



AI-BASED SMART IRRIGATION SYSTEM

Shravani Parade¹

¹ Dattakala Group of Institutions, Faculty of Engineering, Pune, Maharashtra, India, India

Corresponding author: shravaniparadepatil48@gmail.com

Guide / Supervisor: Mrs. Dhage T.S.

ARTICLE ID IJREK-2026-00099	ARTICLE TYPE Research Article	RESEARCH AREA Artificial Intelligence	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 19 October 2026	REVISED 20 October 2026	ACCEPTED 19 October 2026	PUBLISHED 20 October 2026

Abstract

Water availability, irregular rainfall, changing temperatures, and inefficient irrigation practices create pressure on farm productivity and operating costs. This paper presents a research-based design for an AI-based smart irrigation system that combines soil-moisture sensing, local weather information, crop and soil characteristics, and a decision engine to recommend or automate irrigation. The proposed architecture uses sensor nodes to collect field measurements, a controller to validate readings, and a rule-based safety layer supported by machine-learning prediction where suitable labelled data are available. Irrigation is triggered only when estimated crop water need, sensor evidence, and safety constraints agree; rainfall forecasts, sensor faults, pump limits, and manual override are considered before actuation. A review of recent literature indicates that machine learning can support irrigation scheduling, but performance depends on data quality, calibration, crop context, and validation across locations. Unlike an experimental report, this paper does not claim measured water savings or crop-yield gains. It defines a practical architecture, control workflow, evaluation protocol, and deployment risks that can guide a student prototype and subsequent field testing.

Keywords: Smart irrigation; artificial intelligence; Internet of Things; soil-moisture sensing; machine learning; precision agriculture; water-use efficiency

I. INTRODUCTION

A. Background and Context Irrigation decisions are often made using fixed schedules, visual observations, or a farmer's experience. These methods remain valuable, but they may not fully reflect short-term changes in soil moisture, rainfall, crop growth stage, or local microclimate. Water applied too early can drain below the root zone, while delayed irrigation can expose plants to water stress. A digital system can combine repeated field measurements with weather and crop information to make irrigation decisions more transparent and responsive. The Food and Agriculture Organization of the United Nations highlights efficient irrigation methods and water management as practical components of water conservation and climate-resilient agriculture [1]. B. Problem Statement A useful smart irrigation system must solve more than the task of switching a pump on when a sensor crosses a threshold. Soil-moisture sensors can drift or fail; weather feeds may be unavailable; different crops require different moisture ranges; and pumps have electrical and hydraulic limits. A model trained on one field may not generalize to another soil type or season. The central problem is therefore to design a dependable decision process that uses data intelligently while preserving safe operation, farmer control, and understandable recommendations. C. Aim and Objectives • Design a modular architecture for collecting soil, weather, crop, and irrigation-system data. • Define a transparent baseline controller and show where machine learning can improve water-need estimation. • Specify safeguards for sensor faults, rainfall, water availability, pump runtime, and manual override. • Propose measurable evaluation criteria without inventing experimental results. • Identify deployment barriers relevant to small and medium farms, including cost, connectivity, maintenance, and usability. D. Scope and Contribution The work is a literature-informed system proposal rather than a claim of completed field experimentation. Its contribution is the integration of a measurable baseline, a forecast-aware decision layer, a clear feedback loop, and an evaluation plan. The design can be implemented with a microcontroller, soil-moisture probes, a relay or motor driver, a pump or valve, and a web/mobile dashboard. AI is treated as a component that must be justified by evidence—not as a substitute for agronomic knowledge or safe control logic.

