Scientific Research Publishing

Advances in Artificial Intelligence and Robotics Research, 2026, 2(4), 209-224
https://www.scirp.org/journal/airr
ISSN Online: 3143-3995
ISSN Print: 3143-3987

Machine Learning-Based Crop Recommendation Using Tuned Classifiers and Ensemble Methods

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

Department of Computer Science & Engineering, City University, Dhaka, Bangladesh
Email: faizaniha0062@gmail.com, hasibul5611@gmail.com, mahfizamunny@gmail.com, shahria.zularbine@gmail.com, sraboni.joya06@gmail.com

How to cite this paper: Priya, F.N., Hasan, M.Md.H., Akter, M., Badhon, S. and Joya, S.G. (2026) Machine Learning-Based Crop Recommendation Using Tuned Classifiers and Ensemble Methods. *Advances in Artificial Intelligence and Robotics Research*, **2**, 209-224.
https://doi.org/10.4236/airr.2026.24012

Received: September 18, 2026
Accepted: November 15, 2026
Published: November 18, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

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 compared two publicly available crop recommendation datasets, the Ingle dataset and the Garg dataset, and selected the balanced Ingle dataset, containing 22 crops and 2200 records, as the sole dataset. Five models were trained under identical conditions: Logistic Regression, KNN, Random Forest, XGBoost, 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 sets with a clean split that ensured at no point did we have the same observations split between the three sets. The final results indicate that a well-calibrated Random Forest model obtained the same weighted F1 score (0.9954) as the three ensemble models. This demonstrates that if a single model is tuned correctly, introducing additional models will not improve performance 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 traditional intuition. Prior work has found good accuracy with machine learning for crop selection using soil and climatic data [1]-[3]. However, many studies rely on default hyperparameters and a single train-test split, which gives little sense of reliability or generalizability. The task has been tackled by many classifiers, including boosted and tree-based models [4], deep neural networks [5], and comparisons of tree ensembles [6], yet few studies compare classifiers under systematic tuning. Ensembling is often used on the assumption that combining learners gives greater accuracy, rather than by comparing against a carefully optimized single learner [7] [8]. This leads to our main question: what happens when one systematically tunes the ensembled classifiers; does this lead to measurable improvements, or are the models simply reaching a limit set by the data? We explore this by systematically tuning each 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 [9] [10], especially for smallholder farmers who have limited flexibility to correct errors on their farms. 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 [11] [12], and the lack of high-end computing infrastructure in many locations [13], simplicity and deployability are just as important as accuracy, and the question is: Can a simple model perform a task as well as an ensemble?

This study evaluates models on the Ingle benchmark dataset [14], consisting of 2200 samples across 7 features and 22 crop categories. Five models were tested and optimized using the Grid method and then ensembled in an appropriate way. The findings indicate that 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.

2. Related Works

It has been observed that ensemble-based methods are very effective for dealing with 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 [15], boosting models like XGBoost and CatBoost [16], and neural networks [17]. For further improvement in accuracy, some recent studies stack or vote for more than one type of model: Ahmed *et al.* added overfitting controls to the models [18]; Hasan *et al.* used weighted averaging of a regression model and tree model [19]; Senapaty *et al.* stacked SMOTE balancing and multiple models [20]. Other efforts are focused on practical deployment with basic hardware, for example, a

low-cost IoT device that pairs soil sensors with a machine learning model for crop recommendation [13], and a hybrid GRU-SVR model for time-varying weather by Janani *et al.* [21]. 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. Table 1 summarizes the representative prior work in crop recommendation.

Table 1. Comparison of representative prior work in crop recommendation.

Paper	Dataset	Model	Result	Limitations
Ahmed *et al.* (2025) [18]	Two crop datasets with 28,242 records	Complex 18-model ensemble	Higher accuracy, lower overfitting	High computational cost
Hasan *et al.* (2023) [19]	Bangladesh farm data (1969-2021)	KNN + RF + Ridge	Better than SVR, NB, CatBoost	Few crops; costs not checked
Mahesh & Soundrapandiyan (2024) [16]	Rainfall, pesticide, and temperature data	CatBoost, LightGBM, XGBoost	CatBoost worked best	Needs many settings to be tuned
Paithane (2023) [15]	Soil and crop data	DT, SVM, LR, RF, NB, XGBoost	Less overfitting; stable	Not flexible; uneven data
Ramzan *et al.* (2024) [17]	Live sensor + standard data	Tuned Artificial Neural Network (ANN)	99% accurate on both	Needs data scaling; it is hard to explain
Senapaty *et al.* (2024) [20]	Dataworld & Odisha (246,000 records)	SMOTE + SGDC	100% Accurate; high F1 score	Good accuracy but poor specificity
Janani *et al.* (2026) [21]	Cauvery Delta (10 years' data)	Hybrid GRU + SVR (SKRDX)	Predicts rapid weather changes	Low accuracy when the weather changes rapidly

