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

José Isidro, Rafael Teixeira, João Costa, Carolina Gonçalves, Diogo Ferreira, Pedro Azevedo, Pedro Simões, Rui Pinto, Gil Gonçalves

Presenter: José Miguel Isidro, Universidade do Porto
up202006485@up.pt

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

IARIA

Presenter Profile

Hello! My name is José Miguel Isidro, I am a Software Engineer pursuing a Master's in Informatics and Computing Engineering at **FEUP**. Software Developer at **JuniFEUP**, AI Researcher at **INESC TEC**.

JMI
JOSÉ MIGUEL ISIDRO
Website

in
LinkedIn

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
up202006485@up.pt

Context & Motivation

Water contamination poses severe risks to economic stability, environmental safety, and public health. The ability to rapidly differentiate between potable and non-potable water is an essential operational goal for immediate intervention.

Economic Risk

Contamination events disrupt supply chains and impose significant remediation costs.

Public Health

Delayed detection translates directly into preventable illness and loss of life.

Decentralised Need

Affordable, autonomous systems must operate independently of centralised infrastructure.

The Problem with Current Solutions

Existing approaches leave critical gaps between detection and intervention. Traditional and cloud-centric systems both introduce unacceptable delays.

Manual Monitoring Latency

Laboratory analyses of field samples can take **days**, making real-time response impossible.

Cloud-Centric Architecture

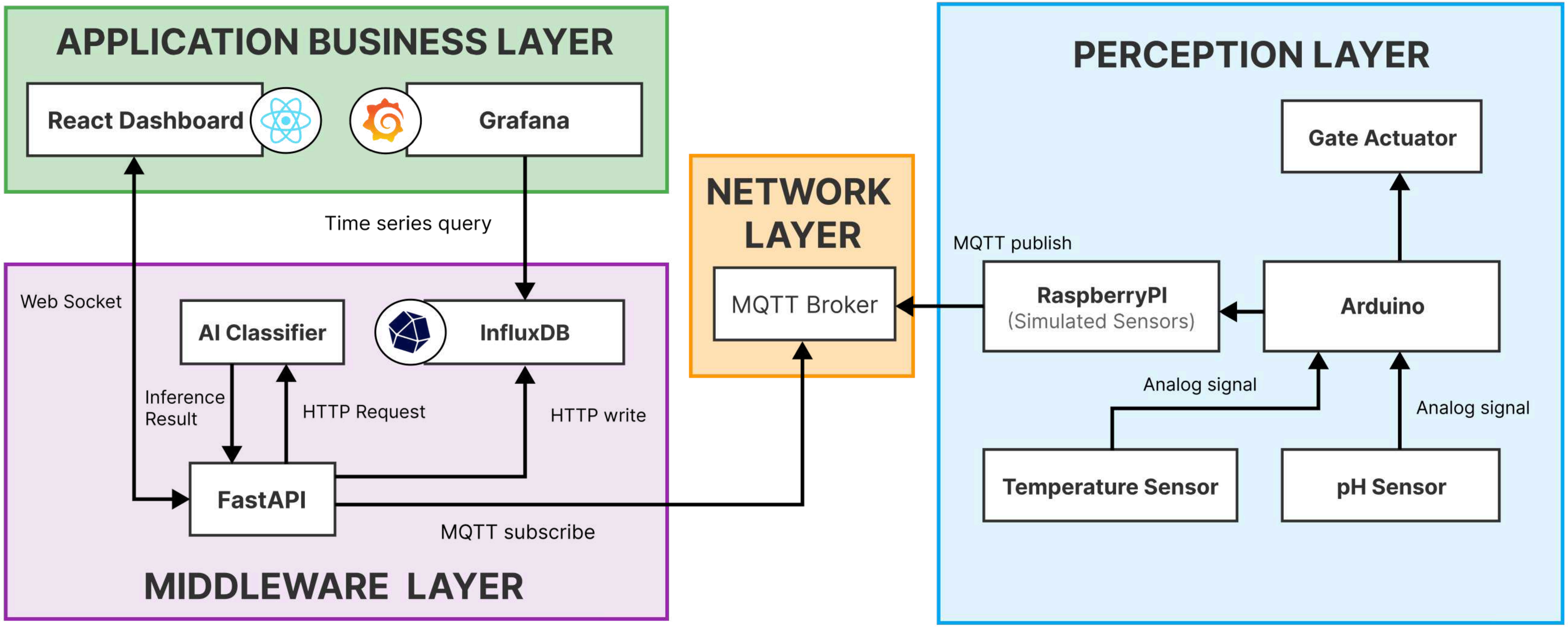
Most modern Smart Water Quality Monitoring Systems (SWQMS) route all intelligence through remote data centres, creating structural fragility.

