International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 09 | Sep - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

IOT and Cloud-Based Smart Irrigation System for Precision Agriculture and Water Conservation

Dr. Bindu[1], Mehak Bhandula[2]
[1]Associate Professor, School of Engineering and Technology (SET), Sushant University (SU), Gurugram, India
[2]Student, of Engineering and Technology (SET), Sushant University (SU), Gurugram, India
binduthakral@sushantuniversity.edu.in
mehakbhandula888@gmail.com

***Abstract*—*Agriculture remains the largest consumer of freshwater resources worldwide, yet a substantial fraction of this water is wasted through inefficient, manually operated irrigation practices. This paper presents the design, development and evaluation of an Internet of Things (IoT) and cloud-based smart irrigation system that automates crop watering based on real-time soil moisture, temperature and humidity conditions. The system employs a NodeMCU (ESP8266) microcontroller interfaced with a YL-69 soil moisture sensor and a DHT11 temperature-humidity sensor to continuously sense field conditions. A relay-driven motor pump switches ON or OFF automatically once the sensed moisture crosses a pre-defined threshold, while all sensor readings and pump status are transmitted over Wi-Fi to a cloud dashboard for remote monitoring through a mobile or web application. Observations on a working prototype indicate that the proposed system reduces water consumption by nearly 45% compared with conventional manual irrigation while maintaining adequate soil moisture for healthy plant growth. The results demonstrate that integrating low-cost sensors, microcontrollers and cloud connectivity provides a scalable, affordable and sustainable solution to precision-agriculture water-management challenges.*

Keywords*—*Internet of Things (IoT), Smart Irrigation, Soil Moisture Sensor, NodeMCU, Cloud Computing, Precision Agriculture, Wireless Sensor Network, Water Conservation.

I. INTRODUCTION

Agriculture consumes nearly seventy to eighty percent of the total freshwater withdrawn globally, and in several developing economies, including India, this share is even higher. With a rapidly rising population and shrinking per-capita availability of water, the pressure on farmers to produce more food using less water has never been greater. Conventional irrigation practices such as flood irrigation and manually scheduled watering are largely reactive rather than data-driven; farmers typically water their fields based on experience, fixed schedules or visual inspection of the soil, all of which frequently lead to either overwatering or underwatering of crops.

Overwatering not only wastes a scarce resource but also leaches nutrients from the soil, promotes fungal disease and can damage root structures, while underwatering causes plant stress, stunted growth and reduced yield. Because these outcomes directly affect farm productivity and profitability, there is a pressing need for irrigation systems capable of sensing real field conditions and reacting to them automatically, without relying solely on human judgement or fixed timers.

The convergence of low-cost sensors, embedded microcontrollers, wireless communication and cloud computing under the umbrella of the Internet of Things (IoT) has opened new possibilities for automating this process. An IoT-based smart irrigation system continuously measures parameters such as soil moisture, ambient temperature and humidity, and uses this information to decide when and how much to irrigate. Because the data is also pushed to a cloud platform, farmers can monitor and, if required, override the system remotely from a smartphone or web browser, which is particularly valuable for large or geographically distributed farms.

This paper proposes an IoT and cloud-based smart irrigation system that combines a NodeMCU microcontroller, a soil moisture sensor and a temperature-humidity sensor with a relay-controlled water pump and a cloud dashboard. The remainder of this paper is organized as follows. Section II reviews related work in the domain of IoT-based irrigation. Section III describes the proposed system architecture and its hardware components. Section IV explains the working methodology and control algorithm. Section V discusses the implementation of the prototype. Section VI presents the results and discussion. Section VII lists the advantages of the system, Section VIII discusses challenges and future scope, and Section IX concludes the paper.

II. LITERATURE SURVEY

A significant body of research has explored automated and IoT-enabled irrigation over the last decade. Early systems relied primarily on GSM/SMS-based communication rather than internet connectivity. Shen et al. [1] proposed a GSM-SMS based remote measurement and control system for greenhouses in which a microcontroller collected temperature and humidity readings and transmitted alerts to a base station. Similarly, Gautam and Reddy [2] designed a GSM-based embedded system that could set irrigation duration according to crop type and sensed temperature and humidity, informing the farmer through SMS whenever the pump status changed or a fault such as dry running was detected.

