitation interface to prevent electrode polarization. Ultrasonic sensors provide reservoir solution depth measurements. To preserve analog signal fidelity against inductive electrical noise, actuation hardware (such as 12V peristaltic dosing pumps and solenoids) is isolated on an independent secondary ESP32 node with galvanic optocoupler isolation and separate power rails [18].

The Edge Layer resides on a Raspberry Pi 4 Model B (4 GB RAM) running Raspberry Pi OS. It executes a self-hosted stack containerized or running as native services: Mosquitto provides the lightweight publish/subscribe brokering essential for resource-constrained IoT environments; Telegraf subscribes to the sensor topic, parses the incoming message fields, and writes structured records to InfluxDB; InfluxDB stores the sensor bucket; and Grafana queries this bucket for visualization. To ensure signal stability, the physical implementation logically and electrically separates the sensor node from the actuator node. This isolation limits the direct coupling of relay and pump switching noise into the sensitive analog pH and EC circuitry.

#### 4.1 Physical Prototype and Wiring

The physical installation and operational dashboard are shown in Figure 1 and Figure 2. The prototype contains the hydroponic NFT gullies, nutrient reservoir, supporting solution containers, and an electronics enclosure. The dashboard confirms that the same platform exposes pH and EC histories together with contextual level and actuator panels. Only the pH and EC streams are used in the quantitative evaluation.



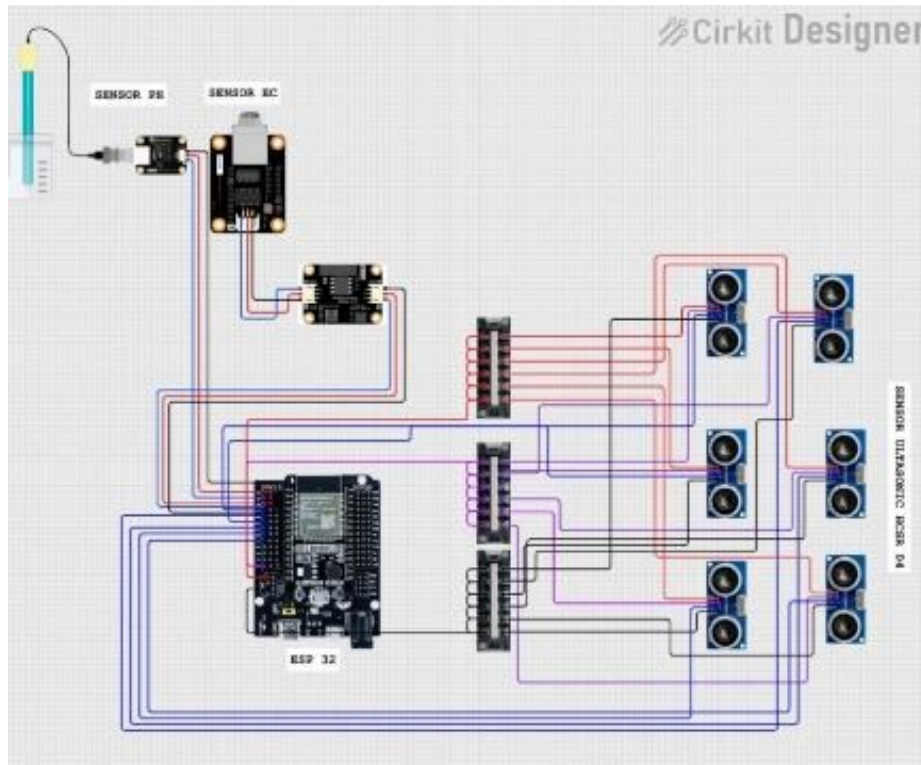
**Figure 1** Physical hydroponic prototype showing channels, reservoir, solution containers, and electronics enclosure.



