International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

AquaVision BI: An Intelligent AI-Based Irrigation Monitoring System Using IoT Sensors

Vrushabh Jitendra Patil, Pratik Prakash Patil, Ganesh Rajendra Mote
D. Y. Patil Technical Campus, Talsande
Dept. of Computer Science & Engineering

Abstract- — Traditional irrigation systems depend heavily on manual inspection to detect leaks, pipe blockages, and abnormal water flow, resulting in significant water wastage, reduced irrigation efficiency, and increased maintenance expenditure. This paper presents AquaVision BI, an intelligent IoT-enabled irrigation monitoring system that integrates three Hall-effect flow-rate sensors, an ESP32 Wi-Fi microcontroller, and an AI-driven differential-threshold anomaly-detection algorithm to achieve real-time surveillance of irrigation pipelines. The system continuously samples sensor pulse counts at one-second intervals, computes volumetric flow rates, and applies pairwise differential analysis to localise leakage to specific pipeline segments (upstream, mid-stream, or downstream). Upon anomaly detection, automated alerts are dispatched via the Blynk IoT cloud dashboard and a local buzzer actuator. Experimental evaluation on a controlled testbed confirms accurate leak localisation across all three sensor nodes, with end-to-end alert latency consistently below two seconds. The proposed system significantly reduces water wastage, lowers operational costs, and promotes sustainable agricultural water management practices.

Key Words: Irrigation monitoring; anomaly detection; Internet of Things (IoT); ESP32; AI-based threshold detection; water leakage localisation; smart agriculture; MicroPython; Blynk IoT platform.

I. INTRODUCTION

A. Background and Motivation

Agriculture accounts for approximately 70% of global freshwater withdrawals, yet significant volumes are lost through leaking or poorly managed irrigation infrastructure [1]. In many developing agricultural regions, irrigation networks are still monitored through manual inspection, an approach that is labour-intensive, inconsistent, and incapable of providing early warning of incipient faults. The consequences include water wastage, crop loss from under-irrigation in fault-affected zones, and elevated maintenance expenditure.

The convergence of low-cost IoT microcontrollers, Hall-effect flow sensors, and cloud-based monitoring platforms has created a compelling opportunity to automate irrigation fault detection. By deploying multiple flow sensors at strategic points along a pipeline and applying differential-analysis algorithms, it becomes possible to detect and spatially localise leaks in real time without manual intervention.

B. Problem Statement

Traditional irrigation systems rely on periodic manual inspection to detect leaks, blockages, and irregular water distribution. These methods fail to provide timely detection, especially in extensive field networks, resulting in sustained water wastage and degraded crop yield. There is therefore a clear and urgent need for an automated, intelligent monitoring solution capable of continuous real-time analysis and immediate fault notification.

C. Contributions

The principal contributions of this work are:

- A three-node differential flow-rate algorithm that localises pipeline leakage to upstream, mid-stream, or downstream segments.
- An end-to-end embedded implementation on the ESP32 platform using MicroPython and interrupt-driven pulse counting for precise volumetric flow measurement.
- Cloud-dashboard integration via Blynk IoT providing sub-2-second alert delivery to remote users.
- An empirical evaluation on a physical testbed demonstrating 100% leakage localisation accuracy across all test conditions.

D. Paper Organisation

The remainder of this paper is organised as follows. Section II reviews related work. Section III presents the system analysis. Section IV describes the system architecture and design. Section V details the implementation. Section VI reports testing and evaluation results. Section VII discusses findings. Section VIII concludes and outlines future work.

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

II. RELATED WORK

Research into automated irrigation monitoring has progressed along two broad tracks: machine-learning-based anomaly detection and IoT-enabled sensor integration.

A. Machine Learning Approaches

Ahmed et al. [1] applied Decision Trees and Support Vector Machines (SVM) to sensor data from smart irrigation systems, demonstrating substantial accuracy gains over rule-based thresholds. Liu et al. [2] extended this work using Convolutional Neural Networks (CNNs) to analyse time-series flow and soil-moisture data, exploiting CNN spatial-feature extraction to identify complex temporal anomaly patterns that simpler classifiers miss.

Zhang et al. [3] proposed Long Short-Term Memory (LSTM) networks for predictive maintenance, training on historical sensor logs to forecast component failures before they occur. Wang et al. [4] compared Random Forests and Gradient Boosting Machines (GBMs), finding both effective for maintenance prediction, with Random Forests yielding slightly higher accuracy on field-collected data.

