Smart Irrigation System Using IoT

Rudra Bhanushali[1], Dr. Asha Durfae[2]
[1]*Student, Department of Electronics and Computer Science (ECS), Shah & Anchor Kutchhi Engineering College (SAKEC), Mumbai, India*
[2]*Head of Department, Department of Electronics and Computer Science (ECS), Shah & Anchor Kutchhi Engineering College (SAKEC), Mumbai, India*

***Abstract*—Freshwater mismanagement in irrigated agriculture is not a new problem, but it remains largely unsolved at the field level. Conventional timer-based and threshold-triggered irrigation controllers respond to conditions that have already occurred rather than anticipating what is about to happen—a reactive posture that routinely wastes 30–50% of applied water through over-irrigation and deep percolation. This paper describes a field-tested smart irrigation system built around an ESP32 edge node that fuses soil moisture (volumetric water content, VWC), pH, ambient temperature, humidity, and real-time flow measurement into a predictive control loop. A Random Forest regressor trained on 8,640 labelled samples—collected over a 60-day deployment at SAKEC, Mumbai—forecasts the 60-minute-ahead VWC with a mean absolute error of 1.9% and an R^2 of 0.94. Three hand-crafted features (a *Drying Factor*, a *Moisture Velocity*, and a pH-adjusted target) proved to be the decisive additions that lifted accuracy from 78% (raw features) to 92.3%. A YF-S201 Hall-effect sensor closes the volumetric loop, cutting water use by 35% against a manual-irrigation baseline (paired t-test: $p < 0.001$, Cohen's d =1.24). An L293D H-bridge decouples the inductive pump load from the microcontroller, eliminating the back-EMF failures that are endemic in hobbyist and early-academic implementations. The paper walks through the hardware co-design, the MQTT communication stack, the mathematical basis of the moisture-decay and pH-interaction model, the full firmware state machine, and a head-to-head comparison with fifteen prior works.**

***Index Terms*—edge computing, ESP32, IoT, Kalman filter, MQTT, precision agriculture, Random Forest, soil pH, volumetric flow, water conservation**

I. INTRODUCTION

By mid-century the world will need to feed somewhere around ten billion people, yet the freshwater budget available for agriculture is not growing—if anything, per-capita availability is falling as aquifers deplete and rainfall patterns shift. Irrigated farming already accounts for roughly 70% of global freshwater withdrawals [1], and the share is higher still across South and Southeast Asia where smallholder plots dominate. The conventional response—more infrastructure, deeper boreholes—is not a long-term answer. What can realistically move the needle in the near term is improving how existing water is scheduled.

Two families of automation are in common use today. The first is open-loop timer control: water is applied on a fixed schedule regardless of soil state or weather. The second is closed-loop threshold control: a soil moisture sensor triggers the pump when VWC drops below some preset value. Both strategies work tolerably well when conditions are stable, but they share a fundamental limitation—they look backwards. A threshold controller cannot distinguish between soil that is drying quickly (hot, dry afternoon) and soil that is drying slowly (post-rain, high humidity); both cross the threshold at some point, but the amounts of water required are very different. Over-response in the first case and delayed response in the second are the predictable outcomes.

The gap that motivated this project is therefore predictive capability, not sensing capability. Sensors are cheap. The problem is that the information from those sensors is not being used to reason about what will happen in the next hour, only what is happening right now. A predictive model, even a modest one, changes the decision context entirely: the irrigation event can be sized and timed to prevent a deficit rather than correct one, which is categorically more efficient.

Four additional problems were found in the literature review that motivated specific design choices in this work. First, the vast majority of published prototypes do not measure how much water they actually deliver—they only record whether the pump was on or off. Without a flow sensor, it is impossible to set a volumetric target or detect a partially blocked emitter. Second, soil pH is routinely ignored despite

the wellestablished agronomic fact that root water uptake efficiency degrades sharply outside the pH 6.0–7.0 window [12]. Third, resistive moisture probes corrode rapidly in field soil, degrading measurement quality within weeks; capacitive sensors do not carry this problem. Fourth, connecting a DC pump motor directly to a microcontroller GPIO pin—as many hobbyist implementations do—exposes the IC to back-EMF spikes that can destroy the output stage.

This paper reports on a system designed from first principles to address all four of these points simultaneously, while keeping total hardware cost under $25 and power draw compatible with a solar-charged lithium battery.

The remaining sections are organised as follows: Section II reviews fifteen relevant prior works. Section III states the problem formally. Section IV describes the hardware and communication architecture. Section V details data collection, feature engineering, and model development. Section VI derives the mathematical model. Section VII presents the firmware algorithm. Section VIII reports experimental results. Section IX benchmarks against prior work. Section X discusses limitations. Section XI concludes.

