A Microservices-Based IoT Analytics Architecture for Real-Time Environmental Monitoring

Pedro Manoel Herminio Alves[1], Cláudio de Souza Baptista[1], André Luiz Firmino Alves[2]

[1]Federal University of Campina Grande (UFCG) – Campina Grande – Paraíba – Brazil

[2]Federal Institute of Education of Paraíba (IFPB) – Picuí – PB – Brazil

`pedro.alves@copin.ufcg.edu.br, baptista@computacao.ufcg.edu.br`

`andre.alves@ifpb.edu.br`

Abstract. *This paper presents an IoT Analytics architecture for real-time environmental monitoring, designed to overcome the limitations of solutions that provide limited support for operational and historical data analysis. The proposal integrates continuous collection, stream processing, temporal analytical storage, and interactive visualization via dashboards within a microservices framework. An end-to-end pipeline was implemented using open-source tools, and an evaluation was conducted in a real-world scenario in the state of Acre, Brazil, in comparison with the platform currently used for air quality monitoring. In the usability evaluation, the proposed solution achieved mean scores between 6.20 and 6.65 (on a Likert scale from 1 to 7), while the reference solution ranged between 2.15 and 2.50. The results indicate consistent gains in real-time indicator retrieval, historical time-series exploration, and the execution of analytical tasks.*

1. Introduction

Environmental monitoring has become central to public health and ecosystem protection. Environmental sensors enable the continuous collection of metrics such as air quality, temperature, humidity, and atmospheric pressure. Among these metrics, air quality stands out due to its association with respiratory and cardiovascular diseases [Dapper et al. 2016]. International guidelines also establish exposure limits for atmospheric pollutants [World Health Organization 2021]. In this context, the Internet of Things (IoT) enables the tracking of these variables in real time by connecting devices and services capable of continuously collecting, transmitting, and integrating data [Raja et al. 2023].

Despite these advances, many monitoring scenarios still underutilize the collected data due to limitations in ingestion, analytical storage, and visualization, which compromises decision-making, especially in situations requiring a rapid response. In addition, low-cost sensor networks require extra care regarding data quality, interpretation, and measurement reliability [Castell et al. 2017, Williams et al. 2014]. This scenario is observed in the state of Acre, where a sensor network for air quality monitoring exists, but the solution currently employed, available through the Qualidade do Ar – Acre portal[1],

[1]`https://acre.qualidadedoar.net.br`

offers limited support for real-time functionalities, analytical queries, and interactive visualizations typical of Business Intelligence (BI) tools.

Although there are relevant IoT-based environmental monitoring initiatives, open-source solutions that integrate continuous ingestion, stream processing, temporal analytical storage, and interactive visualization—while being evaluated in real-use scenarios—are still less frequent. Given this context, this work proposes and implements an IoT Analytics solution based on open-source tools and microservices, aiming to collect, process, store, and visualize environmental data in real time and as historical series. The solution was conceived as a generic and replicable model for similar environmental monitoring scenarios and, in this work, was evaluated in the context of the state of Acre using air quality data. The implementation of the developed architecture is publicly available in a repository[2].

The main contributions of this work are: (i) the definition of a microservices-based IoT Analytics architecture organized into ingestion, processing, analytical storage, and visualization layers; (ii) the implementation of a continuous stream data pipeline using open-source tools, integrating environmental sensors with analytical dashboards; and (iii) a comparative evaluation of the visualization layer, based on the PSSUQ, relative to the system currently used in Acre, highlighting gains in usability and analytical task execution.

The remainder of the article is organized as follows: Section 2 discusses related work; Section 3 describes the proposed architecture, its components, and flows; Section 4 presents the evaluation of the solution; and Section 5 summarizes the conclusions and points to future work.

2. Related Work

To organize this section, the studies were grouped into two analytical axes: (i) end-to-end architectures and integration for the ingestion, processing, and visualization of environmental data; and (ii) measurement quality and reliability of low-cost sensors. This thematic organization facilitates the comparison between contributions with distinct objectives [Snyder 2019].

