IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SMART IRRIGATION SYSTEM: A REAL-TIME SENSOR-BASED IOT SOLUTION FOR SUSTAINABLE PRECISION FARMING

[1]Sushant Navle, [2]Dr. Nita Patil, [3]Vrishab Shenvi, [4]Ganesh Suryawanshi, [5]Anjali Gupta

[1]Student, [2]Professor, [3] Student, [4]Student, [5]Student

Department of Computer Engineering,

K.C. College of Engineering, Thane, India

Abstract: The Smart Irrigation System is designed to make farming easier and more efficient by using modern IoT technology. It helps farmers provide the right amount of water to their crops by continuously monitoring soil moisture, temperature, humidity and rainfall conditions.

The system uses sensors to collect real-time data and an ESP32 microcontroller to automatically control irrigation based on the needs of the soil. It is connected to web application, allowing farmers to check field conditions, receive alerts via email and control the system remotely.

By reducing water wastage, saving time and improving crop health, this system provides a practical and affordable solution for smarter and more sustainable farming.

Index Terms - Smart Irrigation System, Internet of Things, IoT, Precision Farming, Automation, Sensors.

I. INTRODUCTION

Agriculture plays a vital role in the economy, but one of the biggest challenges faced by farmers today is efficient water management. Traditional irrigation methods often depend on manual effort or fixed schedules, which do not consider real-time conditions of the soil and environment. As a result, crops may receive either too much or too little water, leading to water wastage, increased costs and reduced productivity.

With the advancement of technology, especially in the field of the Internet of Things (IoT), it is now possible to make irrigation systems smarter and more efficient. IoT enables devices to collect real-time data from the environment and take decisions automatically without constant human intervention.

The Smart Irrigation System is developed to address these challenges by providing an automated and intelligent solution for water management in agriculture. The system uses sensors to monitor soil moisture, temperature, humidity and rainfall conditions. Based on this data, it determines when irrigation is required and controls the water supply accordingly.

In addition to automation, the system also offers remote monitoring and control through a mobile or web application. This allows farmers to stay updated about field conditions and manage irrigation from anywhere. The integration of cloud technology further helps in storing data, generating insights and improving decision-making over time.

Overall, this paper aims to reduce water wastage, minimize manual effort and improve crop health by applying modern technology to traditional farming practices. It represents a step towards precision agriculture and sustainable farming.

II. LITERATURE SURVEY

Morchid et al. [1] developed an IoT-based smart irrigation system using an ESP32 microcontroller integrated with sensors such as soil moisture, temperature, humidity, and water level. The system transmits data to a cloud platform using HTTP and controls irrigation based on predefined thresholds, enabling efficient monitoring and automation of irrigation processes.
Kunt et al. [2] proposed an advanced IoT and AI-based irrigation system that incorporates a Long Short-Term Memory (LSTM) model to predict irrigation needs based on historical data. The system uses ESP32 and multiple sensors along with a mobile application for monitoring and control, demonstrating the integration of predictive analytics in smart irrigation.
Zhang et al. [3] introduced a smart irrigation system using LoRa communication and edge computing techniques. Their system processes data locally using models like the Penman-Monteith equation, enabling efficient data handling and long-range communication capabilities.
Panchal et al. [5] developed a cloud-based irrigation system using a microcontroller and Thing Speak platform for data analysis and automation. The system enables automatic pump control based on soil moisture data, supporting real-time monitoring and decision-making.
Karthick et al. [6] proposed an automated irrigation system using Arduino and ESP8266 modules to send sensor data to the cloud and control the pump via relay modules. The system demonstrates effective integration of hardware and cloud technologies for irrigation management.
Kim et al. [7] introduced a distributed irrigation system using Wireless Sensor Networks (WSN) with Zigbee communication. The system uses a local base station to monitor and control irrigation, illustrating the use of wireless communication for agricultural automation.
Ghosh et al. [8] proposed a conceptual framework combining IoT, cloud computing, and AI for smart agriculture applications. The study emphasizes the role of predictive analytics in irrigation and crop management.
Rawal et al. [9] developed an IoT-based irrigation system using Arduino and the Blynk platform for monitoring and manual control. The system allows users to remotely manage irrigation through a mobile application, enhancing accessibility and control.
The Food and Agriculture Organization (FAO) [10] provided a comprehensive report on the impact of climate change on agriculture and emphasized the importance of smart irrigation solutions for sustainable water management and food security.

