Sanskar Kadam , 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

AI-Based Crop Prediction and Recommendation System Using IoT Sensors and Ensemble Machine Learning

Sanskar Kadam, Varad Jamdar, Krushna Kapse, Snehal Shere, and Shrawani Mule.
Department of Applied Science and Engineering at AISSMS Institute of Information Technology, located in Pune, Maharashtra, India.
Guide: Dr. Ashish Apate

Abstract- Farming is a big part of India's economy, making up about 17% of the country's GDP. It also helps support the lives of more than half of the people living in rural areas. But even with its importance, many Indian farmers face crop failures every year. This often happens because they don't have access to affordable tools that can give them good advice on which crops to plant, based on scientific research. To solve this problem, we've developed a system that uses artificial intelligence to predict and recommend crops. It combines special hardware that senses the condition of the soil with a type of machine learning model that considers many factors at once. This allows the system to give farmers personalized advice in real time, helping them make informed decisions about which crops to plant. Here's a rewritten version of the input text in a more human-like tone, similar to the provided reference human samples: When it comes to measuring soil nutrients, we've developed a hardware prototype that's pretty impressive. It's made up of an RS-485 Modbus NPK sensor, a DHT11 temperature and humidity sensor, a capacitive soil moisture sensor, and an Arduino UNO microcontroller. In the early stages, we even experimented with a TDS sensor as a low-cost alternative for estimating soil nutrient characteristics. This approach helped us keep costs down while still testing the system's architecture. But in the end, we decided to use actual NPK sensor readings as the primary input for soil nutrients. We also trained and compared seven different machine learning models using a dataset of 2,200 agricultural samples covering 22 different crop classes. These models included Random Forest, XGBoost, LightGBM, SVM, Gradient Boosting, KNN, and Logistic Regression. And what we found was that a soft-voting ensemble combining Random Forest, XGBoost, and LightGBM achieved an impressive 99.77% test accuracy and 99.73% mean cross-validation accuracy. But here's the thing: we didn't just stop at soil nutrients. We also incorporated real-time weather data for temperature, humidity, and rainfall into our model, using the OpenWeatherMap API. This allows us to provide location-aware recommendations that take into account the specific weather conditions in a given area. And the best part? The entire system is deployed as a user-friendly Gradio web application, with three different output tabs: crop recommendation with confidence bars, soil health analysis with fertiliser advice, and a seasonal crop planner. What's really exciting about this system is that it directly supports the United Nations' Sustainable Development Goal 2 - Zero Hunger. By providing farmers with accurate and reliable recommendations, we can help increase crop yields and reduce hunger around the world. It's a big goal, but we're hopeful that our system can make a real difference.

Keywords — Crop recommendation, ensemble learning, IoT, NPK sensor, precision agriculture, Random Forest, XGBoost, LightGBM, Gradio, OpenWeatherMap API.

I. INTRODUCTION

Farming today is at a crossroads, where old ways of doing things meet new technology. Some farmers have been working on the same land for years and have a good sense of the soil and the seasons. But on the other hand, science has given us powerful tools like satellite pictures, special sensors, and machine learning. These tools can analyze the land

in ways that no one person can do on their own. The problem, especially in countries like India that are still developing, is finding a way to bring these two worlds together in a way that's affordable, easy to use, and actually helps farmers. We need to make sure that the new technology is something that farmers can really use, not just something that's expensive and hard to understand. By combining traditional knowledge with modern tools, we can create a better future for farming.

Crop failure in India is often caused by three main issues: picking the wrong crops, unpredictable rainfall, and damage from pests. In fact, the government's agricultural surveys show that choosing the wrong crops leads to losses of thousands of crores of rupees every year. The main reason for this is that many small and marginal farmers don't have access to reliable information about which crops will do well in their soil at a particular time of year. The current systems that advise farmers on which crops to plant have some major flaws. Either they rely on laboratory soil tests that are done periodically, which are expensive and take a long time, or they provide general recommendations that don't take into account the current weather conditions. This means that farmers are not getting the specific guidance they need to make informed decisions about which crops to plant, leading to a higher risk of crop failure.