B. Sensor Integration and IoT Platforms

Raza et al. [5] conducted a comprehensive review of sensor types for precision irrigation, analysing trade-offs between accuracy, reliability, and cost-effectiveness. Their work underlines the importance of sensor placement strategy — a principle central to the AquaVision BI multi-node design. Recent work has also explored LoRaWAN and MQTT protocols for low-power wide-area irrigation telemetry; however, Wi-Fi-based ESP32 deployments remain dominant for field-scale prototypes owing to their low cost and ease of integration with cloud platforms such as Blynk [6][7].

C. Research Gap

Despite advances in ML-based detection and IoT sensing, most existing prototypes address only one of: (i) intelligent anomaly detection, (ii) real-time multi-node sensing, or (iii) user-accessible cloud dashboards. Systems that integrate all three in a single low-cost embedded platform — while also providing spatial localisation of faults — remain scarce. AquaVision BI addresses this gap.

III. SYSTEM ANALYSIS

A. Functional Requirements

The following functional requirements were elicited from stakeholder interviews and analysis of existing system shortcomings:

- FR1) The system shall continuously monitor water flow in irrigation pipelines using calibrated flow-rate sensors.
- FR2) The system shall detect leaks, blockages, and abnormal flow conditions in real time using threshold-based differential analysis.
- FR3) The system shall transmit sensor data and alert status to a cloud dashboard within two seconds of anomaly onset.
- FR4) The system shall provide a visual and audio alert mechanism upon fault detection.
- FR5) The system shall log all sensor readings and alert events for retrospective analysis.

B. Non-Functional Requirements

- NFR1) Performance: Alert latency □ 2 s; sensor sampling interval = 1 s.
- NFR2) Reliability: Continuous operation □ 72 h without manual reset.
- NFR3) Scalability: Architecture extensible to N sensor nodes without firmware redesign.
- NFR4) Usability: Dashboard accessible to non-technical agricultural workers via mobile application.
- NFR5) Cost: Total BOM cost □ USD 25 per monitoring node.

C. Feasibility Assessment

Dimension	Assessment
Technical	All components commercially available; ESP32 + Blynk well-documented.
Operational	Blynk dashboard requires no specialised training for field operators.
Economic	Estimated BOM □ USD 18; well within smallholder budget constraints.

Table I. Feasibility Assessment Summary

IV. SYSTEM DESIGN

A. Architecture Overview

AquaVision BI adopts a three-tier IoT architecture: (i) a Perception Layer comprising three YF-S201 Hall-effect flow sensors; (ii) a Processing Layer implemented on the ESP32 microcontroller running MicroPython; and (iii) a Presentation Layer hosted on the Blynk IoT cloud with a mobile-accessible

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

dashboard. A solenoid valve and buzzer actuator constitute the response subsystem.

B. Sensor Placement Strategy

Three sensors are installed at the pipeline inlet (S1), mid-point (S2), and outlet (S3). Under leak-free conditions, flow rates at all three nodes are approximately equal. A leak between S1 and S2 manifests as a significant positive deviation of S1 from both S2 and S3; a leak between S2 and S3 produces a positive deviation of S2 from S3 with S1 remaining elevated. This pairwise differential logic enables single-segment leak localisation.

C. Detection Algorithm

Let fi denote the flow rate at sensor node i (i □ {1,2,3}), and let □ = 0.5 L/min be the configurable leakage threshold. A leak at node k is declared when:

$$|f_k - f_j| > \delta \quad \forall j \neq k$$

Flow rate is derived from pulse count C accumulated

over a one-second window:

$$f_i = C_i / 7.5 \quad [L/min]$$

The divisor 7.5 is the sensor-specific calibration constant (pulses per litre per minute) for the YF-S201 at nominal operating pressure.

D. Database Schema

Five entities are maintained in cloud and local storage:

- User: user_id (PK), username, password_hash, contact.
- Sensor: sensor_id (PK), type, pipeline_position, calibration_k.
- IrrigationReading: reading_id (PK), sensor_id (FK), flow_rate, timestamp.
- Alert: alert_id (PK), reading_id (FK), type, severity, timestamp.
- SystemStatus: status_id (PK), pump_state, valve_state, health_flag, timestamp.

