International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 4, July-August 2025

Impact Factor: 8.152

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 4, July - August 2025 ||

DOI:10.15680/IJARETY.2025.1204086

Real Time Crop Prediction and Fertilizer Recommendations System using Machine Learning and IOT

Prasad S

Department of Masters of Computer Applications, CMR Institute of Technology, Bengaluru, India

ABSTRACT: Lack of appropriate facilities to test soil and lack of agronomic recommendations often make farmers to find it difficult to choose on the right crop and fertilizer to use. Old practices of measuring soil quality and level of fertilizers needed are time-consuming and expensive. The present paper suggests a real-time system capable of predicting crops and calculating a fertilizer recommendation with the use of Internet of Things (IoT) devices and Machine Learning (ML) algorithms. The system uses sensors to gather soil and other environmental information that is further processed with models like Naive Bayes and random forest that predict optimal crop. It also prescribes fertilizers that suit the soil nutrient balance. The system showed more than 80% real-life prediction accuracy aspiring to assist farmers in making data-backed decision making, lower the input costs, and increase agricultural output.

KEYWORDS: Crop Prediction, Fertilizer Recommendation, Random Forest, Soil NPK Sensor, IoT in Agriculture, Machine Learning.

I. INTRODUCTION

The agriculture sector in India plays a vital role in the economy and ensures food security, as more than half of the population is involved in farming to support their livelihoods. However, various factors such as crop choice, inefficient use of fertilizers, and irregular weather patterns pose significant risks to productivity. The introduction of precision agriculture, supported by modern technologies like the Internet of Things (IoT) and Machine Learning (ML), presents a transformative solution.

IoT technology enables real-time monitoring through connected sensors that gather essential data on soil composition, temperature, humidity, and rainfall.

This data is then analyzed using machine learning algorithms to provide insights on suitable crops and the right type of fertilizers needed. These insights help farmers make well-informed decisions about their farming practices.

Traditionally, soil testing involves sending samples to laboratories, which is expensive and not always accessible to small-scale farmers.

Moreover, fertilizers are often applied based on assumptions, leading to either excessive or insufficient use, both of which negatively affect crop yield and the environment. This study introduces an intelligent, real-time system aimed at overcoming these challenges.

The proposed system uses affordable and portable components such as an Arduino Uno, NPK sensors, and soil moisture sensors to collect data directly at the farm site.

This data is processed using machine learning models like random forest and Naive Bayes to recommend the best crops and fertilizers for the specific conditions.

By automating these decisions, the system enhances productivity, optimizes input usage, and reduces environmental impact, contributing to sustainable agricultural practices.

II. LITERATURE REVIEW

Machine Learning (ML) and the Internet of Things (IoT) have been used in recent developments in agriculture to tackle problems that involve crop yield-anticipation and fertilizer-prescription. Some researchers have investigated different ML models to modify agricultural output basing on real-time environmental information.

Vasily Martin and Jean-Franois Bonnaud [12] employed different ML models to predict crop yield in numerous environments (New Zealand, China, Australia, etc.) using the dataset that was provided by the Australian government. Their research found out that the results of this Random Forest Regressor model proved the most accurate candidate in predicting the data when temperature, rainfall, and seasonal data were entered as input features. Sequential models such as Recurrent Neural Networks and LSTM also performed well on predicting the environment parameters.

Kaur and Soni [2] surveyed statistical models as well as remote sensing crop yield forecasting methods. Their survey noted that combining satellite images and vegetation indices with classical models (e.g., linear regression and Support Vector Machines or SVM) increases the accuracy of prediction.

Garg et al. [3] performed a thorough study of ML methods in predicting yield. They also remarked that Decision Trees (DT), SVM, and Artificial Neural Networks (ANN) models can be effective when coupled with weather and soil data, however, they each come with their own trade-offs relating to data requirements and interpretability.

The article by Saha et al. [4] suggested the need to integrate various data inputs-weather, soil characteristics, and irrigation habits, to have a reasonable forecast. Their survey endorses the implementation of ensemble methods in order to eradicate the shortfalls of individual algorithms.