This project is all about finding a complete solution to a problem. It's a system that collects important information about the soil, like how much Nitrogen, Phosphorus, and Potassium it has. This is done using a special sensor connected to a tiny computer called an Arduino UNO. The system also measures the temperature and humidity using another sensor, and it gets rainfall data from a weather forecast website. All this information - seven things in total - is then used by a machine learning program to figure out which crop would be best to plant in that soil. The program can choose from 22 different crops that are commonly grown in India, and it also says how confident it is in its choice. The result is shown in a simple web application that anyone can use, even if they're not tech-savvy.

Here's how the paper is laid out. First, we've got a section that looks at what other people have said about the topic - that's Section II. Then, in Section III, we get into what the actual problem is. After that, Section IV explains how we think we can solve it. The next part, Section V, is all about how the different parts of the system fit together. We've also got a section on the hardware and software we used, which is Section VI. When it comes to getting and preparing the data, that's all in Section VII. The models we used to make sense of it all are explained in Section VIII. What we found out when we ran the experiments is in Section IX, along with what it all means. We talk about what's good and not so good about our approach in Section X. Looking ahead, Section XI is about what we might do next. And finally, we wrap everything up in Section XII.

II. LITERATURE SURVEY

A structured review of published work in precision agriculture, IoT-based soil monitoring, and machine learning for crop prediction was conducted to identify existing approaches and their limitations.

A. Machine Learning for Crop Recommendation

Researchers found a way to predict which crops will grow well in certain soils. They used a computer program to look at lots of data from different soils and crops. The program was able to guess correctly about 94% of the time. It found that the levels of nitrogen, phosphorus, and potassium in the soil were the most important things to consider when deciding which crop to plant. However, the system had some limitations - it didn't take into account what was happening in the field at the time, like the weather or the condition of the soil. This makes it less useful for farmers who need to make decisions based on current conditions.

B. IoT-Based Soil Monitoring

Soil is a pretty big deal when it comes to growing crops, and knowing what's going on with it can make all the difference. A couple of researchers, Patil and Bhosale, came up with a way to keep an eye on soil moisture, pH, and temperature using a special device called NodeMCU, which is connected to the internet. This system was good at collecting data in real-time

and could even send alerts to the cloud. But, it had some limitations - it couldn't measure the levels of nitrogen, phosphorus, and potassium (NPK) in the soil, which is kind of a big deal for choosing the right crops. And, it didn't have any machine learning capabilities, which means it couldn't give farmers any advice on what crops to plant based on the soil conditions.

C. NPK Sensing and Crop Selection

Researchers looked at how well a certain type of sensor, called RS-485 Modbus NPK, could measure the nutrients in soil directly. They tested these sensors on five different soil samples and found that they were pretty accurate, with an average error of only 3.2% compared to the results from a laboratory. This means that these sensors are good enough to use in farming. The study also gave some useful information about how to connect the sensors to a device called Arduino, which was helpful for designing the hardware for this project. But one thing that was missing was a way to use machine learning to make sense of the data from the sensors.

D. Ensemble Methods in Agricultural Prediction

Researchers Ramesh and Vardhan made a breakthrough in 2022 by combining Random Forest and XGBoost techniques to accurately identify 22 different crop classes from a large dataset of 2,200 samples, achieving an impressive 99.3% accuracy rate. Their work showed that using multiple methods together can be more effective than relying on a single approach for complex agricultural problems. Building on this idea, our current project takes it a step further by using a softer approach to combining the results, incorporating real-time weather data, and linking the model to physical sensors to create a more dynamic and responsive system.

E. Gradient Boosting Frameworks

Researchers Chen and Guestrin came up with XGBoost in 2016, a system that's really good at boosting trees and has regularization built right in. This means it can handle classification and regression tasks with the best of them. Then, in 2017, Ke and his team introduced LightGBM, a gradient boosting algorithm that uses histograms and trains way faster than XGBoost on big datasets, all while keeping the accuracy pretty much the same. We decided to use both of these frameworks in our project, combining them into a single ensemble.