Single Point of Failure

Loss of internet connectivity stalls cloud-reliant systems entirely — a critical vulnerability in remote or disaster-affected deployments.

Proposed Edge-Based Solution

Our architecture relocates the intelligence layer from remote data centres to the network Edge, enabling autonomous AI classification without requiring continuous internet connectivity.



By executing inference locally, the system maintains operational resilience under network disruption while delivering continuous, low-latency water quality assessment.

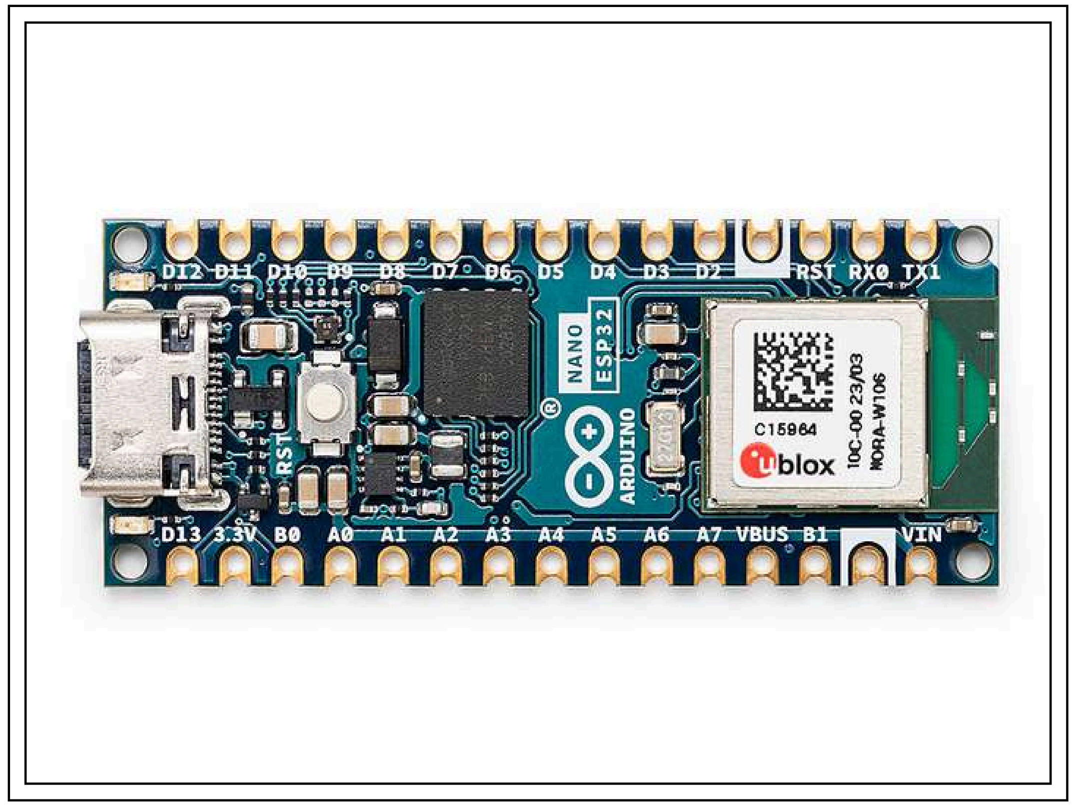
Hardware Architecture

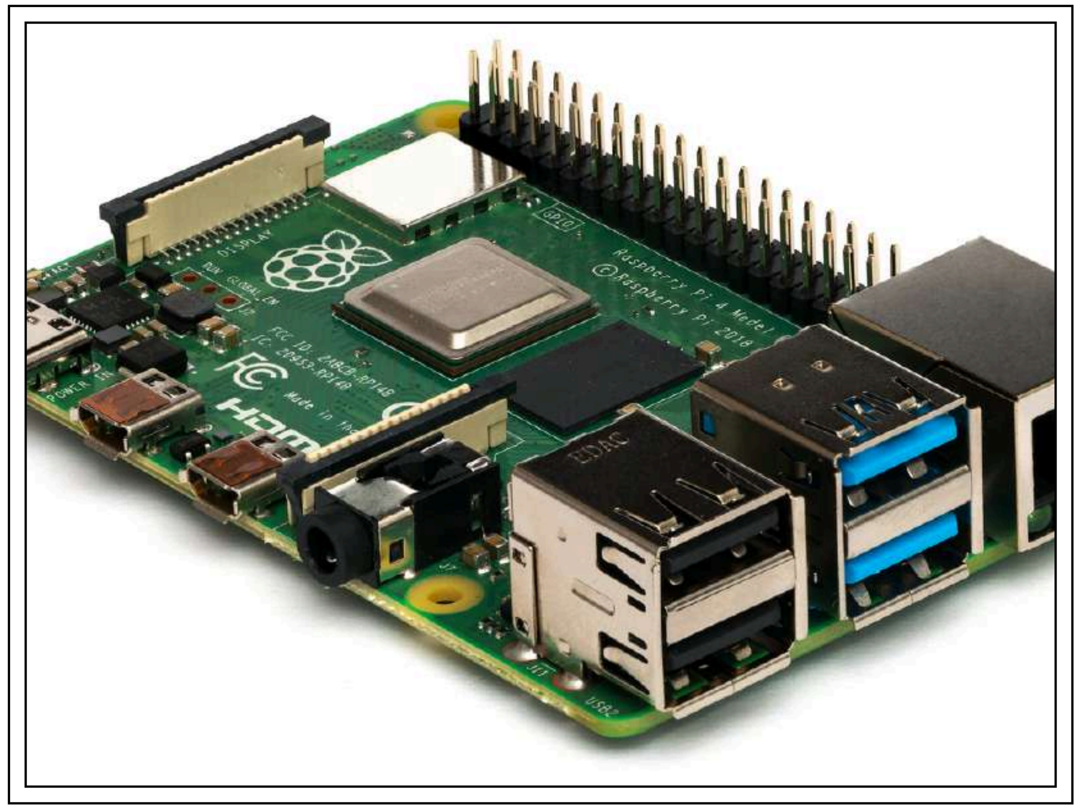
Arduino Nano ESP32

- **Role**: Physical Interface & Data Acquisition.
- **Function**: Handles real-time, low-level hardware control (Sensors/Actuators).
- **Why**: Chosen for robustness, low power consumption, and reliability in continuous monitoring.

Raspberry Pi 3

- **Role**: Central Processing Hub.
- **Function**: Manages data aggregation, executes high-level processing (AI/ML) and handles network connectivity.
- **Capabilities**: Enables remote monitoring and data logging.





Perception Layer

The foundation of our system lies in its ability to accurately perceive water quality parameters and physically react to them. This layer integrates both real and simulated inputs, alongside actuated outputs.

Physical pH Sensor

Real-time analogue-to-digital conversion directly from the pH probe, providing immediate chemical insights into water potability.