III. RESEARCH METHODOLOGY

The methodology of this paper focuses on using real-time data to make irrigation smarter and more efficient. Sensors continuously monitor factors like soil moisture, temperature, humidity and rainfall, and the collected data is processed to ensure it is accurate and reliable. Based on this information, the system automatically decides whether the water pump should be turned ON or OFF. It also includes safety checks, such as detecting rain, to avoid unnecessary watering. This approach helps save water, reduces manual effort and makes the overall irrigation process more efficient.

3.1 Working Flow

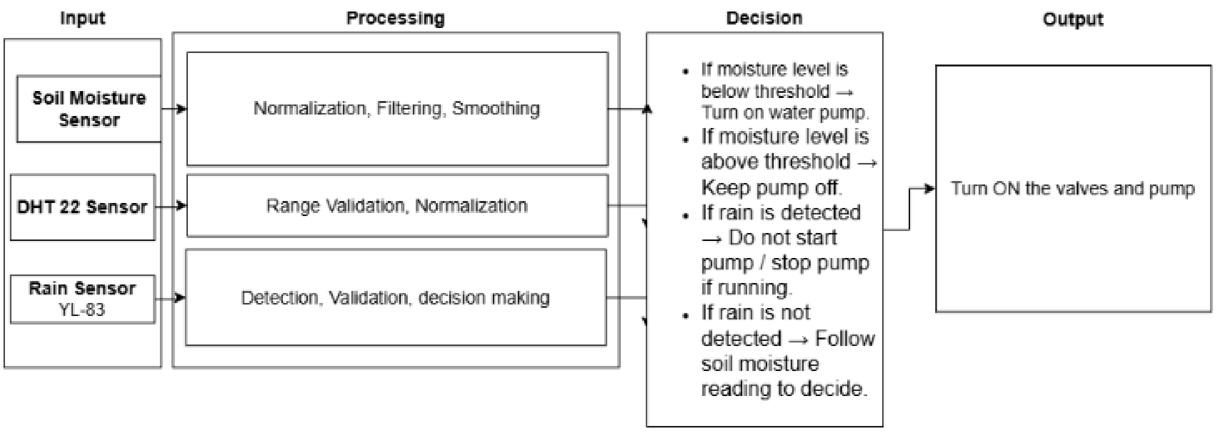


Fig 3.1.1: Methodology Diagram

1. Input Acquisition:

ESP32 Microcontroller: The ESP32 acts as the central control unit of the system. It reads sensor data, processes it, makes decisions, and sends data to the cloud using Wi-Fi for remote monitoring.

Soil Moisture Sensor (YL-69): This sensor measures soil moisture by detecting electrical resistance. It provides analog output to determine whether the soil is wet or dry.

Temperature and Humidity Sensor (DHT22): The DHT22 measures temperature (°C) and humidity (%) with high accuracy. It provides digital output to help analyze environmental conditions.

Rain Sensor (YL-83): The rain sensor detects rainfall and provides both analog (intensity) and digital (rain/no rain) outputs. It helps stop irrigation during rain.

Water Level Sensor (Robodo-SEN18): This sensor measures the water level in the tank using analog output. It ensures sufficient water is available and prevents pump damage.

Water Flow Sensor: The water flow sensor measures the flow rate of water in liters per minute (L/min). It helps monitor water usage and detect leaks or blockages.

2. Processing:

The data preprocessing in the Smart Irrigation System begins with the ESP32 microcontroller fetching raw input data from various sensors. The soil moisture sensor, water level sensor, and rain sensor primarily provide analog input values (integer type, range 0–4095 from the ESP32), while the rain sensor may also provide digital input (boolean type: 0 or 1 for no rain/rain detected). Temperature and humidity sensors (DHT22) provide digital numerical data in float format (temperature in °C and humidity in %). The analog integer values are then converted into meaningful units using scaling and mapping techniques such as:

Soil Moisture (%) = (1 − (Raw Value / 4095)) × 100, Water Level (%) = (Raw Value / 4095) × 100 and Rain Intensity (%) = (Raw Value / 4095) × 100 or Rain Status = 1/0.