There's a big gap in what's been done so far - no one has created a system that brings together a hardware interface using NPK sensors, a highly accurate soft-voting ensemble, and live weather API integration all in one application that can be easily deployed. This project aims to fill that gap.

TABLE I: COMPARATIVE SUMMARY OF RELATED WORK

Ref.	Author/Year	Method	Accuracy	Limitation
[1]	Doshi et al., 2018	Random Forest	93.8%	No real-time sensors or weather API
[2]	Patil & Bhosale, 2020	IoT+NodeMCU	N/A	No NPK sensor; no ML model
[3]	Kumar et al., 2021	RS-485 NPK Sensor	3.2% err	No ML integration
[4]	Ramesh & Vardhan, 2022	RF+XGB Ensemble	99.3%	No hardware sensors or weather data
This work	**Kadam et al., 2025**	**NPK Sensor + Ensemble + API**	**99.77%**	**Complete end-to-end system**

III. PROBLEM STATEMENT

Despite the availability of machine learning models that can predict crop suitability with very high accuracy, these models rarely reach actual farmers in a form they can use. The core problem has three dimensions.

First, conventional soil testing requires laboratory analysis that can take days and costs several hundred rupees per sample — too slow and too expensive for small and marginal farmers who need actionable information before the sowing window closes. Second, existing digital crop advisory tools typically do not integrate real-time soil sensor data with live weather conditions, meaning their recommendations are based on historical averages that may not reflect the current state of the farmer's field. Third, even where machine learning models exist, they are rarely packaged in interfaces simple enough for non-technical users, and even more rarely connected to low-cost physical hardware that can be deployed in the field.

The proposed system addresses all three dimensions: it uses a low-cost IoT sensor setup to acquire soil parameters in real time, integrates live weather data for location-aware recommendations, and presents results through a simple web interface accessible to any user regardless of technical background.

IV. PROPOSED METHODOLOGY

The system was developed in four sequential phases.

A. Hardware Sensing Phase

The sensing layer consists of an Arduino UNO microcontroller interfaced with an RS-485 Modbus NPK sensor, a DHT11 digital sensor, and a capacitive soil moisture sensor. The NPK sensor is the primary soil nutrient acquisition device; it inserts into the soil and returns nitrogen, phosphorus, and potassium readings in mg/kg via the Modbus RTU protocol over an RS-485 serial interface. The DHT11 provides ambient temperature and relative humidity. The moisture sensor provides volumetric water content as an analog voltage proportional to soil wetness.

During the initial prototype stage, before the RS-485 NPK sensor was procured, a TDS (Total Dissolved Solids) Meter V1.0 sensor was used as a low-cost alternative to estimate relative soil dissolved-ion concentration, which served as a proxy indication of overall nutrient loading. This approach was explicitly treated as a prototype-level approximation and not as a scientifically rigorous NPK measurement. It allowed the team to validate the complete software pipeline — including the Arduino-to-Python serial communication, the Gradio web app, and the ML inference logic — before the final NPK hardware was available. All quantitative accuracy results reported in this paper are from the full NPK sensor setup.

B. Data and Model Phase

A publicly available agricultural dataset of 2,200 records covering 22 crop classes was used for model training. Seven ML models were trained in scikit-learn pipelines incorporating StandardScaler for feature normalization. A soft-voting ensemble of the three best-performing models (RF, XGBoost, LightGBM) was constructed by averaging their predicted class probability vectors and selecting the class with the highest mean probability as the final recommendation.

C. Software Integration Phase

The weather_api.py module fetches a five-day, three-hourly forecast from the OpenWeatherMap API for any specified city. Average temperature, average humidity, and total rainfall over the forecast period are computed and passed as inputs to the ML model alongside the sensor-acquired soil parameters. Error handling covers network timeouts, invalid city names, and API failures.

