# Smart Agricultural System Using IoT and Machine Learning

[1]Dhakshaya M, [2]Ramaraj M

Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamil Nadu, India

[1]dhakshayavcm@gmail.com, [2]ramaraj.phdcs@gmail.com

***Abstract-****Precision agriculture has become an important approach for improving crop productivity, optimizing resource utilization, and supporting sustainable farming practices. This project presents a smart agricultural system that combines Internet of Things technologies with machine learning methods to monitor field conditions and generate intelligent recommendations for farmers. The system collects real-time and near-real-time data using sensors such as soil moisture, temperature, and water-level sensors connected through an ESP32-based setup. It analyzes both sensor inputs and historical agricultural features to recommend suitable crops, seed varieties, and fertilizers while also supporting irrigation alerts based on threshold conditions. A Streamlit-based interface is used to present environmental readings, recommendations, and alerts in a simple and interactive form. By integrating IoT monitoring, predictive analytics, cloud communication, and a user-friendly dashboard, the system reduces manual effort and improves agricultural decision-making.*

***Keywords:*** *Precision Agriculture, Internet of Things, Machine Learning, Random Forest, Streamlit, ESP32, Crop Recommendation, Fertilizer Recommendation.*

## I. Introduction

Traditional agriculture often depends on experience-based decisions, delayed observations, and generalized recommendations that may not suit local soil and environmental conditions. This can lead to inefficient irrigation, suboptimal crop choices, poor fertilizer use, and reduced productivity.

The proposed Smart Agricultural System uses IoT sensors and machine learning to create a data-driven recommendation platform for modern farming. The project integrates field sensing, cloud-based data access, predictive modelling , and dashboard-based visualization so that farmers can make timely and informed decisions.

The system focuses on real-time environmental awareness and multi-output agricultural prediction. In addition to monitoring conditions, it predicts the best crop, seed variety, and fertilizer for given inputs, which makes it more useful than systems that only display sensor data.

## II. Existing and Proposed System

Existing agricultural systems mostly rely on traditional farming practices, manual observation, or isolated digital tools that provide only limited monitoring without intelligent prediction. Many of these systems lack tight integration between IoT devices and machine learning models, which reduces adaptability to changing field conditions.

The proposed system combines IoT sensing, cloud communication through ThingSpeak, a Multi-Output Random Forest model, and a Streamlit-based interface. This architecture allows the system to perform real-time monitoring, predictive recommendation, and basic automation support such as sprinkler-related alerts.

### Advantages of the Proposed System

- Improved crop prediction accuracy using machine learning models.
- Real-time monitoring through IoT sensors and cloud connectivity.

- Simultaneous prediction of crop, seed variety, and fertilizer.
- Better resource optimization for water, fertilizers, and soil usage.
- User-friendly dashboard for farmers and non-technical users.
- Support for scalable deployment in small and large farms.

## III. System Specification

### Hardware Requirements

- ESP32 development board for IoT integration and Wi-Fi communication.
- DHT11 or DHT22 sensor for temperature and humidity sensing.
- Soil moisture sensor for measuring soil water content.
- Water level sensor for irrigation monitoring.
- Intel Core i5 or above, minimum 8 GB RAM, and 500 GB storage for training and development.

### Software Requirements

- Operating System: Windows 10/11, Linux, or macOS.
- Programming Language: Python 3.7 or above.
- Embedded Tools: Arduino IDE or ESP-IDF for ESP32 programming.
- IDEs: Jupyter Notebook, Visual Studio Code, or PyCharm.
- Libraries and Frameworks: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, Joblib, Requests, and Streamlit.
- Cloud Platform: ThingSpeak for IoT data storage and monitoring.

## IV. Methodology

The methodology begins with two forms of data acquisition: a structured agricultural dataset and live sensor readings from IoT hardware. The dataset contains features such as soil type, pH level, temperature, moisture, and nutrient values, while ESP32-connected sensors collect environmental values from the field and transmit them through ThingSpeak.

Before training, the data is preprocessed by handling categorical variables through label encoding, checking for null values, and splitting the dataset into training and testing partitions. The project uses an 80/20 split and prepares the data for multi-output prediction, where a single pipeline supports multiple recommendation targets.

For prediction, the system evaluates several models and selects a Multi-Output Random Forest classifier as the final algorithm. The trained model predicts the best crop, seed variety, and fertilizer recommendation from soil and environmental features, while the Streamlit dashboard displays results and real-time sensor values.

## V. Module Description

### Data Gathering Module

This module combines historical agricultural records and IoT sensor readings. Dataset values support training, while real-time values from sensors improve live monitoring and practical relevance.

### Data Preprocessing Module

This module encodes categorical fields, structures numerical inputs, and checks for consistency. It ensures the dataset is ready for model training and prediction.