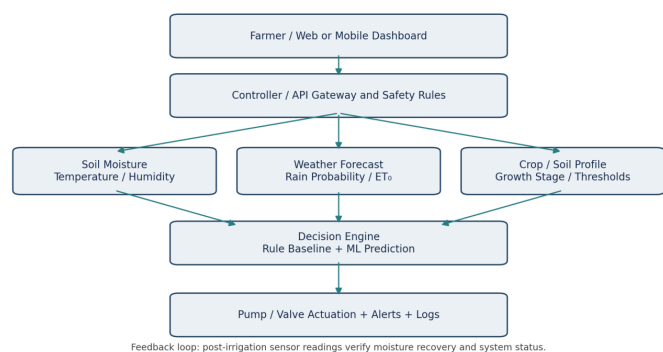
II. LITERATURE REVIEW AND RELATED WORK

TABLE I

Research theme	Implication for the proposed system
Soil-moisture sensing [2,7]	Calibrate probes and use representative sensor placement; do not rely on one unverified reading.
ML-based water-need estimation	Compare models against a threshold baseline and validate on different days or plots.
[3,4,6]	
Multi-source intelligent irrigation [5]	Combine local sensing with weather and crop context while tracking source quality.
Water conservation [1]	Evaluate applied water and crop condition together; less water is not automatically better if plants become stressed.

III. PROPOSED SYSTEM AND METHODOLOGY

A. System Architecture The proposed system follows a sensing–decision–actuation feedback loop. Sensor nodes measure root-zone soil moisture and, where useful, temperature and humidity. A weather service supplies forecast rainfall and atmospheric conditions. Crop metadata describes crop type, growth stage, rooting depth, and locally validated moisture thresholds. The controller validates incoming data, estimates irrigation need, checks safety rules, and either actuates a pump/valve or sends a recommendation to the farmer. After irrigation, new sensor readings are collected to verify whether moisture has moved toward the target range.

**Fig. 1. Proposed architecture of the AI-based smart irrigation system.****TABLE II**

Component	Purpose	Important design consideration
Soil-moisture sensors	Estimate water availability in the root zone.	Calibration, placement, corrosion, and sensor drift.
Weather input	Provide rainfall forecast and relevant atmospheric variables.	Forecast uncertainty and API outage handling.
Controller / edge node	Read sensors, validate data, execute local rules.	Continue safe baseline operation during network loss.
AI decision module	Estimate water need or irrigation probability.	Use only validated data and expose confidence/limits.
Pump / valve driver	Apply irrigation for a bounded duration.	Electrical isolation, dry-run protection, maximum runtime.
Dashboard and alerts	Show readings, recommendations, status, and overrides.	Readable language, low-bandwidth design, privacy.
Database / logs	Store readings, decisions, and irrigation events.	Timestamps, units, missing-value flags, backup.

C. Data Inputs and Pre-processing Each record should include a timestamp, sensor identifier, field or zone identifier, soil-moisture value and unit, sensorhealth status, crop profile, recent irrigation event, and relevant weather fields. Data validation checks physically implausible readings, stale timestamps, duplicate records, and abrupt changes inconsistent with sensor behaviour. Missing values should be marked as missing rather than silently replaced with a normal value. If a critical sensor is invalid, the controller should suspend autonomous watering for that zone or revert to a documented conservative policy and alert the farmer.

D. Decision Strategy The recommended approach is hybrid. First, a transparent rule-based baseline uses calibrated moisture thresholds and a maximum runtime. Second, an AI model may estimate short-horizon moisture decline or irrigation need from historical sensor readings, weather, crop stage, and prior watering. Third, a safety layer checks rainfall, water availability, actuator state, and pump limits before any command is issued. The model may recommend an action, but it must not bypass safety rules. If the model confidence is low or the inputs fall outside the training range, the system should use the validated baseline or request human confirmation.

E. Working Principle At each scheduled decision interval, the controller obtains fresh readings, verifies sensor status, and estimates whether the root-zone moisture is approaching a crop-specific lower limit. It then considers expected rainfall and the time since the last irrigation. If the moisture deficit is meaningful and no blocking condition is present, the controller calculates a bounded watering duration or asks the farmer to approve it. The pump is switched off at the end of the permitted interval, and a post-irrigation reading is recorded. The process repeats, allowing the system to learn from observed outcomes only after data quality has been checked.

IV. CONTROL ALGORITHM AND SAFETY LOGIC

TABLE III

Step	Operation
1	Read moisture sensors, weather data, crop profile, and actuator status.
2	Check timestamps, sensor ranges, connectivity, and data quality flags.
3	If a critical reading is invalid, stop automatic actuation and notify the farmer.
4	Estimate moisture deficit and near-term irrigation need using baseline rules and, if validated, an ML model.
5	Check rainfall forecast, water availability, pump status, maximum runtime, and minimum off-time.
6	If all conditions permit, actuate for a bounded duration or send a recommendation for approval.
7	Stop the pump, record water/runtime estimates, and re-read moisture after an appropriate delay.
8	Log the decision and outcome for monitoring and later model evaluation.

