International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Smart Precision Agriculture Using Machine Learning and IOT: An IOT-Enabled, Soft-Voting Ensemble Framework for Data-Driven Crop Recommendation, Fertiliser Advisory and Soil-Condition Alerting

Lavanya B
SJB Institute of Technology
Bengaluru, INDIA

P Nandushree
SJB Institute of Technology
Bengaluru, INDIA

Spoorthy Kalmata
SJB Institute of Technology
Bengaluru, INDIA

Prof.Jasma Bhavani
SJB Institute of Technology
Bengaluru, INDIA

Abstract— Crop selection and fertiliser decisions in smallholder farming still rely on experience and occasional laboratory soil tests, so field conditions that change within days go unexamined. This paper presents the implementation of a Smart Precision Agriculture System that connects soil and weather sensing, machine-learning crop recommendation and rule-based advisory in one Flask web application backed by MySQL. An IoT sensor simulator streams eight parameters (nitrogen, phosphorus, potassium, temperature, humidity, soil moisture, pH and rainfall) to a relational store every 30 seconds. A soft-voting ensemble of Random Forest, Support Vector Machine, Gaussian Naive Bayes and Logistic Regression classifies the seven agronomic features into 22 crops and returns the three most probable crops with predicted class probabilities. A deterministic rule engine produces nutrient-deficiency advice and raises alerts for low soil moisture, acidic or alkaline pH and high temperature. On the public Crop Recommendation dataset (2,200 samples, 22 balanced classes) the ensemble reaches 99.32% accuracy and 99.37% macro-precision on a stratified 20% hold-out set and 99.41% under five-fold cross-validation; top-3 accuracy is 100%. These results demonstrate the functionality of the implemented software pipeline on the curated benchmark dataset but do not establish real-world agricultural effectiveness or field-level recommendation accuracy. It does not establish field accuracy, because the dataset is a curated benchmark and the sensor stream in this prototype is simulated. Both limitations are stated and addressed in the future-work plan.

Keywords— precision agriculture, crop recommendation, ensemble learning, soft voting, IoT, soil nutrients, fertiliser advisory, Flask, MySQL, decision support.

I. INTRODUCTION

Agriculture employs close to half of India's workforce, yet most holdings are small and farmers choose crops and fertiliser from habit, neighbours' practice or a soil-test report that may be a season old. Over-application of nitrogen, phosphorus and potassium (NPK) raises input cost and degrades soil and water, while under-application limits yield. Precision agriculture addresses this by measuring variability in the field and acting on it at the right time and place [1], [2].

Two technology trends make this practical at low cost. First, inexpensive sensors and microcontrollers can report soil and weather variables continuously, and recent surveys document their use in irrigation, nutrient management and crop monitoring [3], [4], [5]. Second, supervised machine learning can map soil and climate measurements to suitable crops with high accuracy, as reviewed in [6] and [7]. Most published work treats these as separate problems: a classifier trained on a static table, or a sensor dashboard with no recommendation logic. Farmers need one tool that ingests readings, explains the state of the soil, recommends crops, and warns when conditions leave a safe range.

This paper presents such a tool. The system collects eight soil and weather parameters, stores them in MySQL, and serves a web dashboard that reports the latest reading, the top three recommended crops with confidence percentages, NPK-based fertiliser advice and condition alerts.

A. Problem Formulation

Given a vector of agronomic measurements x = (N, P, K, T, H, pH, R), where N, P and K are soil nutrient levels, T is air temperature in °C, H is relative humidity in %, pH is soil acidity and R is rainfall in mm, the system must (i) rank the crops that suit x, (ii) judge whether each nutrient is adequate, and (iii) flag unsafe soil-moisture, pH and temperature values as they arrive, all through an interface that a non-technical user can operate.