However, the existing studies are not without their drawbacks. While often high accuracy is often reported, minor performance problems such as low specificity for minority crops are neglected, as in the case of almost 100% accuracy but little specificity, which has been reported in Senapaty *et al.* [20]. Many ensemble methods, such as the 18-model ensemble of Ahmed *et al.* [18], are impractical for use due to expensive implementation in terms of the number of classifiers. Janani *et al.*'s "hybrid" model compromised (relatively low) R2 quality because of the variation in climate [21]. 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 their operation more complicated than tree-based models. Other work has addressed interpretability through explainable AI [22], evaluated a range of learning-based models [23], or treated recommendation as a multi-criteria problem [24] [25]. However, whether systematic tuning closes the gap between single classifiers and ensembles remains open. This study addresses that gap by investigating, through a systematic and direct comparison, whether a whole ensemble can outperform a well-tuned individual model.

3. Methodology

3.1. Overview

The proposed study is divided into the specified five stages, namely: dataset selec-

tion, data preprocessing, implementation of baseline models on five classifiers, hyperparameter tuning on the top three models of each group using the GridSearchCV approach, and ensemble construction. The overall workflow used in the process is shown in Figure 1.

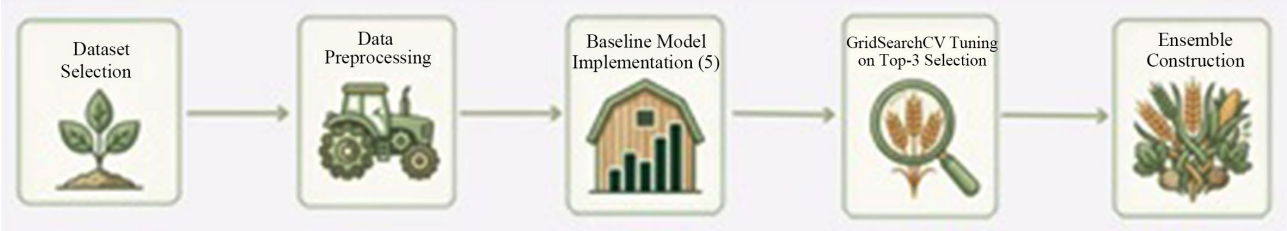


Figure 1. Proposed architecture for crop recommendation system.

All experiments were implemented using Python 3.13.15 with NumPy 2.1.3, Pandas 2.2.3, scikit-learn 1.6.1, and XGBoost 3.4.1.

3.2. Dataset Description and Selection

Two out of the publicly available two crop datasets were tested, which were acquired from Kaggle. The Ingle dataset [14] has 2200 rows evenly spread across 22 crop types, with seven features: Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall. The Garg dataset [26] has 6596 rows across 51 crop types with the same features, but the classes are not evenly distributed. The data was selected from the Ingle dataset due to the balance, which would mean that the class size would not affect the testing and tuning of different models used on it. A comparison of both datasets is given in Table 2.

Table 2. Comparison of the Ingle and Garg datasets.

Attribute	Ingle dataset	Garg dataset
Samples	2200	6596
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

3.3. Preprocessing

Only the training set (70%, 1540 samples) was used for normalization of all the numerical features. The validation set (20%, 440 samples) and test set (10%, 220 samples) were kept separate to avoid data leakage. The data were split using stratified sampling to preserve the class distribution, with a random seed of 42. Crop names were encoded as numerical labels. To ensure every dataset was transformed in exactly the same way, the data preparation and model training were combined into a single pipeline.

3.4. 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.

