Edge-Based IoT and AI Framework for Real-Time Wastewater Potability Classification

José Isidro[1], Rafael Teixeira[1], João Costa[1], Carolina Gonçalves[1], Diogo Ferreira[1], Pedro Azevedo[1], Pedro Simões[1], Rui Pinto[1,2], Gil Gonçalves[1,2]
Dept. de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal[1]
SYSTEC, ARISE, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal[2]
Email:{up202108831, up202006485, up202108714, up202108781, up202205295, up201905966, up202403063} @edu.fe.up.pt
{rpinto, gil}@fe.up.pt

***Abstract*—Access to safe drinking water is a critical public health requirement. However, conventional water quality monitoring approaches remain labor-intensive, intermittent, and heavily dependent on delayed laboratory analyses or cloud-centric infrastructures. Such architectures introduce latency, connectivity dependencies, and limited operational resilience in decentralized or bandwidth-constrained systems. This paper presents a proof-of-concept edge-based Internet of Things (IoT) framework for real-time wastewater potability classification. The proposed system integrates an Arduino-based sensing node for physicochemical data acquisition with a Raspberry Pi edge gateway executing local machine learning inference. By relocating data processing and decision-making from the cloud to the edge, the system enables low-latency, autonomous classification while preserving data locality and operational continuity under limited connectivity. Experimental results demonstrate that resource-constrained edge hardware is capable of supporting real-time water quality assessment and classification, validating the feasibility of edge-centric architectures as a scalable and cost-effective alternative for resilient water quality monitoring systems.**

Keywords-Internet of Things; Edge Computing; Water Quality Monitoring; Artificial Intelligence; Wastewater Treatment.

I. INTRODUCTION

Access to safe drinking water is critical for public health, yet water quality monitoring systems remain largely reactive, relying on labor-intensive sampling procedures and delayed laboratory analyses. In wastewater and industrial water infrastructures, contamination events can evolve rapidly, posing significant risks to environmental safety, economic stability, and community health if not promptly detected and mitigated [1][2]. Consequently, the core problem addressed in this work is the inability of existing Smart Water Quality Monitoring Systems (SWQMS) [3] to support timely, autonomous decision-making under strict latency, reliability, and data-governance constraints.

Recent advances in the Industrial Internet of Things (IIoT) have improved the granularity and frequency of water quality data acquisition. However, the majority of deployed SWQMS architectures remain fundamentally cloud-centric, outsourcing data processing and decision logic to remote data centers. This design introduces a critical latency bottleneck, defined by the round-trip time required for sensor data to reach the cloud and for control actions to be issued in response [4]. In safety-critical wastewater scenarios, such delays can prevent the system from reacting within the narrow temporal window required to contain contamination events, undermining its ability to function as a real-time "digital reflex arc" [5][6].

In addition to latency issues, cloud-based SWQMS solutions suffer from limited operational resilience. The reliance on continuous connectivity creates a single point of failure: when communication with the cloud is disrupted, intelligent monitoring and control capabilities are degraded or lost entirely. Furthermore, the continuous transmission of raw sensor data to external infrastructures raises concerns related to data sovereignty, privacy, and regulatory compliance within industrial environments. Collectively, these limitations reveal a misalignment between the architectural assumptions of cloud-centric SWQMS and the real-time, reliability-critical requirements of water quality management.

To address this gap, this paper proposes an Edge-based SWQMS grounded in edge analytics, where Machine Learning (ML) inference is performed directly at the network perimeter. By relocating the intelligence layer from Cloud to Edge, the proposed system enables low-latency, autonomous decision-making while preserving full functionality under intermittent or absent external connectivity. This architectural shift directly targets the identified problem by prioritizing speed, reliability, and data locality as first-class design objectives.

Accordingly, the main contributions of this work are:

1) The design and implementation of an Arduino-based sensor node for continuous, real-time acquisition of physicochemical water quality parameters.
2) The deployment of a localized ML inference engine, hosted on a Raspberry Pi and exposed through a FastAPI backend, enabling autonomous and low-latency decision-making at the Edge.
3) The development of an integrated local dashboard that provides real-time operational insights while ensuring that all sensitive telemetry remains confined to the on-premise network.