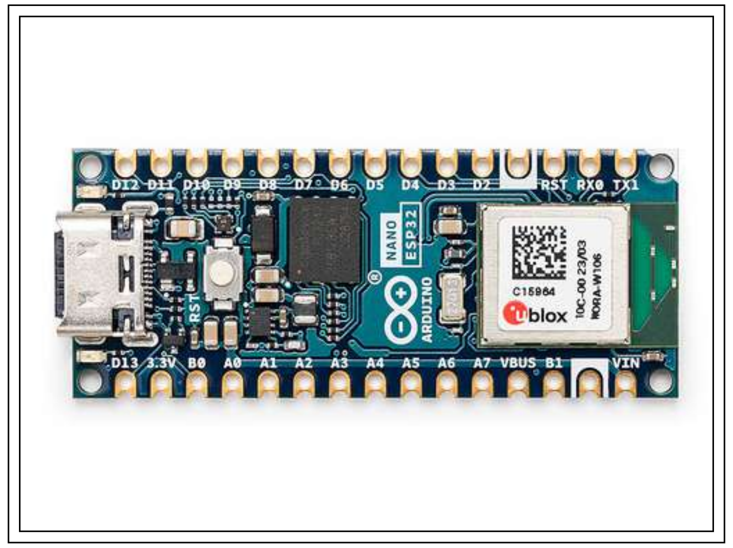
Simulated Sensor Data

A dedicated software layer on the Raspberry Pi generates bounded synthetic values, allowing for comprehensive pipeline testing without additional physical sensors.

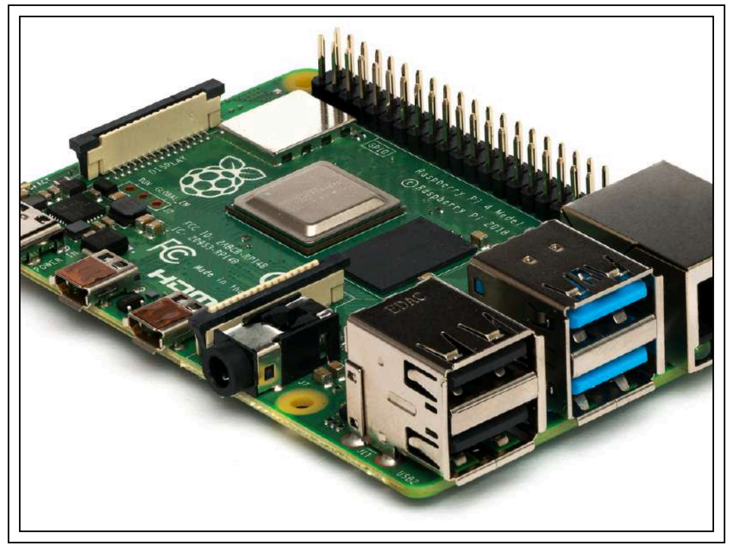
Electric Valves

Electronically actuated valves manage water routing based on the AI model's classification, directing potable and non-potable water streams.

Network Layer



Arduino Nano ESP32



Raspberry Pi

MQTT via Mosquitto

The lightweight **MQTT protocol** reduces network overhead and preserves data integrity under unstable connections through a publish/subscribe pattern.

Synthetic Data Engine

A simulation layer on the Pi generates **bounded synthetic values** for TDS and turbidity, enabling full end-to-end pipeline testing without additional physical sensors.

Middleware Layer

Two specialised components form the data backbone of the edge node, prioritising throughput and write speed for high-frequency sensor streams.

FastAPI

Asynchronous Python backend handling **high-throughput data ingestion** and low-latency processing of incoming sensor payloads.