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

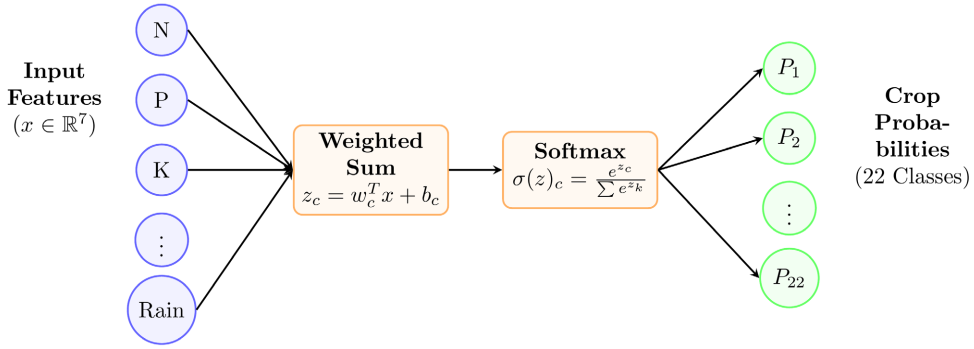


Figure 2. Logistic regression architecture.

2) 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 about how the data is distributed, and it captures local patterns of soil and climate data. The model architecture is illustrated in Figure 3.

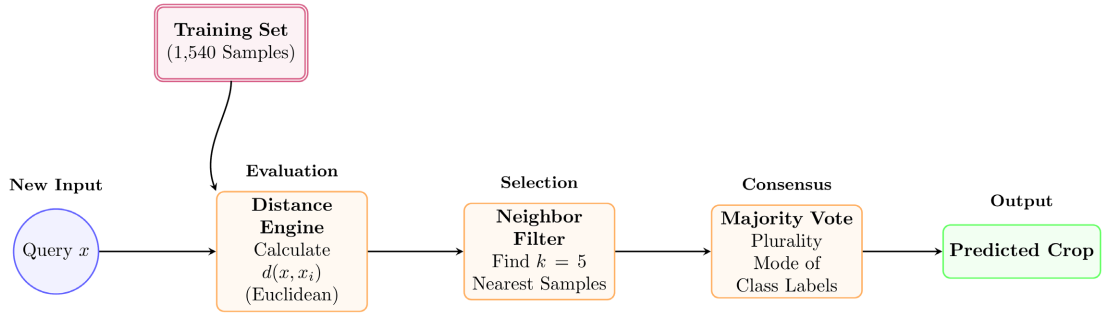


Figure 3. K-nearest neighbors architecture.

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

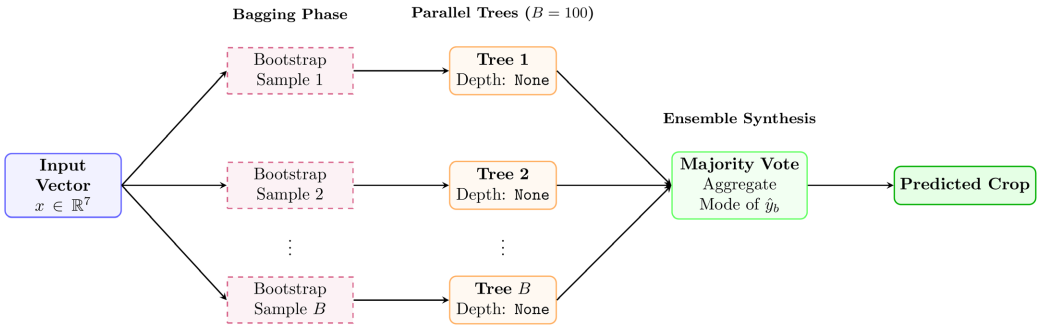


Figure 4. Random forest architecture.

4) XGBoost: XGBoost is a gradient boosting algorithm in which boosted decision trees are created one by one in sequence with the aim of correcting the 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**.

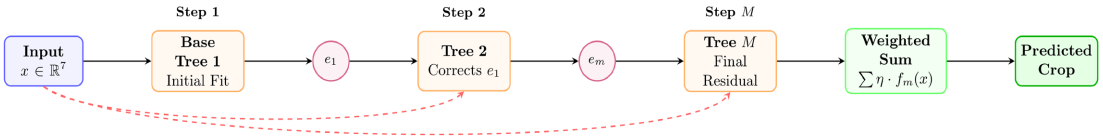


Figure 5. XGBoost architecture.

5) Feedforward Neural Network: A Feedforward Neural Network 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**.