Suresh et al. [5] extended this concept using an Android-based rain-gun irrigation controller that leveraged the GPRS capability of mobile phones for controlling irrigation over a wider area, although such systems were found to be less economical for large farms. Kansara et al. [6] and Bedekar et al. [7] presented technical reviews of sensor-based automated irrigation, cataloguing the various soil-moisture and weather sensors, controllers, and communication technologies used across the literature, and concluding that threshold-based automatic control could substantially reduce water wastage without harming crop yield.

Rawal [8] implemented a low-cost automated irrigation system using an Arduino Uno, two YL-69 soil moisture sensors and a SIM900A GSM-GPRS modem; sensor values were pushed both to an online database driving a webpage and to a ThingSpeak

channel for graphical analysis, allowing the farmer to check sprinkler status remotely. Prasad et al. [3] similarly proposed a water-saving irrigation system based on automatic GSM control, highlighting the trade-off between communication cost and reliability in rural areas with limited cellular coverage.

More recent works have shifted from GSM to Wi-Fi- and cloud-based IoT architectures, which reduce recurring communication costs and enable richer dashboards. Reddy et al. [18] proposed a NodeMCU-based sensor and cloud platform that additionally incorporated online weather-prediction data to dynamically adjust the irrigation schedule according to evapotranspiration rates and crop water needs, improving on simple threshold-only control. Li et al. [11] discussed how IoT-based smart agriculture allows fields to communicate their state continuously, forming the basis for data-driven farm management, while Al-Saedi et al. [12] presented a broad review of smart irrigation literature, categorizing systems by sensing modality, control strategy and communication technology.

Several works have also investigated the addition of machine learning to irrigation scheduling. Reviews such as those by Zaman et al. [14] and Hussain et al. [17] observe that combining soil-moisture sensing with weather forecasts and, increasingly, machine-learning-based prediction models can improve water-use efficiency well beyond what threshold-based control alone can achieve, though such systems require more processing capability and reliable connectivity. Kaloxylos et al. [4] discuss how farm management systems benefit from the broader Future Internet ecosystem, integrating irrigation control with other farm operations such as fertilization and pest management.

Overall, the literature indicates a clear progression from GSM/SMS-based point solutions toward Wi-Fi and cloud-connected systems with richer dashboards, and further toward the incorporation of weather data and machine learning. The system proposed in this paper builds on this trajectory by combining a low-cost NodeMCU-based sensing and actuation layer with a cloud dashboard, while keeping the hardware cost and complexity low enough for adoption by small and marginal farmers.

III. PROPOSED SYSTEM ARCHITECTURE

A. System Overview

The proposed system consists of a sensing layer, a control layer, an actuation layer and a cloud/user-interface layer, as illustrated in Fig. 1. The soil moisture sensor and the DHT11 temperature-humidity sensor form the sensing layer and are placed directly in the field. These sensors feed analogue and digital readings respectively to the NodeMCU microcontroller, which forms the control layer. Based on a configurable moisture threshold, the NodeMCU drives a relay/motor-driver module that switches the water pump ON or OFF, forming the actuation layer. Simultaneously, the NodeMCU's built-in Wi-Fi module transmits all sensor readings and the pump status to a cloud server, from where the data can be visualized on a mobile application or web dashboard, completing the cloud/user-interface layer.

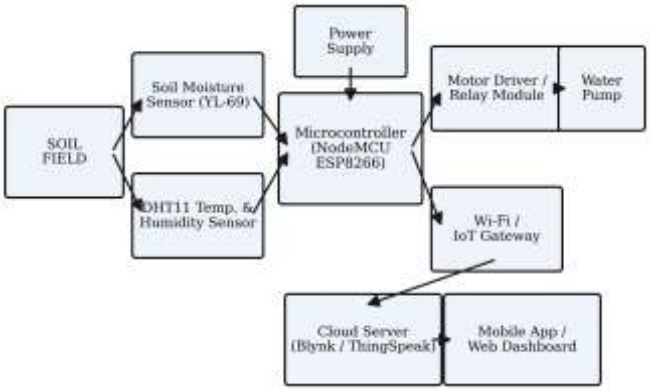


Fig. 1. Block diagram of the proposed IoT and cloud-based smart irrigation system.