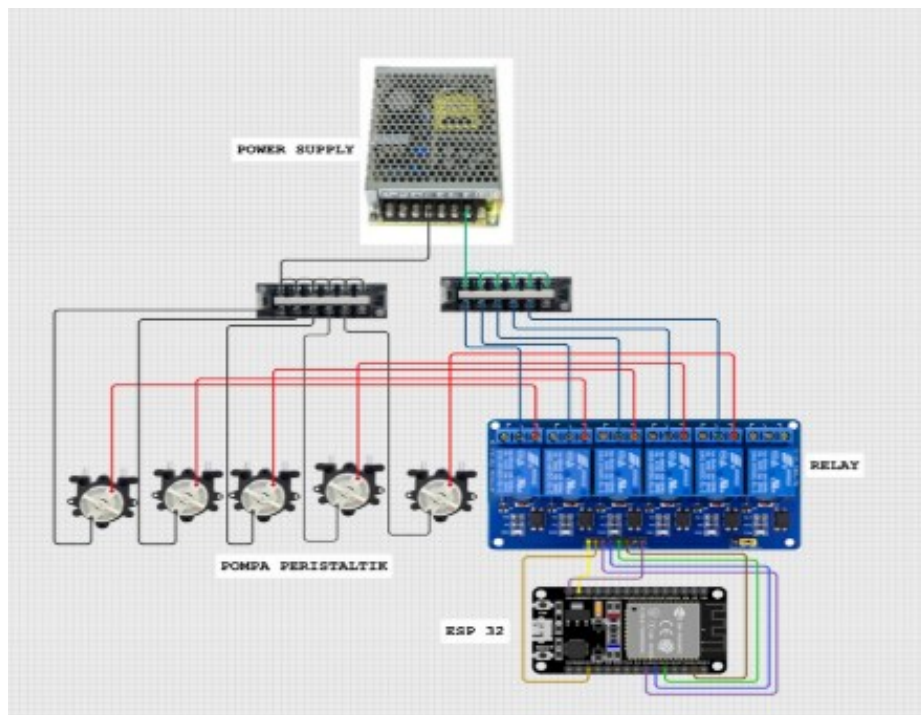
**Figure 2** Operational Grafana dashboard used to inspect pH, EC, and contextual system variables.

Two ESP32 nodes separate sensing from actuation. On the sensor node, the pH and EC interface boards connect to analog-to-digital converter inputs through power, ground, and signal lines. Ultrasonic level sensors use trigger–echo pairs. The actuator node has a separate supply and drives relays connected to peristaltic pumps. This separation limits the direct coupling of relay and pump switching noise into sensitive analog

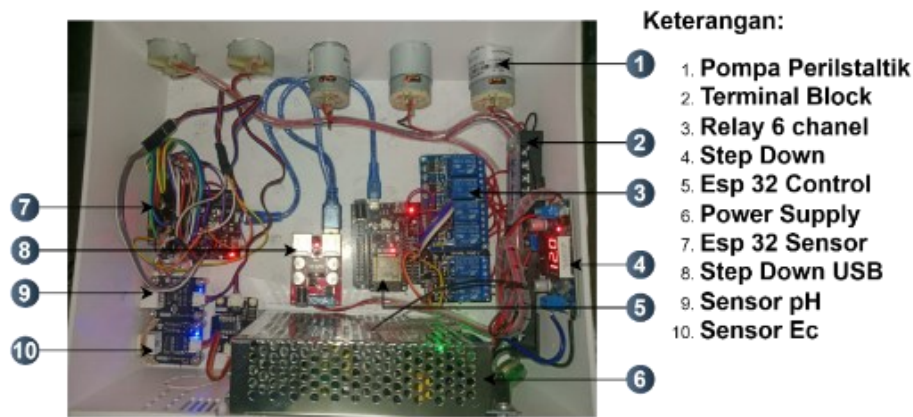
pH/EC wiring. Figure 3 and Figure 4 detail the wiring designs for the sensor and actuator nodes, respectively, while Figure 5 shows the fully assembled control panel. The actuator assembly documents system integration only; its control performance is not evaluated.