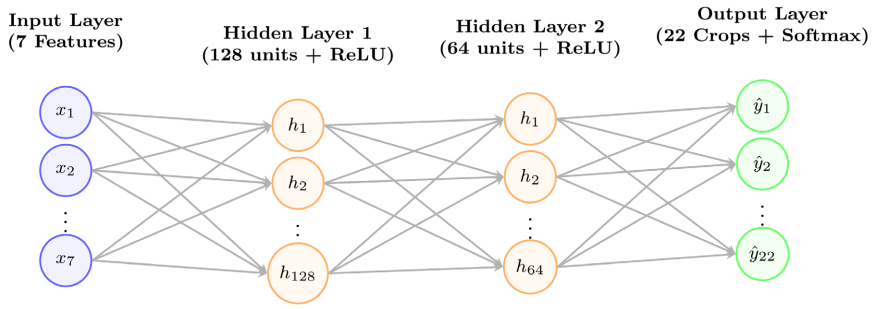


Figure 6. Feedforward neural network architecture.

3.5. Hyperparameter Tuning

To optimize the three models that were chosen, hyperparameter tuning was performed on them using GridSearchCV with five-fold cross-validation targeting the weighted F1 score. For Random Forest, tree count, max depth, and split criteria

were explored. Tuning of XGBoost involved tuning and changing the number of trees, learning rate, depth, and subsample ratio. Adjusting hidden layer sizes, L2 regularization, and learning rates was performed for the neural network. For each model, the best configuration was selected by taking the one that achieved the best mean cross-validation score as shown in Equation (1).

$$\boldsymbol{\theta}^{*} = \arg\max_{\theta\in\Theta}\left(\frac{1}{\mathbf{K}}\right)\sum_{\mathbf{k}=1}^{\mathbf{K}}\mathbf{F}1_{\mathbf{w}}\left(\boldsymbol{\theta},\mathbf{D}_{\mathbf{k}}\right) \tag{1}$$

3.6. Ensemble Construction

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 wi, and picks the class with the highest average probability, as defined in Equation (2).

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

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

$$\hat{y}_{\text{hard}} = \text{mode}\left\{\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_n\right\} \tag{3}$$

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).

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

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 strengths for the Stacking meta-learner, and tuning the components of the meta-learner accordingly for each of the methods.

3.7. 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 predictions for a class out of the total number of predictions made for a class is the precision measure, as shown in Equation (5).

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{5}$$

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).

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{6}$$

The weighted F1 score is simply the harmonic average of the precision and re-

call 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 that occurred during cross-validation, as given in Equation (7).

$$F_{1w} = \sum_{c=1}^{C} \left(\frac{n_c}{N} \right) \cdot 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{7}$$

4. Results and Discussion

4.1. Dataset Selection Outcome

From the two sets of data considered, the "balanced" 2200-sample Ingle dataset (22 classes) was selected for the remaining experiments. All results from here on are based on this dataset. This is a comparison of the baseline models.

4.2. Baseline Model Comparison

Table 3. Baseline model performance on the Ingle dataset.

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



Figure 7. Baseline model performance comparison across four metrics.

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 evaluated using five-fold stratified cross-validation on the training set. The 1540 training samples were divided into five stratified folds, with four folds used for training and the remaining fold used for validation in each iteration. This process was repeated five times so that each fold served as the validation fold once. The reported baseline perfor-

mance represents the average performance across the five folds. The separate validation set (440 samples) and test set (220 samples) were not used during this cross-validation. The performance metrics are shown in Table 3 and Figure 7.

Random Forest and XGBoost achieved the 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 a lack of good performance with non-linear patterns. Based on these results, Random Forest, XGBoost, and the Neural Network were chosen for hyperparameter tuning.

4.3. The Hyperparameter Tuning

The three highest baselines were selected and optimized further. GridSearchCV with five-fold stratified cross-validation was used to search for the best hyperparameters by maximizing the weighted F1 score. The search was performed using the training set only. The complete hyperparameter search spaces are presented in Table 4, while Table 5 lists the best configurations identified by GridSearchCV.

Table 4. Complete hyperparameter search spaces.