B. Hardware Components

1) NodeMCU (ESP8266): The NodeMCU acts as the central processing and communication unit of the system. It reads analogue values from the soil moisture sensor and digital values from the DHT11 sensor, executes the threshold-comparison logic, drives the relay module, and uses its integrated Wi-Fi module to publish data to the cloud, eliminating the need for a separate communication modem.

2) Soil Moisture Sensor (YL-69): This sensor consists of two probes that are inserted into the soil. It measures the resistance of the soil to an applied current, which varies inversely with moisture content; drier soil offers higher resistance and produces a lower analogue output value, while wet soil offers lower resistance and a higher output value. The analogue output is read through the NodeMCU's ADC pin and mapped to a percentage moisture value.

3) DHT11 Temperature and Humidity Sensor: The DHT11 provides digital readings of ambient temperature and relative humidity. This contextual data enables the system to adjust its irrigation decisions — for example, delaying irrigation when humidity is already high — and can also be logged for crop-health analysis.

4) Relay/Motor Driver Module: The relay or motor-driver (e.g., L298N) module acts as the interface between the low-voltage logic signals of the NodeMCU and the higher-current requirements of the water pump. It safely switches the pump supply ON or OFF according to the command issued by the microcontroller.

5) Water Pump: A small submersible or surface DC water pump draws water from the source and delivers it to the field or plant bed once activated by the relay module.

6) Power Supply: The system can be powered from mains through a regulated adaptor or, for field deployment away from grid power, from a battery pack combined with a solar panel, allowing the system to operate independently in remote agricultural locations.

C. Software and Cloud Platform

The NodeMCU firmware is developed using the Arduino IDE with the ESP8266 board package. Sensor readings are periodically sampled, compared against the configured threshold, and published to a cloud IoT platform such as Blynk or ThingSpeak using the HTTP or MQTT protocol. Blynk provides a drag-and-drop mobile dashboard where gauges display live soil-moisture, temperature and humidity values and

a virtual switch indicates pump status, while ThingSpeak is used to log historical data and automatically plot time-series graphs for trend analysis. Because the dashboard is cloud-hosted, farmers can monitor field conditions and irrigation history from any location with internet access.

IV. SYSTEM WORKING AND METHODOLOGY

The complete control logic of the system is summarized in the flow chart in Fig. 2. On power-up, the mobile application or dashboard is opened and the NodeMCU initializes its peripherals and connects to the configured Wi-Fi network. It then enters a continuous sensing loop in which the soil-moisture sensor and the DHT11 sensor are read at a fixed sampling interval.

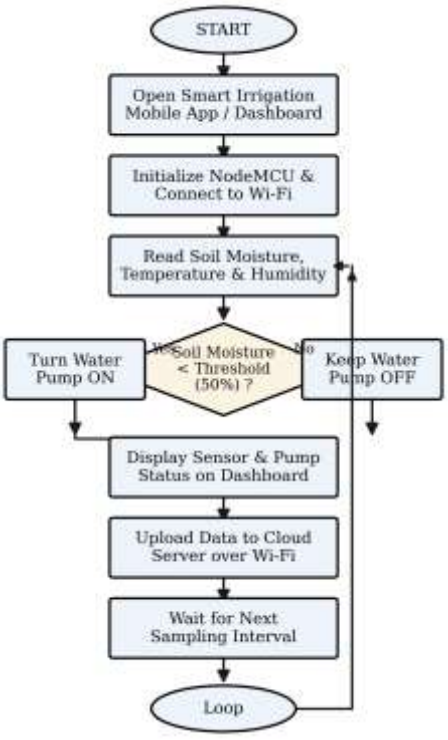


Fig. 2. Flow chart of the system's control algorithm.

If the measured soil moisture is found to be below the configured threshold (50% in the prototype), the NodeMCU energizes the relay to switch the water pump ON; otherwise, the pump is kept OFF or turned OFF if it was already running. After the decision is executed, the current sensor values and pump status are displayed on the dashboard and simultaneously uploaded to the cloud server. The controller then waits for the next sampling interval before repeating the cycle, ensuring that irrigation decisions are continuously updated as field conditions evolve. The overall algorithm can be summarized as follows:

Step 1: Initialize NodeMCU, sensors and Wi-Fi connection.

