ISSN: 2994-032X

Research Article

International Journal of Preventive Medicine and Care

Machine Learning Based Crop Recommendation Using Tuned Classifiers and Ensemble Methods

Faiza Noushin Priya, Shahria Badhon, Mirza Md. Hasibul Hasan, Sraboni Ghosh Joya* and Mahfiza Akter

Computer Science & Engineering City University Dhaka, Bangladesh

***Corresponding Author**
Sraboni Ghosh Joya, Computer Science & Engineering City University Dhaka, Bangladesh.

Submitted: 2026, Sept 01 ; **Accepted:** 2026, Sept 30; **Published:** 2026, Oct 29

Citation: Priya, F. N., Badhon, S., Hasan, M. M. H., Joya, S. G., Akter, M. (2026). Machine Learning Based Crop Recommendation Using Tuned Classifiers and Ensemble Methods. *Int J Prev Med Care, 4*(2), 01-09.

Abstract

The choice of a suitable crop for a given situation is determined by soil and various climatic parameters. For choosing these crops, we commonly use machine learning models, but combining multiple models, called ensembling, and checking whether an ensemble model can beat one well-tuned model is still an important question. In this paper this is directly checked. We have chosen two public 'crop recommendations' datasets for comparison, and selected a balanced sample consisting of 22 crops and 2,200 records. Five models were trained under identical conditions; Logistic Regression, KNN, Random Forest, XG Boost, and Feedforward Neural Network. Next, we fine-tuned the hyperparameters of the top three models using Grid Search and incorporated them into the soft voting, hard voting and stacking ensembles. To avoid data leakage, we used train, validation and test set with a clean split that ensures at no point we had the same observations split between the three sets. The final results indicate that a well calibrated Random Forest model obtained the same accuracy as the three ensemble models. This demonstrates that if a single model is tuned right, this won't help performance if one introduces additional models on this dataset. This therefore provides a cost-effective, lightweight and easy-to-install answer without running redundant models side by side for precision farming and for automatic crop selection applications.

Keywords: Crop Recommendation, Machine Learning, Ensemble Learning, Random Forest, Hyperparameter Tuning, Precision Agriculture

1. Introduction

Selecting what to grow requires balancing soil chemistry (N, P, K, pH) and environment (humidity, temperature, rainfall) in complicated, non-linear ways, which is not very practical for a traditional intuition. Prior work has found good accuracy with machine learning for crop selection using soil and climatic data, though most often good accuracy is obtained with default hyperparameters on a single train-test split so there is no true sense of reliability (or generalizability) in the results. The task has been tackled by many classifiers, linear and deep neural networks to name a few, and not many classifiers have been compared on an empirical basis. It is typically by the case that ensembling is used only under the assumption that the combination of learners results in greater accuracy but not by a carefully optimized single learner. This leads to our main question: What happens when one starts to improve the ensembled classifiers optimally, does this lead to any measurable improvements or are just models reaching a limit of their capacity determined by the actual data? We explore this by systematically tuning any of the crop recommendation models and comparing its performance with ensembling.

This problem is important not just for correctness, though. Unsuitable inputs use and practices can waste inputs, harm soil health, and constrain productivity, especially for smallholder farmers who have limited flexibility to correct errors on their fares. So a good system will be conducive to good food security, sustainable use of land resources and rationalization of the allocation of resources. With the increasing maturity of precision agriculture, and the lack of high-end computing infrastructure in many locations, simplicity and deployability is just as important as accuracy, and the question is: Can a simple model perform a task as well as an ensemble? It is this

work that utilizes a benchmark dataset of 2200 data samples over 7 features and 22 crop categories. Five models were tested and optimized by using the Grid method and then ensembled in an appropriate way. The findings indicate that an efficient crop recommendation can be achieved with the simple model "Random Forest" and that complexity is not always necessary to provide high levels of efficiency, since only linear limits exist to achieving accuracy.

1.1. Related Works

It has been observed that ensemble-based are very effective for dealing with the complex soil and weather data, and have replaced the simpler set of algorithms such as KNN, Decision Trees, and SVM for precision agriculture systems so far. These simpler models are outperformed by methods like Random Forest, boosting models like XGBoost and CatBoost and neural networks [1-3]. For further pushing of accuracy, some recent studies stack or vote more than one type of model: Ahmed et al. added to the models overfitting controls used weighted averaging of regression model and tree model; stacked SMOTE balancing and multiple models [4-6]. Other efforts are focused on practical deployment with basic hardware, for example, the light weight neural networks introduced by for basic hardware, and hybrid GRU-SVR model for time varying weather by [7]. In conclusion, this body of work reveals that generally tree or boosting models perform better than traditional classifiers and tuning plus data balancing (such as SMOTE) can increase prediction stability [8].