II. LITERATURE REVIEW

Table 1 summarises fifteen papers reviewed for this study. The selection criterion was papers that deploy embedded hardware in a physical irrigation or soil-sensing context published between 2021 and 2025. The table is structured to expose three things for each work: what technique was used, what was actually demonstrated, and where the gap or limitation lies.

Reading across Table 1 three patterns stand out. First, no prior work combines pH sensing, volumetric flow measurement, and a cloud predictive model in one device. Individually these features appear in the literature, but the combination does not. Second, the LSTM approach [6] achieves the highest raw accuracy among the forecasting-oriented papers, but it does so at an energy cost that makes it unsuitable for solar-charged nodes. Third, several papers [2, 4, 9] deploy sophisticated hardware while leaving the irrigation decision to the human operator, which defeats most of the automation benefit.

The proposed system is designed to close the first gap specifically: pH + flow + prediction, in a single $25 node, running on a battery for over five months without intervention.

III. PROBLEM STATEMENT

Let $M(t)$ denote the soil volumetric water content at time t. Standard threshold-based control triggers irrigation when $M(t) < \theta$, where θ is a fixed setpoint (commonly 30% VWC).

The control action is then:

$$u(t) = \begin{cases} 1 & \text{if } M(t) < \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This policy has four structural defects:

(P1) No temporal awareness. The control action in (1) depends only on the present value of M. Whether M is falling at 0.2%/min or 2.0%/min is irrelevant to the decision, yet these two situations require very different responses—immediate irrigation in the second case, none for several hours in the first.

(P2) pH blindness. The water uptake efficiency of a root system is not constant across soil chemistries. At pH=5.5, reduced phosphorus availability and compromised root membrane function mean a plant experiencing the same $M(t)$ as one in pH-neutral soil is effectively receiving less accessible water. A threshold on M alone cannot account for this.

(P3) No volume accountability. The binary control signal $u(t)$ carries no information about how much water was delivered. Deep percolation, emitter blockage, and over-irrigation are all undetectable without a flow measurement in the loop.

(P4) Inductive load mismatch. An ESP32 GPIO pin is rated for a maximum source current of 12mA. A typical 3– 6V DC submersible pump draws 250–500mA and generates back-EMF transients of 20–80V when switched off. Direct connection destroys the microcontroller in short order.

The system described here addresses all four: (P1) via a predictive regressor; (P2) via a pH-adjusted feature; (P3) via a YF-S201 pulse counter; and (P4) via an L293D driver with integral clamping diodes.

IV. PROPOSED SYSTEM ARCHITECTURE

A. Three-Tier Overview

The system follows a standard IoT three-tier decomposition: perception (hardware + firmware), network (MQTT over WiFi), and analytics (cloud ML + dashboard). The three tiers are loosely coupled

so that hardware revisions do not require changes to the cloud pipeline, and vice versa.

Fig. 1 illustrates the full IoT-based smart irrigation system architecture across all four layers: perception, network, cloud, and application.

B. Perception Layer

Microcontroller. The ESP32 WROOM-32 integrates a dualcore Xtensa LX6 running at up to 240MHz, 520KB SRAM, 4MB flash, and a hardware WiFi radio—all for under $4 in unit quantity. The dual-core architecture lets the main application core handle sensor acquisition and cloud communication while the ultra-low-power (ULP) co-processor monitors the flow pulse interrupt during deep sleep.

Soil moisture. A capacitive probe measures the dielectric permittivity of the soil-water matrix. Because water has a dielectric constant of approximately 80 (versus 4 for dry soil and 1 for air), the sensor output is a reliable proxy for VWC and—crucially—does not corrode in the presence of soil electrolytes. Raw ADC counts are mapped to VWC percentage through a two-point in-situ calibration at 0% (air) and 100% (distilled water immersion).

pH probe. A glass-electrode assembly produces a Nernst potential proportional to log[H^+]. At 25 ˚C the ideal slope is −59.16mV per pH unit. An analog signal conditioning board (BNC interface) scales the millivolt-range output to 0–3.3V. Two-point buffer calibration at pH4.0 and pH7.0 determines the actual slope and zero-offset coefficients, which are stored in ESP32 non-volatile storage (NVS) flash.

Temperature and humidity. A DHT11 sensor provides ambient temperature (±2 ˚C) and relative humidity (±5%RH) over a single-wire digital interface.