Model	Hyperparameter	Search values
Random Forest	n_estimators	100, 200
	max_depth	None, 10, 20
	min_samples_split	2, 5
	min_samples_leaf	1, 2
XGBoost	n_estimators	100, 200
	max_depth	3, 6, 9
	learning_rate	0.01, 0.1, 0.3
	subsample	0.8, 1.0
FFN (MLP)	hidden_layer_sizes	(64), (128, 64), (128, 64, 32)
	alpha	0.0001, 0.001, 0.01
	learning_rate_init	0.001, 0.01

Table 5. Best hyperparameter configurations identified via GridSearchCV.

Model	Best configuration	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

The FFN used ReLU activation, a maximum of 500 training iterations, and a random state of 42. Table 6 and Figure 8 illustrate the effect of the tuning. These present the cross-validation weighted F1 scores of each model before and after

hyperparameter tuning.

Table 6. Model performance before vs. after hyperparameter tuning.

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

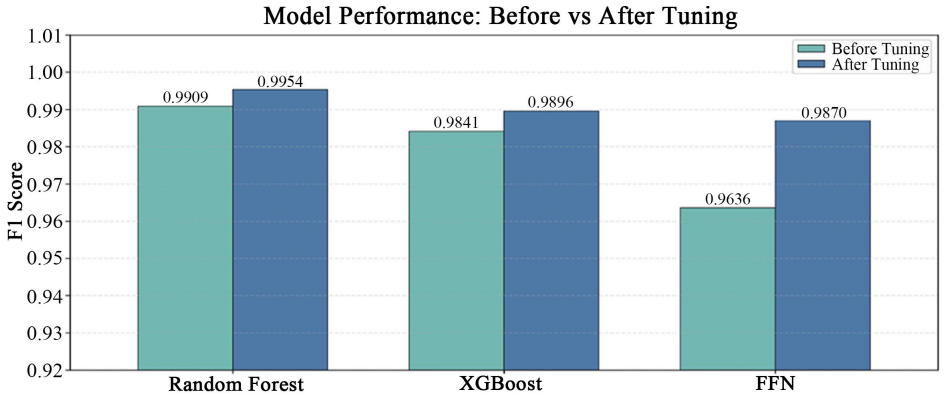


Figure 8. Before vs. after hyperparameter tuning comparison.

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. Overall, the results show that all three models were already performing near the upper range observed on this dataset before tuning.

4.4. Ensemble Learning

The following three ensemble methods were built: Soft Voting, Hard Voting, and Stacking (with Logistic Regression as the meta-learner). For stacking, we employed a careful approach to prevent data leakage during meta-learner training. The base models (RF, XGBoost, and FNN) were trained using 5-fold cross-validation, where predictions on held-out folds in each iteration served as training data for the Logistic Regression meta-learner. This out-of-fold approach ensured that the meta-learner was trained using predictions generated from held-out samples rather than predictions from samples used to train the corresponding base models. Once trained, the meta-learner combined base model predictions on the final test set. Before and after the ensemble tuning, the results of the performance are shown in Table 7 and Figure 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.

For Soft Voting, the weight combinations [1, 1, 1], [2, 1, 1], [1, 2, 1], [1, 1, 2],

[2, 2, 1], and [3, 2, 1] were evaluated using GridSearchCV. For Stacking, the Logistic Regression metalearner was tuned using C values of 0.1, 1.0, and 10.0, and the passthrough option was evaluated as False and True. Both searches used 5-fold cross-validation and the weighted F1 score as the scoring metric.

Table 7. Ensemble performance before vs. after ensemble-level tuning.

Ensemble	CV F1 (before)	CV 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

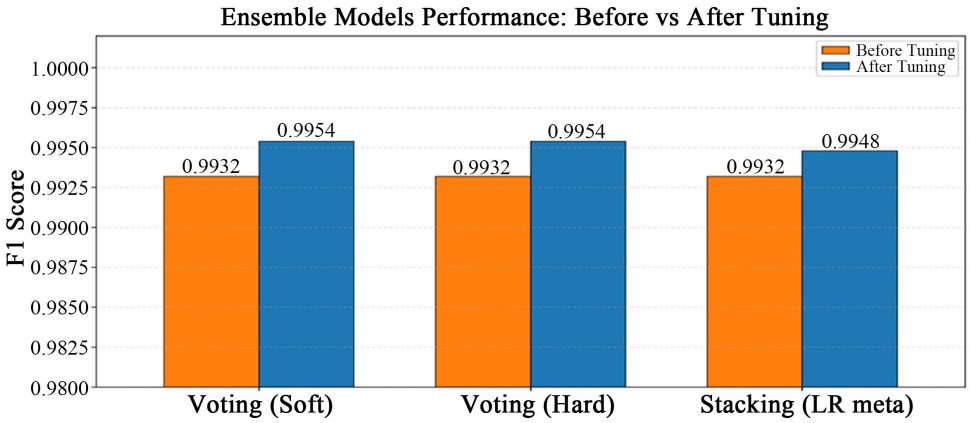


Figure 9. Before vs. after ensemble tuning comparison.