III. ANALYSIS

Problem Statement
The unpredictability of crop selection and the use of fertilizers has long been a significant challenge for Indian farmers.

Factors such as erratic rainfall, imbalanced soil nutrients, and the lack of timely agricultural advice lead to reduced crop outputs and financial losses. Conventional farming often relies on outdated practices or assumptions, which can result in poor yields and inefficient use of resources.

This proposal aims to tackle two main issues: crop forecasting and fertilizer recommendations.

It proposes the use of an IoT and machine learning-based smart system that operates in real time. The objective is to create a model that analyzes environmental and soil data to suggest the best crops to grow and the appropriate fertilizers needed. By automating this process, farmers can optimize their input costs and improve both the quality and quantity of their harvest.

System Requirements
To ensure the proper functioning of the proposed system, both hardware and software components must be in place to make the equipment functional as described:

Software Requirements
Python Libraries: NumPy, Pandas, SciPy, and Scikit-learn (for machine learning algorithms), Flask, and Jinja (for deploying the system on the web)
Tools: PyCharm IDE, Arduino IDE
API: Weather API to obtain real-time temperature and humidity data

Hardware Requirements
Microcontroller: Arduino Uno
Sensors: Soil NPK sensor, Soil Moisture sensor
System: A laptop or PC with an Intel i3, i5, or i7 processor, 4GB of memory, and 1GB of free disk space
C.7 Functional Requirements

International Journal of Advanced Research in Education and TechnologY(IJARETY)
| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |
|| Volume 12, Issue 4, July - August 2025 ||
DOI:10.15680/IJARETY.2025.1204086

The system should:
- Collect environmental and soil data in real time using sensors
- Clean and normalize the data
- Utilize machine learning algorithms to predict suitable crops and recommend necessary fertilizers
- Present the current results through a web-based user interface

The flow of data is illustrated using Data Flow Diagrams (DFD):
Level 1 DFD outlines the main inputs (sensor data), processing (machine learning model), and outputs (recommendations and predictions).D. Non-Functional Requirements

Performance: Data must be entered, and the system should respond in a few seconds even in cases of large quantities of data
Accuracy: The system to have an accuracy of predicting days of at least 80%
Scalability: Able to accommodate new sensors or greater numbers of users
Security: Guarantees secure data processing and controlling access on a role basis
Usability: Accessible to farmers with very low levels of digital literacy, the web interface is clean and well-designed
Maintainability: A codebase can support future updates and introduce bug fixes and scaled modular features
Reliability: It has to work all the time with the least amount of disruption in the rural network environments
Compatibility Where it runs on widely used browsers and devices

Feasibility Study
Viability of the proposed project has been rendered in several aspects:
Feasibility: Testing with both tech-savvy and non-tech users has proven that it is user-friendly and practically applicable
Technical viability: The system is built of open-source tools including widely available, low-cost hardware
Feasibility of Implementation: The system can operate in already limited infrastructures, especially in the countryside
Scheduling Feasibility: This cycle of development can be performed in a modular, phased-style to ensure flexibility in field testing as well as adjustments

Apply Use Case Analysis
Operationally Key operations are defined in the use case diagram
1. Data Collection: Sends data to app in real time that is monitored with Soil and climate sensors
2. User Input: Manual: option of manual input by non-sensing user
3. Prognosis: Back end machine learning model reads data to recommend crops
4. Output Display: The aim is the user will get predictions and recommendations on fertilizers through a dashboard
These examples of use cases illustrate flow of decision-making through the system monitoring data gathering to the output that can be acted upon

IV. SYSTEM DESIGN

A. Design Objectives
1. The primary design challenge of this system is to offer an efficient, scalable, and user-friendly tool that assists in predicting the best crops and suggesting suitable fertilizers based on real-time field conditions.

It is essential that the system:
Supports multiple sensor inputs and user contributions,
Integrates machine learning models for predictive analysis,
Provides interactive feedback to users via an internet-based interface.
To achieve this, a modular and component-based design approach was adopted to ensure ease of future upgrades and maintenance.