Paper	Dataset	Model	Result	Limitations
Ahmed et al. (2025) [1]	Two crop dataset with 28242 records	Complex 18-model ensemble	Higher accuracy, lower overfitting	High computational cost
Hasan et al. (2023) [3]	Bangladesh farm data (1969-2021)	KNN + RF + Ridge (mixed)	Better than SVR, NB, CatBoost	Few crops; cost not checked
Mahesh & Soundra-pandiyan (2024) [5]	Rainfall, pesti-cide, and tem-perature data	CatBoost, Light-GBM, XGBoost	CatBoost worked best	Needs many set-tings tuned
Paithane (2023) [6]	Soil and crop data	DT, SVM, LR, RF, NB, XGBoost	Less overfitting; stable	Not flexible; un-even data
Ramzan et al. (2024) [7]	Live sensor + standard data	Tuned Artificial Neural Network (ANN)	99% accurate on both	Needs data scal-ing; hard to ex-plain
Senapaty et al. (2024) [9]	Dataworld & Odisha (246,000 records)	SMOTE + SGDC	100% accurate; high F1 score	Good accuracy but poor specificity
Romanuke et al. (2019) [8]	Farm sensor data	Simple neural networks	Fast; works on basic devices	Too simple; basic features only
Janani et al. (2026) [4]	Cauvery Delta (10 years data)	Hybrid GRU + SVR (SKRDX)	Predicts fast weather changes	Low accuracy when weather changes fast

Table 1: Comparison of Representative Prior Work in Crop Recommendation

However, the existing studies are not without their draw-backs. While often high accuracy is reported in many studies, minor performance problems such as a low specificity for minority crops are neglected, as in the case of just about 100% accuracy, but little specificity, that has been reported in [6]. Many ensemble methods, such as the Ahmed Ensemble of 18 models in are impractical for use due to expensive implementation in terms of number of classifiers [4]. A shallow model was selected for speeding up the processing, but was also known to sacrifice some of other attributes. used a shallow model, which emphasized speed at the expense of other attributes's "hybrid" model compromised with (relatively low) R2 quality because of the variation in climate [7,8]. Many studies also use just one train-test split instead of leakage-free cross-validations, so it is unclear if the models generalize, and the scaling of features in neural networks is a problem that makes them work more complicated than tree-based models. After the previous work, this remaining lack is here filled: it is investigated for complex ensembles, whether the whole ensemble can be made better than a good individual with proper tuning, using a systematic and direct comparison.

2. Methodology

2.1Overview

The proposed study is divided into the specified five stages, namely: dataset selection, data preprocessing, implementation of baseline models on five classifiers, hyperparameter tuning on top three models of each group using the Grid Search CV approach and ensemble construction. The overall workflow used in the process is shown in Figure. 1.

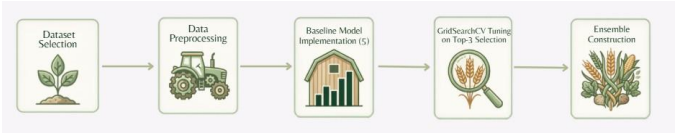


Figure 1: Proposed Architecture for Crop Recommendation System

2.2 Dataset Description and Selection

Two out of the public two crop datasets were tested which were acquired from Kaggle. The Ingle dataset has 2,200 rows evenly spread across 22 crop types, with seven features: Nitrogen (N), Phosphorus (P), Potassium (K), temperature, hu-midity, pH and rainfall [9]. The Garg dataset has 6,596 rows across 51 crop types with the same features, but not evenly distributed [10]. The data was selected from the Ingle dataset, due to the balance that would mean that the class size wouldn't affect the test and tuning of different models used on it. A comparison of both datasets is given in Table II.

