RFarm

(Remote Farming Application)

Dr G Dharani Devi
Department of CSE
Rajalakshmi Engineering College, Chennai,India
dharanidevi.g@rajalakshmi.edu.in

Tharun Raj
Department of CSE
Rajalakshmi Engineering College, Chennai,India
230701362@rajalakshmi.edu.in

Sreya G
Department of CSE
Rajalakshmi Engineering College, Chennai,India
230701334@rajalakshmi.edu.in

Veronica Regina Paul
Department of CSE
Rajalakshmi Engineering College, Chennai,India
230701377@rajalakshmi.edu.in

ABSTRACT:

In many rural areas, farming still relies on manual labor and traditional irrigation methods. This becomes difficult when farmers face irregular rainfall, water scarcity, or cannot be physically present to manage irrigation. In this project, we developed a smart irrigation system using a Wi-Fi-enabled microcontroller (ESP8266), connected to soil moisture, rain, and ultrasonic sensors to monitor field conditions. Data is sent to a web dashboard built with the MERN stack (MongoDB, Express.js, React.js, Node.js), allowing users to control irrigation remotely.

A Rain Sensor along with a Weather API was used to pause irrigation if rain is detected or predicted. We followed a Design Thinking approach by collecting feedback from farmers and adjusting the system to be simple, cost-effective, and user-friendly. Testing showed a clear reduction in water use and manual effort, making the solution suitable for small and medium-scale farms.

Also the sensors might get errorful so there exists a Isolation Forest ML Model to detect this sensor bugs.

KEYWORDS :

IoT, Smart Irrigation, ESP8266, MERN Stack, Remote Monitoring, Weather API, ML Model ,Design Thinking

I.INTRODUCTION

In many developing countries, farming is still a major part of daily life, especially in rural areas. But a lot of farmers still use old methods for watering crops — either fixed schedules or manual checking. This can be time-consuming and often wastes water, especially now that rainfall has become unpredictable and labor is harder to find. So there's a clear need for smarter solutions that can help manage irrigation better.

That's what this project is about. We created a smart irrigation system using IoT to help monitor and control watering automatically. The system is built around an ESP8266 (NodeMCU) microcontroller, which collects real-time data from sensors and sends it to a web dashboard. We used the MERN stack — MongoDB, Express.js, React.js, and Node.js — to create this dashboard. It lets users check moisture levels, rainfall, and water level, and turn the pump on or off, all from their phone or computer. Also there is an emergency notification system powered by python ML Model to detect sensor failures if any.

To make sure this system is actually useful to farmers and not just a tech project, we used the Design Thinking process. We spoke to real users, listened to their challenges, and made changes based on their feedback. This helped us build something simple, practical, and easy to use.

II.LITERATURE REVIEW

A lot of recent research has focused on how IoT can improve farming. Using sensors to track real-time conditions makes it easier to save water and avoid unnecessary irrigation. One study by Gupta et al. [1] showed that using soil moisture and weather data reduced water usage by 35%. Another by Patel and

Sinha [2] used rainfall sensors to stop irrigation when it rains, which helped reduce overwatering.

In terms of software, MERN stack (especially React and Node.js) has been used to build dashboards that are easy to use, even for people without much tech experience. Jain et al. [3] highlighted how user-friendly and responsive these systems can be when built properly.

On the hardware side, the ESP8266 is a popular choice because it's affordable, works on Wi-Fi, and supports real-time control. Kumar and Rao [4] showed that ESP8266-based systems were reliable and suitable for smart farming setups.

But many of the systems we found either didn't include weather data or had complicated dashboards. Also, very few projects actually used a design approach focused on the real needs of farmers. That's why we made sure to use Design Thinking in our process — so that everything we built made sense to the people using it.

III.PROPOSED SYSTEM

The system we built is designed to solve common irrigation problems faced by farmers, especially in rural areas where fields are far apart and hard to monitor regularly. Our idea was to create a setup that lets farmers check field conditions in real-time and water their crops without needing to be physically present.

We divided the system into three main parts:

<u>Sensing Layer</u> – This includes soil moisture sensors, a rain sensor, and an ultrasonic sensor that checks water levels. These sensors collect live data from the field.

<u>Control & Communication Layer</u> – This is handled by the ESP8266 NodeMCU, which reads the sensor data and sends it to the cloud using Wi-Fi.

<u>Application Layer</u> – This is the dashboard we built using the MERN stack. It shows all the sensor data and also allows users to control the pump, check the weather forecast, and get crop suggestions.