B. Objectives

1. Acquire eight soil and weather parameters at fixed intervals and persist them with timestamps in a relational database.
2. Train and compare four classical classifiers and combine them into a soft-voting ensemble for 22-class crop recommendation.
3. Return a ranked top-three list with probabilities so the user sees alternatives and model confidence, not only a single label.
4. Provide transparent rule-based fertiliser advice and condition alerts that do not depend on the learned model.
5. Expose a web dashboard, a manual-entry form and a JSON API so that real sensors can replace the simulator without changing the rest of the system.
6. Report evaluation honestly, including what the benchmark metrics do and do not prove.

C. Contributions

The work contributes (a) an integrated sensing, prediction and advisory architecture in a single deployable Flask application; (b) a reproducible comparison of four base learners and their soft-voting combination under identical splits; (c) a top-k output design that exposes model uncertainty to the farmer; and (d) a separation between probabilistic recommendation and deterministic alerting, so a safety-relevant warning such as low soil moisture never waits on a model.

The remainder of the paper is organised as follows. Section II reviews related work. Section III describes the proposed method. Section IV reports results and discussion. Section V concludes and lists future work.

II. RELATED WORK

Precision agriculture and IoT. Zhang et al. give an early worldwide overview of precision agriculture and its dependence on spatial and temporal measurement [1]. Elijah et al. survey IoT and data analytics in agriculture and identify sensing, connectivity and analytics as the three layers any deployment needs [3]. Ayaz et al. review IoT platforms, sensors and open challenges including cost, power and interoperability [4], and Tzounis et al. catalogue field, greenhouse and livestock applications [5]. These surveys motivate the sensing layer used here, a low-cost stream of soil and weather parameters.

Machine learning for crop and soil decisions. Liakos et al. review machine learning across crop, livestock, water and soil management and report that support vector machines, random forests and neural networks dominate crop-management studies [6]. Kamilaris and Prenafeta-Boldú survey deep learning in agriculture and note that many agricultural tasks have small labelled datasets, which favours classical learners over deep models [7]. The Crop Recommendation dataset [8] is widely used as a benchmark for this task because it combines N, P, K, temperature, humidity, pH and rainfall for 22 crops. The present work uses this dataset and adopts the learner families that the reviews identify as most effective on tabular agronomic data.

Ensemble methods. Dietterich explains why combining diverse classifiers reduces variance and error relative to individual members [9]. Breiman's Random Forest averages decorrelated trees and is robust to feature scale and noise [10]. Cortes and Vapnik's support vector machine finds a maximum-margin boundary and, with kernels, handles non-linear class structure [11]. Soft voting averages predicted class probabilities and so preserves each member's confidence, which suits a system that reports top-k probabilities. Scikit-learn provides tested implementations of all four learners and the voting wrapper [12].

Gap. The cited work gives mature sensing platforms and accurate offline classifiers, but few implementations tie them into one running application that stores readings, recommends crops with confidence, advises on nutrients and raises alerts. This paper implements that integration at student-project scale and evaluates it reproducibly.

III. PROPOSED METHOD

A. System Architecture

The system has four layers, shown in Fig. 1. The *sensing layer* is a Python simulator that generates one reading every 30 seconds and inserts it into MySQL; a manual-entry form and a JSON endpoint accept readings from other sources. The *data layer* is a MySQL database whose sensor_data table stores temperature, humidity, soil moisture, rainfall, nitrogen, phosphorus, potassium, pH and a timestamp. The *intelligence layer* has two independent components: the soft-voting crop classifier and the rule engine for fertiliser advice and alerts. The *presentation layer* is a Flask application that renders the dashboard, result, alerts and history pages and exposes REST endpoints.