Attribute	Ingle Dataset	Garg Dataset
Samples	2,200	6,596
Crop Classes	22	51
Features	7 (N, P, K, temp., humidity, pH, rainfall)	7 (N, P, K, temp., humidity, pH, rainfall)
Class Distribution	Balanced (100 samples/class)	Imbalanced

Table 2 : Comparison of the Ingle and Garg Datasets

- **Preprocessing**
  Only the training set was used for normalization of all the numerical features, at no time even data leakage was incorporated. Simple numbers were used for crop names. To ensure every dataset was transformed the exact same way, the data preparation and model training were combined into a single pipeline.

- **Baseline Model Implementation**
  In this proposed method, five machine learning models were used which are Logistic Regression, K-Nearest Neighbors, Random Forest, XGBoost and Feedforward Neural Network. They all underwent the same processing and were tested on the training set with the five fold cross validation method.

2.3 Logistic Regression: Logistic Regression is an easy linear model that estimates the probability of class as the result of a Sigmoid function applied to a linear combination of the input variables. It serves as a simple, interpretable to compare against more complex algorithms. The model architecture is illustrated in Figure. 2.

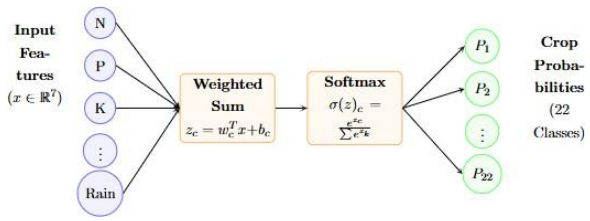


Figure 2: Logistic Regression architecture

2.4 K-Nearest Neighbors: KNN is an instance-based classification approach that uses the majority class of the k nearest neighbors of the sample to classify it. It is suitable for this project because it does not require any assumption of how the data is distributed and it captures local patterns of soil and climate data. The model architecture is illustrated in Figure. 3.

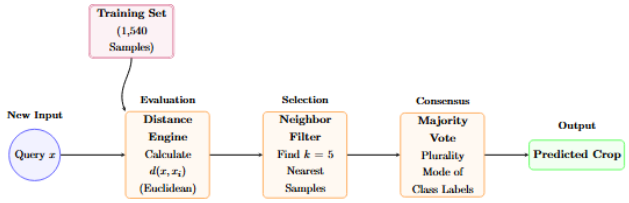


Figure 3: K-Nearest Neighbors Architecture

2.5 Random Forest: Random Forest is an ensemble of decision trees which improves the accuracy by using the majority voting principle for the final prediction. It has reduced over-fitting over a single decision tree and is able to accommodate non-linear soil and environmental data. The model architecture
is illustrated in Figure. 4.

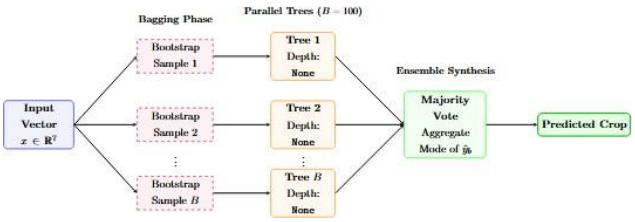


Figure 4: Random Forest architecture

2.6 XGBoost: XGBoost is a gradient boosting algorithm in which boosted decision trees are created one by one in se-quence with the aim of correcting errors of their predecessors. It can also be used to regularize in order to reduce the risk of overfitting and extract more complex patterns from the data set. The model architecture is illustrated in Figure. 5.

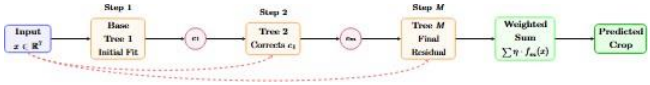


Figure 5: XGBoost Architecture

2.7 Feedforward Neural Network: Feedforward Neural Net-work is a network of connected neurons with weights and functions that process the input data. This is a framework that enables the network to learn non-linear features, which allows it to better capture non-linear relationships between soil and climate properties that can be weaker in simpler linear models. The model architecture is illustrated in Figure. 6.