Proposed control workflow (conceptual; requires field validation)**Fig. 2. Proposed decision and feedback workflow.**

V. IMPLEMENTATION PLAN

TABLE IV

Layer	Candidate implementation	Notes
Sensing	Capacitive soil-moisture sensor; optional air temperature/humidity sensor.	Select rugged components and calibrate in the target soil.
Edge control	ESP32 or comparable microcontroller; relay/driver and pump/valve.	Electrical protection and enclosure are required.
Backend	Python API (FastAPI/Flask) or local service.	Validate requests and separate model logic from actuation.
Data store	SQLite for prototype; PostgreSQL/MongoDB for larger deployments.	Store time series, events, device health, and configuration.
AI analysis	Python with scikit-learn or another validated library.	Train/test split must avoid leakage across time and plots.
User interface	Mobile-friendly dashboard with readings and alerts.	Display units, freshness, reason for action, and override.

B. Software Workflow The firmware should acquire measurements at a defined interval and attach timestamps and sensor-health flags. The backend validates payloads and stores observations. The decision service combines the latest valid readings with the crop configuration and weather input. It then produces a structured recommendation containing the reason, expected duration, data freshness, and confidence or uncertainty note. Before a physical command is sent, the edge controller verifies local interlocks. Irrigation events and post-event readings are logged so the operator can inspect why an action occurred.

C. Data Management and Security The data model should distinguish sensor observations, crop/zone configuration, weather snapshots, model predictions, irrigation commands, and actual actuator outcomes. Each event should include a timestamp and device identifier. API keys must remain server-side; device communication should use authentication and encrypted transport where supported. Firmware and model

updates should be versioned, and changes to thresholds should be logged. A farmer should be able to export records and delete unnecessary personal information. Data retention should be proportional to operational and evaluation needs. D. Deployment Considerations For small farms, the system should minimize recurring data costs, tolerate intermittent connectivity, and make calibration and maintenance instructions clear. Solar power may be considered where appropriate, but battery life and pump energy must be measured rather than assumed. Sensor placement should represent the root zone and relevant irrigation zones; one probe may not represent a heterogeneous field. Before scale-up, a prototype should be tested in a controlled plot and then compared with current farmer practice under similar crop, soil, and weather conditions.

VI. EVALUATION PLAN AND DISCUSSION

TABLE V

Applied irrigation water	Flow meter or calibrated volume per plot and period.	Compare water use only with comparable crop and weather conditions.
Soil-moisture stability	Time within a crop-specific target band; duration below threshold.	Indicates whether water availability is maintained.
Crop response	Yield, plant-health observations, or a suitable agronomic measure.	Avoid water savings that are achieved by harming the crop.
Control reliability	Successful commands, missed triggers, false triggers, fault events.	Measures system safety and operational robustness.
Prediction quality	MAE/RMSE for numeric estimates; precision/recall for events.	Use time-aware validation and compare with a simple baseline.
Usability and cost	Setup time, maintenance, energy, connectivity cost, user feedback.	Assesses suitability for the intended farm context.

C. Analysis Plan For continuous outcomes, report averages together with variability and sample counts; for irrigation-event classification, report precision, recall, and false-trigger rate. Separate training, validation, and test data chronologically where time series are involved, and keep data from the same plot or sensor from leaking across splits when the goal is cross-field generalization. Compare the AI model with a threshold-only baseline. If the difference is small or unstable, the simpler controller may be preferable because it is easier to explain, maintain, and audit. All reported gains must be derived from recorded measurements, not illustrative values. D. Expected Behaviour and Interpretation The design is expected to make decisions more responsive to field conditions than a rigid timer because it incorporates measured moisture and contextual information. However, actual benefits will vary with soil, crop, irrigation method, sensor quality, weather forecast accuracy, and farmer practices. Forecast-aware control can reduce unnecessary watering when rainfall is likely, but forecasts may be wrong. A system that prioritizes water conservation without tracking crop stress may under-irrigate. Consequently, water-use metrics should always be interpreted alongside crop condition and reliability measures.