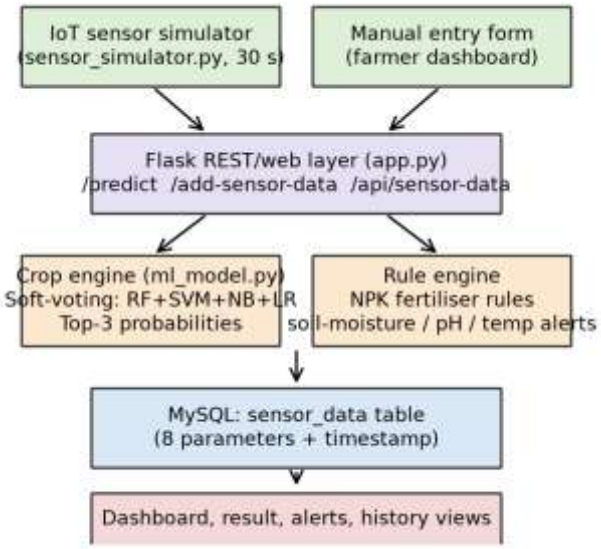


Fig. 1. Layered architecture of the Smart Precision Agriculture System.

The dashboard route retrieves the latest available reading and determines the sensor status based on the freshness of that reading. It passes the seven model features to the classifier and lists the recommended and alternative crops. The prediction route accepts the same values from a form, runs both engines, stores the reading and renders a result page. A POST endpoint accepts JSON so that a physical device can replace the simulator without changes elsewhere.

B. Dataset

The classifier is trained on the public Crop Recommendation dataset [8]. It contains 2,200 records, 100 per crop, for 22 crops: rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mung bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute and coffee. There are no missing values. Table I summarises the seven input features.

TABLE I
INPUT FEATURE RANGES IN THE TRAINING DATASET

Feature	Unit	Min	Mean	Max
Nitrogen (N)	Data Specific	0	50.6	140
Phosphorus (P)	Data Specific	5	53.4	145
Potassium (K)	Data Specific	5	48.1	205
Temperature	°C	8.8	25.6	43.7
Humidity	%	14.3	71.5	100
Soil pH	-	3.5	6.5	9.9
Rainfall	mm	20.2	103.5	298.6

The dataset reports nitrogen, phosphorus and potassium values as dataset-specific nutrient measurements without explicitly specifying physical units. Therefore, these values are treated as numerical input features for crop classification and should not be interpreted as fertiliser quantities or validated soil-test measurements. Soil moisture is captured by the sensing layer and used by the alert rules, but it is not a feature of the crop model because the benchmark dataset does not contain it.

C. Soft-Voting Ensemble

Four base learners are trained on the same data. The Random Forest uses 200 trees. The SVM uses a radial-basis kernel with probability estimates enabled, in a pipeline that standardises features first. Gaussian Naive Bayes models each feature per class with a normal distribution. Logistic Regression is a multinomial model with standardised inputs and up to 5,000

iterations. All use a fixed random seed of 42 for reproducibility. Let p_m(c | x) be the probability that learner m assigns to crop c. With M = 4 equally weighted learners, the ensemble probability is

$$P(c \mid x) = (1/M) \Sigma_m p_m(c \mid x) \quad (1)$$

and the recommended crop is

$$\hat{c} = argmax_c P(c \mid x) \quad (2)$$

Soft voting averages the predicted class probabilities from all four equally weighted learners. The final probability distribution represents their combined predictions and is used to rank the crops. These probabilities are displayed as probability-based confidence scores in the user interface; however, they are not guaranteed to be statistically calibrated.

The service sorts P(c | x) in descending order and returns the first three crops, with percentage values rounded to two decimals. The first is labelled the recommendation, the other two alternatives. The trained ensemble is serialised once with joblib and loaded at server start; if loading fails, prediction requests raise an explicit error instead of returning a default crop.