In the first axis, Lo et al. [Lo et al. 2019] propose a generic component-based IoT architecture aimed at interoperability and data-driven feedback across heterogeneous devices, but without emphasizing an end-to-end analytical pipeline for environmental data. De Vito et al. [De Vito et al. 2021] address air-quality crowdsensing through a hybrid fixed/mobile architecture centered on sensing nodes, calibration, and long-term validation. Martínez et al. [Martínez et al. 2023] focus on the hardware design of a low-cost monitoring device coupled to an IoT platform for visualization and data sharing. In contrast, our work emphasizes the analytical orchestration of environmental monitoring data through an open-source microservices architecture that explicitly separates continuous ingestion, stream processing, temporal analytical storage, and interactive dashboards, while also incorporating enrichment/correction routines and operational indicators.

Complementarily, Ramadan et al. [Ramadan et al. 2024] emphasize real-time monitoring and AI-based forecasting in industrial scenarios, while Marche et

[2] `https://github.com/pedro-manoel/iot-analytics-solution-tcc`

al. [Marche et al. 2025] discuss security and trust issues in crowdsensing architectures. In the second axis, Snyder et al. [Snyder et al. 2013] analyze the paradigm shift associated with low-cost sensors, highlighting both their potential and the limitations in data interpretation. Budde et al. [Budde et al. 2013] also discuss the feasibility of particulate matter measurements with low-cost sensors and stress the need for careful experimental validation. More recent studies reinforce this perspective through calibration and data quality improvement strategies, such as machine learning-based correction methods and AI techniques for atmospheric sensing [Balagopal et al. 2025, Montalbán-Faet et al. 2025].

Taken together, the literature advances important aspects of interoperability, participatory sensing, calibration, affordability, forecasting, and security. The main contribution of this work is to combine a different set of priorities in a single open-source solution: microservices-based organization, continuous ingestion and stream processing, analytical storage optimized for time-series queries, and interactive dashboards evaluated in a real-use public monitoring context. This positions the proposal as an IoT Analytics architecture focused on the analytical lifecycle of environmental data—from acquisition to time-series exploration—rather than primarily on generic interoperability, crowdsensing infrastructure, or low-cost device design.

3. Architecture

The proposed architecture was designed to meet the requirements of scalability, decoupling, and continuous processing characteristic of IoT-based environmental monitoring environments. Considering the progressive growth in the volume of data generated by sensor networks and the need for both real-time and retrospective analysis, a microservices-based approach was adopted, in which components are independent, specialized, and communicate through asynchronous mechanisms [Rath et al. 2023].

The adoption of a microservices-oriented architecture favors the independent evolution of modules, fault isolation, and the horizontal scalability of specific components according to demand, overcoming the limitations of monolithic approaches in IoT scenarios [Ouyang et al. 2023]. Furthermore, the solution was designed to favor interoperability and flexibility, which are relevant attributes in ecosystems with multiple data sources and heterogeneous analytical requirements [Lo et al. 2019].

The solution was organized into four functional layers: ingestion, processing, analytical storage, and visualization. This separation of responsibilities increases the clarity of the data flow and facilitates the replacement or expansion of components without significant impact on other services. Figure 1 presents the overview of the architecture.

3.1. Architecture Components

The architecture integrates well-established open-source tools from the data engineering ecosystem, selected for their technological maturity and suitability for continuous data stream scenarios. In the ingestion layer, Apache NiFi orchestrates the collection flows and offers backpressure, prioritization, and monitoring mechanisms, contributing to the reliability of periodic sensor reading acquisition. The decoupling between ingestion and processing is handled by Apache Kafka, employed as an event bus to absorb load variations and allow asynchronous consumption of readings.

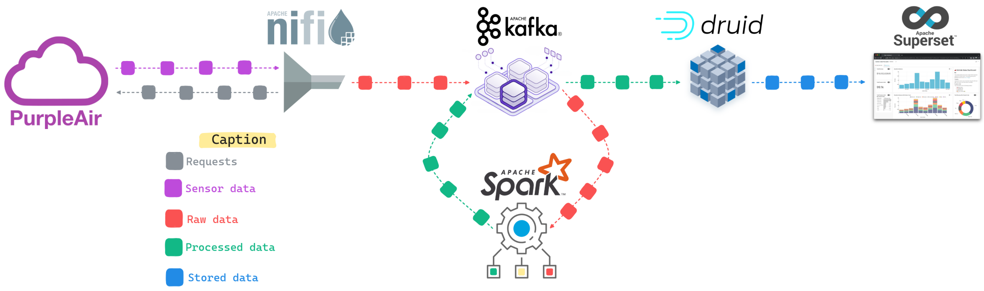


Figure 1. Architecture of the IoT Analytics solution.