Table 8. Final comparison of tuned single models vs. tuned ensembles.

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

In addition, ensemble tuning was more beneficial than performance with single model tuning. Random Forest and XGBoost outperformed the neural network—Soft Voting 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 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 significant improvement was achieved, whose parameters were definitely measurable. In Table 8, and in Figure

10, the final rankings of all six tuned models are given. This includes the three individual base models along with three ensemble strategies (all ordered by the cross-validation weighted F1).

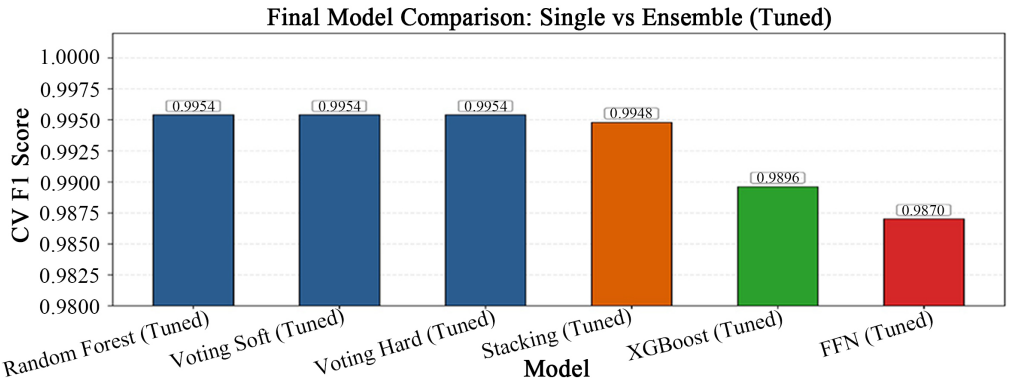


Figure 10. Final comparison of tuned single models vs. tuned ensembles.

After tuning, the Random Forest and the voting ensembles achieved the highest cross-validation weighted F1 score of 0.9954. This finding reveals that the combination of multiple models does not necessarily outperform a single well-tuned model. With stacking, the outcome was similar, as the models at the bottom level predicted similarly 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 correct modelling, however, the gain from ensembling is almost negligible.

To further evaluate the generalization performance of the final selected models, each model was evaluated once on the untouched test set containing 220 samples. The test set was not used during model or ensemble selection. The final test-set performance is reported in Table 9.

Table 9. Final performance on the untouched test set.

Model	Accuracy	Precision	Recall	Weighted F1
RF (tuned)	1.0000	1.0000	1.0000	1.0000
XGB (tuned)	0.9864	0.9876	0.9864	0.9863
FFN (tuned)	0.9909	0.9917	0.9909	0.9909
Soft Vote	0.9955	0.9959	0.9955	0.9954
Hard Vote	1.0000	1.0000	1.0000	1.0000
Stacking	1.0000	1.0000	1.0000	1.0000

On the untouched test set, Random Forest, Hard Voting, and Stacking achieved 1.0000 across all reported metrics. Soft Voting achieved a weighted F1 score of 0.9954, while FFN and XGBoost achieved weighted F1 scores of 0.9909 and 0.9863, respectively. These results provide an independent evaluation of the final selected

models after the model and ensemble selection process.

5. Conclusions

This study compared two publicly available crop recommendation datasets from Kaggle and selected the Ingle dataset as the sole dataset for model evaluation, containing 2200 samples across 22 classes as a balanced benchmark. 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 achieved the same best cross-validation weighted F1 Score as the tuned Random Forest (0.9954). The main contribution of this work is a systematic comparison of single classifiers against ensembles, demonstrating that a well-tuned Random Forest can match complex ensembles on the balanced Ingle dataset used as the evaluation benchmark.