During preprocessing, noise is reduced using smoothing techniques like the moving average: Smoothed Value = $(x_1 + x_2 + \ldots + x_n) / n$, which helps eliminate sudden spikes and fluctuations. Validation checks are also applied to ensure all values fall within acceptable ranges, such as 0–100% for moisture and humidity, rejecting any abnormal readings. After conversion and filtering, the final processed data is structured into a standardized format such as JSON before being sent to the cloud (Firebase). At this stage, all data types are well-defined: soil moisture and water level are float percentages, temperature is float (°C), humidity is float (%), and rain status is boolean or integer (0/1), ensuring accurate real-time monitoring and reliable system performance.

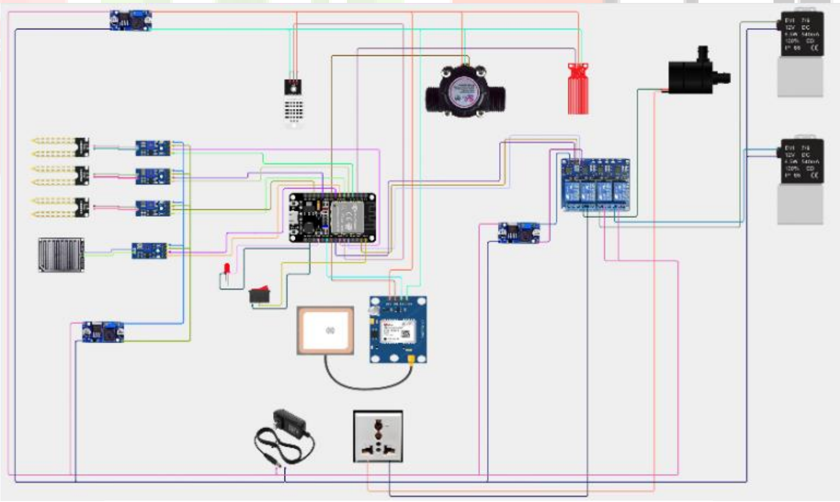
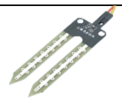


Fig 3.1.2: Circuit Diagram

TABLE I. COMPONENT DETAILS:

Sensor Images	Sensor Description
 Fig 3.1.3: Esp 32 Microcontroller	A powerful microcontroller board with built-in Wi-Fi and Bluetooth, commonly used for IoT and smart device projects. It can read sensors, control devices, and connect to the internet easily.
 Fig 3.1.4: Soil Moisture Sensor	This sensor measures the moisture level in soil by detecting water content. It is commonly used in smart irrigation systems to know when plants need watering.
 Fig 3.1.5: DHT22 Sensor	Measures temperature and humidity in the environment for basic monitoring applications.
 Fig 3.1.6: GPS Module	This module determines the exact location (latitude and longitude) using satellite signals. It is commonly used in navigation systems, tracking devices, and location-based applications.

3. Decision:

The Smart Irrigation System makes decisions by continuously comparing real-time sensor values with predefined threshold values. The primary comparison is between current soil moisture (%) and the threshold soil moisture (%) set for a specific crop and location. The basic decision condition is: if (Soil Moisture < Threshold Moisture) → Irrigation ON, otherwise Irrigation OFF. Similarly, additional conditions are checked such as Rain Status = 1 → Irrigation OFF and Water Level < Minimum Threshold → Irrigation OFF (safety condition). Temperature and humidity are also considered to adjust irrigation needs indirectly (e.g., high temperature → faster evaporation → may require higher moisture threshold).

The threshold values are dynamically determined based on the selected crop and environmental conditions. For example, crops like rice may require 70–80% soil moisture, while crops like wheat may need around 50–60%, and vegetables may vary between 60–75%. These values are stored in the system (or cloud database) and are automatically updated when the user selects a crop for a specific sector. Additionally, location-based factors such as temperature and humidity can slightly modify these thresholds (e.g., hotter regions may increase the required moisture threshold). Thus, the system uses a dynamic threshold = crop-based value ± environmental adjustment.

The final decision is taken using a combination of all conditions: if (Soil Moisture < Threshold AND Rain = 0 AND Water Level is sufficient) → Pump ON & Valve OPEN, else Pump OFF & Valve CLOSED. This logical decision-making ensures that irrigation happens only when necessary, prevents overwatering during rainfall, and avoids system damage due to low water availability. By continuously evaluating these parameters in real time, the system achieves intelligent, precise, and automated irrigation control.