In the processing layer, Apache Spark was adopted for its integration with Kafka and for allowing the implementation of transformation and enrichment routines with low coupling. For analytical storage, Apache Druid offers support for low-latency queries over time series, while Apache Superset composes the visualization layer, enabling interactive dashboards and analytical exploration. The exclusive adoption of open-source tools also promotes transparency, auditability, and reduced operational costs, factors particularly relevant in institutional and governmental contexts. Regarding deployment, services were executed in Docker containers with orchestration via Docker Compose and persistent volumes.

3.2. Flows

The operation of the architecture is structured into four flows corresponding to the functional layers: ingestion, processing, storage, and visualization. Each flow defines the interaction between two or more services to fulfill a specific stage of the data pipeline.

3.2.1. Ingestion

The ingestion flow performs periodic and automated collection of sensor readings. At this stage, Apache NiFi executes requests at regular intervals, applies initial validations, formats and transforms the data into JSON, and then publishes them to Apache Kafka, making them available to the processing flow.

For the real-time scenario, the collection mechanism incorporates an auxiliary service for obtaining temporary credentials to access the data source. Although this strategy allowed for the operationalization of integration in the studied case, it evidences an external dependency that affects the portability of the solution and should be treated as an architectural limitation.

Raw records remain in Kafka for a limited retention period, a strategy that reduces storage costs in this layer and reinforces the decoupling between ingestion and processing while keeping the system responsive to load variations.

3.2.2. Processing

In Apache Spark, processing routines were implemented using PySpark to transform raw data into analytical structures suitable for temporal storage and visualization. To reduce traffic in the ingestion layer, readings from Kafka are combined with immutable metadata files in JSONL format, containing static information such as identifiers, municipalities, and geographic coordinates of the sensors. This approach shifts to the processing layer, reducing the redundancy of transmitted data.

As low-cost sensor networks require additional care regarding interpretation and quality [Castell et al. 2017, Williams et al. 2014], a central processing step consists of applying the LRAPA correction formula to the particulate matter field ($PM_{2.5}$), following the reference formulation reported by Barkjohn et al. [Barkjohn et al. 2021].

In addition to this correction, the flow performs attribute rounding, time zone adjustment, transformation of the RSSI value into a positive Wi-Fi signal strength indicator, and calculation of the time elapsed since the last reading, allowing for the derivation of operational indicators to monitor sensor network availability. Figure 2 illustrates the difference between raw data and data after processing.

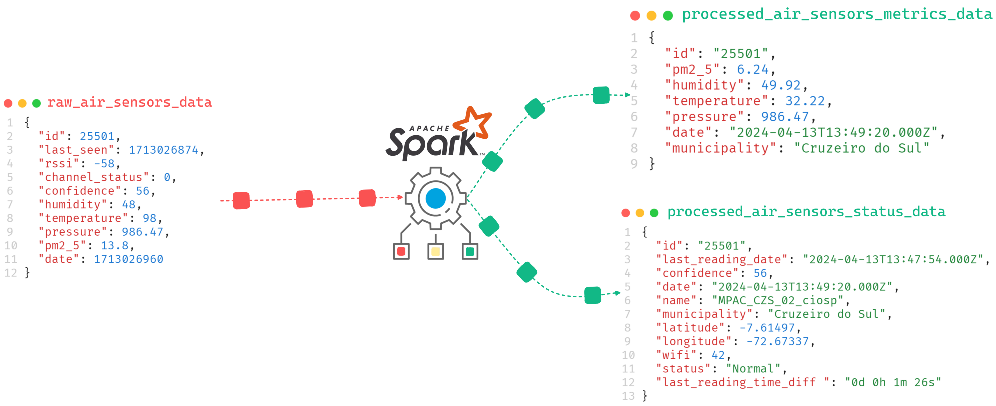


Figure 2. Sensor data before and after Apache Spark processing.

3.2.3. Storage

The storage flow receives the processed data produced in the previous step and makes it available for analytical queries. This flow is composed of Apache Kafka and Apache Druid.

