Volume 2, 2026, Pages 103-121 *Journal of Smart and Sustainable Farming*
| Regular Article | Open Access

An Explainable Stacked Ensemble Learning Framework for Crop Recommendation Using Soil and Climatic Parameters

Fen Huang
Agronomy, National Taiwan University, Roosevelt Road, Taipei 10617, Taiwan.
yafenhua@ntu.edu.tw

Benjamin Hutchinson
Crop and Soil Sciences, Faculty of Agriculture, College of Agriculture and Natural Resources, Kumasi, Ghana.
anbenjamin@knust.edu.gh

Article Info
Journal of Smart and Sustainable Farming
https://www.ansispublications.com/jssf/jssf.html

© The Author(s), 2026.
https://doi.org/10.64026/JSSF/2026010

Received 12 February 2026
Revised from 26 March 2026
Accepted 20 May 2026
Available online 02 June 2026
Published by Ansis Publications.

Corresponding author(s):
Benjamin Hutchinson, Faculty of Agriculture, College of Agriculture and Natural Resources, Kumasi, Ghana.
Email: anbenjamin@knust.edu.gh

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Abstract – Choosing the right crop for a piece of land relies on soil nutrients, soil acidity and local climate and if the wrong crop is selected, the yield will not be as good and inputs will be wasted. In this paper, an explainable stacked ensemble learning (EELE) model is presented which recommends one of the 22 crops based on seven soil and climatic parameters, namely, Nitrogen, Phosphorus, Potassium, Temperature, Relative Humidity, Soil pH and Rainfall. The base layer consists of random forest, histogram-based gradient boosting and a support vector machine with an RBF kernel, with the out-of-fold class probabilities being combined by a meta-learner, a multinomial logistical-regression. Permutation importance is introduced to indicate which parameters are important for a recommended classification. The system is tested on a test dataset consisting of 2200 samples generated by simulation from the ranges of agronomic requirements and is compared against eight classifiers serving as baseline classifiers. The held-out test set shows that the proposed model achieves 94.55% accuracy and 94.56% macro average F1 score, surpassing all baseline models, while a macro averaged AUC of 0.9994. It ranks second to a tuned SVM (94.59%) under 5×2 repeated cross validation and an ablation study reveals that the Gradient Boosting member makes the least contribution. The most important inputs are humidity, nitrogen and phosphorus. Learning Curves, noise robustness and computational cost are also reported and it discusses the limitations of what can be concluded from simulated data.

Keywords – Crop Recommendation, Precision Agriculture, Stacked Ensemble, Random Forest, Gradient Boosting, Support Vector Machine, Explainable Machine Learning.

I. INTRODUCTION

Agriculture is a key sector of food security, livelihoods and economic development. Crop selection is one of the most critical decisions made during the growing season as it has been linked to productivity, use of resources and profitability. The type of soil, its nutrient content, pH, temperature, moisture and precipitation all play a role in determining the suitability of a crop. Farmers base these decisions on experience, indigenous knowledge, and agricultural extension services; however, the variations in soil properties and changes in climatic conditions can make crop selection difficult to determine. Precision agriculture has, therefore, attracted growing interest in machine learning (ML) to assist in various applications including prediction of crop yield, detection of plant diseases, irrigation management and assessment of soil fertility [1] and [2]. In this context, the problem of crop recommendation is a multi-class classification problem, where the input are measurements of the environment and soil and the output is a classification of the crop into a suitable class [3]. A good recommendation

system can help farmers to take data-driven decisions, while agricultural advisory services can provide uniform advices in various farming contexts.

Although significant advances have been made in the field of machine learning for agriculture, crop recommendation is still a difficult task to do reliably. Different crops can have similar environmental needs, and categorisation of crops can be dependent on the interaction between many input variables. The relationships can pose a challenge for conventional classifiers due to their assumptions and learning processes. Decision Trees can have large variance, Support Vector Machines (SVMs) can be sensitive to the features' scaling or the selection of the parameters, and boosting algorithms can be susceptible to noisy observations. Thus, a single classifier may not make consistent predictions in various evaluation situations. A possible approach is ensemble learning that combines complementary predictive models. Stacking, for example, uses multiple basic models to make a prediction and the secondary model extracts the differences in the classification behavior of the basic models. The stacking is effective, however, if the right meta-features are generated, they are not leaked and regularisation is used at training.