The remainder of this paper is organized as follows: Section II reviews Related Work; Section III details the Methods and System Architecture; Section IV describes the Experimental Setup; Section V presents Results and Discussion; and Section VI concludes the study and outlines future work.

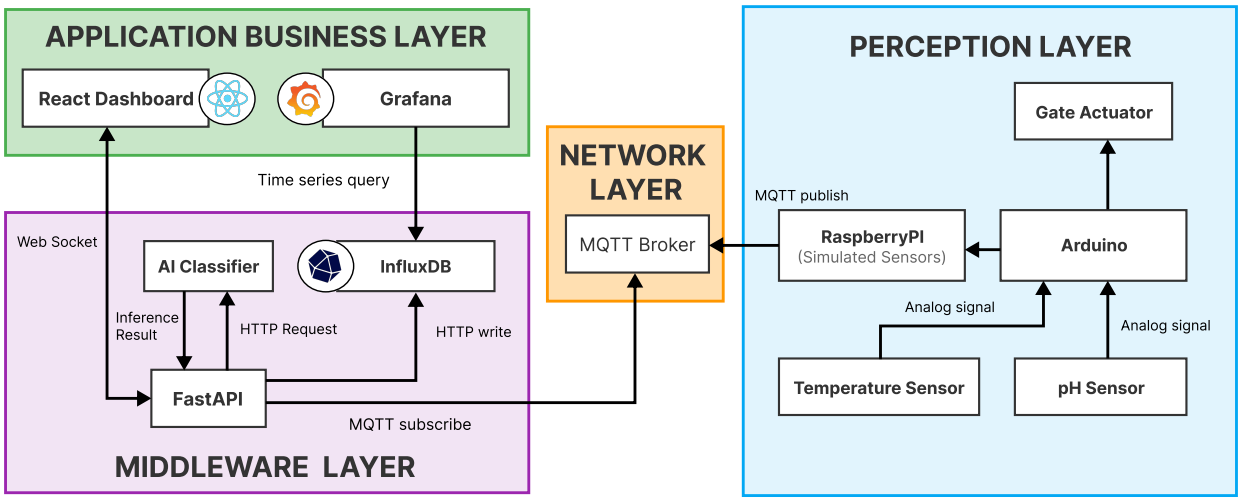

Figure 1. High-level architecture diagram illustrating the overall system design.

II. Related Work

The management of industrial and municipal wastewater is subject to strict environmental regulations, requiring frequent and reliable spatiotemporal monitoring to ensure public health protection and environmental sustainability [7]. Within this context, *Water Quality Monitoring* (WQM) refers to the systematic assessment of physicochemical and biological parameters—such as pH, temperature, turbidity, conductivity, and dissolved solids—to determine water safety and compliance with regulatory thresholds. Traditionally, WQM has played a central role in pollution detection, regulatory enforcement, and operational control in water treatment and distribution systems.

Historically, WQM has relied predominantly on manual sampling followed by laboratory-based analysis. While this approach provides high measurement accuracy, it presents several limitations [8]. Sampling campaigns are typically sporadic and spatially sparse, resulting in limited coverage and reduced sensitivity to short-lived contamination events. Moreover, the inherent delay between sample collection and laboratory results introduces significant latency, preventing timely responses in critical situations where rapid intervention is required [9]. In addition to these constraints, laboratory-based monitoring is labor-intensive and costly, requiring specialized equipment and trained personnel, which limits its feasibility for continuous or large-scale deployment [10].

To address these shortcomings, recent research and industrial practice have increasingly embraced the IIoT paradigm, giving rise to SWQMS [3]. SWQMS leverage distributed sensing, embedded computing, and networked communication to enable continuous, automated, and near real-time monitoring of water quality parameters. By replacing manual sampling with in situ sensors and automated data acquisition, these systems improve temporal resolution, enhance situational awareness, and reduce operational costs. Furthermore, IIoT-based approaches promote system decentralization, interoperability, and scalability, supporting heterogeneous devices and facilitating integration with higher-level data analytics and decision-support tools [11].