InfluxDB

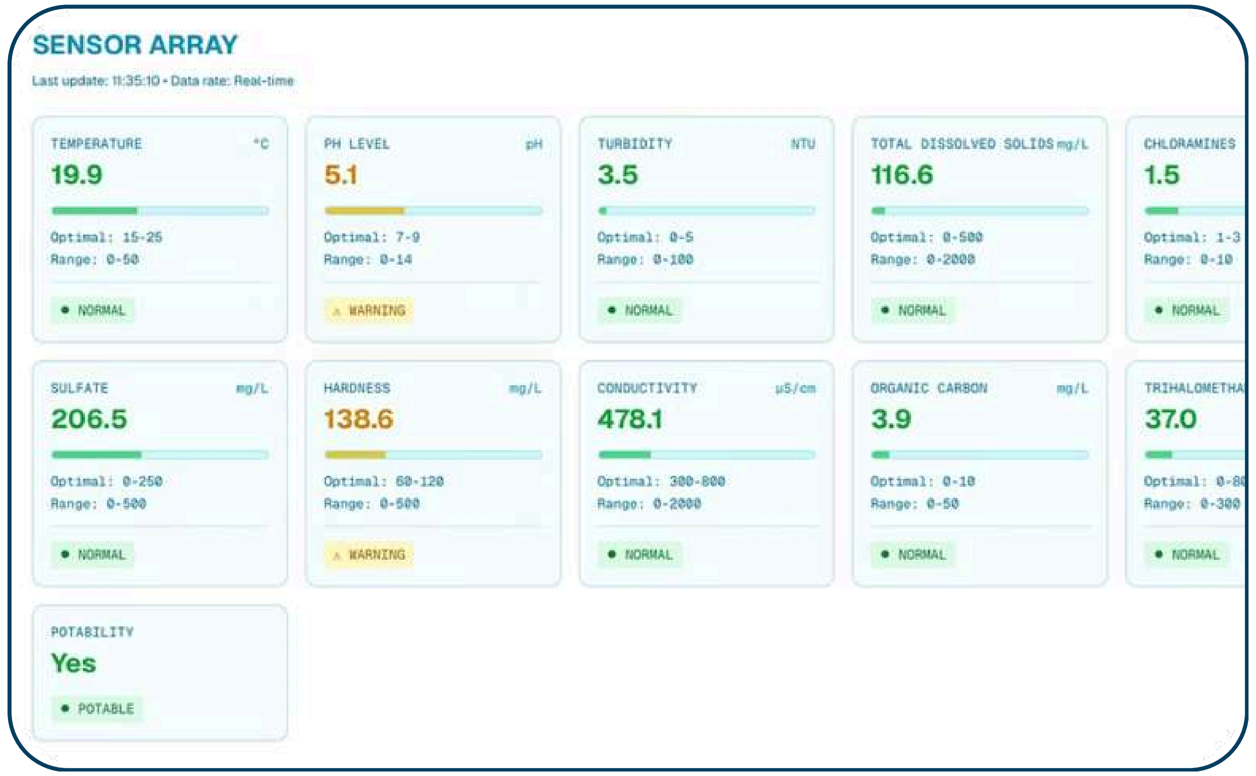
Local **time-series database** optimised for high-frequency writes at the edge, enabling efficient querying of historical sensor records without egress costs.

Application & Visualisation

A dual-view strategy serves two distinct audiences: engineers requiring deep analytical insight, and end-users needing immediate status awareness.

Grafana — Engineer View

Detailed historical visualisations and sensor health metrics for operational diagnostics and long-term trend analysis.