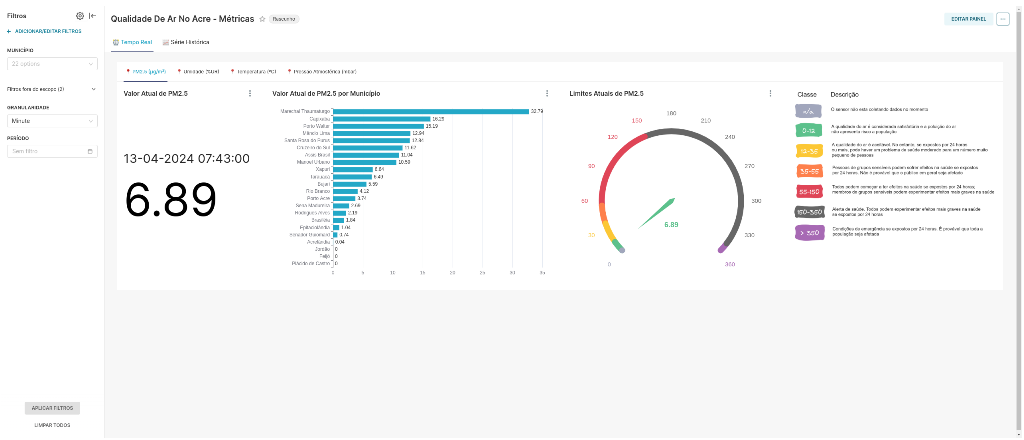
In Apache Druid, streaming ingestion tasks were configured and connected to the processed data topics in Kafka. For real-time queries, users directly access segments currently being ingested. For historical queries, Druid moves completely ingested segments to deep storage and loads them on demand into historical servers to answer queries. This organization reconciles continuous updates with support for low-latency historical series.

3.2.4. Visualization

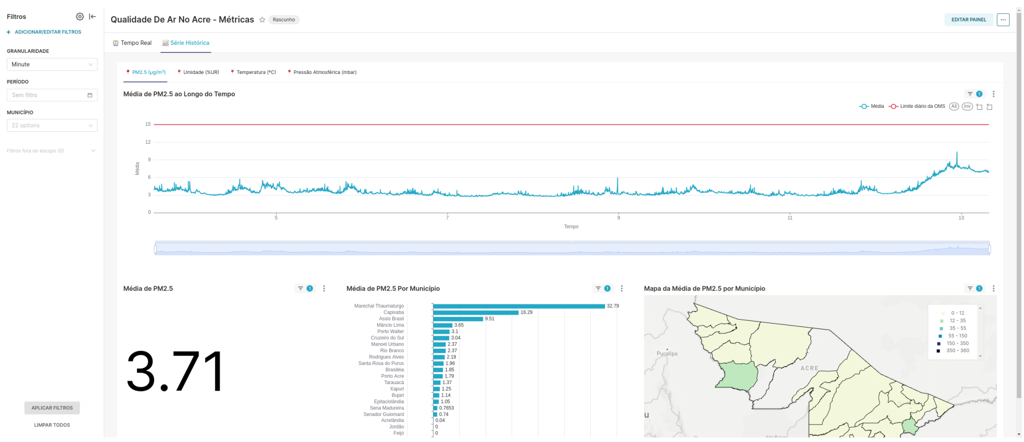
The visualization flow constitutes the analytical layer of the solution and is composed of Apache Druid and Apache Superset. In Superset, a connection to Druid is established and datasets are used to build the dashboards. Two dashboards were developed with complementary objectives: one focused on environmental metrics and another on tracking the status of the sensors. The design of the visualizations considered the existing solution in Acre and incorporated functionalities absent or difficult to explore in the legacy interface, such as dynamic filtering and greater analytical flexibility.

Metrics Dashboard This dashboard meets the primary demand of the solution: allowing for a simple and intuitive analysis of the environmental metrics collected by the sensors. It was organized into two views — real-time and historical series — and, in each one, there are four tabs corresponding to the metrics: $PM_{2.5}$, humidity, temperature, and atmospheric pressure.