B. Design Strategy
The development strategy was structured into several logical steps to streamline the implementation process.
Each phase contributed progressively to the system's overall functionality:
1. Hardware Acquisition and Setup
Key IoT devices such as the Arduino Uno, Soil NPK Sensor, and Moisture Sensor were selected and configured.
Each component was tested individually before being integrated to ensure accuracy and responsiveness.

2. Dataset Development and Assessment
Publicly accessible datasets were used alongside real-time sensor data.
Initial analysis helped identify the most relevant features for training the model, such as temperature, humidity, and levels of nitrogen, phosphorus, and potassium.

3. Data Preprocessing
Statistical techniques were applied to clean and normalize the data.
Missing values were addressed through imputation, and case-control data were balanced to prevent model bias.

4. Algorithm Selection
A comparison of classifiers, including Naïve Bayes, K-Nearest Neighbors, Decision Trees, and Random Forest, was conducted**.**
Random Forest was chosen for its high accuracy and ability to handle high-dimensional data effectively.

5. Model Training
The model was trained and validated using labeled datasets to ensure reliability and generalization, with cross-validation employed to enhance performance

1.Web interfaces Development
The front-end of the system was created on the basis of Flask and Jinja templates. The dashboard will offer a graphical display of understandable data relating to crop recommendation and fertilizer schedules.

2. Testability and Integration.
The last step involved the combination of the hardware with the software stackage and confirmation that the system was functional in a variety of manners end-to-end

System Architecture

- The system is architected into four fundamental modules:
- 1. Input Layer: Characterises sensor and user inputs.
- 2.ProcessingLayer: Performs preprocessing of data, and calls ML Algorithm.
- 3. Prediction Engine: Uses models that have been trained to produce results.
- 4. User Interface: Presentation of results in user friendly web interface.
- The architecture is presented in Fig. 1 (the architecture diagram).

UML Class Diagram

- A class diagram gives a structural design of the program modules. It specifies classes like,
- Sensor Reader: A real-time capture of data
- Preprocessor: Cleans, formats data
- Crop Predictor: Uses a Random Forest logic
- Fertilizer Recommender: Recommends optimum fertilizers
- Dash board UI: Interface tool to deal with front end interaction
- Each of the classes contains characteristics that relate to its functionality and is chained so as to promote clean object-oriented design.

Sequence Diagram

- The sequence diagram describes the running interaction of system components. This is achieved using a process that first collects sensor data which is in turn fed to the ML model to make a prediction. The end results are then read to the web dashboard so that the farmer can see and take appropriate action.
- Collaboration Diagram

This diagram plots the interconnections between the objects and message flows taking place. It also aids in visual imagery of how various parts like sensors, data processors, and user interface are connected in terms of share of communication during the runtime.

State Chart Diagram
G. The state chart diagram shows the different states of the system.

- Idle (willing to work),

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 4, July - August 2025 ||

DOI:10.15680/IJARETY.2025.1204086

- Collecting (gathering sensor data),
- ML computation,
- Outputting (displays results),
- And Failure (Error state).
- It is to provide that the software is able to engage with transitions and resiliently recover in the event of faults.

Activity Diagram
Activity diagram depicts a normal user flow. It will start with the log in page (if it is necessary), move on to data entry or sensor detection and conclude with suggestions.

High-quality data collection is achieved through the use of environmental sensors. Predictions are made using machine learning techniques based on negotiation processes. An intuitive web portal is provided to deliver useful and actionable results. The system's back-end was developed using Python due to its extensive support for machine learning. For the front-end, lightweight web frameworks were integrated with Flask. The predictive models were developed and trained using Pandas, NumPy, and Scikit-learn, which are open-source machine learning tools. Flask and Jinja2 templates were used to ensure a quick and responsive deployment of the front-end, enabling efficient communication between the user interface and the machine learning components.