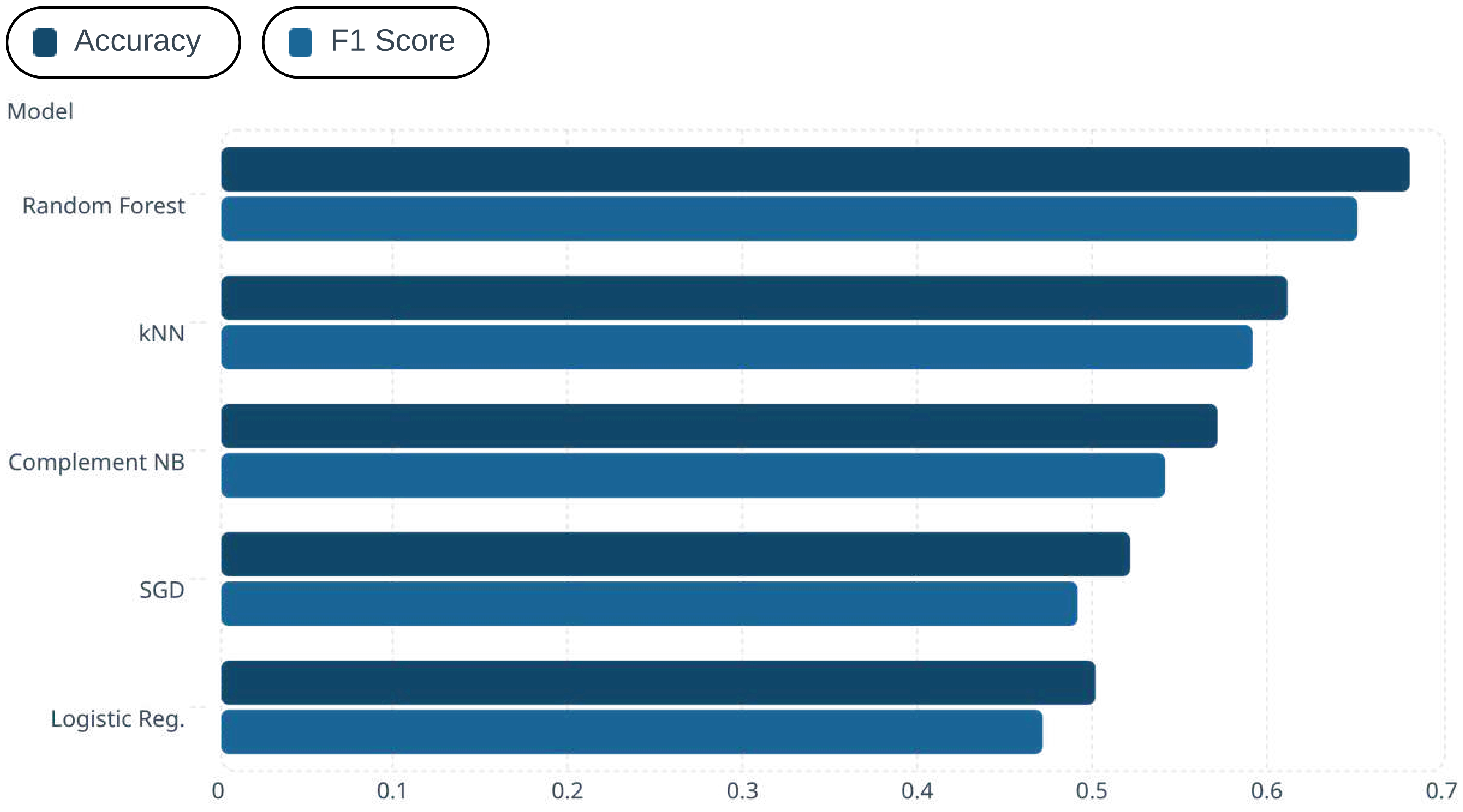


React Web App — User View

Intuitive dashboard leveraging **WebSockets** for instant, zero-latency potability status updates.

Machine Learning Models & Results

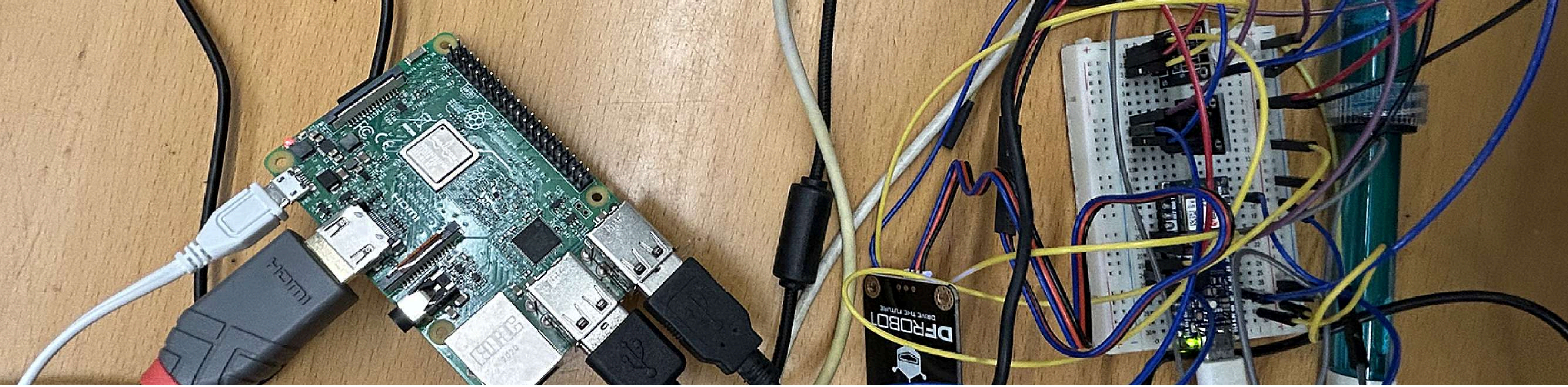
We evaluated multiple machine learning algorithms to identify the best model for water potability classification. Random Forest emerged as the clear winner due to its ability to capture non-linear relationships between chemical parameters.