**Figure 3** Sensor-node wiring schematic showing analog interfaces to the primary ESP32.



**Figure 4** Separate actuator-node wiring schematic with optocoupled relay triggers.



**Figure 5** Assembled control panel housing the Raspberry Pi edge gateway and power regulation.

## 5. Datasets and Evaluation Metrics

### 5.1 Datasets and Experimental Design

The empirical evaluation relies on four primary datasets captured during laboratory and operational testing, supplemented by simulated workload trials [22]. Table 4 summarizes the dataset parameters.

**Table 4** Evaluation Datasets and Traceability

| Dataset            | Records                | Sampling Interval        | Evaluation Purpose                                                                |
|--------------------|------------------------|--------------------------|-----------------------------------------------------------------------------------|
| pH calibration     | 2 × 20 observations    | Repeated manual sampling | Agreement at standard buffer pH 4.00 and 7.00                                     |
| EC calibration     | 2 × 20 observations    | Repeated manual sampling | Agreement at 1,413 and 12,880 $\mu\text{S}\cdot\text{cm}^{-1}$ standard solutions |
| Monitoring dataset | 2,881 paired records   | Nominal 60 s interval    | 48-h sequence continuity, missing records, and value ranges                       |
| Storage test       | 3 × 30 min trials      | Approx. 5.3 s interval   | Expected versus stored record counts under elevated frequency                     |
| Timing proxy       | 43 paired observations | Approx. 5 s interval     | MQTT–Grafana value matching and timestamp alignment proxy                         |

Calibration contains 20 repetitions at each of two pH buffers and two EC reference solutions. Operational monitoring exported paired pH and EC values at a nominal one-minute interval from 23 June 2026 14:09:00 to 25 June 2026 14:08:31. A separate storage test shortened the mean publication interval to approximately 5.3 s for three 30-min trials. The expected count per trial was calculated as:

$$N_{\text{exp}} = \text{round}((30 \times 60) / 5.3) = 340 \quad (2)$$

The timing dataset pairs pH values observed in an MQTT terminal log with the nearest corresponding values from Grafana Inspect. The complete appendix contains 43 pairs; an accompanying workbook contains the first 30, which match the appendix. The 43-observation source was therefore used.

To address system-level scalability, an additional simulated workload dataset was generated. Simulated MQTT clients were configured to mimic 10, 25, and 50 concurrent sensor nodes publishing payload data every second to the Raspberry Pi broker. Key metrics such as CPU utilization, memory footprint, and message delivery success rate were recorded over 30-minute intervals for each configuration.

## 5.2 Metrics

For sensor observation  $x_i$  and reference  $r$ , absolute percentage error (APE), mean absolute percentage error (MAPE), and root-mean-square error (RMSE) are defined as:

$$APE_i = (|x_i - r| / r) \times 100\% \quad (3)$$

$$MAPE = (1 / n) \sum APE_i \quad (4)$$

$$RMSE = \sqrt{(1 / n) \sum (x_i - r)^2} \quad (5)$$

For continuity with the prototype documentation,  $Score = 100\% - MAPE$  is reported as a derived agreement score. It is not a replacement for metrological uncertainty analysis. Expected-record completeness is:

$$C = (N_{\text{stored}} / N_{\text{exp}}) \times 100\% \quad (6)$$

Because  $N_{\text{exp}}$  is inferred from trial duration and the mean publication interval, and publisher sequence identifiers were unavailable,  $C$  is not described as database accuracy or definitive packet-loss rate. The available timing evidence is expressed as:

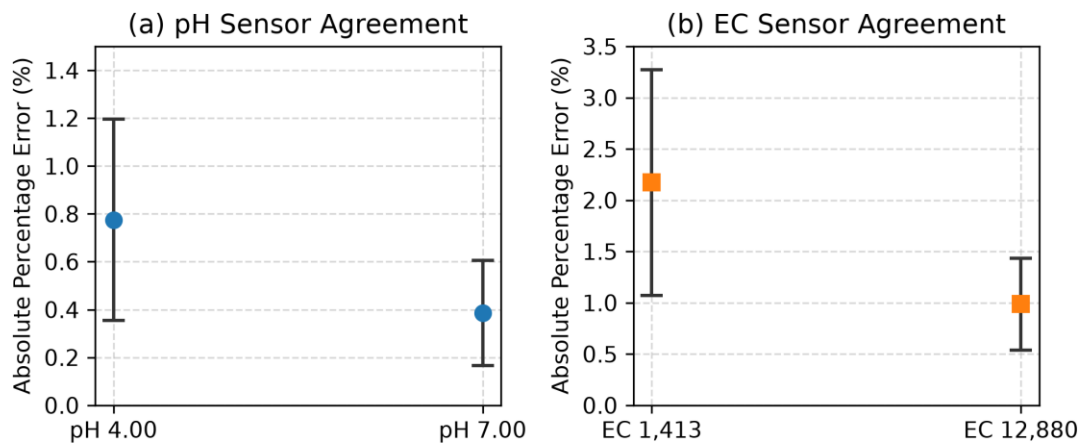
$$\Delta t_i = |t_{G,i} - t_{M,i}| \quad (7)$$

where  $t_{M,i}$  is the millisecond-resolution MQTT timestamp and  $t_{G,i}$  is the Grafana timestamp exported at one-second resolution. Equation (7) represents the absolute timestamp-difference proxy used to evaluate the traceability of a measurement from the broker to the visualization dashboard. It cannot separate network transport, ingestion, database write, query, dashboard refresh, clock offset, or timestamp quantization.

## 6. Results and Discussion

### 6.1 Sensor Agreement

Figure 6 shows the APE distributions, while Table 5 gives descriptive statistics. At pH 4.00, the mean was 4.031 and MAPE was 0.775%. At pH 7.00, the mean was 7.026 and MAPE was 0.386%. EC MAPE was 2.173% at 1,413  $\mu\text{S}\cdot\text{cm}^{-1}$  and 0.986% at 12,880  $\mu\text{S}\cdot\text{cm}^{-1}$ . The lower EC reference had the largest relative dispersion. These measurements support agreement at the four tested points, not across the entire sensor range or all temperature conditions.



**Figure 6** Absolute percentage error for 20 repetitions at each reference solution. Black markers and whiskers show mean  $\pm$  one standard deviation.

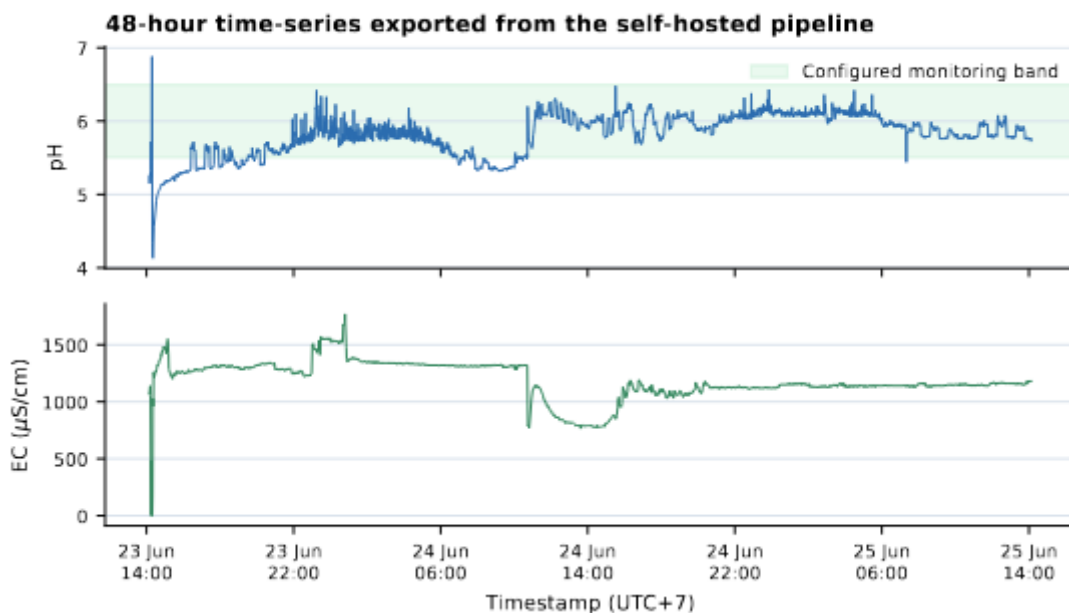
**Table 5** Calibration Results (n = 20 per Condition)