Figure 3 presents the two dashboard views side by side: real-time (Figure 3(a)) and historical series (Figure 3(b)). In the real-time visualization, each tab presents the current value of the metric, a bar chart for comparison between municipalities in Acre, and a gauge to contextualize the current value within appropriate ranges. In the historical series visualization, each tab displays a line chart, a textual summary of the aggregated value, a bar chart by municipality, and a choropleth map showing the spatial distribution of the metric in the state.



(a) Real-time $PM_{2.5}$ dashboard.



(b) Historical series $PM_{2.5}$ dashboard.

Figure 3. Views of the environmental metrics dashboard.

Sensor Dashboard This dashboard addresses a complementary need: tracking the status of the sensors and supporting analyses of availability, reliability, and operational performance. This monitoring is relevant because devices remain exposed to adverse conditions, which can affect the continuity of measurements and communication quality [Aikhuele et al. 2022].

The panel gathers indicators such as Wi-Fi signal strength, reliability, operational status, and the record of the last reading, enabling rapid identification or communication degradation. In addition to the real-time and historical series views, this dashboard includes an overview with a sensor location map, a bar chart with counts by municipality, a pie chart with distribution by status, and a table with supporting information, such as sensor identification, reliability, and time since the last reading, as illustrated in Figure 4.

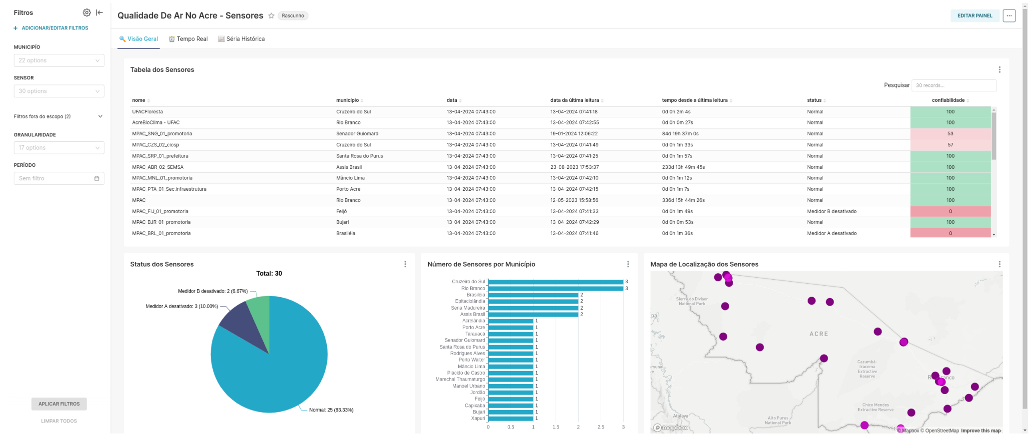


Figure 4. Overview of the sensor dashboard.

4. Evaluation

The evaluation focused on the visualization layer of the architecture through a comparative study between the solution developed in this work and the tool currently in use in Acre, available on the Qualidade do Ar – Acre portal. The objective was to analyze to what extent the proposed interface offers better support for real-time information retrieval, historical data exploration, and the execution of analytical tasks.

To this end, a questionnaire based on the Post-Study System Usability Questionnaire (PSSUQ), version 3, was applied and adapted to the research context. Since the focus was on functionalities most directly related to interface use, the instrument was reduced to seven statements, covering simplicity of use, ease of learning, speed of task execution, information retrieval, screen organization, interface satisfaction, and intention to recommend. Responses were recorded on a Likert scale from 1 to 7, where higher values indicate more positive evaluations. Thus, the results were analyzed descriptively, comparing the two interfaces.

Twenty users participated in the evaluation. Each participant used both solutions and, after a brief explanation of their purposes, performed five tasks: identifying the current value of $PM_{2.5}$ in Rio Branco; identifying the average temperature in the state; checking how many sensors had meter A deactivated; consulting the current Wi-Fi signal value of sensor `MPAC_SNM_01_ifac`; and identifying, from historical data from 2019 to

2023, the date of the highest average $PM_{2.5}$ in Acre. Subsequently, they responded to the questionnaire for each interface.