4. Output:

The output of the Smart Irrigation System is presented through both physical actions and a digital dashboard interface. On the hardware side, the system controls the water pump and irrigation valves, turning them ON or OFF based on the decision logic. For example, when irrigation is required, the pump is activated and the corresponding sector valves are opened, and when conditions are satisfied (sufficient moisture or rain detected), the pump is turned OFF and valves are closed. The system also reflects safety states such as pump OFF during low water levels or rainfall conditions.

On the digital side, the output is displayed on the dashboard in the form of real-time sensor values and system status. This includes soil moisture values for Sector 1, Sector 2, and Sector 3, their average values, temperature (°C), humidity (%), rain status or intensity, water level (%), and weather API data. It also shows the selected crop for each sector along with its threshold values (required soil moisture, temperature, and humidity). Additionally, the dashboard displays system status indicators such as pump status (ON/OFF), valve status for each zone, and connectivity status.

The output also includes user interaction features, where users can manually control the pump and valves, select crops for each sector, and set or view irrigation schedules. Alerts or notifications may also be shown for conditions like low water level or rainfall detection. Overall, the system output provides a complete view of both current field conditions and system actions, making it easy for users to monitor, control, and manage irrigation efficiently.

In the backend, all system data is stored and managed using Firebase, which acts as a real-time database and cloud storage solution. It maintains both current and historical sensor data for all sensors, including soil moisture, temperature, humidity, rain, and water level. The database also stores the relay status (pump and valve ON/OFF states) and periodic health ping information to track the connectivity and proper functioning of the ESP32 device. Additionally, it contains detailed sector-wise information, including the selected crop for each sector and the dynamically updated threshold values for parameters like soil moisture, temperature, and humidity. Along with this, the system keeps logs of all activities and events, such as irrigation actions, sensor updates, and system alerts, ensuring traceability, debugging support, and better system management.

Working Algorithm

Algorithm: Smart Irrigation Control System

Input: Soil Moisture, Temperature, Humidity, Rainfall, Water Level, Crop Threshold Values

Output: Pump Status, Valve Status, Cloud Updates, Alerts

Step 1: Initialize sensors, ESP32 microcontroller, Wi-Fi connection, and cloud database.

Step 2: Continuously collect data from soil moisture, temperature, humidity, rainfall, and water level sensors.

Step 3: Process and validate the collected sensor data.

Step 4: Compare the sensor readings with predefined crop-specific threshold values.

Step 5: Check safety conditions such as rainfall detection and low water level.

Step 6: If soil moisture is below the threshold and no rainfall is detected, activate the water pump and open the corresponding irrigation valve.

Step 7: If the required soil moisture level is achieved or rainfall is detected, stop irrigation by turning OFF the pump and closing the valve.

Step 8: Update sensor readings, irrigation status, and alerts to the Firebase cloud database.

Step 9: Display real-time information on the web dashboard for monitoring and control.

Step 10: Repeat the process continuously for efficient and automated irrigation management.

END

Description:

The algorithm initializes system thresholds and continuously monitors environmental conditions using sensors. It processes and validates the data, applies safety checks like rain detection and low water level, and automatically controls irrigation based on soil moisture. It also ensures data synchronization with the cloud or stores it locally during offline operation for reliable performance.

The ESP32 processes the sensor data and controls the irrigation system using a relay module. The relay module is connected to a water pump and solenoid valves, which regulate the flow of water to different zones. Based on the sensor readings, the ESP32 sends signals to the relay to automatically turn the pump and valves ON or OFF.

Additionally, the system includes a GPS module for location tracking and supports cloud connectivity for real-time monitoring. All components are powered through a regulated power supply, ensuring stable operation. This integrated setup enables automated, efficient and intelligent irrigation with minimal human intervention.

3.2 Comparison of the Models

To evaluate the effectiveness of the proposed Smart Irrigation System, a comparison was made with traditional irrigation methods. Traditional irrigation relies on manual observation and fixed watering schedules, which often result in over-irrigation or under-irrigation. In contrast, the proposed system uses real-time sensor data and automated decision-making to provide precise irrigation based on crop requirements and environmental conditions.