From a systems perspective, SWQMS typically build upon the generic layered architecture widely adopted in IoT applications. In its most common form, this architecture consists of five conceptual layers [12]–[15]. The *Sensing Layer* interfaces directly with the physical environment and includes sensors and embedded devices responsible for data acquisition. The *Network or Gateway Layer* ensures data transmission and aggregation, using wired or wireless communication technologies, such as Wi-Fi, cellular networks, or industrial field-buses. Collected data are then handled by a *Data or Cloud Layer*, where information is ingested via protocols, such as Message Queuing Telemetry Transport (MQTT) or Hypertext Transfer Protocol (HTTP), and stored, often in scalable Not Only SQL (NoSQL) databases. On top of this, the *Analytics Layer* processes raw telemetry into actionable knowledge, performing tasks such as anomaly detection, threshold-based alerts, and performance indicator computation [16]. Finally, the *Presentation Layer* provides user-facing interfaces for visualization, monitoring, and control, enabling operators to observe system status in real time and, in some cases, trigger remote actuation [17][18].

While this layered architecture is effective, most implementations remain cloud-centric. Recent efforts have begun exploring edge computing for environmental monitoring. For instance, Yadav *et al.* [19] proposed a framework for simultaneous air and water monitoring, emphasizing the reduction of network load. Similarly, Ren *et al.* [20] demonstrated how edge nodes can perform preliminary data filtering to enhance system longevity. Unlike these approaches, which often still treat the edge as a pre-processing buffer for the cloud, our solution proposes a fully autonomous edge gateway capable of localized Machine Learning (ML) inference and decision-making, ensuring operation even during total backhaul failure. This architectural shift prioritizes speed and data locality.

III. Methods and System Architecture

Unlike traditional cloud-centric approaches, this proof of concept adopts a local edge-based architecture, illustrated in Figure 1. The system operates on a Raspberry Pi coupled with an Arduino Nano acting as the sensor node, enabling low-latency data processing and AI inference without reliance on external cloud services [21][22]. This design prioritizes responsiveness, privacy, and operational continuity in environments with limited or unreliable internet connectivity.

The Arduino Nano serves as the primary interface with the physical environment, handling analog signal acquisition and actuator control. It performs analog-to-digital conversion for connected sensors and directly actuates elements, such as LED indicators, based on classification feedback received from higher layers. While the pH sensor is fully integrated into the hardware, the system accounts for missing sensing equipment—namely, Total Dissolved Solids (TDS), turbidity, and hardness sensors—through a custom simulation engine. This engine, hosted on the Raspberry Pi, generates bounded synthetic sensor data, enabling full system validation and end-to-end testing despite partial hardware availability.

Data exchange between the embedded hardware and processing components is achieved using the MQTT protocol via a Mosquitto broker [23]. This event-driven communication model was selected due to its lightweight overhead and robustness under unstable network conditions. The Arduino Nano publishes raw telemetry data to predefined topics, which are asynchronously consumed by backend services. This decoupled design supports scalability and flexibility, allowing additional sensor nodes or parallel dashboards to be integrated with minimal changes to the communication layer.

At the core of the system, a middleware component functions as the central intelligence layer, bridging data acquisition and user interaction. Implemented using FastAPI [24], this layer enables high-throughput data ingestion and low-latency processing. Incoming data streams are analyzed in real time using a local scikit-learn classification model, which evaluates water chemistry and produces potability predictions within seconds. Simultaneously, all measurements are persisted in InfluxDB [25], a time-series database optimized for high-frequency sensor data. By performing AI inference locally and avoiding cloud-based computation, the system ensures data privacy and remains fully operational in offline scenarios.