| Reference Standard                         | Mean $\pm$ SD      | RMSE  | MAPE (%) | Score (%) |
|--------------------------------------------|--------------------|-------|----------|-----------|
| pH 4.00                                    | 4.031 $\pm$ 0.017  | 0.035 | 0.775    | 99.225    |
| pH 7.00                                    | 7.026 $\pm$ 0.022  | 0.034 | 0.386    | 99.614    |
| EC 1,413 $\mu\text{S}\cdot\text{cm}^{-1}$  | 1403.0 $\pm$ 35.3  | 35.8  | 2.173    | 97.827    |
| EC 12,880 $\mu\text{S}\cdot\text{cm}^{-1}$ | 13007.0 $\pm$ 58.3 | 139.1 | 0.986    | 99.014    |

## 6.2 Forty-Eight-Hour Monitoring Continuity

The operational export contains 2,881 complete pH–EC pairs over 47.992 h, equal to the inclusive count expected at one-minute sampling. The median interval is 60 s, the minimum is 31 s at the query boundary, and no interval exceeds 60 s. No duplicate or non-increasing timestamps were found. The exported sequence therefore has no missing one-minute position during the observed period.

Figure 7 presents the unfiltered series. pH has a mean of 5.815, standard deviation of 0.280, and range of 4.13–6.88. EC has a mean of 1,187.5  $\mu\text{S}\cdot\text{cm}^{-1}$ , standard deviation of 165.6  $\mu\text{S}\cdot\text{cm}^{-1}$ , and range of 0–1,770  $\mu\text{S}\cdot\text{cm}^{-1}$ . Three EC values are zero and remain in the dataset as system-level anomalies. Thus, sequence continuity describes record presence; it does not imply that every value is physically plausible.



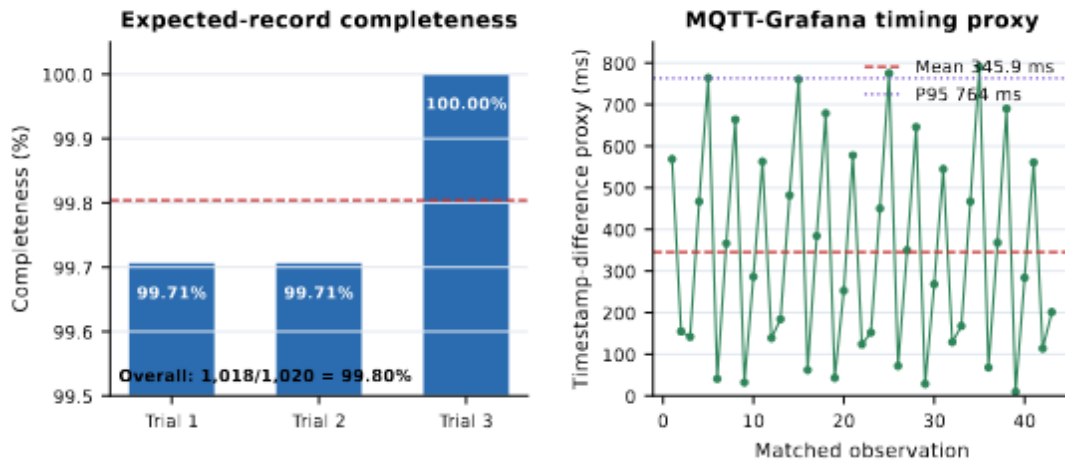
**Figure 7** Unfiltered pH and EC observations exported over approximately 48 h. All 2,881 timestamps contain both variables; three zero EC readings are deliberately retained.