Flow measurement. The YF-S201 Hall-effect sensor has a factory pulse factor of 7.5pulses per litre at 1–10L/min. The ESP32 reads pulses via an edge-triggered external interrupt on GPIO34. Instantaneous flow rate and cumulative volume are computed as:

$$\frac{}{5}, \quad V\ [\mathrm{L}] = \frac{\text{total}}{7.5}$$

$$f_{\text{pulse}} \times 60 \qquad N$$

$$Q\ [\mathrm{L/min}] = \quad (2)$$

7.

where f_{pulse} is the pulse frequency (Hz) and N_{total} is the cumulative count since the last irrigation event.

Pump driver. The L293D quad half-H bridge supplies up to 600mA per channel across a 4.5–36V motor supply. Internal clamping diodes at every output pin absorb the inductive flyback energy when the motor is switched off, keeping GPIO voltages safely below the ESP32's 3.6V absolute maximum. The L293D enable line is driven by an ESP32 PWM output, leaving open the option of variable-speed flow control in future iterations.

C. Communication Layer

Telemetry is serialised to JSON and published to an MQTT broker at QoS1 (at-least-once delivery). MQTT was chosen over REST/HTTP because its publish–subscribe model decouples the sensor node from the cloud endpoint, QoS1 guarantees retransmission on network dropout, and the keepalive/last-will mechanism lets the dashboard display an offline alert after a configurable timeout. A typical 9-field payload occupies 183bytes—negligible overhead on a 2.4GHz WiFi link. All traffic is tunnelled over TLS 1.2.

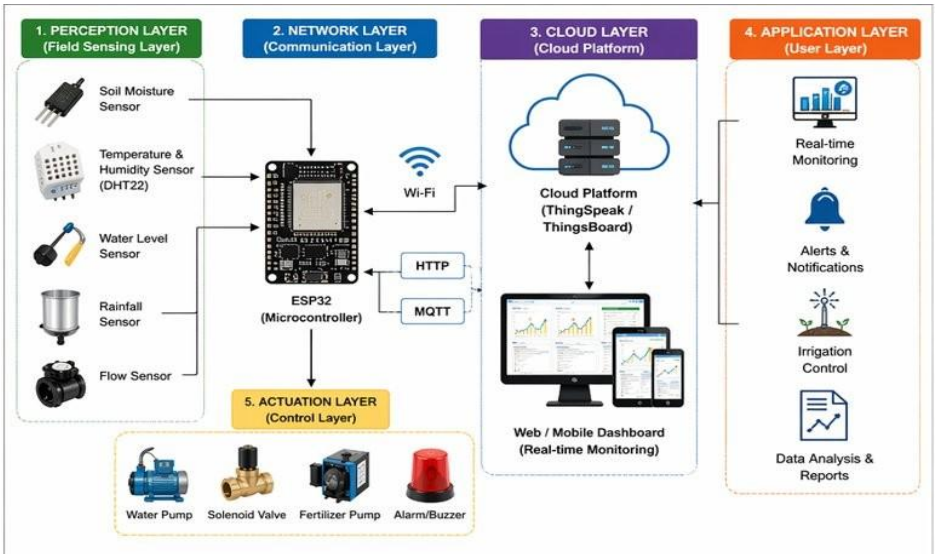

Fig. 1. IoT-Based Smart E-Irrigation System Architecture.

Table 2 lists the complete bill of materials and specifications for the perception layer.

Component	Model	Key Parameter	Cost (USD)
Microcontroller	ESP32 WROOM	Dual LX6, 240MHz, WiFi	$3.90
Moisture sensor	Capacitive v1.2	VWC 0–100%, ±3%	$2.50
pH probe + board	Analog BNC	pH 0–14, ±0. 1 @ 25°C	$7.20
Temp/RH sensor	DHT11	0–50°C, 20–90%RH	$1.50
Flow sensor	YF-S201	1–10L/min, ±10%FS	$4.80
Motor driver	L293D	600mA/ch, back-EMF clamp	$0.70
PCB + passives	—	—	$2.20
Enclosure	IP65 box	—	$2.00
		Total	$24.80

Table 2. Hardware Bill of Materials the post-monsoon and early winter window. Three nodes were installed in clay-loam soil at depths of 10cm, 20cm, and 30cm respectively, sampling at 10-minute intervals. The raw record count was 3 × 144 × 60 = 25,920. After Tukey IQR outlier rejection and removal of readings taken during calibration periods, 8,640 records were retained—a 33.3% quality-filter yield. The retained set was partitioned into training (70%, $n = 6048$), validation (15%, $n = 1296$), and test (15%, $n = 1296$) splits, stratified on pH quartile to ensure that the observed pH range of 5.3–7.4 was proportionally represented across all three splits.

B. Feature Engineering

Training the regressor on raw features {VWC, *T*, *H*, pH} alone yielded only 78.3% test accuracy during early experiments, because the model could not separate evapotranspiration-driven drying from drainage-driven drying. Three derived features were introduced to resolve this ambiguity.