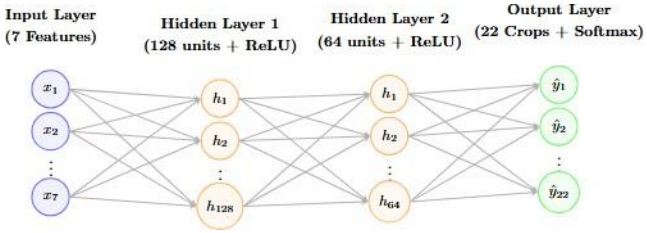


Figure 6: Feedforward Neural Network Architecture

3. Hyperparameter Tuning

To optimize the three models that were chosen, hyperparameter tuning was done on them using Grid Search CV with fivefold cross validation targeting weighted F1 score. For Random Forest, tree count, max depth, and split criteria were explored. Tuning of XG Boost involved in tuning and changing the number of trees, learning rate, and depth and subsample ratio. Adjusting hidden layer sizes, L2 regularization, and learning rates were done for the neural network. For each model, the best configuration was selected by taking the one that scored the best mean cross validation as shown in Equation (1).

$$\vartheta^* = \arg\max_{\vartheta \in \Theta} \frac{1}{K} \sum_{k=1}^{K} F1_w(\vartheta, D_k) \quad (1)$$

- **Ensemble Construction**

  Soft Voting, Hard Voting and Stacking are three types of ensemble strategies constructed from the tuned base learners. Soft Voting takes the predicted probabilities from each base model, multiplies them by individual weights w_i, and picks the class with the highest average probability, as defined in Equation (2).

$$\hat{y}_{soft} = \arg\max_{c} \sum_{i=1}^{n} w_i P_i(c) \quad (2)$$

Hard Voting looks at the final predicted class from each base model and chooses the class that gets the most votes, as shown in Equation (3).

$$\hat{y}_{hard} = \text{mode}\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\} \quad (3)$$

Stacking uses a Logistic Regression model as a meta-learner on the predictions of the base learners. Instead of using a simple voting or averaging rule, it learns how to combine the base models' outputs, as given in Equation (4)

$$\hat{y}_{stack} = g\left(f_1(X), f_2(X), \ldots, f_n(X)\right) \quad (4)$$

After setting up these ensembles, grid search was used to tune them. This included trying out different types of voting configurations in the voting configuration and different regularization strength for the Stacking meta-learner and tuning the components of the meta-learner accordingly for each of the methods.

- **Evaluation Metrics**

The performance of the model was assessed by accuracy, precision, recall and weighted F1 score on both validation and test sets.
The percentage of correct prediction for a class out of the total number of prediction made for a class is precision measure as shown in Equation (5).

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

Recall is defined as the ratio of the number of samples that have been correctly predicted out of the total number of

positive samples of a particular class, as defined in equation (6).

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

The weighted F1 score is simply the harmonic average of precision and recall values obtained per class, weighted by the number of samples in the class. It was adopted as the key criterion for model tuning and comparison in the presence of minor class imbalances occurred during cross validation as given in Equation (7).

$$F1_w = \sum_{c=1}^{C} \frac{n_c}{N} \cdot \frac{2 \cdot Precision_c \cdot Recall_c}{Precision_c + Recall_c} \quad (7)$$

4. Results and Discussion

4.1 Dataset Selection Outcome

From the two sets of data considered, the "balanced" 2,200 sample Ingle dataset (22 classes) was selected for the remaining experiments.

All results from here on are on the basis of this dataset. This is a comparison of the baseline models.
There was little difference on this dataset with tuning. Random Forest improved by 0.0004 F1 score and Neural Network by 0.0011 F1 score, and XGBoost decreased by 0.0031 F1 score after tuning. It is a small decrease that indicates the use of the default hyperparameters of XGBoost was near the optimum. In all, the results demonstrate that all three models now reached almost the ceiling of the dataset prior to tuning.

4.2 Baseline Model Comparison

The following models were tested as baseline models: Logistic Regression, K-Nearest Neighbors, Random Forest, XGBoost, and a Feedforward Neural Network. All models were preprocessed in the same way and used fivefold stratified cross validation. The performance metrics are shown in Table III and Figure. 7.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9955	0.9957	0.9955	0.9955
XGBoost	0.9932	0.9935	0.9932	0.9931
FFN	0.9909	0.9915	0.9909	0.9909
KNN	0.9795	0.9804	0.9795	0.9793
Logistic Regression	0.9727	0.9740	0.9727	0.9725