D. Interface Deployment Phase

The Gradio web application (app.py) accepts city name and soil NPK and pH values via interactive sliders and a text field. On submission, it fetches weather data, runs model inference, and populates three output tabs: the Recommendation tab (top-5 crops with confidence bars), the Soil Analysis tab (NPK status, soil health score 0–100, fertiliser recommendations), and the Crop Calendar tab (seasonal planting schedule for the top-5 predicted crops).

V. SYSTEM ARCHITECTURE

The system is organized into three layers: a sensing layer, a processing layer, and a presentation layer.

A. Sensing Layer

The sensing layer consists of the physical hardware: the RS-485 NPK sensor for soil nutrients, the DHT11 for temperature and humidity, and the capacitive moisture sensor for volumetric water content, all interfaced with an Arduino UNO. The Arduino reads data from each sensor, assembles a comma-separated data string, and transmits it to the host computer at 9600 baud over USB serial. A serial reading script (serial_read.py) on the host parses the incoming string and passes the sensor values to the application layer.

B. Processing Layer

The processing layer consists of the Python backend running on a laptop or embedded single-board computer. It has three sub-modules: the weather API

module that fetches and averages forecast data, the ML inference module that loads the saved Random Forest pipeline (crop_model_enhanced.pkl) and runs predict_proba() to generate per-class confidence scores, and the soil analysis module that evaluates NPK levels and pH against agronomic thresholds to generate a soil health score and fertiliser recommendations.

C. Presentation Layer

The presentation layer is the Gradio web application, which renders a browser-based user interface accessible on the local network. The interface provides sliders for soil NPK and pH inputs, a text field for the city name, and three tabbed output panels displaying prediction results, soil analysis, and seasonal planning information.

D. Data Flow

Here's how the whole process works: first, you enter the city name and details about the soil in a special interface. Then, a part of the program called weather_api.py uses a service called OpenWeatherMap to get information like the average temperature, how humid it is, how much rain there is, and if there's a risk of bad weather for harvesting. All this information, along with details about the soil like its nitrogen, phosphorus, and potassium levels, is put together.

The program then uses a special set of instructions it learned from data to make predictions, and it shows you the top 5 possibilities with how confident it is in each one. At the same time, it looks at each aspect of the soil to figure out how healthy it is, giving it a score from 0 to 100. Finally, all the results are displayed in three separate sections for easy viewing.

VI. HARDWARE COMPONENTS

A. Hardware Components

TABLE II: HARDWARE COMPONENT SPECIFICATIONS

#	Component	Key Specifications	Role in System
1	RS-485 Modbus NPK Sensor	12V DC, 1–1999 mg/kg, ±2% accuracy	Direct soil N, P, K measurement
2	DHT11 Sensor	3.3–5V, 0–50°C, 20–90% RH, ±2°C/±5% RH	Ambient temperature and humidity
3	Capacitive Moisture Sensor	3.3–5V, 0–100% VWC, Analog0–3.3V out	Soil volumetric water content
4	TDS Meter V1.0 (Prototype only)	5V, 0–1000 ppm, analog voltage output	Low-cost proxy for soil ion concentration during prototype phase
5	Arduino UNO	ATmega328P, 16 MHz, 32KB Flash, 14 I/O pins	Central data acquisition and serial transmission
6	RS-485 to UART Adapter	MAX485 chip, 5V, half-duplex	Protocol conversion for NPK sensor
7	12V DC Power Supply	12V 1A regulated	Power for NPK sensor module

The TDS sensor is worth mentioning. It measures the total dissolved solids in soil leachate, which is somewhat related to the total dissolved mineral ion concentration in the soil solution. Although it's not possible to use TDS to accurately determine individual NPK values, the team used it as a prototype to test the entire software pipeline and identify any integration issues before moving forward. This approach allowed them to work out any kinks in the system and make sure everything was working together seamlessly. By using the TDS sensor in this way, the team was able to refine their process and prepare for more precise measurements in the future.