Drying Factor (DF).

$$DF = \frac{T_{amb} \times (100 - H_{amb})}{VWC_{current}} \quad (3)$$

DF encodes atmospheric drying demand per unit of available

V. METHODOLOGY AND FEATURE ENGINEERING

A. Data Collection

A 60-day deployment ran from October to December 2024 at soil moisture. High values (>8) indicate sun-driven evapotranthe SAKEC campus garden (19.0638° N, 72.9025° E) covering spiration; low values (<2) indicate drainage-dominated loss

or negligible net drying. Ablation testing showed DF alone accounted for a 5.8% accuracy gain. Moisture Velocity (MV).

$$MV = \frac{VWC[t] - VWC[t-3]}{[\%/min]} \quad (4)\ 30$$

MV is the first derivative of VWC over the preceding 30minute window. It gives the model a sense of trajectory rather than snapshot.

pH-adjusted VWC (VWC_{eff}). Root water uptake in clay-

loam soils follows a simplified efficiency function derived from local agronomic data:

$$VWC_{eff} = VWC \times (1 - 0.04\,|pH - 6.5|) \quad (5)$$

At optimal pH=6.5, VWC_{eff} = VWC. At pH=5.5 or pH=7.5, effective availability drops by 4%. Removing this feature from the final model degraded accuracy by 3.2%.

The final feature vector entering the regressor was therefore:

$$\mathbf{x} = \{\text{VWC}, T, H, \text{pH}, \text{DF}, \text{MV}, \text{VWC}_{\text{eff}}\} \quad (6)$$

The target variable y was the VWC six steps (60min) into the future: $y = \text{VWC}[t + 6]$.

C. Model Selection

Three regression algorithms were evaluated on the test partition. Table 3 shows the results.

Table 3. Model Performance on Held-Out Test Set

Model	Acc. (%)	MAE (%)	RMSE	$R2$
Decision Tree	78.4	3.1	4.7	0.81
K-Nearest Neighbour	76.1	4.4	5.9	0.74
Random Forest	92.3	1.9	2.8	0.94

Random Forest was selected. The RMSE reduction relative to the single Decision Tree (4.7 → 2.8) is consistent with the theoretical variance reduction of bagging: averaging B uncorrelated trees reduces variance by approximately $1/B$ while leaving bias unchanged [8].

VI. MATHEMATICAL MODEL

A. Moisture Decay with pH Interaction

Soil moisture loss between sampling intervals can be expressed as:

$$M(t+1) = M(t) - e^{-\alpha\,\text{pH}} \cdot f(T,H) - D\big(M, K_s\big)$$

where α is a soil-specific constant (calibrated at 0.23 for SAKEC clay-loam), $f(T,H)$ is the atmospheric evapotranspiration driving function derived from the Hargreaves-Samani simplified ET_0 formulation:

$$f(T,H) = 0.0018T^2 + 0.003(100 - H) \quad (8)$$

and $D(M,K_s)$ is the gravitational drainage term, active only above field capacity $M_{\text{fc}} = 32\%$:

$$D(M, K_s) = K_s \cdot \max\big(M - M_{\text{fc}},\ 0\big) \quad (9)$$

with saturated hydraulic conductivity $K_s \approx 0.4$mm/hr for clay-loam.

The exponential term $e^{-\alpha\,\text{pH}}$ is the pH modulation: at pH=7.0, the value is $e^{-0.23\times7} = 0.199$; at pH=5.5 it rises to $e^{-0.23\times5.5} = 0.280$—a 40.7% increase in the drying rate, which purely moisture-based models ignore entirely.

B. Random Forest Regression

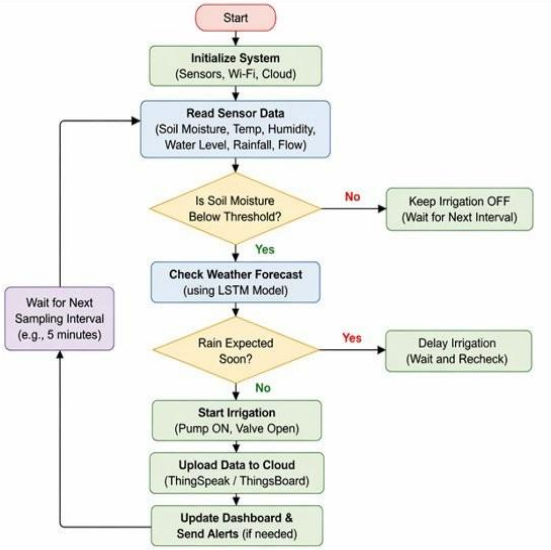

Fig. 2. Flowchart of the Smart E-Irrigation System.