E. System Workflow

1	**Data Collection**: Interrupt-driven pulse counting on GPIO 13/12/14; counts accumulated per 1 s window.
2	**Flow Computation**: pulse_count / 7.5 → L/min for each node.
3	**Anomaly Detection**: Pairwise differential test against threshold δ.
4	**Alert Dispatch**: Blynk virtual pin V4 updated; buzzer activated.
5	**Data Logging**: Readings and alert status written to Blynk cloud datastream.
6	**Dashboard Update**: Gauge widgets on V1–V3 reflect live flow; V4 shows status.

Table II. System Workflow Stages

V. IMPLEMENTATION

A. Hardware Configuration

Component	Specification	Role
ESP32-WROOM-32	Xtensa LX6, 240 MHz, 520 KB SRAM	Central controller
YF-S201 (×3)	Hall-effect, 1–30 L/min, GPIO 13/12/14	Flow sensin g
Solenoid Valve	12 V DC, normally open	Automated shutoff
Water Pump	5 V mini submersible	Flow generation
Buzzer	5 V active, 85 dB	Local alert
Power Supply	5 V / 2 A USB adapter	Syste m power

Table III. Hardware Bill Of Materials

B. Software Stack

Layer	Technology
Firmware	MicroPython 1.20 on ESP32
IoT Cloud	Blynk IoT (MQTT over TLS)
Dashboard	Blynk Web + Mobile App
Dev Tools	Thonny IDE, Arduino IDE, VS Code
VCS	GitHub

Table IV. Software Stack

C. MicroPython Implementation

The firmware is structured into five modules: (i) wifi_connect.py — Wi-Fi initialisation; (ii) sensors.py — interrupt handler registration and pulse counting; (iii) flow_calc.py — flow-rate computation and differential anomaly logic; (iv) blynk_client.py — Blynk virtual-pin write calls; and (v) main.py — orchestration loop. The core detection logic is shown in Listing 1.

```
# ■■ Listing 1: Core leakage-detection loop
(main.py) ■■■■■■■■■■■■
THR = 0.5 # leakage threshold [L/min]
INT = 1000 # sampling interval [ms]

while True:
blynk.run()
if ticks_diff(ticks_ms(), t0) >= INT:
t0 = ticks_ms()
f1 = cnt[0] / 7.5; cnt[0] = 0
f2 = cnt[1] / 7.5; cnt[1] = 0
f3 = cnt[2] / 7.5; cnt[2] = 0
blynk.virtual_write(1, f1) # Gauge S1
blynk.virtual_write(2, f2) # Gauge S2
blynk.virtual_write(3, f3) # Gauge S3

leak = False
if abs(f1-f2)>THR and abs(f1-f3)>THR:
msg='■ Upstream Leak (S1)'; leak=True
elif abs(f2-f1)>THR and abs(f2-f3)>THR:
msg='■ Mid-Stream Leak (S2)'; leak=True
elif abs(f3-f1)>THR and abs(f3-f2)>THR:
msg='■ Downstream Leak (S3)'; leak=True
else:
msg='✓ System Normal'

blynk.virtual_write(4, msg)
buzzer.value(1 if leak else 0)
```

Listing 1. Core leakage-detection loop (condensed).

VI. EXPERIMENTAL EVALUATION

A. Test Environment

Experiments were conducted on a 3-metre PVC irrigation pipeline testbed instrumented with YF-S201 sensors at inlet (S1), mid-point (S2), and outlet (S3). A 5 V submersible pump supplied water at approximately 2.5 L/min nominal flow. Simulated leaks were introduced by opening calibrated apertures at three pre-defined locations corresponding to each pipeline segment. All tests were repeated five times; results are reported as mean ± standard deviation.

B. Test Types and Results

Test Type	Description	Outcome
Unit	Individual module verification — sensor interrupt, flow calc, alert logic	All Pass
Integration	ESP32 □ Blynk data-path; end-to-end virtual-pin update	Pass
System	Full pipeline under normal, low-pressure, and leak conditions	Pass
UAT	Dashboard usability with five non-technical users	Pass
Leak Localisation	Upstream / mid / downstream leak; 5 trials each segment	100% accuracy
Latency	Time from anomaly onset to Blynk notification	1.4 ± 0.3 s
Stability	Continuous 72-hour operation without reboot	No failure

Table V. Evaluation Results Summary

VII. RESULTS AND DISCUSSION

A. Real-Time Flow Monitoring

Under normal operating conditions, all three sensor nodes reported flow rates in the range 2.3–2.7 L/min. The Blynk dashboard gauge widgets refreshed consistently within one second, confirming that the 1 s sampling loop imposed negligible overhead on the ESP32. No spurious alerts were generated during the 72-hour stability test, demonstrating a zero false-positive rate at □ = 0.5 L/min.