The user interface was designed to address the needs of both technical and non-technical stakeholders. Grafana [26] supports in-depth engineering analysis through detailed visualizations of raw sensor data and system behavior over time. In parallel, a lightweight React-based web application [27], shown in Figure 2, provides an intuitive interface for end-users. This application uses WebSockets to maintain a persistent connection with the middleware, ensuring that sensor updates, classification results, and alerts are reflected instantly on the dashboard without requiring manual page refreshes.

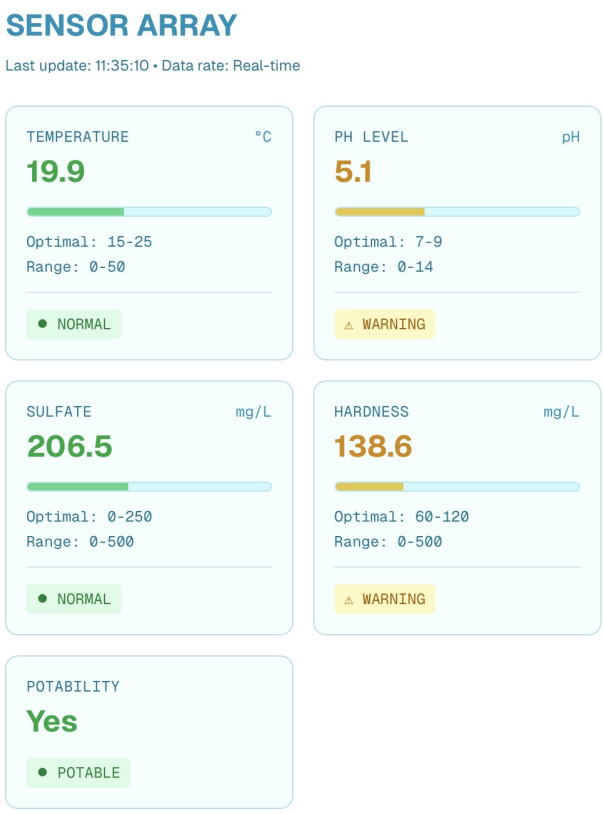

Figure 2. React Web App Dashboard sample.

IV. Experimental Setup

The physical layer of the system comprises two main computational units: an ESP32-based Arduino Nano and a Raspberry Pi 3, as illustrated in Figure 3. The Arduino Nano operates as the sensing node, interfacing directly with the physical environment through analog data acquisition. It is connected to a set of analog sensors, including a pH probe, enabling real-time measurement of key physicochemical parameters from the water source. The Raspberry Pi functions as the edge gateway, aggregating sensor data received via MQTT and executing the Python-based AI inference engine.

The prototype enclosure was implemented using a low-cost, easily reproducible design intended to manage and route water flow based on the system's classification outcome. A cut plastic bottle serves as the primary container for receiving incoming water samples and housing the sensing process.

At the bottle outlet, a flexible hose segment connects to a manual two-way valve that acts as a physical bifurcation point. This valve is linked to two additional hose segments, each leading to a separate container. One container is designated for water classified as potable, while the other collects water identified as non-potable. The routing decision is conceptually driven by the inference result: samples classified as safe are directed to the potable container, whereas unsafe samples are routed to the alternative container.

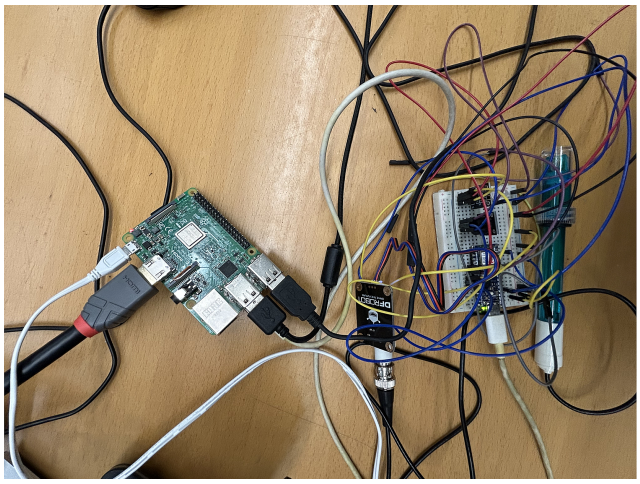

Figure 3. Overall implemented system integrating the Arduino and Raspberry Pi with all sensors.