### Prediction Module

The prediction engine uses a Multi-Output Random Forest classifier to generate crop, seed, and fertilizer recommendations together. This makes the output more useful for real farming decisions than a single-target model.

### IoT Integration Module

The IoT component collects sensor readings from the field through ESP32 hardware and publishes them to ThingSpeak. These readings are later fetched by the application for monitoring and recommendation support.

### Dashboard Module

The Streamlit interface allows users to view sensor data, input soil parameters, and receive recommendations in an understandable format. This improves accessibility for non-technical users.

## VI. System Design

The project follows a modular design where datasets, trained models, encoders, IoT data, and application code are stored in separate folders. This structure improves readability, maintainability, and ease of extension.

The input design supports both manual user inputs and automatically fetched IoT values. Users can enter soil type, pH, nitrogen, phosphorus, and potassium, while the system retrieves temperature, moisture, and water-level values from the cloud-connected sensor setup.

The output design focuses on clarity and decision support. It displays crop recommendations, seed suggestions, fertilizer outputs, irrigation-related status, and visual summaries through the dashboard.

## VII. Data Flow and ER Design

The data flow starts with the collection of inputs such as soil type, pH, NPK values, temperature, moisture, and water level. Some values come directly from users, while others are retrieved from the ThingSpeak API and combined into a feature vector for the machine learning layer.

The Random Forest model processes this feature set and generates three main outputs: crop recommendation, seed suggestion, and fertilizer recommendation. In parallel, the system checks water-level status and can support sprinkler-related automation alerts as part of the output layer

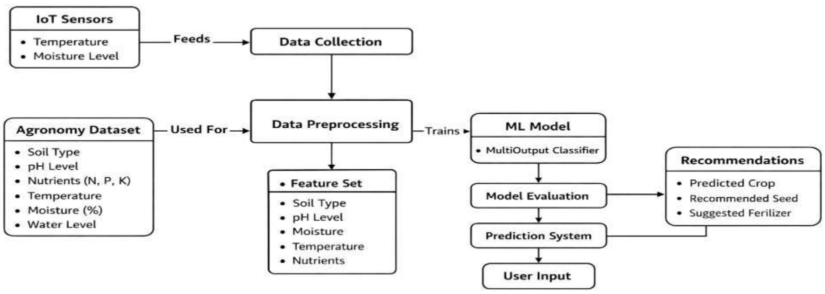


Figure 1. Data Flow Diagram of the smart agricultural system using IoT and machine learning.

## VIII. Testing and Implementation

The project includes unit testing, integration testing, and system testing to verify data preprocessing, model training, prediction generation, and module interaction. The report also highlights practical maintenance considerations such as version control, documentation, cloud storage, and resource management.

System implementation combines ESP32-based IoT sensing, ThingSpeak-based cloud communication, a Python machine learning workflow, and a Streamlit user interface. The complete flow starts from sensor data collection and ends with dashboard-based recommendations and alerts.

## IX. Results and Discussion

The system demonstrates that combining IoT with machine learning can improve farming decisions through continuous monitoring and data-driven recommendations. Its ability to suggest crop type, seed variety, and fertilizer at the same time makes it suitable for practical precision agriculture use.

The Streamlit-based interface also adds usability by presenting predictions and environmental status in a clear form. By linking sensor monitoring with predictive analytics, the project reduces manual effort and supports more efficient and sustainable agricultural practices.


*Figure 2. Abstract page from the uploaded final project report.*

## X. Conclusion and Future Enhancement

The Smart Agricultural System using IoT and Machine Learning provides an intelligent farming support platform that integrates field sensing, cloud connectivity, and predictive modeling. It improves productivity, helps optimize resource usage, and supports sustainable agriculture through timely recommendations and irrigation awareness.

Future enhancement can include additional sensors such as pH and nutrient sensors, deep learning models for improved prediction quality, real-time weather forecasting integration, mobile application support, fully automated irrigation, edge computing, drone-based monitoring, satellite imaging, and expansion into a broader agricultural decision platform

ISSN: 3107-6513

**REFERENCES**

K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," Sensors, vol. 18, no. 8, p. 2674, 2018.

A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018.

M. S. Mekala and P. Viswanathan, "A survey: Smart agriculture IoT with cloud computing," in Proc. Int. Conf. Microelectronic Devices, Circuits and Systems, 2017, pp. 1-7.

S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika, and J. Nisha, "Crop recommendation system for precision agriculture," in Proc. IEEE Int. Conf. Advanced Computing, 2017, pp. 32-36.

F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

R. Patel, M. Patel, and D. Patel, "Multi-output crop and fertilizer recommendation system using machine learning," Int. J. Eng. Res. Technol., vol. 9, no. 5, 2020.[1][2][3][4][5][6]