Why Random Forest Won

Linear architectures (Logistic Regression, SGD) failed to capture the **non-linear chemical synergies** governing potability. **Random Forest**'s ensemble of decision trees successfully models high-order parameter interactions.

- Best model: **Random Forest**
- **Accuracy 0.68**
- **F1 Score 0.65**



Proof of Concept Prototype

The physical prototype demonstrates the full classification pipeline, routing water based on the AI's potability output. A pivot from electronic to **manual valves** was made under time constraints to guarantee a reliable, reproducible demonstration of the core classification logic.

Reproducible Design

Modular layout allows components to be swapped or recalibrated independently.

AI-Driven Routing

Classification output directly governs the physical water path: potable or non-potable.

Pragmatic Pivot

Manual valves ensured demonstration reliability without compromising system logic integrity.

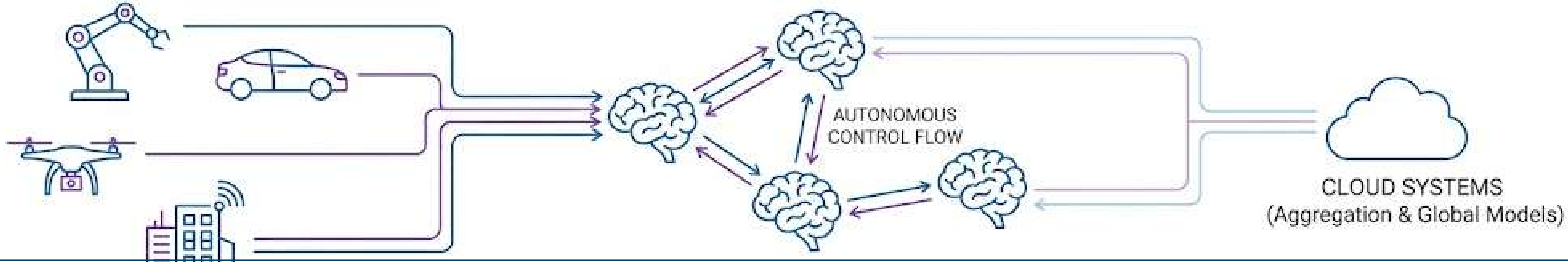
Challenges & Lessons Learned

Challenges Encountered

- Restricted availability of specialised IoT components delayed hardware assembly
- Sensor reliability and calibration drift required iterative troubleshooting
- Steep learning curve integrating MQTT, FastAPI, InfluxDB, and ML inference on constrained hardware

Key Lessons

- **Component sourcing:** Plan procurement well ahead of build phases
- **Sensor calibration:** Design explicit calibration routines from the outset
- **Integration buffer:** Allocate dedicated time for cross-layer system integration and regression testing



Conclusion

Our project successfully demonstrated that relocating intelligence to the edge significantly improves responsiveness and proves that low-cost hardware can execute complex ensemble models. This represents a robust foundation for decentralised data processing.

Future Work

Physical Sensor Integration

Future iterations will prioritise full physical sensor integration to eliminate the need for simulated inputs, ensuring real-world data accuracy.

Autonomous Closed-Loop Control

The system will transition from a monitoring tool to a fully autonomous, closed-loop control system, utilising electronically actuated valves for active management.