The project was able to use the RS-485 NPK sensor, which was a key part of the plan to show that the system could work while keeping costs down. In fact, the NPK sensor was used to get all the results that were reported in the end.

TABLE III: SOFTWARE TOOLS

Module / File	Technology	Function
train_model.py	scikit-learn, XGBoost, LightGBM, joblib	Train 7 ML models, 5-fold CV, save .pkl pipeline
weather_api.py	requests	Fetch 5-day forecast; compute avg. temp., humidity, total rainfall
app.py	Gradio, pandas, numpy, joblib	Web UI, input validation, model inference, soil analysis, calendar
serial_read.py	pyserial	Read Arduino USB serial, parse NPK and sensor values
OpenWeatherMap API	REST / JSON	Live weather data: temperature, humidity, 5-day rainfall forecast

VII. DATA COLLECTION AND PROCESSING

A. Dataset Description

We used a machine learning model that was trained on a big set of data about farming. This data had 2,200 records and each record had seven things that described the soil and environment, plus one label that said what kind of crop it was. The data was pretty evenly spread out, with 100 examples of each of the 22 different kinds of crops, so we didn't have to worry about some crops being overrepresented. Also, none of the records were missing any information, which made things easier.

TABLE IV: DATASET FEATURE DESCRIPTION

Feature	Full Name	Unit	Range in Dataset
N	Nitrogen	mg/kg	0 – 140
P	Phosphorus	mg/kg	5 – 145
K	Potassium	mg/kg	5 – 205
temperature	Ambient Temperature	°C	8.8 – 43.7
humidity	Relative Humidity	%	14.3 – 99.9
ph	Soil pH	—	3.5 – 9.9
rainfall	Annual Rainfall	mm	20.2 – 298.6
label	**Crop Class (Target)**	**—**	**22 classes, 100 samples each**

The 22 crop classes are: rice, maize, chickpea, kidneybeans, pigeonpeas, mothbeans, mungbean, blackgram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, and coffee.

B. Data Preprocessing

The dataset was loaded into a Pandas DataFrame and split into an 80/20 stratified train-test partition using random_state=42, ensuring all 22 classes are proportionally represented in both subsets. StandardScaler was embedded inside each scikit-learn Pipeline object to perform zero-mean, unit-variance normalization of all feature values before they reach the classifier. Embedding the scaler inside the pipeline eliminates data leakage: the scaler is fit only on the training fold during cross-validation, not on the full training set. For XGBoost and LightGBM, which require integer class labels, a LabelEncoder was applied to convert string crop names to integers before training; the encoder was saved separately as label_encoder.pkl for use at inference time.

VIII. MACHINE LEARNING MODEL

A. Individual Models

We trained and tested seven separate classifiers. Each one was set up with a pipeline that started by scaling the data to a standard format.

Random Forest is a way of making lots of decision trees that aren't too similar to each other. It does this by randomly choosing which features to look at when deciding how to split the data. If we make 300 of these trees and only look at the square root of the features when splitting, the model is really good at handling noisy data and giving us reliable probabilities. Plus, it tells us which features are most important, so we can understand what's going on.

XGBoost constructs trees sequentially, with each tree correcting the residual errors of the preceding ensemble. The implementation used 200 estimators with a maximum depth of 6 and a learning rate of 0.1. Built-in L1 and L2 regularisation terms prevent overfitting.

LightGBM uses a special way to handle features, called histogram-based feature bucketing, and it grows trees in a leaf-wise manner, not depth-wise. This makes it really fast and good with memory, especially when dealing with huge datasets. For this setup, we used 200 estimators, each with 63 leaves, and a learning rate of 0.05. This combination helps LightGBM work efficiently and make accurate predictions. Compared to XGBoost, LightGBM is often faster and uses less memory, which is a big advantage when working with large amounts of data.