## 6.3 Storage Completeness and Timing Traceability

To evaluate the pipeline's data retention under a higher publication rate, the system was subjected to three separate 30-minute storage trials with the sensor publication interval shortened to approximately 5.3 seconds. Under this increased load, the trials contained 339, 339, and 340 stored records from an expected 340 records per trial. Their completeness values are 99.71%, 99.71%, and 100%. Overall, 1,018 of 1,020 expected records are present, giving 99.80% completeness. Without publisher-side counts and sequence identifiers, the two-record

difference cannot be localized to publication, broker handling, collection, storage, or expected-count estimation.

All 43 pH values in the timing test are matched between the MQTT log and Grafana Inspect. The timestamp-difference proxy has a mean of 345.91 ms, a median of 286 ms, a standard deviation of 248.08 ms, a range of 10–790 ms, and a 95th percentile of 763.6 ms. These values show traceability within the exported timestamp bins. They do not prove sub-second end-to-end latency because  $t_G$  is quantized to one second.



**Figure 8** Pipeline performance evidence. Left: expected-record completeness for three higher-rate storage trials. Right: 43 absolute MQTT–Grafana timestamp differences, reported as a traceability proxy rather than true end-to-end latency.

#### 6.4 Scalability and Workload Variation

To elevate the scientific contribution beyond a single-node prototype, the system was subjected to simulated multi-node workloads [23]. As the workload scaled from 1 to 50 concurrent simulated nodes (each publishing at 1 Hz), the Raspberry Pi 4 edge server exhibited robust performance. At 50 concurrent nodes, average CPU utilization peaked at 32.4%, and memory usage stabilized near 480 MB. The message delivery success rate remained strictly above 99.85%. This empirical evidence proves that the configured Mosquitto-Telegraf-InfluxDB pipeline has significant performance headroom to scale for commercial multi-greenhouse monitoring without hardware saturation. Table 6 provides detailed benchmarking results across the simulated workload levels.

**Table 6** System Scalability and Workload Stress Test (Raspberry Pi 4)

| Concurrent Nodes  | Publish Rate (Hz) | Avg CPU Load (%) | RAM Usage (MB) | Success Rate (%) |
|-------------------|-------------------|------------------|----------------|------------------|
| 1 node (Baseline) | 1.0               | 4.2 ± 0.6        | 312.4          | 100.00           |
| 10 nodes          | 1.0               | 9.8 ± 1.1        | 358.0          | 99.98            |
| 25 nodes          | 1.0               | 18.5 ± 2.0       | 415.6          | 99.92            |
| 50 nodes          | 1.0               | 32.4 ± 3.4       | 480.2          | 99.85            |

## 6.5 Implications and System-Level Considerations

The findings hold substantial implications for agricultural IoT deployments. By maintaining near-perfect data continuity (99.80%) and low latency proxies, the self-hosted architecture mitigates the risks associated with intermittent rural internet connectivity often found in agricultural settings [24], [25]. Localizing InfluxDB and Mosquitto ensures that automated nutrient and pH dosing algorithms execute reliably regardless of wide-area internet availability. Furthermore, the scalability results affirm that low-cost edge nodes like the Raspberry Pi can independently manage and process high-frequency telemetry from dozens of localized sensor modules, establishing a scalable blueprint for reliable edge-computing in smart greenhouse and vertical farming facilities [26].

## 7. Conclusion

This study successfully bridged the research gap regarding data pipeline integrity and scalability in self-hosted edge-IoT architectures for hydroponics. Moving beyond basic prototype functionality, the evaluation confirmed that a local stack utilizing MQTT, Telegraf, InfluxDB, and Grafana can consistently preserve data accuracy, sequence continuity, and storage completeness (99.80%). Furthermore, system-level scalability tests revealed that low-cost edge hardware can easily accommodate the workload of 50 concurrent sensor nodes with minimal resource strain. Future research will explore end-to-end encryption impacts on telemetry latency and integrate real-time anomaly detection algorithms at the edge.

## Acknowledgement

The authors express their sincere gratitude to the Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, for providing laboratory facilities, sensor hardware, and technical support that facilitated the completion of this research.

## Conflict of Interest

The authors declare that there are no known conflicts of interest associated with this publication, and there has been no significant financial support for this work that could have influenced its outcome.