This design provides a clear physical separation between potable and non-potable water, demonstrating the practical applicability of the proposed system in real-world scenarios. At the same time, it intentionally avoids permanent structural housing or refined mechanical fabrication, reinforcing the prototype's focus on functional validation, affordability, and reproducibility rather than industrial-grade deployment.

A. Dataset and Metrics

The proposed models were trained and evaluated using the Water Quality dataset [28], which comprises 3,276 samples described by the physicochemical indicators listed in Table I. This dataset is well-suited to the study for three main reasons: (i) its multivariate structure reflects the complexity of water potability assessment; (ii) the weak linear correlations among features capture chemical interdependencies, motivating learning-based approaches over fixed threshold laboratory checks; and (iii) its alignment with global regulatory standards supports future scalability and deployment.

The dataset contains nine numerical features and a binary target variable. Missing values were identified in several attributes, most notably pH (14.99%), sulfate (23.84%), and trihalomethanes (4.95%). Based on feature distribution analysis, mean imputation was applied to address these gaps.

A primary limitation of this study is the synthetic origin of the dataset. The high density of unique values across multiple features suggests limited underlying source diversity, which may reduce the granularity of learned decision boundaries and affect model generalization.

As the physical prototype currently supports only pH sensing, the remaining chemical parameters required by the model were synthetically generated to enable end-to-end testing. These values were constrained to World Health Organization (WHO) recommended ranges, ensuring complete feature vectors for inference. Model performance was evaluated using complementary classification metrics [29], including accuracy (ACC), precision ($PREC$), recall ($Recall$), $F1$-score, and the area under the Receiver Operating Characteristic (ROC) curve (AUC), providing a comprehensive assessment under class imbalance and non-linear decision boundaries. The binary potability label follows standard encoding, where 0 denotes unsafe water, and 1 indicates water suitable for human consumption.

TABLE I. CHARACTERISTICS OF WATER QUALITY MEASUREMENTS.

Material	Description	WHO Range
pH	Acidity or alkalinity	6.5 to 8.5
Hardness	Ca/Mg salts	-
Solids (TDS)	Dissolved minerals	500 to 1000 mg/L
Chloramines	Disinfectants	$\leq$ 4 mg/L
Sulfate	Mineral compound	3 to 30 mg/L
Conductivity	Electrical ability	$\leq$ 400 μS/cm
Organic Carbon	Pollutant indicator	$<$ 4 mg/L
Trihalomethanes	Chlorination by-product	$\leq$ 80 ppm
Turbidity	Suspended solids	$\leq$ 5 NTU
Potability	1=Safe; 0=Unsafe	-

B. ML models and code architecture

The proposed system requires an ML model capable of assessing water potability; however, limitations related to data availability, data quality, and the gap between synthetic and real-world data prevent strong guarantees on model performance. To address these constraints and ensure long-term adaptability, a flexible model management architecture was implemented to support the seamless introduction of new datasets and alternative models as they become available.

The architecture is structured around two core components: *Model_Class* and *Controller_Models*. The *Model_Class* encapsulates model-specific functionality, including training, evaluation, persistence, and metadata management. Each model, along with its configuration and performance metrics, can be serialized and restored from a JSON representation. The *Controller_Models* component orchestrates model usage by loading available models from a predefined repository, selecting a designated primary model, and executing inference requests. This separation of concerns enables transparent model replacement and simplifies future system evolution.

To explore water potability from diverse algorithmic perspectives, the model selection strategy spans linear baselines to non-linear ensemble methods. Logistic Regression (LR) and a Stochastic Gradient Descent (SGD) classifier with log loss were implemented as baseline models to evaluate whether a global linear decision boundary can effectively separate the nine-dimensional chemical feature space.