Hardware Platform B.
The hardware was inexpensive, easy to obtain components and real-time sensing and data acquisition was performed using the system:
Arduino Uno: This will be used as a microcontroller to interface sensors and transmit the data.
Soil NPK Sensor: Measures nutrient (Nitrogen, Phosphorus, Potassium).
Soil Moisture Sensor: It monitors the amount of water within soil so as to make decisions relating to irrigation.
Computer System Requirements
Constituent of process chip: Intel i3 / i5 /i7 /// AMD equivalent
RAM: must have 4 GB at least
Storage: 1 GB or more free disk space
These elements were chosen in light of their affordability and their application to open-source platforms, which makes them usable even in impoverished settings.

Software Platform C
The software stack was chosen because it supports scientific computing, was flexible and easy to integrate:
Back-end/ML:
Python 3.x
o Scikit-learn
o Pandas
o NumPy
o SciPy
• Frontend:
flask (web framework)
Jinja2 (templating engine)
• IDE: PyCharm in the backend logic
Arduino programming language IDE microcontroller
• APIs:
weather information (temperature, humidity) using Open Weather Map API in real-time
This stack will facilitate fast development and facilitated communication with components, ranging all the way down to the hardware sensors and all the way up to the user interface output.

Hardware Specifications
1. Arduino Uno
2. The core of the system is Arduino Uno, a powerful easy to use microcontroller board. Operating at 16 MHz on the ATmega328P chip, the microcontroller runs the project as its brain taking input reading of soil sensors and transmitting it to be processed. The Arduino has digital and analog pins so that it is easy to build and test the setup, and has USB connector that has been made to connect it to a computer with ease.
3. NPK Sensor Soil

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 4, July - August 2025 ||

DOI:10.15680/IJARETY.2025.1204086

4. A Soil NPK sensor is used to determine the health of the soil. The sensor can detect the three most important nutrients required by all plants, including Nitrogen (N), Phosphorus (P) and Potassium (K). It operates on special probes to read the level of the nutrient and broadcasts it on the use of RS-485 networking procedure. The sensor is then well calibrated to ensure that the values that are given are enough and reliable.
5. Moisture Sensor (In naissance Run Down Model)
6. Because water has as much significance as nutrients, there is capacitive soil moisture sensor that is also incorporated. This sensor examines the amount of water in the soil and converts it to a simple signal, which can be read by the Arduino. Based on this information, the system can make useful recommendations on when and how much to irrigate to prevent the possibility of under or over-watering.
7. Software Specifications
8. Python Libraries
9. scikit-learning: It powers the machine learning models that learns using soil and weather information to give predictions and recommendations.
10. Pandas and NumPy: Assist in the cleaning, organization and analysis of large amounts of data within a swift and effective manner.
11. SciPy: Its additional assistance is aimed at calculations and preprocessing of data.
12. SQL Alchemy: Enables easy connection to database, which in turn allows information to be stored, retrieved and managed easily.

14. Arduino IDE

Was used to program sensor reading and serial communication of the sensor data to the PC in a C++ language.

15. Web App Flask

The front end presents a form-based input interface where users could manually enter information or they can use sensor integration. The Flask server loads the ML model in the backend and renders the predictions retrieved to the UI.

Deployment Diagram

The deployment structure is shown in Fig. 5.12 (not shown here), and is as follows:
Client Device (Mobile phone of farmer or PC with browser)
Server (Flask App on local/ cloud instance)
Arduino device (sensors, sending to the server)

Details of the Implementation

1) Dataset
This data was on 2200 instances that covered 11 typical Indian crops. The soil nutrients, temperature, humidity and pH level are the attributes. The data was randomly separated into two sets (train and test; 80:20): it was used to assess the generative capabilities of models.