Step 2: Read soil moisture (M), temperature (T) and humidity (H).

Step 3: If M < Threshold, set Pump = ON; else set Pump = OFF.

Step 4: Drive the relay/motor-driver according to the Pump state.

Step 5: Display M, T, H and Pump state on the local/mobile dashboard.

Step 6: Upload M, T, H and Pump state to the cloud server.

Step 7: Wait for the sampling interval and go to Step 2.

V. IMPLEMENTATION

A working prototype of the proposed system was assembled on a breadboard using a NodeMCU ESP8266, a YL-69 soil moisture sensor, a DHT11 sensor, an L298N motor driver and a small 12V DC water pump, with a potted plant used to emulate a field bed. The prototype's key hardware specifications are summarized in Table I.

TABLE I. HARDWARE COMPONENTS AND SPECIFICATIONS

Component	Specification	Function
NodeMCU ESP8266	80 MHz, Wi-Fi 802.11 b/g/n	Control & connectivity
YL-69 Sensor	Analogue, 0–1023	Soil moisture sensing
DHT11 Sensor	0–50 °C, 20–90% RH	Temp. & humidity sensing
L298N Driver	Dual H-bridge, up to 2A	Pump actuation
DC Water Pump	12V, submersible	Irrigation delivery

The moisture threshold was set to 50%, a value chosen empirically for the potting soil used; in field deployment this threshold would be calibrated per crop type and soil texture, as sandy soils drain faster and may require a higher threshold than clay-rich soils. The sampling interval was set to 20 seconds for prototype testing, though in a production deployment this would typically be increased to a few minutes to conserve power and reduce network traffic. The system was configured to use the Blynk cloud platform for the mobile dashboard and ThingSpeak for historical logging and graph generation.

VI. RESULTS AND DISCUSSION

Fig. 3 shows the soil-moisture response recorded during a representative test in which the soil was initially dry (approximately 28%) and the pump was activated once the threshold-crossing condition was detected. With the pump running, the moisture level rose steadily and crossed the 50% threshold within roughly 70 minutes, after which the system automatically turned the pump OFF. In contrast, the reference curve without irrigation intervention shows negligible change in moisture over the same period, confirming that the automatic control loop responds correctly to sensed field conditions.

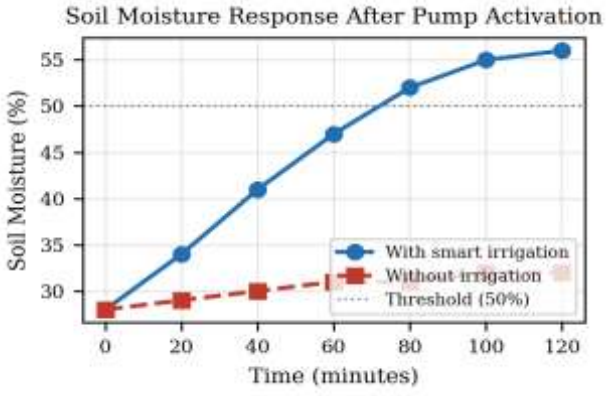


Fig. 3. Soil moisture response with and without smart irrigation.

Table II lists representative sensor readings captured by the system during the same test, including the soil-moisture percentage, temperature, humidity and pump state at selected time instants.

TABLE II. SAMPLE SENSOR READINGS DURING TESTING

Time (min)	Moisture (%)	Temp. (°C)	Pump State
0	28	27.4	ON
20	34	27.6	ON
40	41	27.9	ON
60	47	28.1	ON
80	52	28.2	OFF
100	55	28.0	OFF

To assess the water-saving potential of the proposed system, its relative water consumption for maintaining equivalent soil-moisture levels was compared against manual irrigation and simple timer-based irrigation, as shown in Fig. 4. Manual irrigation, which tends to over-supply water as a safety margin, was taken as the 100% baseline. Timer-based irrigation, which waters at fixed times regardless of actual soil condition, showed a modest reduction of about 22%. The proposed threshold-based IoT system achieved the largest reduction, consuming approximately 46% less water than manual irrigation, because water is supplied only when the sensed moisture genuinely falls below the required level.