Complementing these global approaches, the k-Nearest Neighbors (kNN) algorithm was employed to capture local data structure, operating under the assumption that chemically similar samples exhibit similar potability outcomes. To further

TABLE II. COMPREHENSIVE PERFORMANCE METRICS AND TRAINING TIME (IN SECONDS) FOR TESTED MODELS.

Model	$F1$	ACC	AUC	$PREC$	$Recall$	$Time$
LR	0.48	0.63	0.52	0.39	0.63	0.424
SGD	0.48	0.63	0.50	0.39	0.63	0.013
kNN	0.49	0.63	0.51	0.50	0.62	0.0117
CNB	0.54	0.53	0.53	0.55	0.53	0.002
RF	**0.65**	**0.68**	**0.69**	**0.67**	**0.68**	**0.884**

address class imbalance—a common characteristic of environmental datasets—Complement Naive Bayes (CNB) was included. Although the current dataset exhibits limited imbalance, CNB is architecturally suited to mitigate the dominance of the majority (non-potable) class and is expected to be particularly relevant in real-world deployments.

Finally, a Random Forest (RF) model was used to capture complex, non-linear relationships among features. By aggregating multiple decision trees through ensemble learning, RF enables the identification of intricate decision boundaries and mitigates variance introduced by data imputation. This multi-model strategy provides a comprehensive assessment of whether water safety can be adequately described by linear thresholds or instead emerges from highly specific, non-linear chemical interactions.

V. RESULTS AND DISCUSSION

This section evaluates the system from two perspectives: the predictive accuracy of the localized AI models and the operational performance of the integrated edge architecture.

A. Model Performance Evaluation

The comparative analysis of the tested algorithms is summarized in Table II.

The empirical results demonstrate that water potability classification is a significantly non-linear task, as evidenced by the failure of linear architectures. Both LR and SGD converged to an AUC of approximately 0.5, indicating that linear hyperplanes possess near-zero discriminatory power for this feature space and effectively default to majority-class guessing. While CNB achieved a marginally higher $F1$ score of 0.54, its poor accuracy of 0.53 reveals a high false-positive rate, failing to reliably distinguish chemical safety margins.

The RF emerged as the best architecture, achieving an AUC of 0.69 and a balanced $F1$ score of 0.65. Its success is tied to its ensemble nature, which mitigates the impact of noise and the variance introduced by missing value imputation, factors that crippled the distance-based kNN. Using recursive partitioning, the RF successfully captured high-order interactions between variables. This performance gap confirms that water potability is not governed by independent feature thresholds but by a multifaceted chemical synergy that only the ensemble-based decision trees could effectively resolve.

B. System Integration and Latency Analysis

Beyond algorithmic accuracy, the system's ability to function as an integrated control unit was validated through real-time testing. The Edge AI model demonstrated the capability to process incoming MQTT packets and return the classification within milliseconds. By hosting the inference engine locally via FastAPI, the system eliminated the "Cloud Round-Trip Time", which in industrial IoT environments can fluctuate between 100ms and several seconds depending on network congestion [4].

The React-based dashboard successfully maintained a persistent WebSocket connection, reflecting potability changes instantly. This responsiveness is critical for the "Control Rule" aspect of the project, as any delay in detecting non-potable water could lead to the contamination of downstream reservoirs before a manual or cloud-based intervention could occur.

VI. CONCLUSION AND FUTURE WORK

This work presents a proof of concept for a wastewater detection system combining local IoT sensing with edge-level AI inference. The results demonstrate that relocating intelligence from the cloud to the edge improves responsiveness and operational reliability, providing deterministic latency suitable for control logic. The study further validates that low-cost platforms, such as the Raspberry Pi 3, can execute complex ensemble models, confirming that non-linear water quality interactions can be effectively resolved at the edge in a scalable and accessible manner.

However, several limitations define the current boundary conditions. The system relies on a partially synthetic dataset, as several chemical parameters were generated to compensate for incomplete hardware integration. In addition, missing values in the source data required mean imputation, potentially simplifying learned decision boundaries. While sufficient for validating the edge architecture and middleware logic, the resulting model may not fully reflect the variability, noise, and stochastic behavior of real wastewater streams.