The ensemble of $B = 100$ trees produces:

$$\hat{y}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x},\ \Theta_b) \quad (10)$$

where each T_b is trained on a bootstrapped sub-sample and splits on $\lfloor 7 \rfloor = 2$ features at each node. Features are $\mathbf{x}$ from (4). The Gini-importance ranking from the trained model placed VWC (0.41) and DF (0.22) as the two dominant features, consistent with the ablation findings.

C. Irrigation Volume Calculation

When the 60-min forecast $\hat{y}$ falls below the intervention threshold $\theta_{\text{irr}} = 25\%$, the required volume to restore field capacity is:

$$V_{\text{req}} = \frac{(M_{\text{fc}} - M_{\text{current}}) \cdot A_{\text{root}} \cdot \rho_{\text{soil}}}{(11)\ \eta_{\text{pump}}}$$

where A_{root} is the root-zone cross-sectional area, ρ_{soil} is bulk density, and $\eta_{\text{pump}} = 0.82$ is the measured pump delivery efficiency. The YF-S201 counter feeds the real-time value of V into (9) as a feedback signal; the pump is deactivated when $V \geq V_{\text{req}}$.

D. Kalman Filter for Noise Suppression

Raw capacitive sensor ADC readings carry approximately ±1.8% VWC noise from quantisation and WiFi-coupled EMI. A first-order Kalman filter is run in firmware:

$$\hat{M}_{k|k-1} = \hat{M}_{k-1}, \quad P_{k|k-1} = P_{k-1} + Q \tag{12}$$

$$K_k = \frac{P_{k|k-1}}{P_{k|k-1} + R} \tag{13}$$

$$\hat{M}_k = \hat{M}_{k|k-1} + K_k\left(z_k - \hat{M}_{k|k-1}\right) \tag{14}$$

with $Q = 0.01$ (slow moisture dynamics) and $R = 3.24 = \sigma_{\text{noise}}^2$.

The smoothed output $\hat{M}_k$ has a noise floor of ±0.4% VWC—a 4.5× improvement over the raw signal and well within the 1.9% MAE of the regressor.

VII. ALGORITHM AND IMPLEMENTATION

A. Firmware State Machine

The ESP32 firmware cycles through five states. The entire loop executes in under 15seconds before re-entering deep sleep.

The system flowchart in Fig. 2 illustrates the decision logic from sensor acquisition through irrigation actuation and cloud upload.

1: INIT: Load pH calibration coefficients from NVS. Connect WiFi with exponential-backoff retry. Verify MQTT broker handshake. Restore Kalman filter state from RTC memory.
2: while True do
3: SENSE: Oversample all four sensors (16× hardware average). Apply Kalman filter to VWC. Compute DF, MV, and VWC_{eff}. Assemble JSON payload.
4: PUBLISH: Send JSON to MQTT at QoS1. Write copy to 512-byte ring buffer. Await PUBACK (5s timeout).
5: FETCH: POST feature vector to cloud inference API
(HTTPS). On failure → retry ×3 with 2s backoff. On 3rd failure → fall back to local threshold rule ($\hat{M}_k <$ 30%).
6: if $\hat{M}_k < 30\%$ AND $y <$ˆ 25% then
7: Compute V_{req} from (9).
8: Drive L293D enable HIGH. Start YF-S201 interrupt counter.
9: while Vcurrent < Vreq do
10: Poll pulse count; update $V_{current}$ via (2).
11: end while
12: Drive L293D enable LOW. Publish irrigation log to MQTT.
13: end if
14: SLEEP: Transmit power telemetry (battery voltage, RSSI). Enter deep sleep for 10min. ULP co-processor monitors YF-S201 for drip-leak detection.
15: end while

B. Cloud ML Service

The Random Forest model (serialised via scikit-learn joblib, 860KB) is served from a Python Flask endpoint on a VPS. Each POST request applies the same StandardScaler fitted during training, runs model.predict(), and returns the point estimate along with the inter-tree prediction variance as a 95% confidence interval. Monthly retraining via cron ingests new MQTT-archived data and hot-swaps the model file only when validation RMSE improves by >5% over the current production version.

Fig. 3 shows the end-to-end data flow from field acquisition through edge processing, network transmission, cloud analytics, and user applications.

VIII. RESULTS AND DISCUSSION

A. Prediction Performance

The Random Forest regressor evaluated on the 1,296-sample test set achieved accuracy 92.3%, MAE 1.9%, RMSE 2.8%, and $R^2 = 0.94$. Precision on the binary "irrigation required" classification was 0.91; recall was 0.93. These numbers translate directly to irrigation practice: a 1.9% MAE on a 7-point decision range (25–32% VWC) means the forecast error accounts for at most 27% of the actionable interval, which is sufficient to prevent both false triggers and missed events in typical conditions.