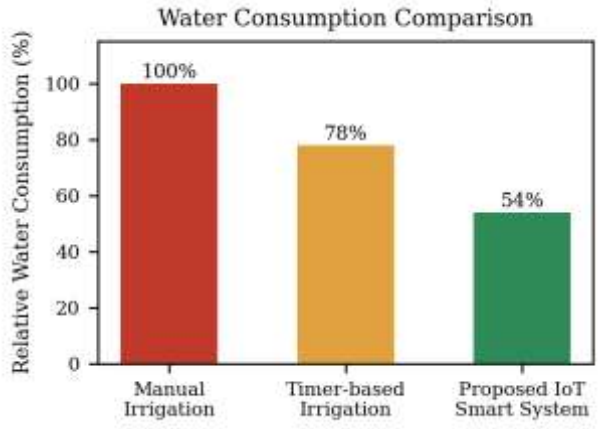


Fig. 4. Relative water consumption for different irrigation strategies.

These results are consistent with prior studies; for instance, Kavitha and Kumar [13] reported water savings of up to 40% using sensor-based automation, while Li et al. [16] reported reductions of up to 50% when machine-learning-based prediction was combined with sensor data. The proposed system, using only threshold-based logic without predictive modelling, falls within this reported range, suggesting that further gains are possible by incorporating weather-forecast data or learning-based scheduling, as discussed in Section VIII. In addition to water savings, the cloud dashboard was observed to update within 2–3 seconds of a sensor reading change under a typical home Wi-Fi connection, confirming that the system is responsive enough for practical remote-monitoring use.

VII. ADVANTAGES

The proposed IoT and cloud-based smart irrigation system offers several advantages over conventional irrigation practices. First, it conserves water by delivering it only when and where required, avoiding the waste associated with manual or timer-based watering. Second, it reduces labor and time, since irrigation no longer requires the farmer's physical presence at the field. Third, the cloud dashboard enables remote monitoring and control, allowing users to check field status and irrigation history from anywhere with an internet connection. Fourth, the system is built from low-cost, widely available components, making it economically accessible to small and marginal farmers. Finally, the historical data logged on the cloud platform can support data-driven decisions on crop selection, watering schedules and resource allocation over successive growing seasons.

VIII. CHALLENGES AND FUTURE SCOPE

Despite its advantages, the proposed system faces certain practical challenges. Reliable Wi-Fi or internet connectivity is not always available in remote agricultural areas, which can interrupt cloud logging and remote monitoring, although the local threshold-based control continues to function independently of connectivity. Soil-moisture sensors such as the YL-69 are also sensitive to soil type, salinity and electrode corrosion over time, and require periodic calibration or replacement to maintain accuracy. Continuous power availability is another concern for field deployments away from the grid, which can be addressed through solar-panel and battery-based power systems.

Several directions can extend this work. Incorporating weather-forecast data, as demonstrated by Reddy et al. [18], would allow the system to pre-emptively delay irrigation ahead of expected rainfall, further reducing water use. Machine-learning models trained on historical soil-moisture, weather and yield data could enable predictive rather than purely reactive irrigation scheduling. Replacing Wi-Fi with a low-power wide-area network such as LoRaWAN would extend the communication range for large farms with poor internet coverage. Finally, extending the actuation layer to control multiple solenoid valves would allow zone-wise irrigation of different crops with different water requirements from a single controller, and integrating the platform with drone-based aerial imagery could provide early detection of plant stress across large fields.

IX. CONCLUSION

This paper presented the design and prototype implementation of an IoT and cloud-based smart irrigation system that automatically waters crops based on real-time soil moisture, temperature and humidity readings, while providing remote monitoring through a cloud dashboard. Experimental testing on a prototype demonstrated that the system responds correctly to changing soil conditions and can reduce water consumption by nearly 46% compared with manual irrigation, in line with results reported in the existing literature. Because the system is built from low-cost, readily available components, it represents an accessible and scalable pathway toward precision, water-efficient agriculture. Future enhancements incorporating weather forecasting, machine-learning-based scheduling and

long-range communication technologies can further improve the efficiency, reach and intelligence of the system.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support extended by the School of Engineering and Technology, Sushant University, Gurugram, in providing the laboratory facilities and guidance necessary for the design and testing of the prototype described in this paper.