D. Rule-Based Fertiliser Advisory

The fertiliser module compares each measured nutrient with a fixed lower bound and reports its status. A nutrient is flagged when

$$N<50 \ \vee \ P<40 \ \vee \ K<40 \quad (3)$$

and the message for that nutrient states that its requirement is high. If none is flagged the user is told that NPK levels are within the defined range. These are project-defined decision thresholds, not agronomic dose recommendations. They identify which nutrient needs attention and defer actual fertiliser quantities to a soil-test report. Although crop-specific nutrient rules are available separately in the project repository, they are not currently integrated into the Flask application. Therefore, the present system provides only general NPK deficiency alerts and does not generate crop-specific fertiliser quantities or dosage recommendations. The thresholds are applied to the dataset-specific numerical NPK values used by the system and are intended only for identifying potential nutrient deficiencies within the prototype; they do not represent physical fertiliser quantities or validated agronomic soil-test limits.

E. Condition Alerts

The alerts page scans the ten most recent readings and raises a warning for each of the following conditions: soil moisture below 30%, pH below 5.5 (acidic soil), pH above 8.0 (alkaline soil) and temperature above 35 °C. Because these checks are plain comparisons, each alert can be traced to the exact reading and threshold that caused it, and none depends on the machine-learning model.

F. Sensor Simulation

The simulator generates sensor readings at 30-second intervals and inserts them into the MySQL database. The simulated readings are used to demonstrate the data flow between the sensing layer, database and dashboard. Since the readings are generated through software simulation rather than collected from physical agricultural sensors, they do not represent actual field measurements. The sensor simulation is intended to demonstrate the software workflow and should not be considered field validation.

TABLE II
SIMULATED SENSOR RANGES

Parameter	Range	Parameter	Range
Nitrogen	40–120	Humidity (%)	50–90
Phosphorus	20–80	Soil moisture (%)	20–80
Potassium	20–100	Soil pH	4.5–9.0
Temperature (°C)	20–40	Rainfall (mm)	50–200

G. Evaluation Protocol

The data are split 80/20 with stratification on the crop label, giving 1,760 training and 440 test records, with random seed 42. Each base learner and the ensemble are fitted on the training portion and scored on the test portion. Reported measures are accuracy, macro-averaged precision, recall and F1-score, and top-3 accuracy, which counts a prediction as correct if the true crop is among the three most probable. Five-fold stratified cross-validation on the full dataset is reported as a second estimate to reduce dependence on a single split. All values reported in Section IV were computed by re-running the project's own training configuration on the supplied dataset.

The hold-out evaluation and five-fold stratified cross-validation are conducted as separate evaluation procedures. The hold-out results are obtained using an 80:20 stratified train-test split, while five-fold cross-validation provides an additional estimate of model performance across different data partitions. The five-fold accuracy values reported in Table III represent the average accuracy across the five folds.

IV. RESULTS AND DISCUSSION

A. Classifier Performance

Table III compares the four base learners and the ensemble on the 440-record hold-out set and under five-fold cross-validation. All models exceed 97% accuracy.

TABLE III
HOLD-OUT AND CROSS-VALIDATED PERFORMANCE (%)

Model	Acc.	Prec.	Rec.	F1	Top-3	5-fold
Logistic Reg.	97.27	97.40	97.27	97.25	100	97.09
SVM (RBF)	98.41	98.56	98.41	98.40	100	98.36
Random Forest	99.55	99.57	99.55	99.55	100	99.59
Naive Bayes	99.55	99.59	99.55	99.54	100	99.45
Soft voting	99.32	99.37	99.32	99.32	100	99.41

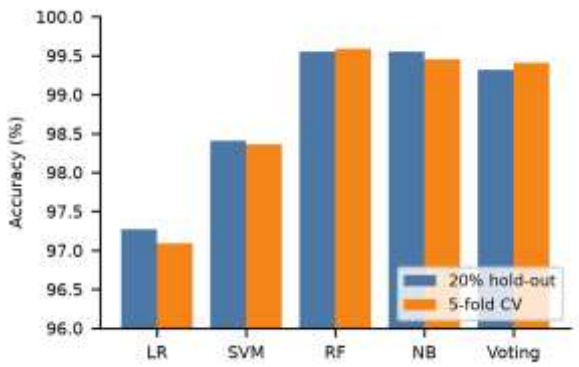


Fig. 2. Accuracy of each model on the hold-out set and under five-fold cross-validation.