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

- Final Model Selection: Considering both the tuned Random Forest and the tuned Voting ensemble, as well as 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, and CatBoost, and extend the pipeline comparisons.
- Real World Data Validation: Apply the 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, lightweight application for farmers or local agricultural workers to use easily with the selected model.

Data Availability

The datasets analyzed during the current study are publicly available in the Kaggle repository. The primary dataset used for model development is available at https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset [14]. The comparison dataset is available at https://www.kaggle.com/datasets/sashankgarg23/crop-dataset [26].

Acknowledgements

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

Author Contributions

Conceptualization, S.G.J., F.N.P., M.M.H.H., M.A. and S.B.; methodology, F.N.P.; software, F.N.P.; validation, F.N.P. and M.M.H.H.; formal analysis, M.M.H.H.

and S.B.; investigation, M.A.; resources, S.B. and M.A.; data curation; writing original draft preparation, F.N.P., M.A., S.B. and M.M.H.H.; writing review and editing, S.G.J. and F.N.P.; visualization, F.N.P.; supervision, S.G.J.; project administration, S.G.J.; funding acquisition, F.N.P., M.M.H.H., M.A., S.B. and S.G.J. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Gosai, D., Raval, C., Nayak, R., Jayswal, H. and Patel, A. (2021) Crop Recommendation System Using Machine Learning. *International Journal of Scientific Research in Computer Science*, *Engineering and Information Technology*, **7**, 558-569. https://doi.org/10.32628/cseit2173129

[2] Apat, S.K., Mishra, J., Raju, K.S. and Padhy, N. (2023) An Artificial Intelligence-Based Crop Recommendation System Using Machine Learning. *Journal of Scientific & Industrial Research*, **82**, 558-567.

[3] Afzal, H., Amjad, M., Raza, A., Munir, K., Villar, S.G., Lopez, L.A.D., *et al.* (2025) Incorporating Soil Information with Machine Learning for Crop Recommendation to Improve Agricultural Output. *Scientific Reports*, **15**, Article No. 8560. https://doi.org/10.1038/s41598-025-88676-z

[4] Shams, M.Y., Gamel, S.A. and Talaat, F.M. (2024) Enhancing Crop Recommendation Systems with Explainable Artificial Intelligence: A Study on Agricultural Decision-Making. *Neural Computing and Applications*, **36**, 5695-5714. https://doi.org/10.1007/s00521-023-09391-2

[5] Venkateswara, S.M. and Padmanaban, J. (2025) Interpretable Deep Learning Models for Independent Fertilizer and Crop Recommendation. *Scientific Reports*, **15**, Article No. 41721. https://doi.org/10.1038/s41598-025-26910-4

[6] Yadav, S., Singh, H.K., Garg, A. and Khan, M.M. (2026) Comparative Analysis of Machine Learning Approaches for Crop Recommendation in Sustainable Agriculture in India. In: *Atlantis Highlights in Intelligent Systems*, Atlantis Press International BV, 578-586. https://doi.org/10.2991/978-94-6239-693-7_57

[7] Acharya, N., Khatiwada, P., Pandey, R., Niroula, S. and Chapagain, P. (2024) Crop Recommendation System Using Machine Learning: A Comparative Study. *International Journal on Engineering Technology*, **1**, 302-311. https://doi.org/10.3126/injet.v1i2.66708

[8] Rinayat, B.S. and Dharaskar, B. (2025) A Robust Ensemble Approach for Crop Recommendation Based on Environmental Parameters. *EPJ Web of Conferences*, **328**, Article 01024. https://doi.org/10.1051/epjconf/202532801024

[9] Khaliq, A., Khan, A., Jan, S., Umair, M., Gulshair, A., Ali, A., *et al.* (2025) AI-Driven Smart Agriculture: An Integrated Approach for Soil Analysis, Irrigation, and Crop-Fertilizer Recommendations. *IEEE Access*, **13**, 141124-141138. https://doi.org/10.1109/access.2025.3594162

[10] Goap, A., Sharma, D., Shukla, A.K. and Krishna, C.R. (2018) An IoT Based Smart Irrigation Management System Using Machine Learning and Open Source Technologies. *Computers and Electronics in Agriculture*, **155**, 41-49. https://doi.org/10.1016/j.compag.2018.09.040