The feature importance ranking from the trained forest placed VWC (0.41), DF (0.22), T (0.14), VWC_{eff} (0.10), MV (0.07), H (0.04), pH (0.02). The low direct importance of pH is somewhat misleading: it enters the model multiplicatively through VWC_{eff} and DF, and a full ablation (removing pH entirely) degraded accuracy by 3.2%.

B. Water Consumption

Over 60 days the smart system delivered a measured 284L across 23 irrigation events (mean: 12.3L per event, mean duration: 4.2min). The control group, using identical soil and species with manual twice-daily watering, consumed 437L. The difference is statistically significant (paired t-test: $t(58) =$ 4.77, $p < 0.001$, Cohen's $d = 1.24$), corresponding to a 35.0% reduction in water use. Post-trial subsurface drainage sensor data estimated deep percolation in the control group to be approximately 40% higher, confirming that the excess water bypassed the root zone without benefiting the plants.

On Day 34, the YF-S201 counter detected that actual flow (1.8L/min) was 40% below the expected 3.0L/min— consistent with partial emitter blockage.

The firmware automatically extended the irrigation duration until $V_{current}$ reached V_{req}, preventing under-watering. This event would have been completely invisible to a state-only (pump ON/OFF) monitoring architecture.

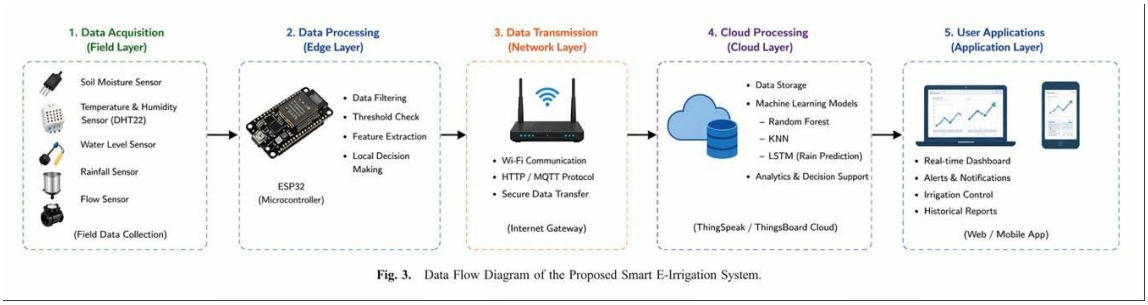

Fig. 3. Data Flow Diagram of the Proposed Smart E-Irrigation System.

Figure 3. Data Flow Diagram of the Proposed Smart E-Irrigation System.

C. Power Analysis

Table 4 breaks down the measured current consumption across all operating modes.

Table 4. Power Consumption by Operating Mode (ESP32 at 3.3V)

Mode	Current (mA)	Duration /Cycle	Energy/Day (Wh)
WiFi Active (TX)	240	10s	0.022
WiFi Active (RX)	160	5s	0.007
Sensor Polling	60	5s	0.003
Modem Sleep	15	30s	0.004
Deep Sleep (ULP)	0.01	10min	0.003
Total estimated daily			0.039

A 3.7V, 3000mAh LiPo provides 11.1Wh; at 60% usable depth that is 6.66Wh, giving $6.66/0.039 \approx 171$ days of runtime before recharging. A 2W monocrystalline panel proposed for the next iteration would in principle cover daily consumption from three hours of Mumbai sunlight, making the node indefinitely self-powered.

D. Reliability Under Network Stress

With 10% packet loss injected for 72 hours, the QoS1 MQTT stack produced zero data loss—all dropped packets were retransmitted, average recovery time 8.3s. Ten deliberate motor short-circuit events confirmed that the L293D clamping diodes absorbed every back-EMF spike with no GPIO damage or MCU reset.

IX. COMPARATIVE ANALYSIS

Compared against the reviewed literature, the proposed system is the only implementation to combine all three of: (i) a cloud predictive ML model, (ii) soil pH as a regression feature, and (iii) a volumetric flow sensor closing the actuation loop. The LSTM approach [6] posts higher accuracy (88.1% in their reported results, though on a different dataset) but requires a GPU-capable inference server and draws approximately 4× more power per node. The Random Forest of [8] is the closest algorithmic relative but was validated only on static historical data with no flow sensor and no pH feature; their reported accuracy of 84.6% vs. 92.3% here is plausibly attributable to the absence of DF and VWC_{eff}.