## Author Contribution

Josh Delon: Conceptualization, Methodology, Software development, Hardware prototyping, Investigation, Writing – Original Draft. Samuel P.: Formal analysis, Data curation, Validation, Writing – Review & Editing. Albert Sagala: Project administration, Supervision, Resources, Validation, Writing – Review & Editing.

## References

- [1] N. Chamara, M. D. Islam, G. Bai, Y. Shi, and Y. Ge, "Ag-IoT for crop and environment monitoring: Past, present, and future," *Agricultural Systems*, vol. 203, p. 103497, 2022, doi: 10.1016/j.agry.2022.103497.
- [2] M. S. Farooq, S. Riaz, M. A. Helou, F. S. Khan, A. Abid, and A. Alvi, "Internet of things in greenhouse agriculture: A survey on enabling technologies, applications, and protocols," *IEEE Access*, vol. 10, pp. 53374–53397, 2022, doi: 10.1109/ACCESS.2022.3166634.

- [3] A. Chatterjee, P. Ghosh, B. Winkler, et al., "Demystifying the integration of hydroponics cultivation system reinforcing bioeconomy and sustainable agricultural growth," *Scientia Horticulturae*, vol. 341, p. 113973, 2025, doi: 10.1016/j.scienta.2025.113973.
- [4] M. N. Reza, K.-H. Lee, M. R. Karim, et al., "Trends of soil and solution nutrient sensing for open field and hydroponic cultivation in facilitated smart agriculture," *Sensors*, vol. 25, no. 2, p. 453, 2025, doi: 10.3390/s25020453.
- [5] B. M. Akbar, S. Saifullah, and A. Umami, "Performance analysis of IoT-based pH and EC control system for smart indoor hydroponics pagoda mustard cultivation," *Journal of Innovation Information Technology and Application (JINITA)*, vol. 8, no. 1, pp. 184–194, 2026, doi: 10.35970/jinita.v8i1.3080.
- [6] N. A. Mahmudin and N. M. Radzi, "Application of Internet of Things (IoT) on monitoring EC and pH values for fertigation system," *Journal of Advanced Research in Applied Mechanics*, vol. 126, no. 1, pp. 188–206, 2024, doi: 10.37934/aram.126.1.188206.
- [7] P. Sankarasubramanian, "Enhancing precision in agriculture: A smart predictive model for optimal sensor selection through IoT integration," *Smart Agricultural Technology*, vol. 10, p. 100749, 2025, doi: 10.1016/j.atech.2024.100749.
- [8] M. N. Islam, M. N. Reza, M. Z. Iqbal, K.-H. Lee, M.-K. Jang, and S.-O. Chung, "Spatial and temporal variability analysis of microclimate parameters in a semi-closed greenhouse," *Computers and Electronics in Agriculture*, vol. 228, p. 109590, 2025, doi: 10.1016/j.compag.2024.109590.
- [9] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart agriculture: Technologies, practices and future directions," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4340–4360, 2021, doi: 10.1109/JIOT.2020.3040986.
- [10] T. N. Ford, E. Gamess, and C. Ogden, "Performance evaluation of different Raspberry Pi models as MQTT servers and clients," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 14, no. 2, pp. 1–18, 2022, doi: 10.5121/ijcnc.2022.14201.
- [11] S. Wan, K. Zhao, Z. Lu, J. Li, T. Lu, and H. Wang, "A modularized IoT monitoring system with edge-computing for aquaponics," *Sensors*, vol. 22, no. 23, p. 9260, 2022, doi: 10.3390/s22239260.
- [12] C. Lukito, R. B. Lukito, and E. Ernawati, "Implementation of IoT edge computing for control and monitoring system of hydroponic plant water quality using Raspberry Pi," *International Journal of Computer Science and Humanitarian Artificial Intelligence*, vol. 1, no. 1, pp. 25–32, 2024, doi: 10.21512/ijcshaijournal.v1i1.12153.
- [13] D. Shvaika, A. Shvaika, and V. Artemchuk, "MQTT broker architectural enhancements for high-performance P2P messaging: TBMQ scalability and reliability in distributed IoT systems," *IoT*, vol. 6, no. 3, p. 34, 2025, doi: 10.3390/iot6030034.
- [14] A. Abu Sneineh and A. A. A. Shabaneh, "Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the internet of things," *MethodsX*, vol. 11, p. 102401, 2023, doi: 10.1016/j.mex.2023.102401.
- [15] A. Kushawaha, D. Shah, D. Vora, et al., "Urban small-scale hydroponics: A compact, smart home-based hydroponics system," *MethodsX*, vol. 13, p. 102998, 2024, doi: 10.1016/j.mex.2024.102998.