Table 3 : Baseline Model Performance on the Ingle Dataset

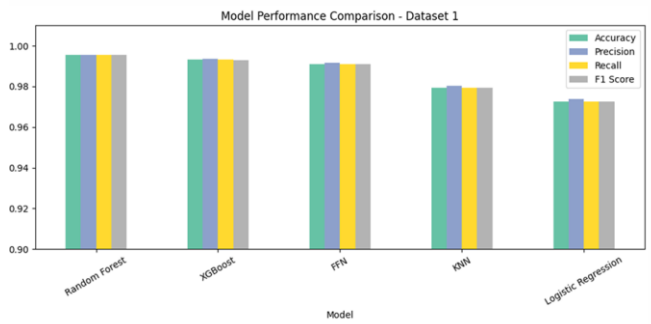


Figure7: Baseline Model Performance Comparison Across Four Metrics

Random Forest and XGBoost achieved best results with an average weighted F1-score of 0.9955 and 0.9931 respectively. Logistic Regression had the highest standard deviation across the folds (std = 0.0099), possibly due to lack of

(6) good performance with non-linear patterns. Based on these results, Random Forest, XGBoost and the Neural Network were chosen for hyper parameter tuning.

Model	Best Hyperparameters	CV F1
Random Forest	n_estimators=100, max_depth=None, min_samples_split=5, min_samples_leaf=1	0.9954
XGBoost	n_estimators=200, max_depth=6, learning_rate=0.3, subsample=1.0	0.9896
FFN (MLP)	hidden_layer_sizes=(64,), alpha=0.01, learning_rate_init=0.01	0.9870

Table 4: Best Hyperparameter Configurations Identified Via Gridsearchcv

Model	CV F1 (Before)	CV F1 (After)	Change
Random Forest	0.9950	0.9954	+0.0004
XGBoost	0.9927	0.9896	-0.0031
FFN (MLP)	0.9859	0.9870	+0.0011

Table 5 : Model Performance Before Vs. After Hyperparameter Tuning

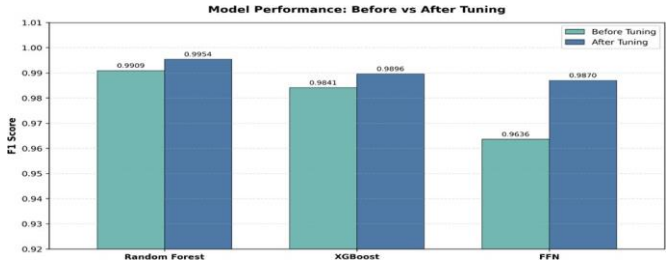


Figure 8: Before vs. After Hyperparameter Tuning Comparison

6. Ensemble Learning

The following three ensemble methods were built: Soft Voting, Hard Voting and Stacking (with Logistic Regression as the meta-learner). Before and after the ensemble tuning, the results of the performance are shown in Table VI and Fig. 9. In this configuration the term "before tuning" includes those ensembles where all base models had equal weights. After tuning" means the voting weights and the parameters of the meta-learner were tuned with the grid search.

Ensemble	F1 (Before)	F1 (After)	Change
Soft Voting	0.9932	0.9954	+0.0022
Hard Voting	0.9932	0.9954	+0.0022
Stacking	0.9932	0.9948	+0.0016

Table 6 : Ensemble Performance Before Vs. After Ensemble-Level Tuning

In addition, ensemble tuning was more beneficial than performance with single model tuning. Random Forest and XGBoost were more than the neural network—Soft Voting

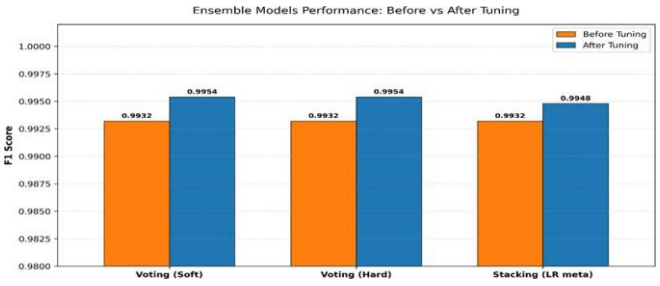


Figure 9: Before Vs. After Ensemble Tuning Comparison

improved by 0.0022 when these two models received double the weight of the neural network and equal weights were the best for Hard Voting. The empowerment of both voting schemes was 0.9954. Stacking was improved by 0.0016 after tuning its regularization to obtain a value of 0.9948. There were not many opportunities for improvement with the individual models, but if these were all combined with the optimum weighting, a fairly massive improvement was achieved whose parameters were definitely measurable.