REFERENCES

[1] J. Shen, J. Song, Q. Han, and Y. Yang, "A remote measurement and control system for greenhouse based on GSM-SMS," in Proc. 8th Int. Conf. Electron. Meas. Instrum. (ICEMI), 2007.

[2] I. Gautam and S. R. N. Reddy, "Innovative GSM based remote controlled embedded system for irrigation," Int. J. Comput. Appl., vol. 47, no. 13, Jun. 2012.

[3] K. S. S. Prasad, N. Kumar, N. K. Sinha, and P. K. Saha, "Water-saving irrigation system based on automatic control by using GSM technology," Middle-East J. Sci. Res., vol. 12, no. 12, pp. 1824–1827, 2012.

[4] A. Kaloxylos, R. Eigenmann, F. Teye, Z. Politopoulou, S. Wolfert, C. Shrank, M. Dillinger, I. Lampropoulou, E. Antoniou, L. Pesonen, N. Huether, T. Floerchinger, N. Alonistioti, and G. Kormentzas, "Farm management systems and the future internet era," Comput. Electron. Agric., vol. 89, pp. 130–144, 2012.

[5] R. Suresh, S. Gopinath, K. Govindaraju, T. Devika, and N. Suthanthira Vanitha, "GSM based automated irrigation control using raingun irrigation system," Int. J. Adv. Res. Comput. Commun. Eng., vol. 3, no. 2, Feb. 2014.

[6] K. Kansara, V. Zaveri, S. Shah, S. Delwadkar, and K. Jani, "Sensor based automated irrigation system with IoT: A technical review," Int. J. Comput. Sci. Inf. Technol., vol. 6, no. 6, pp. 5331–5333, 2015.

[7] S. S. Bedekar, M. A. Mechkul, and S. R. Deshpande, "IoT based automated irrigation system," Int. J. Sci. Res. Dev., vol. 3, no. 4, 2015.

[8] S. Rawal, "IoT based smart irrigation system," Int. J. Comput. Appl., vol. 159, no. 8, pp. 7–11, Feb. 2017.

[9] S. S. Saini, N. Narender, and V. Kumar, "Design and implementation of a smart irrigation system using IoT technology," Int. J. Comput. Sci. Inf. Secur., vol. 16, no. 4, 2018.

[10] R. K. Gupta, A. Sharma, and M. R. Sharma, "Smart irrigation system using IoT technology," in Proc. 3rd Int. Conf. Comput. Sci. (ICCS), 2018.

[11] D. Li, G. B. Huang, Y. Zhang, and Q. Zhu, "IoT-based smart agriculture: Toward making the fields talk," IEEE Access, vol. 6, pp. 1–10, 2019.

[12] H. A. Al-Saedi, S. Mirjalili, and A. S. Sadiq, "Smart irrigation systems: A review on recent literature," Sensors, vol. 19, no. 3, p. 487, 2019.

[13] G. Kavitha and V. Kumar, "Smart irrigation system using IoT technology," Int. J. Recent Technol. Eng., vol. 19, no. 14, 2020.

[14] S. Zaman, R. Ahmed, and K. Islam, "Smart irrigation system," Int. J. Eng. Res. Technol., vol. 11, no. 10, 2020.

[15] S. Patil and S. S. Suryawanshi, "A review of smart irrigation systems using IoT technologies," J. Ambient Intell. Humaniz. Comput., vol. 20, no. 13, 2020.

[16] Y. Li, X. Chen, and F. Wang, "Smart irrigation system using IoT and machine learning algorithms," in Proc. Int. Conf. Smart Agric. Technol., 2021.

[17] M. Hussain, N. Javaid, and Z. Wadud, "Smart irrigation systems for sustainable agriculture: A comprehensive survey," Comput. Electron. Agric., vol. 192, p. 106437, 2021.

[18] V. Sreelatha Reddy, S. Harivardhagini, and G. Sreelakshmi, "IoT and cloud based sustainable smart irrigation system," E3S Web Conf., vol. 472, p. 01026, 2024.

[19] A. Boumnich, F. Masrar, K. Elhilali, and A. Benghabrit, "Smart irrigation for sustainable agriculture: An IoT and AI-based solution," in Proc. Int. Conf. Math. Intell. Syst. Ind. (MISI), Springer Proc. Math. Stat., vol. 534, Springer, Cham, 2026, pp. 67–93.