We also added a weather API to check if it's going to rain — if so, the system holds off on watering. Farmers can open the dashboard on their phone or computer from anywhere, giving them full control. The whole setup is modular and budget-friendly, which makes it perfect for both small farms and bigger lands.

IV.SYSTEM ARCHITECTURE AND IMPLEMENTATION

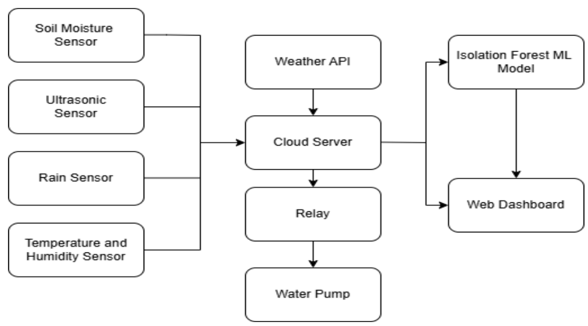


Fig no.1 Flowchart of working

SYSTEM ARCHITECTURE:

The system architecture, as illustrated in Fig. 1, consists of sensor nodes connected to the ESP8266 microcontroller, which gathers field data and sends it to the cloud database via WiFi. This data **is** visualized through a dashboard built using the MERN stack.

IMPLEMENTATION:

A. Hardware Components

Here's what we used to build the system:

<u>ESP8266 (NodeMCU)</u> – The main controller that connects to sensors and sends data over Wi-Fi.

<u>Soil Moisture Sensor (YL-69)</u> – Checks how dry or wet the soil is.

<u>Ultrasonic Sensor (HC-SR04)</u> – Used to measure water level, especially useful for paddy fields.

<u>Rain Sensor (YL-83)</u> – Detects when it's raining.

<u>Temperature and Humidity Sensor (DHT11)</u> – Detects temperature and humidity of the field area.

<u>Relay Module</u> – Turns the irrigation motor on and off based on sensor readings or manual input.

B. Software Stack

On the software side, we used:

React.js – For building the dashboard interface that shows live sensor values.

Node.js + Express.js – For handling requests from the frontend and managing logic.

MongoDB – Stores all the data from the sensors, crop suggestions, and pump activity logs.

Embedded C (Arduino IDE) – To program the ESP8266 and control how it responds to sensor data.

Python (Isolation Forest Model) – For determining the anomalies in sensor readings. Takes temperature , humidity and sensor data as input.

OpenWeatherMap API – For getting live weather updates so the system knows when to skip wateri

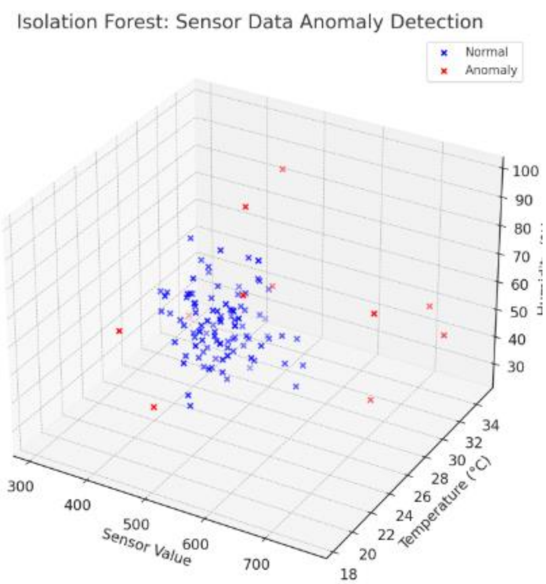


Fig no.2 Isolation Forest model Statistics

C. How It Works (Flow)

- Sensors collect data from the field (like soil moisture, rainfall, water level).
- ESP8266 reads this data and sends it to the backend server.
- The backend provides sensor data to the ML Model for anomaly prediction.
- The backend stores it in MongoDB and updates the frontend dashboard.
- If there's an anomaly , it sends an email to the farmer .The mail tells the level of malfunction, categorized as slight malfunction , moderate malfunctioning and immediate intervention needed.
- If it's raining or rain is expected, the system avoids turning on the motor.
- Farmers can still open the dashboard and control everything manually if needed.

V. RESULTS AND DISCUSSION

We tested our system in a simulated setup that represented a small farm area with two zones — one for paddy and another for vegetables. We used actual soil, water containers, and plants, and the ESP8266 microcontroller was connected to the soil moisture, rain, and water level sensors. A relay module was used to switch the motor on and off, and the entire setup ran on a solar battery so we could check how well it would work in areas with no electricity.