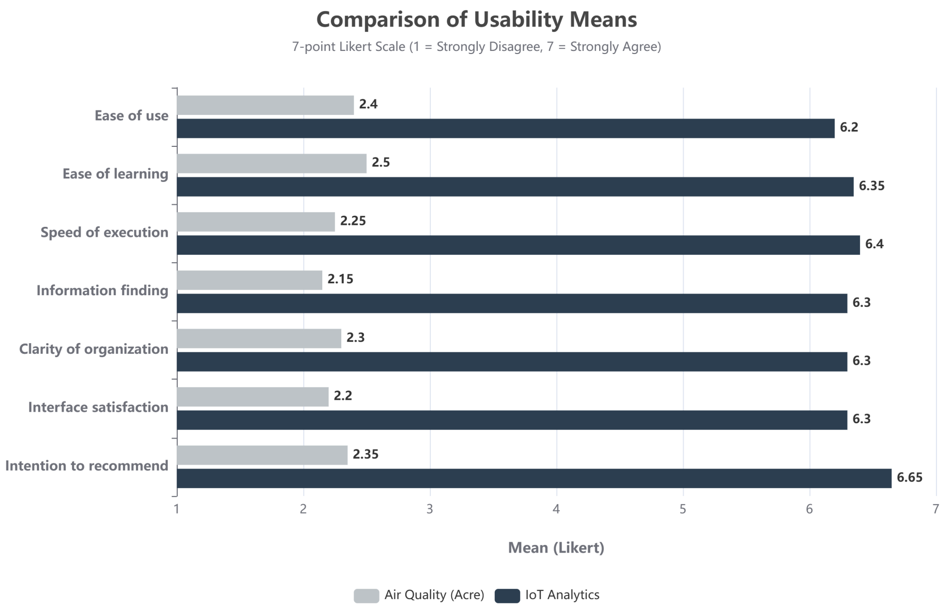


Figure 5. Usability comparison of the solutions.

The results presented in Figure 5 show a consistent advantage for the proposed solution across all evaluated dimensions. While the existing interface presented averages between 2.15 and 2.50, the IoT Analytics solution obtained values between 6.20 and 6.65. The largest difference was observed in “Information location”, a dimension in which the legacy solution reached an average of 2.15, while the proposed interface reached 6.30.

Task performance reinforces this result. It was observed that 58% of the tasks were completed exclusively with the developed solution, 40% could be completed in both solutions, and only 2% were completed exclusively with the legacy interface. Together, these findings indicate that the proposed visualization layer offers better usability and greater support for the execution of analytical tasks than the tool currently in use.

5. Conclusion

This work presented the design and implementation of a generic and replicable IoT Analytics architecture for environmental monitoring, based on microservices and open-source tools, integrating ingestion, processing, analytical storage, and visualization. In the context of this article, the solution was applied and evaluated in the state of Acre, focusing on air quality data. The approach enables the analysis of environmental metrics in real time and in historical series and incorporates dashboards dedicated to monitoring the data captured by sensors and their operational indicators, contributing to analyses of availability, reliability, and measurement behavior.

In the comparative study conducted in Acre, the proposed solution showed a better perception of usability compared to the previously adopted system, especially regarding access to information, interface organization, and support for data exploration. In quantitative terms, the averages for the proposed solution varied between 6.20 and 6.65, while the legacy interface varied between 2.15 and 2.50; additionally, 58% of the tasks were completed exclusively with the developed solution. These results indicate gains that go beyond perceived satisfaction and are reflected in the practical ability to locate and interpret relevant information.

As limitations, the evaluation focused on the visualization layer and did not include specific experiments on architecture latency, scalability, or fault tolerance. Furthermore, although processing includes correction and enrichment mechanisms aimed at improving analytical quality, the work did not include systematic local validation of sensors against reference instruments. Finally, part of the data acquisition mechanism depends on external conditions for accessing the source platform, which may affect reproducibility and maintenance in future deployments.

For future work, the evaluation should be expanded to include a larger and more diverse group of participants, as well as experiments on architectural performance, scalability, and resilience. It is also important to incorporate systematic strategies for sensor calibration and validation and to improve data-source integration in order to reduce external dependencies. In parallel, machine learning models may be integrated to support air quality forecasting, sensor-failure anomaly detection, and automatic calibration routines.