The Smart Irrigation System continuously monitors soil moisture, temperature, humidity, rainfall, and water level through IoT sensors. Based on predefined threshold values, the system automatically controls the water pump and irrigation valves. This approach minimizes water wastage, reduces manual effort, and improves irrigation efficiency.
Furthermore, the proposed system provides cloud connectivity and a web-based dashboard for remote monitoring and control, features that are not available in conventional irrigation methods. The comparison indicates that the proposed system offers better water management, improved crop productivity, and enhanced operational efficiency, making it a more sustainable solution for modern agriculture.

Parameter	**Traditional Irrigation**	**Proposed Smart Irrigation System**
Irrigation Control	Manual	Automatic
Water Usage	High	Optimized
Monitoring	Physical Inspection	Real-Time Remote Monitoring
Decision Making	Fixed Schedule	Sensor-Based
Labor Requirement	High	Low
Water Conservation	Limited	High
Crop Productivity	Moderate	Improved
Cloud Connectivity	Not Available	Available

Table 3.2 Comparison of Traditional and Smart Irrigation Systems

IV. RESULTS AND DISCUSSION

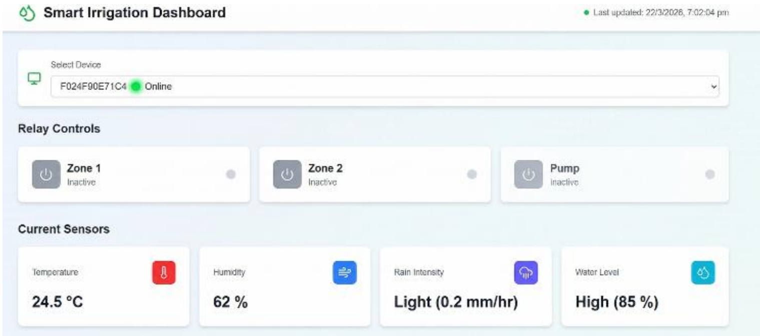
4.1 Hardware Model

A prototype model of the Smart Irrigation System was developed to validate the proposed approach. The setup consists of a water tank, PVC pipeline network, soil moisture sensors, rain sensor, relay modules, water pump, and an ESP32 microcontroller. The field is divided into multiple irrigation zones where sensors continuously monitor environmental conditions.
The ESP32 processes sensor data and automatically controls the pump and valves based on soil moisture levels and rainfall conditions. Water is supplied only to the zones requiring irrigation, thereby reducing water wastage and improving irrigation efficiency. The successful operation of the prototype demonstrates the feasibility of implementing real-time automated irrigation in agricultural fields.

Fig. 4.1: Prototype Model

4.2 Software Interface

A web-based dashboard was developed to provide real-time monitoring and control of the irrigation system. The dashboard displays sensor readings such as soil moisture, temperature, humidity, rainfall



status, and water level. It also shows the operational status of pumps and valves, enabling users to monitor field conditions remotely.

Fig. 4.2: Web-Based Dashboard

One of the key features of the dashboard is crop-based configuration. Users can select crops for different irrigation zones, and the system automatically applies suitable threshold values for soil moisture, temperature, and humidity. This enables precise irrigation according to crop requirements and improves water management efficiency.

4.3 Analytics and Control Features

The system also includes an analytics dashboard that provides historical sensor data and irrigation statistics. Users can analyze environmental conditions, monitor water usage, and evaluate irrigation performance over time.

In addition to monitoring, the dashboard supports manual control of pumps and valves as well as irrigation scheduling. Users can create predefined watering schedules and receive real-time system updates. These features enhance operational flexibility and provide a user-friendly interface for efficient irrigation management.

The experimental results indicate that the proposed Smart Irrigation System effectively automates irrigation decisions, reduces manual effort, conserves water resources, and supports sustainable precision farming.