A. Performance Metrics

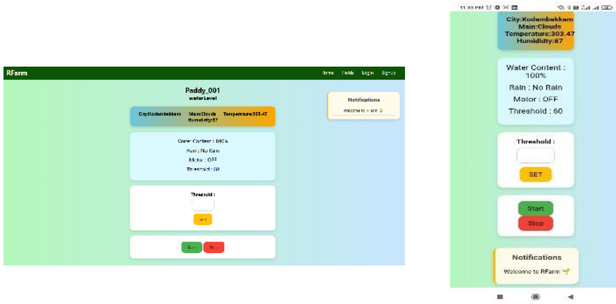
PARAMETER	VALUE
Soil Moisture Sensor Accuracy	±3%
Water Level Detection Accuracy	±5%
Avg. Data Transmission Delay	~2.8 seconds
System Uptime	>98%
Manual Effort Reduction	Around 80%
Estimated Water Saved	0% to 90%

B. Dashboard Utility

The dashboard we built using React worked well during testing. It let us:

- Check live readings from all sensors
- View weather forecast data from the API
- Manually turn the motor on or off
- Switch between the two farm zones
- View graphs showing moisture levels over time

- The interface was simple and responsive, and it worked well on both desktop and mobile browsers.



(a) (b)

Fig 3. Dashboard of fieds

(a) Desktop view
(b) Mobilr view

C. User Feedback

We showed the system to farmers and staff from a local agri-college. They said the dashboard was easy to understand, even for people with little tech knowledge. They liked the auto pause feature during rainfall and suggested adding features like:

- SMS alerts when irrigation is triggered
- Offline mode in case of poor internet
- Voice control for older users or those unfamiliar with reading dashboards
- Overall, they found the concept practical and were interested in using a version of this in real farms.

VI.APPLICATIONS AND ADVANTAGES

A. Where It Can Be Used

- Small Farms – Farmers can check soil moisture and water crops without being physically present every day.
- Paddy Fields – Water level detection helps avoid flooding or under-watering.
- College/Institution Farms – Multiple plots can be managed easily from a single dashboard.

B. Why It's Useful

- Saves Water – Since irrigation only happens when needed, there's no waste.
- Remote Control – Farmers can check and control irrigation from anywhere using a phone.
- Expandable Design – Sensors can be added or removed based on crop type or field size.
- Built with Feedback – We talked to actual farmers during the project, so their needs were considered.
- Affordable – We kept the hardware simple and cheap so that it's practical for rural use.

VII.CONCLUSION AND FUTURE SCOPE

In this project, we built and tested a smart irrigation system that uses IoT to help farmers monitor their fields and control watering more efficiently. We used the ESP8266 microcontroller along with sensors to track soil moisture, rainfall, and water levels. The system sends this data to a web dashboard, which we built using the MERN stack. From this dashboard, users can see live field data and manually control irrigation from their phone or computer.Throughout the project, we followed the Design Thinking process to make sure the system was user-friendly and matched real farmer needs. During testing, the system worked well, helping reduce water usage and the amount of manual effort needed to manage irrigation. Users especially appreciated how simple the dashboard was and the ability to access it in local language.

FUTURE SCOPE:

We believe this system can still be improved with a few upgrades:

- Adding more sensors (like pH, NPK, and temperature) to check soil health more completely
- Creating a mobile app that can work even without an internet connection
- Using AI to suggest when and how much to water, based on patterns in the data
- Sending SMS or voice alerts for important events like low moisture or motor status

- Partnering with government schemes or colleges to try this on larger farms
- Using solar panels more efficiently to save energy and make the system greener

If these improvements are added, this system can go beyond just irrigation — it could become a full farm management tool for farmers everywhere.

VIII. REFERENCES

[1] A. Gupta et al., “IoT Applications in Smart Agriculture,” Journal of AgriTech Research, vol. 12, pp. 89–96, 2022.

[2] R. Patel and K. Sinha, “Automated Irrigation with Soil Moisture Sensors,” IEEE Conf. on Smart Farming, pp. 102–106, 2021.

[3] S. Jain et al., “MERN Stack for Real-Time Agricultural Systems,” IJCA, vol. 180, no. 5, pp. 21–25, 2023.

[4] V. Kumar and A. Rao, “IoT-Based Smart Farming using ESP8266,” International Journal of Embedded Systems, vol. 17, no. 3, 2021.

[5] World Bank Innovation Lab, “Design Thinking in Agricultural Technology Development,” 2020.

[6] K. Elissa, “Cloud-Based Decision Making in Precision Irrigation,” IEEE Cloud Computing, vol. 9, pp. 47–55, 2023.