VII. LIMITATIONS, ETHICS, AND FUTURE SCOPE

A. Limitations • Sensor measurements are local and may not represent the full field; installation and calibration affect reliability. • Weather forecasts and network services may be unavailable, delayed, or inaccurate. • ML models need representative labelled

data and may not transfer across crops, soils, and seasons. • A software-only design cannot directly detect all plant stresses or hydraulic faults without suitable sensors. • Hardware, maintenance, power, connectivity, and farmer training may limit adoption. • The present paper describes a proposed system and evaluation protocol; it does not establish measured water savings, yield improvement, or cost reduction. B. Responsible Use Irrigation recommendations should complement, not replace, local agronomic advice. The interface should explain why watering is recommended, show the freshness of sensor and weather data, and make manual override accessible. Systems should not silently change crop thresholds or apply model updates without testing. Any field study should obtain informed participation where user data are collected and avoid publishing identifiable farm information without permission. C. Future Scope Future development can incorporate evapotranspiration estimates, satellite or drone observations, crop-growth models, multi-zone optimization, solar-powered sensing, and low-connectivity modes. Federated or privacy-preserving learning may be investigated where multiple farms contribute data, but its practical value must be weighed against complexity. Additional research should focus on cross-season validation, robust sensor-fault detection, transparent decision explanations, and affordability in smallholder settings. A later version may integrate regional-language voice prompts and local agricultural advisories.

VIII. CONCLUSION

This paper presents a literature-informed design for an AI-based smart irrigation system that combines soil-moisture sensing, weather context, crop profiles, decision logic, and bounded pump or valve control. Its main design principle is to separate AI prediction from safety-critical actuation: machine learning can assist with estimating irrigation need, while local interlocks, sensor validation, maximum runtime, and manual override protect reliable operation. Recent reviews report growing interest in machine learning and IoT for irrigation decisions, but also identify data quality, generalization, and evaluation as continuing challenges [2]–[7]. The proposed system should therefore be evaluated against a transparent baseline using measured water volume, soil-moisture behaviour, crop response, reliability, prediction quality, and total cost. No performance improvement is claimed until a documented field study supports it.

REFERENCES

[1] Food and Agriculture Organization of the United Nations (FAO), “Water Conservation and Smart Irrigation,” FAO in India, 8 October 2026. Available: <https://www.fao.org/india/home/water-conservation-and-smart-irrigation/en>

[2] R. Togneri, R. Prati, H. Nagano, and C. Kamienski, “Data-driven water need estimation for IoT-based smart irrigation: A survey,” *Expert Systems with Applications*, vol. 225, art. 120194, 2023, doi: 10.1016/j.eswa.2023.120194.

[3] Y. Abiadi et al., “The application of machine learning techniques for smart irrigation systems: A systematic literature review,” *Smart Agricultural Technology*, vol. 7, art. 100425, 2024, doi: 10.1016/j.atech.2024.100425.

[4] L. Umutoni and V. Samadi, “Application of machine learning approaches in supporting irrigation decision making: A review,” *Agricultural Water Management*, vol. 294, art. 108710, 2024, doi: 10.1016/j.agwat.2024.108710.

[5] H. Hammouch et al., “A Systematic Review and Meta-Analysis of Intelligent Irrigation Systems,” *IEEE Access*, vol. 12, pp. 128285–128304, 2024, doi: 10.1109/ACCESS.2024.3421322.

[6] M. Del-Coco, M. Leo, and P. Carcagni, “Machine Learning for Smart Irrigation in Agriculture: How Far along Are We?,” *Information*, vol. 15, no. 6, art. 306, 2024, doi: 10.3390/info15060306.

[7] A. E. G. and G. J. Bala, “IoT and ML-based automatic irrigation system for smart agriculture system,” *Agronomy Journal*, vol. 116, no. 3, pp. 1187–1203, 2024, doi: 10.1002/agj2.21344.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00099