2) Preprocessing
Malfunctioning data were entered using mean substitution Features were standardized such that they fell in close ranges. Outliers were eliminated by an IQR filter.
Random Forest Model: Random Forest
Random Forest classifier was chosen because of its resistance to overfitting and nonlinear feature relations. It obtained more than 85 percent accurate in our tests and was better than the Naive Bayes, KNN and Decision Tree models.
Testing and validating
Validation was carried out on the model.
Cross-validation: To obtain statistical stability, 10-fold cross-validation was used.
Confusion Matrix: To analyse true, and predicted crop labels.
Accuracy and F1-Score: They are used as indicators of performance. Random Forest reached a ~87 % accuracy and approximately a 0.89 F1-score.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

The proposed research aims to provide a real-time solution to crops forecasting and fertilizer prescription to efficiently combine Machine Learning (ML) and Internet of Things (IoT) technologies to enable evidence-based decision-making in agriculture. The system is developed with the objectives of affordability, accuracy and ease of use so that farmers could gain knowhow about better crop types and fertilizer application on the basis of soil and environment parameters that were being measured in real-time.

The system helps farmers gain actionable knowledge by analyzing the information provided by sensors - the level of nutrients in the soil, and the moisture content of the soil, and applying ML algorithms such as Random Forest. The result is the enhanced productivity of farmers and the minimization of agricultural inputs being misused. The obtained prediction accuracy of more than 85 percent confirms that it is possible to apply the model in real-world practice.

The system is modular, thus making it simplistic to deploy and maintain, particularly in rural and resource-constrained environments. In addition, open-source technology and cheap hardware elements mean that such a solution is scalable and affordable.

To conclude, the presentation of the proposed system, in addition to empowering farmers with updated and precise suggestions, will lead to sustainable farming through effective resource management, upstream integration, and decentralized, on-demand management.

B. Future Work

Although the system has proven to be able to deliver on its qualities even with the tested settings, there are gaps that can be exploited later to ensure that there are better ways of expanding the capabilities of this system:
Deep Learning: Insertion of Deep Learning techniques, such as Convolutional Neural Networks (CNNs) or Long Short-Term Memory (LSTM) neural network models, may assist in learning more complex patterns in the data.

Integration of mobile apps: It would be important to create a mobile application that could have offline accessibility to cater to farmers who do not have time to access internet on a regular basis.

Enlarged Crop Database: To make the system more adaptable to different areas of harvesting, it is good to include more types of crops and variations thereof.

Multilingual Support: It will help in reaching more farmers by offering the interface in various languages of regions where it will be implemented.

Satellite and Drone Data: Combining ground sensor data with remote sensing observations can additionally enhance better prediction and ground mapping.

Inventory and Cost Optimization of Fertilizers: Future models will be able to disseminate real-time market information to recommend not only efficient fertilizers but cost effective ones at the local level.

By meeting these additions, the system will be transformed into a universal smart farming system that corresponds to the overall concept of digital farming and food security.

REFERENCES

You will find the References section in standard IEEE citation style below, based on works cited in your document:

1. A. Nigam, S. Garg, and A. Agrawal, Crop Yield Prediction using ML Algorithms, International Journal of Engineering Research & Technology (IJERT), vol. 8, 2019.
2. R. Kaur and S. K. Soni, Review of crop prediction models and remote sensing based methods, International Journal of Agricultural and Environmental Information Systems 8, 1 (2017), pp. 122.
3. R. Garg, P. Jain, and R. K. Jain, Crop yield prediction using machine learning algorithms: A review, Journal of Agricultural Science and Technology, 21/3 (2019), 501-516.
4. S. Saha, S. Ghosh, M. Pal, 2018. A review on machine learning techniques applied to crop yield prognosis International Journal of Computer Applications 182 (27): 23-30.
5. D. Kumar, A. Kumar, and S. Kumar, Crop yield prediction using machine learning techniques: a review, Journal of King Saud University - Computer and Information Sciences, 30, 2, 2018, 163-178.
6. N. Dubey and D. Kumar, A survey of machine learning techniques to predict crop yield, International Journal of Computer Sciences and Engineering, 7 (2019) 416-423.
7. S. K. Roy, S. C. Paul, M. S. Islam, and M. A. Hasan, Crop yield prediction using machine learning algorithms: A comprehensive review, IEEE Access, vol. 7, pp. 7890778926, 2019.
8. P. Patil, A. Shinde and M. Kadam, A survey on crop yield prediction using data mining techniques, International Journal of Advanced Research in Computer Science 8(1), 421-425, 2017.

IJARETY

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com