In the Table VII and in the Fig. 10 the final rankings of all six tuned models are given. This is the three individual base models along with three ensemble strategies (all ordered by the cross validation weighted F1).

Rank	Model	CV Weighted F1
1	Random Forest (Tuned)	0.9954
1	Voting – Soft (2,2,1)	0.9954
1	Voting – Hard (1,1,1)	0.9954
4	Stacking (C=1.0)	0.9948
5	XGBoost (Tuned)	0.9896
6	FFN / MLP (Tuned)	0.9870

Table 7: Final Comparison of Tuned Single Models Vs. Tuned Ensembles

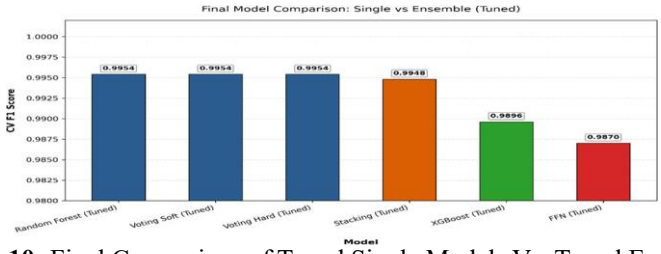


Figure 10: Final Comparison of Tuned Single Models Vs. Tuned Ensembles

After tuning, the model in Random Forest was the top ensemble models with an accuracy of 0.9954, this finding reveals that the combination of multiple models does not necessarily outperform the single well-tuned model. With stacking, the outcome was similar as the models at the bottom level predicted similar and there was little information for the meta-learner to exploit. There is a trade-off between model stability, accuracy, and simplicity: tuned voting ensembles are useful when even more stable predictions are sought after, but Random Forest is the model of choice when it is needed to be not only accurate, but also simple. With the correct modelling though, the gain from ensembling is almost negligible.

7. Conclusion

This study compared the two publicly available datasets for crop recommendation found on Kaggle and further polished them to a balanced 2200 samples in 22 classes for further development of models. Five baseline models were tested, and three of the highest performing (Random Forest, XGBoost, Feedforward NN) were tuned using grid search to create these ensembles: Soft Voting, Hard Voting, and Stacking. There was no gain in combining models as all the well-tuned ensembles had achieve the same best cross validation weighted F1 Score as the tuned Random Forest (0.9954). This work makes the main contribution: A cleaned version of how to compare single classifiers against ensembles to conclude that a well-tuned Random Forest is able to outperform more complex ensembles on a balanced agricultural dataset.

Future Work: These are the areas to be undertaken as future work based on our results:

- Final Model Selection: Considering both the tuned Ran-dom Forest and the tuned Voting ensemble, the equal performance, computational load, and ease of deployment of these models, select the one to be used.
- Extended Model Benchmarking: Try out some other algorithms such as Support Vector Machines, LightGBM, CatBoost and extend the pipeline comparisons.
- Real World Data Validation: Apply selected model to larger datasets of field collected data to test robustness in real world agricultural conditions and not clean data.
- Development: Create a simple, light application for farm-ers or local agriculture workers to use easily using the selected model.

Data Availability

The datasets generated during and/or analysed during the current study are available in the Kaggle repository, https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset
and https://www.kaggle.com/datasets/sashankgarg23/crop-dataset.

Acknowledgment

The authors thank the Department of Computer Science & Engineering, City University, Dhaka, Bangladesh, and grate-fully acknowledge the supervision of Sraboni Ghosh Joya throughout this study.

References

1. Paithane, P. M. (2023). Random forest algorithm use for crop recommendation. *ITEGAM-JETIA, 9*(43), 34-41.
2. Mahesh, P., & Soundrapandiyan, R. (2024). Yield prediction for crops by gradient-based algorithms. *Plos one, 19*(8), e0291928.