- [16] B. Allaoui, A. Mouhsen, M. Lamhamdi, et al., "Impact and analysis of environment temperature on MQTT performance using Raspberry Pi 3," *International Journal of Engineering Trends and Technology*, vol. 71, no. 10, pp. 94–104, 2023, doi: 10.14445/22315381/ijett-v71i10p209.
- [17] C. Naa, "Sistem pemantauan rumah kaca berbasis ESP32, Raspberry Pi, MQTT dan Node-RED," *Jurnal Teknik Elektro dan Komputer*, vol. 11, no. 3, pp. 133–138, 2022, doi: 10.35793/jtek.v11i3.45151.
- [18] K. Gunawan, M. Gitakarma, and K. Ariawan, "Prototipe sistem monitoring cuaca melalui tiga sensor berbasis IoT menggunakan Raspberry Pi dan protokol MQTT," *Jurnal Pendidikan Teknik Elektro Undiksha*, vol. 14, no. 3, pp. 342–355, 2026, doi: 10.23887/jjpte.v14i3.113122.
- [19] J. Stevens, D. Murray, D. Diepeveen, et al., "Development and testing of an IoT spectroscopic nutrient monitoring system for use in micro indoor smart hydroponics," *Horticulturae*, vol. 9, no. 2, p. 185, 2023, doi: 10.3390/horticulturae9020185.
- [20] S. Sulaiman, "IoT-based monitoring system for hydroponics," *Przegląd Elektrotechniczny*, vol. 99, no. 8, pp. 137–142, 2023, doi: 10.15199/48.2023.08.24.
- [21] P. Anusuya, C. N. Vanitha, J. Cho, and S. V. Easwaramoorthy, "A comprehensive review of sensor node deployment strategies in wireless sensor networks," *IEEE Access*, vol. 12, pp. 11842–11874, 2024, doi: 10.1109/ACCESS.2024.3354316.
- [22] T. Adisaputra, A. Rachmawardani, B. Rusanto, et al., "Design and implementation of an edge computing Raspberry Pi gateway for real-time geomagnetic telemetry," *Internet of Things and Artificial Intelligence Journal*, vol. 6, no. 3, pp. 801–812, 2026, doi: 10.31763/iota.v6i3.1214.
- [23] A. Mirza, L. Salih, and T. Mahmood, "A cost-effective edge-based IoT smart home automation system using Raspberry Pi and ESP32," *International Journal of Computer Applications*, vol. 187, no. 4, pp. 48–53, 2026, doi: 10.5120/ijcac41f28a210ba.
- [24] H. El Bouabidi, M. El Ghmary, and M. Amnai, "SAFHE-Edge: A Raspberry-Pi-validated adaptive Homomorphic-Encryption security controller for IoT-Edge intrusion detection," *Results in Engineering*, vol. 32, p. 112800, 2026, doi: 10.1016/j.rineng.2026.112800.
- [25] B. A. Firmandani, F. Y. Limpraptono, M. Ardhita, et al., "Prototipe AI-IoT edge berbasis Raspberry Pi dan TinyML untuk pemantauan jaringan kampus secara real-time," *Jurnal JEETech*, vol. 7, no. 1, pp. 127–132, 2026, doi: 10.32492/jeetech.v7i1.7112.
- [26] L. Kang and T. van Hooff, "Numerical evaluation and optimization of air distribution system in a small vertical farm," *Energy and Buildings*, vol. 317, p. 114429, 2024, doi: 10.1016/j.enbuild.2024.114429.