SVM with an RBF kernel (C=10, probability=True) provides effective margin-based classification but does not natively produce calibrated probabilities; Platt scaling was applied via the probability=True flag. Gradient Boosting, KNN (k=5, distance-weighted), and Logistic Regression (lbfgs solver, max_iter=2000) were included for baseline comparison.

B. Soft-Voting Ensemble

The soft-voting ensemble combines the probability outputs of the three top-performing individual models: RF, XGBoost, and LightGBM. Let P_RF(c), P_XGB(c), and P_LGB(c) denote the predicted probability of class c from each model. The ensemble probability for class c is:

$$P_{ens}(c) = [P_{RF}(c) + P_{XGB}(c) + P_{LGB}(c)] / 3$$

The final predicted class is: $y_{pred} = \text{argmax}_c P_{ens}(c)$. Soft voting is preferred over hard voting because it incorporates the confidence levels of each model, making it more robust in borderline cases where two crop classes have similar soil requirements.

C. Cross-Validation

We tested our models using a special method called five-fold stratified cross-validation. This means we divided our dataset of 2,200 samples into five parts, making sure each part had a good mix of all 22 classes. By doing this, we got a better idea of how well our models would work in general, rather than just relying on one specific split of the data into test and training sets. This approach helps us understand how well our models can be applied to new, unseen data.

IX. EXPERIMENTAL RESULTS AND ANALYSIS

A. Model Accuracy Comparison

TABLE V : MODEL PERFORMANCE COMPARISON

Model	Test Accuracy	CV Mean (5-Fold)	CV Std Dev
Random Forest	99.45%	99.40%	0.0021
XGBoost	99.32%	99.27%	0.0019
LightGBM	99.18%	99.10%	0.0024
SVM (RBF, C=10)	98.86%	98.77%	0.0031
Gradient Boosting	98.41%	98.36%	0.0028
KNN (k=5)	97.73%	97.59%	0.0048
Logistic Regression	95.45%	95.18%	0.0062
Ensemble (RF+XGB+LGB)	**99.77%**	**99.73%**	**0.0015**

The group of models works really well together, getting it right 99.77% of the time on the test, and it's also very consistent, with a standard deviation of just 0.0015 when we try it out in different ways. This is better than any single model, like the Random Forest one that got 99.45% right. It's not a huge difference, but it's consistent across all the different tests we ran, which shows that combining the models in a smart way really helps reduce the chances of getting wrong predictions.

B. Feature Importance Analysis

When you look at what makes crops grow well, two things stand out: rainfall and humidity. Together, they make up about 48.2% of what determines whether a crop will thrive. This shows that having enough water is the main factor in deciding if a crop is suitable for a particular area. Temperature and

potassium are also important, making up around 31% combined. Other factors like nitrogen, pH, and phosphorus are less significant on their own, but together they contribute about 21% and are still crucial for accuracy - you can't ignore them without affecting the outcome.

C. Sample Prediction Test Cases

TABLE VI : SAMPLE PREDICTION RESULT

#	N	P	K	pH	Temp. (°C)	Rain (mm)	Expected	Predicted (Conf.)
1	90	42	43	6.5	22	203	Rice	Rice (98.4%)
2	20	18	20	7.2	26	55	Cotton	Cotton (97.1%)
3	14	22	16	6.0	28	113	Mango	Mango (99.1%)

The results of the three test cases were very promising, with the ensemble model making accurate predictions with confidence levels of over 97%. This shows that the model works well with different combinations of soil parameters and crop types. For example, in the test case in Pune, Maharashtra, the model recommended Muskmelon as the top crop, with a confidence level of 46.3%. This makes sense, given the summer climate in Pune at the time of testing, and the soil parameters used in the test, which were N=90, P=40, K=40, and pH=6.9. Overall, the model seems to be reliable and effective in making predictions across different scenarios.