3. Ramzan, S., Ali, B., Raza, A., Hussain, I., Fitriyani, N. L., Gu, Y., & Syafrudin, M. (2024). An innovative artificial neural network model for smart crop prediction using sensory network based soil data. *PeerJ Computer Science, 10,* e2478.
4. Ahmed, R. A., El-Shafai, W., Ahmed, Z. A., El-Rabaie, E. S. M., & El-Samie, F. E. A. (2025). High-precision crop recommendation system with stacking ensemble classifiers for optimizing agricultural productivity. *Scientific reports, 15*(1), 43377.
5. Hasan, M., Marjan, M. A., Uddin, M. P., Afjal, M. I., Kardy, S., Ma, S., & Nam, Y. (2023). Ensemble machine learning-based recommendation system for effective prediction of suitable agricultural crop cultivation. *Frontiers in Plant Science, 14*, 1234555.
6. Senapaty, M. K., Ray, A., & Padhy, N. (2024). A decision support system for crop recommendation using machine learning classification algorithms. *Agriculture, 14(*8), 1256.
7. Janani, S. et al. (2026). "Dynamic Hybrid GRU-SVR Forecasting for Cauvery Delta Climatic Variations." Agricultural Systems.
8. Romanuke, V. et al. (2019). "Modified Shallow Architectures for Faster Agri-cultural Data Processing." International Journal of Computer Science.
9. Ingle, A. (2020). Crop recommendation dataset, *Kaggle*.
10. Garg, S. "Crop Dataset," *Kaggle.*
11. Bakthavatchalam, K., Karthik, B., Thiruvengadam, V., Muthal, S., Jose, D., Kotecha, K., & Varadarajan, V. (2022). IoT framework for measurement and precision agriculture: predicting the crop using machine learning algorithms. *Technologies, 10*(1), 13.
12. Shastri, S., Kumar, S., Mansotra, V., & Salgotra, R. (2025). Advancing crop recommendation system with supervised machine learning and explainable artificial intelligence. *Scientific Reports, 15*(1), 25498.
13. Baruah, P. J., Prasad, B., Nath, A. J., Paul, A., Paik, S., Raj, A., & Shridayal, V. S. (2026). A Crop Recommendation System Using Ensemble Learning Approach for Sustainable Farming. *DMPedia Lecture Notes in Computer Science & Engineering,* 77-90.
14. Thilakarathne, N. N., Bakar, M. S. A., Abas, P. E., & Yassin, H. (2022). A cloud enabled crop recommendation platform for machine learning-driven precision farming. *Sensors, 22*(16), 6299.
15. Khaliq, A., Khan, A., Jan, S., Umair, M., Gulshair, A., Ali, A., & Shah, U. A. (2025). Ai-driven smart agriculture: An integrated approach for soil analysis, irrigation, and crop-fertilizer recommendations. *IEEE access.*
16. Akkem, Y., Kumar, B. S., & Varanasi, A. (2023). Streamlit application for advanced ensemble learning methods in crop recommendation systems–a review and implementation. *Indian J. Sci. Technol, 16*(48), 4688-4702.
17. Gosai, D. (2021). Crop recommendation system using machine learning.
18. Apat, S. K., Mishra, J., Raju, K. S., & Padhy, N. (2023). An artificial intelligence-based crop recommendation system using machine learning. *Journal of Scientific & Industrial Research (JSIR), 82*(05), 558-567.
19. Bakr, M. A., Khan, A. J., Khan, S. D., Zafar, M. H., Ullah, M., & Ullah, H. (2025). Evaluation of learning-based models for crop recommendation in smart agriculture. *Information, 16*(8), 632.
20. Karthikeyan, M., Vignesh, T., & Sowmiya, R. (2021). Crop recommendation system using machine learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7*(3), 228-234.
21. Nawaz, M., Shafi, I., & Ahmad, J. (2022). Crop recommendation using gradient boosting and comparative analysis with classical machine learning models. *Computers and Electronics in Agriculture, 198*, 107091.
22. Bhosale, S., Patil, R., & Desai, A. (2023). Deep learning approaches for crop recommendation: A comparative study. *Journal of Agricultural Informatics, 14*(1), 1-15.
23. Kumar, R., Sharma, A., & Singh, P. (2022). A multi-model comparison for crop prediction using machine learning. *Sustainable Computing: Informatics and Systems, 35*, 100689
24. Dey, A., Mondal, S., & Roy, S. (2023). Feed-forward neural networks for agricultural crop classification: A comparative approach. *Neural Computing and Applications, 35*(4), 3201-3215.
25. Singh, R., Kumar, R., & Pandey, A. K. (2021). Multi-objective crop recommendation system using machine learning. *Journal of Agricultural Informatics, 12*(1), 1-12.
26. Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2022). An IoT based smart irrigation management system using Machine learning and open source technologies. *Computers and electronics in agriculture, 155*, 41-49.

Copyright: *©2026 Sraboni Ghosh Joya, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*