Future work will prioritize full sensor integration to eliminate reliance on simulated inputs and improve real-world robustness. The architecture is also designed to evolve from a monitoring solution into a closed-loop control system through the replacement of manual routing with electronically actuated solenoid valves. Validating automated actuation within the existing framework will enable real-time, autonomous water diversion, enhancing system resilience and applicability in decentralized environments.

From a broader socio-economic perspective, such edge-enabled systems may support municipal authorities in reducing operational costs and preventing contamination through rapid, localized decision-making. The modular design and use of standardized protocols, including MQTT, facilitate integration with existing industrial infrastructures without requiring large-scale system overhauls.

ACKNOWLEDGMENTS

This work is financially supported by national funds through the FCT/MCTES (PIDDAC), under the Associate Laboratory Advanced Production and Intelligent Systems – ARISE LA/P/0112/2020 (DOI 10.54499/LA/P/0112/2020) and the Base Funding (UIDB/00147/2020) and Programmatic Funding

(UIDP/00147/2020) of the R&D Unit Center for Systems and Technologies – SYSTEC.

REFERENCES

[1] A. Oros, "Bioaccumulation and trophic transfer of heavy metals in marine fish: Ecological and ecosystem-level impacts," *Journal of Xenobiotics*, vol. 15, no. 2, pp. 59–72, 2025, [retrieved: February, 2026], ISSN: 2039-4713. DOI: 10.3390/jox15020059.

[2] Y. Gelaye, "Public health and economic burden of heavy metals in ethiopia: Review," *Heliyon*, vol. 10, no. 19, Oct. 2024, ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2024.e39022.

[3] R. Martínez et al., "On the use of an iot integrated system for water quality monitoring and management in wastewater treatment plants," *Water*, vol. 12, no. 4, 2020, ISSN: 2073-4441. DOI: 10.3390/w12041096. [Online]. Available: https://www.mdpi.com/2073-4441/12/4/1096.

[4] P. Hu, C. He, Y. Zhu, and T. Li, "The product quality inspection scheme based on software-defined edge intelligent controller in industrial internet of things," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 113–128, 2023, [retrieved: February, 2026], ISSN: 2192-113X. DOI: 10.1186/s13677-023-00487-7.

[5] R. Wiryasaputra, C.-Y. Huang, Y.-J. Lin, and C.-T. Yang, "An iot real-time potable water quality monitoring and prediction model based on cloud computing architecture," *Sensors*, vol. 24, no. 4, 2024, ISSN: 1424-8220. DOI: 10.3390/s24041180.

[6] F. Tosi et al., "Enabling image-based streamflow monitoring at the edge," *Remote Sensing*, vol. 12, no. 12, 2020, ISSN: 2072-4292. DOI: 10.3390/rs12122047.

[7] U. Zhalmagambetova, D. Assanov, A. Neftissov, A. Biloshchytskyi, and I. Radelyuk, "Implications of water quality index and multivariate statistics for improved environmental regulation in the irtysh river basin (kazakhstan)," *Water*, vol. 16, no. 15, pp. 2203–2220, 2024, [retrieved: February, 2026], ISSN: 2073-4441. DOI: 10.3390/w16152203.

[8] H. H. Lou et al., "A new area of utilizing industrial internet of things in environmental monitoring," *Frontiers in Chemical Engineering*, vol. 4, p. 842514, 2022.

[9] H. Cao et al., "Advancing clinical biochemistry: Addressing gaps and driving future innovations," *Front Med (Lausanne)*, vol. 12, p. 1521126, Apr. 2025.

[10] A. T. Chafa, G. Chirinda, and S. Matope, "Design of a real–time water quality monitoring and control system using internet of things (iot)," *Cogent Engineering*, vol. 9, 2022.