B. Leak Localisation Accuracy

Across all 15 controlled leak trials (5 per segment), the differential algorithm correctly identified the leakage segment in every instance, yielding a localisation accuracy of 100%. Mean alert latency was 1.4 s (□ = 0.3 s), satisfying the □ 2 s non-functional requirement. The buzzer and Blynk V4 label provided simultaneous local and remote notification.

C. Dashboard Usability

Five non-technical users were asked to interpret the dashboard without prior training. All five correctly identified the system status (normal vs. leak) and the affected segment within 30 seconds, validating the design's usability for agricultural field operators.

D. Comparative Analysis

Compared to manual inspection (detection delay typically measured in hours) and basic single-sensor threshold systems (no spatial localisation), AquaVision BI reduces detection delay by multiple orders of magnitude and adds segment-level fault localisation at minimal additional hardware cost. While ML-based approaches such as LSTM

[3] and CNN [2] offer adaptive threshold learning, they require substantial labelled training data and significant computational resources — barriers that the differential-threshold approach avoids for real-time embedded deployment.

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented AquaVision BI, a low-cost, real-time irrigation monitoring system that achieves accurate leak localisation using a three-node differential flow-rate algorithm executed on the ESP32 microcontroller. The system delivered 100% leakage localisation accuracy in controlled experiments, sub-2-second alert latency, and 72-hour continuous operation without failure. The Blynk IoT dashboard proved immediately usable by non-technical agricultural workers. AquaVision BI demonstrates that robust, intelligent irrigation monitoring is achievable at a hardware cost of approximately USD 18, making it accessible to smallholder farmers and large-scale irrigation network operators alike.

B. Limitations

- Accuracy depends on sensor calibration; drift over time may require periodic recalibration.
- Continuous Wi-Fi connectivity is required; intermittent connectivity delays cloud alerts.
- Current sensing scope is limited to flow rate; pressure and soil-moisture are not yet incorporated.

C. Future Work

Planned extensions include:

1. Integration of soil-moisture, temperature, and pressure sensors for multi-parameter irrigation analysis.
2. Implementation of LSTM-based predictive maintenance to forecast sensor or pipeline failures before onset.
3. Development of a dedicated Android/iOS application with push-notification support and valve-control capability.
4. Cloud-based big-data analytics pipeline for large-scale multi-field irrigation network management.
5. Solar energy-harvesting module to enable fully off-grid deployment in remote agricultural settings.
6. Field trials across diverse soil types, pipe diameters, and flow regimes to validate threshold generalisation.

IX. ACKNOWLEDGEMENT

The authors gratefully acknowledge the guidance and support of the faculty of the Department of Computer Science & Engineering, D. Y. Patil Technical Campus, Talsande, and the laboratory facilities that made this work possible. The authors also thank the anonymous reviewers for their constructive comments.

REFERENCES

1. S. Ahmed, R. Khan, and M. Ali, "Machine learning-based anomaly detection in smart irrigation systems," Int. J. Agric. Eng., vol. 12, no. 3, pp. 45–58, 2019.
2. Y. Liu, X. Chen, and Z. Wang, "Deep learning approach for irrigation monitoring using convolutional neural networks," Comput. Electron. Agric., vol. 174, p. 105432, 2020.
3. W. Zhang, H. Li, and Q. Zhao, "Predictive maintenance for irrigation systems using LSTM networks," IEEE Sensors J., vol. 21, no. 8, pp. 10234–10244, 2021.
4. J. Wang, T. Sun, and F. Xu, "Machine learning models for irrigation system fault detection," Agric. Water Manag., vol. 258, p. 107188, 2022.
5. A. Raza, B. Mehdi, and C. Nawaz, "Sensor integration for precision irrigation: a review," Sensors, vol. 18, no. 11, p. 3701, 2018.
6. Espressif Systems, ESP32 Technical Reference Manual, v5.1. [Online]. Available: https://docs.espressif.com
7. Blynk Inc., Blynk IoT Platform Documentation, 2023. [Online]. Available: https://docs.blynk.io