D. Soil Health Analysis

We put the soil analysis tool to the test with a sample that had very low levels of key nutrients - nitrogen was at 25, phosphorus at 15, and potassium at 18. The pH level was also quite low, at 5.2. The tool correctly identified all three nutrients as being very low and the pH as too acidic. It then calculated a soil health score, which came out to be 28 out of 100. Based on this, the tool recommended adding some fertilizers to the soil: Urea at a rate of 60 kg per hectare, DAP at 40 kg per hectare, MOP at 30 kg per hectare, and Lime at 500 kg per hectare. These recommendations are in line with standard guidelines for fixing highly degraded and acidic soils in India.

X. ADVANTAGES AND LIMITATIONS

A. Advantages

- The new model is really good at predicting what kind of crop is in a picture. It can get it right 99.77% of the time when looking at 22 different types of crops. This is a big improvement over other models that have been tried before, which were only right about 93.8% of the time.
- The system is special because it brings together a few important things: tools that can sense NPK in the physical world, machine learning that can make sense of that data, and weather information. All of these things work together in one application that can be easily used, which is something that hasn't been done before in any published research in this area.
- Location-based suggestions: With real-time weather updates, the advice you get is based on what's happening now, like the current temperature, how humid it is, and if it's going to rain soon, not just what normally happens in your area.
- Using a low-cost approach to build a prototype can be really helpful. In this case, a TDS sensor was used to test the software pipeline without spending too much money on hardware. This way, the system's architecture can be tried out and validated before deciding to use more expensive NPK sensors. It's a good idea to start with something affordable and make sure everything works as it should before investing in more costly equipment. This approach can save time and money in the long run, and it allows for testing and refinement of the system before making a bigger commitment.
- Practical soil guidance: The fertiliser recommendation module offers useful, quantitative advice on correcting soil issues, and this advice is relevant no matter what crop you ultimately decide to plant.
- User-friendly interface: The Gradio web application requires no technical expertise to operate and is accessible on any device with a web browser.

B. Limitations

- A TDS sensor can be used as a rough guide to NPK levels: The sensor used in the early stages of the project can't tell the difference between individual nutrients like nitrogen, phosphorus, and potassium. It's not a replacement for a specialized NPK sensor if you need to know exactly how many nutrients are in the soil.
- The cost of an NPK sensor is a significant factor to consider. For instance, the RS-485 Modbus NPK sensor, although much more affordable than laboratory testing, can still cost between Rs. 2,000 and 3,500. This can be a major obstacle for smallholder farmers who are working individually. However, if these sensors are used by a group of farmers or at the village level, the cost can be shared, making it more manageable. This approach can help make the technology more accessible to those who need it.
- Internet connection is a must: For the weather API to work, you need to be online. If you're in an area with a weak phone signal, this part won't work properly.
- Dataset scope: The training dataset covers 22 crops and 2,200 records. Crops not in this set cannot be recommended, and the dataset may not fully represent all regional soil-climate combinations in India.
- Single-point sensing: The current setup measures soil parameters at a single location. Field-scale spatial variability requires multiple sensor nodes.

XI. FUTURE SCOPE

Several extensions can significantly increase the system's practical impact.

- Mobile Application: Porting the system to an Android or iOS application would allow farmers to access crop recommendations directly from their smartphones, reaching a far larger user base.
- Raspberry Pi Deployment: Replacing the host laptop with a Raspberry Pi 4 would make the system fully portable and field-deployable, with all components powered by a small solar panel and battery.
- No Internet? No Problem: Adding a local weather database or a sensor like BMP/SHT would let the system work even without internet. This way, it can still give you the weather info you need, anytime and anywhere.
- Crop Disease Detection: Adding a camera module and a lightweight CNN trained on leaf disease images would extend the system to detect and diagnose common crop diseases alongside the recommendation function.
- Multi-Language Interface: Adding Marathi, Hindi, and Telugu language support to the Gradio interface would substantially improve accessibility for rural farmers.
- Cloud Dashboard: A cloud-based data dashboard recording historical soil measurements and recommendations over time would help agricultural extension services track trends and target support.
- Automated Fertilizer Dispenser: By linking the soil analysis results to a microcontroller that controls a fertilizer dispensing system, we can complete the precision farming process, from getting data from sensors to actually taking action based on that data. This way, the whole cycle of precision agriculture, from sensing to acting, can be fully automated.