[11] A. Benis, O. Tamburis, C. Chronaki, and A. Moen, "One digital health: A unified framework for future health ecosystems," *J Med Internet Res*, vol. 23, no. 2, e22189, Feb. 2021, ISSN: 1438-8871. DOI: 10.2196/22189. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/33492240.

[12] Y. Dai, Z. Huang, N. Khan, and M. S. Labbo, "Smart water management: Governance innovation, technological integration, and policy pathways toward economic and ecological sustainability," *Water*, vol. 17, no. 13, 2025, ISSN: 2073-4441. DOI: 10.3390/w17131932.

[13] J. Salgado, C. Pizarro, L. Wong, and J. Castillo, "Iot watercare: Water quality control system in unofficial settlements of peru based in an iot architecture," in *2022 31st Conference of Open Innovations Association (FRUCT)*, 2022, pp. 277–288. DOI: 10.23919/FRUCT54823.2022.9770881.

[14] J. Li, X. Yang, and R. Sitzenfrei, "Rethinking the framework of smart water system: A review," *Water*, 2020.

[15] R. M. M. Salem, M. S. Saraya, and A. M. T. Ali-Eldin, "An industrial cloud-based iot system for real-time monitoring and controlling of wastewater," *IEEE Access*, vol. 10, pp. 7096–7121, 2022.

[16] W. Zhang, F. Ma, M. Ren, and F. Yang, "Application with internet of things technology in the municipal industrial wastewater treatment based on membrane bioreactor process," *Applied Water Science*, vol. 11, no. 52, p. 52, 2021.

[17] A. G. Orozco-Lugo et al., "Monitoring of water quality in a shrimp farm using a fanet," *Internet of Things*, vol. 18, p. 100170, 2022, ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2020.100170.

[18] A. Mamani-Saico and P. R. Yanyachi, "Implementation and performance study of the micro-ros/ros2 framework to algorithm design for attitude determination and control system," *IEEE Access*, vol. 11, pp. 128451–128460, 2023. DOI: 10.1109/ACCESS.2023.3330441.

[19] S. Yadav and A. Kumar, "Integrating edge computing and IoT for real-time air and water quality monitoring systems," *International Journal of Engineering Science*, vol. 11, pp. 1507–1511, 2025, [retrieved: February, 2026]. [Online]. Available: https://theaspd.com/index.php/ijes/article/view/4588.

[20] J. Ren, Q. Zhu, and C. Wang, "Edge computing for water quality monitoring systems," *Mobile Information Systems*, vol. 2022, no. 1, p. 5056606, 2022, [retrieved: February, 2026]. DOI: https://doi.org/10.1155/2022/5056606. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/5056606. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/5056606.

[21] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 3rd. Indianapolis, IN, USA: Wiley, 2014, ISBN: 978-1118921661.

[22] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd. Sebastopol, CA, USA: Maker Media, Inc., 2014, ISBN: 978-1449363338.

[23] R. A. Light, "Mosquitto: Server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[24] M. Lathkar, "Getting started with fastapi," in *High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python*. Berkeley, CA: Apress, 2023, pp. 29–64, ISBN: 978-1-4842-9178-8. DOI: 10.1007/978-1-4842-9178-8_2.

[25] InfluxData, *InfluxDB Documentation*, retrieved: February, 2026, 2025. [Online]. Available: https://docs.influxdata.com/influxdb/.

[26] S. Kirešová et al., "Grafana as a visualization tool for measurements," in *2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES)*, IEEE, 2023, pp. 1–5.

[27] Meta Platforms, Inc., *React: A JavaScript Library for Building User Interfaces*, retrieved: February, 2026, 2025. [Online]. Available: https://reactjs.org/.

[28] Aditya Kadiwal and Kaggle Contributors, *Water Potability Dataset*, retrieved: February, 2026, 2020. [Online]. Available: https://www.kaggle.com/datasets/adityakadiwal/water-potability.

[29] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Artificial Intelligence Application in Networks and Systems*, R. Silhavy and P. Silhavy, Eds., Cham: Springer International Publishing, 2023, pp. 15–25, ISBN: 978-3-031-35314-7.