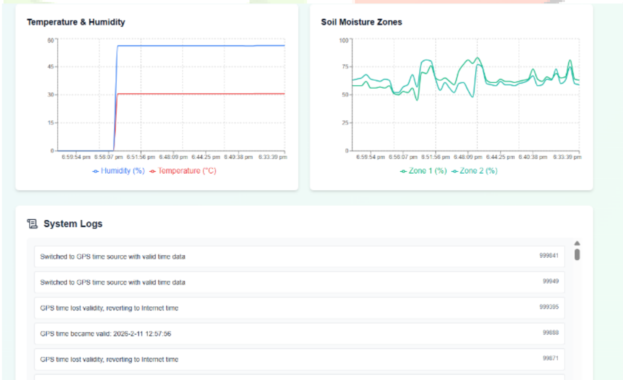


Fig. 4.3: Analytics Dashboard

V. CONCLUSION

The Smart Irrigation System demonstrates the effective use of Internet of Things (IoT) technology in modern agriculture by providing an automated and intelligent irrigation solution. The system continuously monitors important environmental parameters such as soil moisture, temperature, humidity, rainfall, and water level using various sensors integrated with the ESP32 microcontroller.

Based on real-time sensor data and crop-specific threshold values, the system automatically controls irrigation operations, ensuring that crops receive the required amount of water while minimizing wastage. The integration of Firebase cloud services and a web-based dashboard enables remote monitoring, data storage, and user control, making the system more efficient and accessible.

The developed prototype successfully validated the proposed approach by reducing manual intervention, improving water utilization, and supporting precision farming practices. The system also provides flexibility through crop-based configurations, scheduling features, and real-time alerts.

Overall, the Smart Irrigation System offers a cost-effective, scalable, and sustainable solution for agricultural water management. It contributes to increased crop productivity, conservation of water resources, and the advancement of smart farming technologies, making it suitable for future agricultural applications.

VI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to [1]Dr. Nita Patil, Department of Computer Engineering, K.C. College of Engineering and Management Studies and Research, Thane, for her valuable guidance, continuous support, and encouragement throughout the development of this project and research work.

The authors also extend their thanks to the Head of the Department, faculty members, and staff of the Computer Engineering Department for providing the necessary facilities and technical assistance required for the successful completion of this work.

Finally, we are grateful to our institution, K.C. College of Engineering and Management Studies and Research, Thane, for providing the resources and learning environment that contributed significantly to this research.

VII. REFERENCES

[1] A. Morchid, R. Jebabra, H. M. Khalid, R. El Alami, H. Qjidaa and M. O. Jamil, "IoT-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data and cloud computing," *Results in Engineering*, vol. 23, Art. no. 102829, 2024.

[2] Y. E. Kunt, "Development of IoT and AI based Smart Irrigation System," *arXiv preprint*, arXiv:2506.11835, 2025.

[3] Y. Zhang, X. Wang, L. Jin, J. Ni, Y. Zhu, W. Cao and X. Jiang, "Research and development of an IoT smart irrigation system for farmland based on LoRa and edge computing," *Agronomy*, vol. 15, no. 2, Art. no. 366, 2025.

[4] N. Gondchawar and R. S. Kawitkar, "IoT based smart irrigation and crop field monitoring system," *International Journal of Computer Applications (IJCA)*, vol. 135, no. 4, pp. 1–5, 2016.

[5] M. Panchal, A. Patel, P. Shah and N. Patel, "Smart irrigation system for precision agriculture using IoT," in *Proceedings of the IEEE International Conference on Internet of Things (IoT)*, pp. 1–6, 2018.

[6] G. Karthick, M. Karthikeyan and R. Ranjithkumar, "Automated irrigation system using IoT," *International Journal of Engineering Research & Technology (IJERT)*, vol. 6, no. 4, pp. 1–5, 2017.

[7] H. Kim, R. Evans and W. Iversen, "Remote sensing and control of an irrigation system using a distributed wireless sensor network," *ASABE Journal of Irrigation and Drainage Engineering*, vol. 134, no. 2, pp. 223–231, 2008.

[8] S. K. Ghosh, A. Banerjee and S. Mukherjee, "Smart agriculture using IoT, cloud computing and AI," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 17, no. 3, pp. 1–8, 2019.

[9] P. Rawal, S. Singh, H. Choudhary and A. Bhargava, "IoT based smart irrigation system," *International Journal of Computer Applications*, vol. 159, no. 8, pp. 7–10, 2017.

[10] FAO, "The state of food and agriculture: Climate change, agriculture and food security," *Food and Agriculture Organization of the United Nations (FAO)*, Rome, 2016.