XII. CONCLUSION

This paper has presented a complete, end-to-end AI-Based Crop Prediction and Recommendation System that integrates IoT sensor hardware with ensemble machine learning and real-time weather data to provide actionable, location-specific crop recommendations. The system successfully addresses the core problem of incorrect crop selection by combining RS-485 NPK sensor measurements, DHT11 temperature and humidity readings, and OpenWeatherMap forecast data into a unified input vector for a high-accuracy ML model.

Researchers trained and tested seven different machine learning models on a large dataset of 2,200 agricultural samples, which covered 22 different types of crops. They found that combining the strengths of three models - Random Forest, XGBoost, and LightGBM - resulted in the most accurate predictions, with a test accuracy of 99.77%

and a mean cross-validation accuracy of 99.73%. What's more, this combined model was also very stable, with a standard deviation of just 0.0015. When they looked closer at the data, they saw that all seven input parameters played a role in making accurate predictions, but rainfall and humidity were the two most important factors. This suggests that these two factors have a big impact on crop yields, and that machine learning models can be a powerful tool for making predictions in agriculture. By using these models, farmers and researchers may be able to make more informed decisions about planting, harvesting, and crop management, which could lead to better outcomes and more efficient use of resources.

When working on this project, one key decision was to use a TDS sensor as a cheaper option while building the prototype. This allowed the team to test the entire software system before getting the special NPK hardware. It's a smart way to develop hardware in stages, especially for projects with limited resources or for students.

The Gradio web interface is really useful because it makes the system easy to use, even for people who aren't tech-savvy. It shows the results in three different tabs that are full of useful information. The first tab is about crop recommendation, and it has confidence bars to help you make a decision. The second tab is about soil health analysis, and it even gives you advice on fertilizers. The third tab is a seasonal crop calendar, which is super helpful for planning. When you put all of this together, the system is actually helping to achieve the goal of Zero Hunger, which is one of the Sustainable Development Goals. It does this by giving Indian farmers access to science-based information that can help them make affordable decisions about their crops, which can reduce crop failure and improve food security.

REFERENCES

[1] Z. Doshi, S. Nadkarni, R. Agrawal, and N. Shah, "AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms," in Proc. 4th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1–6.

[2] S. A. Patil and M. R. Bhosale, "IoT Based Smart Agriculture System for Monitoring Soil and Environment Parameters," International Journal of Engineering Research and Technology (IJERT), vol. 9, no. 6, pp. 531–534, Jun. 2020.

[3] R. Kumar, M. P. Singh, P. Kumar, and J. P. Singh, "Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique," in Proc. 2021 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021, pp. 1–6.

[4] V. Ramesh and K. Vardhan, "Crop Yield Estimation Using Machine Learning Algorithm," International Journal of Agricultural and Environmental Information Systems, vol. 13, no. 1, pp. 1–18, 2022, doi: 10.4018/IJAEIS.293594.

[5] M. Kang and M. Riegler wrote a paper called "Random Forests for Precision Agriculture" that was published in the IEEE Transactions on Agricultural Engineering. The paper is in volume 7, issue 3, and it's on pages 1023-1032, from 2021. This is a reference to a specific study on using random forests in agriculture.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, 2016, pp. 785–794.

[7] G. Ke et al., " LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, 2017, pp. 3146–3154.

[8] You can find more information about the 5 Day/3 Hour Forecast on the OpenWeatherMap API Documentation page, which is available on the OpenWeather Ltd website at https://openweathermap.org/forecast5 - just head over to this link to learn more, the page was last accessed in May 2025.

[9] Gradio Development Team, "Gradio: Build Machine Learning Web Apps in Python." [Online]. Available: https://gradio.app. [Accessed: May 2025].

[10] scikit-learn Developers, "scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org.