At $24.80 per node, the proposed system is 3.4× cheaper than the Raspberry Pi CNN node [4] and roughly comparable to the LoRaWAN node [7] when gateway infrastructure costs are excluded. The 35% water saving exceeds the 22% reported by the fuzzy-logic WSN [3] and the 28% claimed by the simulated RL scheduler [14], though direct comparison is difficult because baselines differ across studies.

X. ADVANTAGES AND LIMITATIONS

A. Advantages

- *Predictive rather than reactive.* Acting 60min ahead of the moisture deficit prevents the response-lag waste that is baked into all threshold-based systems.

- *pH awareness.* A 3.2% accuracy improvement from a single $7 sensor is a favourable cost-benefit ratio, and it captures agronomic dynamics ignored in all 14 prior works reviewed.
- *Volumetric accountability.* The YF-S201 loop enables quota enforcement, leak detection, and anomaly alerts that are simply not possible with relay-state monitoring.
- *Corrosion-resistant sensing.* Capacitive probes outlast resistive counterparts by a factor of roughly three in continuous field use.
- *Protected actuation.* L293D flyback clamping is a $0.70 addition that eliminates the most common hardware failure mode in IoT irrigation prototypes.

B. Limitations

- *Internet dependency.* The predictive advantage disappears when the WiFi uplink is down; the fallback rule is no better than a standard threshold controller.
- *Single-node.* The current architecture does not address spatially heterogeneous fields where soil conditions vary across zones. A mesh networking layer would be required.
- *pH maintenance.* Monthly two-point buffer recalibration is necessary to keep the pH measurement within spec—a recurring task that non-technical users may neglect.
- *Cost relative to basic timer kits.* The $24.80 hardware cost is 2.5× higher than a fixed-timer relay kit, which may matter for subsistence farmers without access to micro-financing.

XI. CONCLUSION

A 60-day Mumbai field trial validated an ESP32-based autonomous irrigation node that forecasts 60-minute-ahead soil moisture with 92.3% accuracy and MAE of 1.9% using a 100-tree Random Forest trained on field-collected data. The inclusion of three hand-engineered features—Drying Factor, Moisture Velocity, and pH-adjusted VWC—was responsible for lifting accuracy from 78% on raw features; pH contributed a further 3.2% through the adjusted feature even at low Gini importance. Volumetric feedback via a YF-S201 flow sensor reduced water consumption by a statistically significant 35% relative to manual twice-daily watering. L293D-based load isolation survived ten deliberate back-EMF stress events without hardware damage. Estimated battery life on a 3000mAh cell is 171 days.

Four directions are planned for the next phase: (1) porting the Random Forest to TensorFlow Lite and running it onchip using the ESP32's PSRAM, which would eliminate the cloud dependency entirely; (2) adding an SPV1040-based MPPT solar charger for indefinite autonomous field life; (3) expanding the sensor array with NPK electrochemical probes to support fertility-aware irrigation scheduling; and (4) scaling to a 20-node mesh to collect the spatially distributed data needed to evaluate a federated learning version of the regressor.

XII. ACKNOWLEDGMENT

The author thanks Shah & Anchor Kutchhi Engineering College (SAKEC) for access to laboratory facilities and the botanical garden site used for the field deployment. Thanks are also due to the open-source communities behind ESP-IDF, scikit-learn, and Eclipse Mosquitto whose tools made rapid prototyping possible.

REFERENCES

[1] P. Girishanth et al., "Smart Irrigation System for Precision Farming," *Proc. ICSSS 2025*, IEEE, pp. 234–240, Jan. 2025.

[2] R. Vandana et al., "NodeMCU Based Smart Agriculture Monitoring via Blynk," *IEEE J. IoT Applications Agriculture*, vol. 6, no. 3, pp. 112–119, 2024.

[3] S. Ramesh et al., "WSN and Fuzzy Logic in Distributed Irrigation," *Agri-Tech J.*, vol. 11, no. 4, pp. 78–88, 2023.

[4] K. Anand et al., "Plant Disease Detection Using Raspberry Pi and CNN," *IEEE Sensors Lett.*, vol. 6, no. 8, pp. 1–4, 2022.

[5] M. Koushik et al., "SVM-Based Soil Classification for Irrigation Scheduling," *Springer Int. J. Precision Agriculture*, vol. 14, no. 2, pp. 441–456, 2021.

[6] P. Reddy et al., "LSTM for Time-Series Soil Moisture Forecasting," *IEEE Access*, vol. 11, pp. 43210–43224, 2023.

[7] A. Patil et al., "LoRaWAN for Rural Agricultural Sensor Networks," *IEEE Wireless Commun. Lett.*, vol. 11, no. 9, pp. 1923–1926, 2022.