Random Forest and Naive Bayes lead on the hold-out set at 99.55%, equal to two misclassified records of 440, and the soft-voting ensemble follows at 99.32% (three errors). The 0.23-point gap is one record and falls within split-to-split variation:

under five-fold cross-validation the ensemble scores 99.41% against 99.59% for Random Forest and 99.45% for Naive Bayes. The data therefore do not show that voting beats its best member on this benchmark. The ensemble's value is different: it does not achieve the highest accuracy on this benchmark, but its performance remains close to that of the best individual models. It also supplies a probability vector that combines the predictions of four different learning algorithms, which is the quantity displayed in the interface.

. Logistic Regression is weakest at 97.27% because crop classes are separated by non-linear combinations of rainfall, humidity and nutrients that a linear boundary cannot capture fully.

Top-3 accuracy is 100% for every model on the hold-out benchmark dataset, indicating that the actual crop label appears among the three highest-ranked predictions for all test samples. This supports the top-k recommendation approach by providing alternative crop options when the highest-ranked prediction is incorrect. However, this result is specific to the curated benchmark dataset and does not guarantee the same performance under real-world agricultural conditions.

B. Why Accuracy Is So High

Accuracies near 99% should be read with care. The Crop Recommendation dataset is balanced, has no missing values and, as Table I shows, was assembled so that each crop occupies a distinct region of the feature space (for example, rice at high rainfall and humidity, chickpea at low rainfall and humidity). Models separate these regions easily, which explains why even a naive-Bayes classifier reaches 99.55%. Field data would be noisier: soil nutrients vary across a plot, sensors drift, and several crops are agronomically viable on the same soil. The reported results support the functionality of the implemented pipeline, including preprocessing, model training, model serialization, inference and top-k prediction. However, these benchmark results do not establish real-world agronomic effectiveness, field safety or crop recommendation accuracy under actual agricultural conditions. Further validation using locally collected sensor readings and agronomist-labelled field data is required before the system can be considered suitable for practical farm deployment.

C. System Behaviour

For the reference input N = 80, P = 40, K = 45, temperature 25.4 °C, humidity 78%, pH 6.5 and rainfall 120 mm, the service recommends jute (58.13%) with maize (11.05%) and banana (6.86%) as alternatives. The fertiliser module flags no nitrogen deficiency (80 is above 50) and no phosphorus deficiency (40 is not below 40), and potassium at 45 passes its bound of 40, so the advice states that NPK levels are within the defined range. For the dataset's first rice record (N = 90, P = 42, K = 43, 20.9 °C, 82% humidity, pH 6.5, 203 mm rainfall) the ensemble returns rice at 87.83%, followed by jute (10.68%) and coffee (0.38%), consistent with the label.

When the sensor simulator runs, the dashboard retrieves the latest available reading from the MySQL database and refreshes the crop recommendation using the stored values. The sensor status is determined based on the freshness of the latest reading rather than merely the existence of a database record. If no recent reading is available or the database is unreachable, the dashboard displays the sensor as offline. In such cases, any default or fallback values used for display are clearly labelled as simulated or default values and are not presented as live sensor measurements. This approach helps prevent users from confusing outdated or fallback data with actual field measurements.

D. Comparison of Approaches

TABLE IV
QUALITATIVE COMPARISON WITH COMMON PRACTICE

Aspect	Manual / lab test	Proposed system
Data freshness	Per season	Every 30 s
Crop choice	Experience	Top-3 ML, with confidence
Nutrient advice	Lab report	Instant rule check
Warnings	None	Moisture, pH, temp.
Explainability	Expert opinion	Rule trace + probabilities
Cost	Per test	Commodity hardware

Table IV presents a qualitative comparison between conventional manual or laboratory-based practices and the proposed system based on their functional characteristics. The comparison is intended to highlight differences in data availability, crop recommendation, nutrient advisory and condition monitoring. The entries are descriptive and are not derived from controlled field experiments or quantitative cost measurements. Therefore, they should not be interpreted as measured performance improvements or verified cost reductions.