Additional directions include investigating edge computing for local preprocessing under intermittent connectivity, validating the architecture through multi-region deployments, strengthening compliance with WHO guidelines and national regulations, and extending access through public APIs or citizen-facing applications. A cost-benefit analysis comparing the proposed solution with commercial alternatives would also help assess its long-term viability. Nevertheless, due to its modularity, replicability, and observed performance in the evaluated scenario, the solution shows potential for adaptation to other environmental monitoring contexts.

References

Aikhuele, D. O., Nwosu, H. U., and Ighravwe, D. E. (2022). Data-driven model for the evaluation of the reliability of sensors and actuators used in IoT system architecture. *Journal of Reliable Intelligent Environments*, 9(2):135–145.

Balagopal, G., Wijeratne, L., Waczak, J., Hathurusinghe, P., Iqbal, M., Kiv, D., Aker, A., Lee, S., Agnihotri, V., Simmons, C., and Lary, D. J. (2025). Calibration of low-cost LoRaWAN-based IoT air quality monitors using the super learner ensemble: A case study for accurate particulate matter measurement. *Sensors*, 25(5):1614.

Barkjohn, K. K., Gantt, B., and Clements, A. L. (2021). Development and application of a united states-wide correction for pm2.5 data collected with the purpleair sensor. *Atmospheric Measurement Techniques*, 14(6):4617–4637.

Budde, M., El Masri, R., Riedel, T., and Beigl, M. (2013). Enabling low-cost particulate matter measurement for participatory sensing scenarios. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 1–10, New York, NY, USA. Association for Computing Machinery.

Castell, N., Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., Broday, D., and Bartonova, A. (2017). Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, 99:293–302.

Dapper, S. N., Spohr, C., and Zanini, R. R. (2016). Poluição do ar como fator de risco para a saúde: uma revisão sistemática no estado de são paulo. *Estudos Avançados*, 30(86):83–97.

Lo, S. K., Liew, C. S., Soon, T. K., and Mekhilef, S. (2019). An interoperable component-based architecture for data-driven IoT system. *Sensors*, 19(20):4354.

Marche, C., Salvato, M., De Vito, S., Formisano, F., D'Elia, G., D'Auria, P., and Di Francia, G. C. (2025). Secure and trusted crowdsensing for outdoor air quality monitoring: State of the art and perspectives. *Sensors*, 25(12):3573.

Montalbán-Faet, G., Meneses-Albala, E., Felici-Castell, S., Pérez-Solano, J. J., and Segura-Garcia, J. (2025). Improving raw readings from low-cost ozone sensors using artificial intelligence for air quality monitoring. *Atmospheric Measurement Techniques*, 18:4357–4372.

Ouyang, R., Wang, J., Xu, H., Chen, S., Xiong, X., Tolba, A., and Zhang, X. (2023). A microservice and serverless architecture for secure IoT system. *Sensors*, 23(10):4868.

Raja, P., Kumar, S., Yadav, D. S., and Singh, T. (2023). The internet of things (IoT): A review of concepts, technologies, and applications. *International Journal of Information Technology & Computer Engineering*, 3(2):21–32.

Ramadan, M. N. A., Ali, M. A. H., Khoo, S. Y., Alkhedher, M., and Alherbawi, M. (2024). Real-time IoT-powered AI system for monitoring and forecasting of air pollution in industrial environment. *Ecotoxicology and Environmental Safety*, 283:116856.

Rath, A. R., Pradhan, C., Swain, K. S., Iyer, V., Pati, B. K., and Wang, C.-T. (2023). A microservice-based scalable IoT architecture for a highly interoperable environment. *Internet of Things*, 23:100854.

Snyder, E. G., Watkins, T. H., Solomon, P. A., Thoma, E. D., Williams, R. W., Hagler, G. S. W., Shelow, D., Hindin, D. A., Kilaru, V. J., and Preuss, P. W. (2013). The changing paradigm of air pollution monitoring. *Environmental Science & Technology*, 47(20):11369–11377.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339.

Williams, R., Kilaru, V., Snyder, E., Kaufman, A., Dye, T., Rutter, A., Russell, A., and Hafner, H. (2014). Air sensor guidebook. Technical Report EPA/600/R-14/159, U.S. Environmental Protection Agency, Washington, DC.

World Health Organization (2021). *WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*. World Health Organization.