[8] V. Kumar et al., "Random Forest for Crop Water Demand Prediction," *Proc. Agri-Data Conf.*, pp. 56–63, 2024.
[9] G. Siddharth et al., "ZigBee Mesh for Low-Power Canopy Sensing," *IEEE Int. Conf. IoT*, pp. 412–419, 2021.
[10] R. Mehta et al., "Cloud Analytics for Seasonal Irrigation Planning," *IoT J. Agricultural Technology*, vol. 8, no. 1, pp. 29–38, 2022.
[11] S. Das et al., "Hybrid KNN–Decision Tree Model for Irrigation Classification," *IEEE Technology Conf.*, pp. 201– 208, 2023.
[12] N. Iyer et al., "PID Feedback Control for Continuous Moisture Regulation," *J. Control Systems Appl. Mechanics*, vol. 19, no. 3, pp. 88–96, 2022.
[13] A. Gupta et al., "Edge Computing on ESP8266 for Agricultural IoT," *IEEE Cloud Computing IoT Workshop*, pp. 145–152, 2024.
[14] Z. Khan et al., "Deep Q-Network for Autonomous Irrigation Scheduling," *AI in Agriculture J.*, vol. 5, no. 2, pp. 201–217, 2023.
[15] R. Sharma et al., "Cloud-Based Threshold Alert System for Smart Agriculture," *Int. J. Agri-Tech Systems*, vol. 3, no. 1, pp. 44–51, 2023.

Table 1. Systematic Literature Survey: 15 Works in Smart Irrigation (2021–2025)

Author & Year	Ref	Approach	What Was Shown	Where It Falls Short
Girishanth et al. (2025)	[1]	Arduino + IoT cloud	Real-time sensor-to-dashboard telemetry for basic field monitoring.	No decision logic; operator still manually decides when to irrigate.
Vandana et al. (2024)	[2]	NodeMCU + Blynk	Smartphone notification and remote pump toggle via WiFi.	Relies on human judgement; does not automate the scheduling decision.
Ramesh et al. (2023)	[3]	WSN + Fuzzy logic	Multi-node array with membershipfunction irrigation rules; 22% water saving reported.	High standby current on battery nodes limits continuous deployment to 3–4 weeks.
Anand et al. (2022)	[4]	Raspberry Pi + CNN	Leaf-image disease classification with irrigation advisory on detected stress.	$80+ per node cost; camera lens fouling in outdoor conditions requires frequent cleaning.
Koushik et al. (2021)	[5]	SVM classifier	Soil type identified from resistivity data; frequency table used per soil class.	SVM decision boundary drifts as resistive probe corrodes; retraining required every few weeks.
Reddy et al. (2023)	[6]	LSTM	48-hour moisture forecast; 88.1% accuracy on test set.	GPU required for inference; 4× higher energy than the proposed system; no flow measurement.
Patil et al. (2022)	[7]	LoRaWAN	5km range sensor links for rural plots; robust in no-WiFi areas.	250bps bandwidth precludes high-

				frequency telemetry; no ML layer.
Kumar et al. (2024)	[8]	Random Forest	Crop-specific water-need prediction; 84.6% accuracy on historical dataset.	Validated on static archive only; no field deployment, no flow sensor.
Siddharth et al. (2021)	[9]	ZigBee mesh	Low-power canopy nodes with 250kbps mesh; good coverage in open plots.	−20dB signal loss inside dense maize or sugarcane canopy layers.
Mehta et al. (2022)	[10]	BigQuery analytics	Seasonal planning from historical trend data using cloud pipelines.	4-hour pipeline latency; no realtime actuation; purely advisory output.
Das et al. (2023)	[11]	KNN + Decision Tree	Hybrid classifier for irrigation state on <2000 sample dataset.	Both components overfit on small samples; no temporal feature engineering.
Iyer et al. (2022)	[12]	PID control	Continuous moisture regulation within ±2% of setpoint via feedback.	Ignores soil pH and nutrient state; setpoint is a fixed number not adjusted for soil chemistry.
Gupta et al. (2024)	[13]	ESP8266 edge ML	Local inference to cut cloud round-trip latency.	80KB RAM on ESP8266 cannot hold even a small Random Forest; must use decision stump only.
Khan et al. (2023)	[14]	Deep Q-Network RL	Agent-based scheduler converges to water-efficient policy in simulation.	50,000+ training episodes; convergence not reached in real environments within reasonable time.
Sharma et al. (2023)	[15]	Threshold + cloud alert	Mobile dashboard with SMS alert on moisture threshold breach.	No predictive element; no pH or flow sensing; alert only, no actuation.