E. Limitations

1. **Simulated sensors.** The stream is random and uncorrelated, so it exercises the data path but does not represent real field dynamics. No hardware calibration or drift is modelled.
2. **Benchmark data.** The dataset is curated and balanced. Real performance needs validation on locally collected, agronomist-labelled field records.
3. **Point-in-time recommendation.** The model sees one reading. It ignores season, crop history, market price and irrigation availability.
4. **Coarse fertiliser rules.** The thresholds are fixed across crops and give no quantities. Crop-specific rules are present in the repository but not yet connected to the application.
5. **Operational hardening.** Database credentials are set in source code and the Flask server runs in debug mode. These must move to environment configuration before any deployment.

V. CONCLUSION AND FUTURE WORK

This paper presented a Smart Precision Agriculture System that unifies a soil and weather data stream, a four-model soft-voting crop recommender, rule-based fertiliser advice and condition alerts behind a Flask and MySQL web application. On the Crop Recommendation dataset the ensemble reaches 99.32% hold-out accuracy, 99.41% cross-validated accuracy and 100% top-3 accuracy, and its top-three output with confidence scores lets a farmer weigh alternatives. The ensemble does not outperform Random Forest or Naive Bayes on this benchmark, and the benchmark itself is too separable to predict field performance. The contribution is a working, tested integration whose components can each be replaced with real data or better models.

Future work will (1) replace the simulator with physical NPK, moisture, pH and DHT-class sensors reporting over MQTT or HTTP; (2) retrain on regionally collected and agronomist-validated data and report per-class precision and recall; (3) connect live weather forecasts to supply rainfall and

temperature trends instead of a single reading; (4) add SHAP-based explanations [13] so each recommendation shows which nutrients and climate variables drove it; (5) connect the crop-specific CSV fertiliser rules and add dose guidance validated against state agricultural guidelines; (6) add irrigation scheduling from soil-moisture trends; and (7) add authentication, secret management and a production server before field deployment.

REFERENCES

[1] N. Zhang, M. Wang, and N. Wang, "Precision agriculture—a worldwide overview," Computers and Electronics in Agriculture, vol. 36, nos. 2–3, pp. 113–132, 2002, doi: 10.1016/S0168-1699(02)00096-0.

[2] A. McBratney, B. Whelan, T. Ancev, and J. Bouma, "Future directions of precision agriculture," Precision Agriculture, vol. 6, no. 1, pp. 7–23, 2005, doi: 10.1007/s11119-005-0681-8.

[3] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. H. D. N. Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3758–3773, Oct. 2018, doi: 10.1109/JIOT.2018.2844296.

[4] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E. H. M. Aggoune, "Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk," IEEE Access, vol. 7, pp. 129551–129583, 2019, doi: 10.1109/ACCESS.2019.2932609.

[5] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture, recent advances and future challenges," Biosystems Engineering, vol. 164, pp. 31–48, Dec. 2017, doi: 10.1016/j.biosystemseng.2017.09.007.

[6] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," Sensors, vol. 18, no. 8, Art. no. 2674, Aug. 2018, doi: 10.3390/s18082674.

[7] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.

[8] A. Ingle, "Crop Recommendation Dataset," Kaggle. Accessed: Nov. 1, 2026. [Online]. Available: https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

[9] T. G. Dietterich, "Ensemble methods in machine learning," in Multiple Classifier Systems, MCS 2000, Lecture Notes in Computer Science, vol. 1857, pp. 1–15. Berlin, Germany: Springer, 2000, doi: 10.1007/3-540-45014-9_1.

[10] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[11] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[12] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: https://www.jmlr.org/papers/v12/pedregosa11a.html

.

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4768–4777. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html