[11] Bakthavatchalam, K., Karthik, B., Thiruvengadam, V., Muthal, S., Jose, D., Kotecha, K., *et al.* (2022) IoT Framework for Measurement and Precision Agriculture: Predicting the Crop Using Machine Learning Algorithms. *Technologies*, **10**, Article 13. https://doi.org/10.3390/technologies10010013

[12] Thilakarathne, N.N., Bakar, M.S.A., Abas, P.E. and Yassin, H. (2022) A Cloud Enabled Crop Recommendation Platform for Machine Learning-Driven Precision Farming. *Sensors*, **22**, Article 6299. https://doi.org/10.3390/s22166299

[13] Islam, M.R., Oliullah, K., Kabir, M.M., Alom, M. and Mridha, M.F. (2023) Machine Learning Enabled IoT System for Soil Nutrients Monitoring and Crop Recommendation. *Journal of Agriculture and Food Research*, **14**, Article 100880. https://doi.org/10.1016/j.jafr.2023.100880

[14] Ingle, A. (2020) Crop Recommendation Dataset. Kaggle. https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

[15] Paithane, P.M. (2023) Random Forest Algorithm Use for Crop Recommendation. *Journal of Engineering and Technology for Industrial Applications*, **9**, 34-41. https://doi.org/10.5935/jetia.v9i43.906

[16] Mahesh, P. and Soundrapandiyan, R. (2024) Yield Prediction for Crops by Gradient-Based Algorithms. *PLOS ONE*, **19**, e0291928. https://doi.org/10.1371/journal.pone.0291928

[17] Ramzan, S., Ali, B., Raza, A., Hussain, I., Fitriyani, N.L., Gu, Y., *et al.* (2024) An Innovative Artificial Neural Network Model for Smart Crop Prediction Using Sensory Network Based Soil Data. *PeerJ Computer Science*, **10**, e2478. https://doi.org/10.7717/peerj-cs.2478

[18] Ahmed, R.A., El-Shafai, W., Ahmed, Z.A., El-Rabaie, E.M. and El-Samie, F.E.A. (2025) High-Precision Crop Recommendation System with Stacking Ensemble Classifiers for Optimizing Agricultural Productivity. *Scientific Reports*, **15**, Article No. 43377. https://doi.org/10.1038/s41598-025-09640-5

[19] Hasan, M., Marjan, M.A., Uddin, M.P., Afjal, M.I., Kardy, S., Ma, S., *et al.* (2023) Ensemble Machine Learning-Based Recommendation System for Effective Prediction of Suitable Agricultural Crop Cultivation. *Frontiers in Plant Science*, **14**, Article 1234555. https://doi.org/10.3389/fpls.2023.1234555

[20] Senapaty, M.K., Ray, A. and Padhy, N. (2024) A Decision Support System for Crop Recommendation Using Machine Learning Classification Algorithms. *Agriculture*, **14**, Article 1256. https://doi.org/10.3390/agriculture14081256

[21] Janani, K., Alageswaran, R. and Amirtharajan, R. (2026) Dynamic SG-SKRDX Hybrid Framework for Precision Weather Forecasting and Crop Suitability in the Cauvery Delta. *Scientific Reports*, **16**, Article No. 2042.

[22] Shastri, S., Kumar, S., Mansotra, V. and Salgotra, R. (2025) Advancing Crop Recommendation System with Supervised Machine Learning and Explainable Artificial Intelligence. *Scientific Reports*, **15**, Article No. 25498. https://doi.org/10.1038/s41598-025-07003-8

[23] Bakr, M.A., Khan, A.J., Khan, S.D., Zafar, M.H., Ullah, M. and Ullah, H. (2025) Evaluation of Learning-Based Models for Crop Recommendation in Smart Agriculture. *Information*, **16**, Article 632. https://doi.org/10.3390/info16080632

[24] Patel, K. and Patel, H.B. (2023) Multi-Criteria Agriculture Recommendation System Using Machine Learning for Crop and Fertilizers Prediction. *Current Agriculture Research Journal*, **11**, 137-149. https://doi.org/10.12944/carj.11.1.12

[25] Patel, K. and Patel, H.B. (2020) A State-of-the-Art Survey on Recommendation Sys-

tem and Prospective Extensions. *Computers and Electronics in Agriculture*, **178**, Article 105779. https://doi.org/10.1016/j.compag.2020.105779

[26] Garg, S. (2025) Crop Dataset. Kaggle. https://www.kaggle.com/datasets/sashankgarg23/crop-dataset