One of the other factors to consider is whether the agricultural recommendations are interpretable. Farmers and agri-practitioners need explanations that are significant in terms of the model's output to see how a recommended crop is affected by soil or climatic parameters. Explainability techniques like permutation feature importance can help highlight influential variables and make the data-driven decision support more explainable. Moreover, the ability to predict the accuracy of a prediction is not enough to prove it to be practically reliable. Also, the stability of performance, the probability quality, sensitivity to measurement noise, computational needs and statistical significance should be explored.

To overcome these issues, this research suggests a novel explainable stacked ensemble of Random Forest, Gradient Boosting and a kernel-SVM based on a radial basis function (RBF). The meta-learner is trained using out-of-fold probability predictions which limits information leakage and limits overfitting. The study is of four main contributions. At first, it provides a two-level stacking setup that has a mathematically defined probabilistic learning mechanism. Second, it is evaluated and contrasted with eight baseline classifiers using the hold-out evaluation and repeated 5×2 cross-validation methodology, with paired statistical testing. Thirdly, the role of individual base learners, the role of the meta-learner and the role of feature normalization are studied through ablation experiments. Fourth, a learning curve, robustness to measurement noise, crop-specific classification errors, permutation-based feature importance, and computational cost are analyzed. This study not only explores situations where the proposed ensemble is superior but also situations where other competing models are superior, rather than taking a stance that the former is always superior. This evaluation is comprehensive and is designed to provide a clear and repeatable framework to evaluate stacked ensemble learning for crop recommendation and to explore its use for agricultural decision-making support.

II. RELATED WORKS

Machine Learning for Crop Recommendation

In the field of agriculture, machine learning (ML) has proven to be a valuable tool for decision-making, especially in areas such as crop recommendation, yield forecasting, and resource optimization. The use of ML in agriculture is reviewed, focusing on the shift from traditional statistical methods to ensemble learning and deep learning techniques for detecting intricate interactions between environmental and soil metrics [4]. Most of the crop recommendation systems in early harvest used individual classifiers such as Decision Trees, K-Nearest Neighbours (KNN) and Naive Bayes, which were used for linking soil parameters and climate with appropriate crops. In subsequent studies, the focus was on ensembles with the majority voter to increase the prediction accuracy by combining the votes of many classifiers [5]. Yet, some studies do not consider probability calibration, robustness, statistical significance, and performance variation over different partitions of data beyond classification accuracy. Additionally, the limited interpretability can hinder the use of these systems in practice by all stakeholders involved in agriculture.

Ensemble Learning

Ensemble learning is the method of obtaining better prediction by combining several models that will have different strengths and errors. Random Forest uses the bootstrap sampling and random feature selection to build a large number of decision trees with low variance using aggregation of predictions [6]. Gradient Boosting is a method of creating a predictive model in an additive fashion using successive learners on the negative gradients of the loss function [7]. For structured and tabular data, scalable implementations such as XGBoost have shown to be effective [8]. The other line of thought is used by AdaBoost, which boosts the weight of the examples that were incorrectly classified by the previous classifiers, so that the next classifiers pay more attention to them [9].

Stacked generalization is a method proposed by Wolpert, which is based on a second level model that is trained on the predictions generated by the base learners, and that are heterogeneous [10] [11]. Stacking is also able to learn the contribution of individual learners to the final prediction, unlike simple majority voting. For it to work, it requires out-of-fold predictions during the training of the meta-learner, avoiding information leakage and overfitting. Another complementary learning strategy is to use Support Vector Machines (SVMs), which discover the decision surfaces that maximize the margin between classes, and use kernel functions to encode the relationships between the classes [12]. Although KNN is a non-parametric baseline model, it is useful since the prediction of this model relies on how similar

observations are in the feature space [13]. These complementary properties encourage the consideration of the evaluation of heterogeneous learners in a common stacking framework.

Explainability

To build confidence in the use of ML-based recommendations for agriculture, they need to be interpretable. SHapley Additive exPlanations (SHAP) measure the contribution of features with the principles of cooperative game theory, and deliver explanations at an individual prediction level and at an aggregate level [14]. In contrast, permutation importance has the advantage of being a model-agnostic feature that quantifies the decrease in predictive accuracy when one randomly permutes a feature [15]. If there is a significant drop in performance, the model is very dependent on that feature. Permutation importance can be computed for a single classifier and complex stacked ensembles, while model-specific explanation techniques can only be applied to individual classifiers. Therefore, it is used in this study to identify the significant soil and environmental factors and for a clear understanding of the crop recommendations.

Research Gap

Predictive studies of cropping recommendations often focus on prediction and give little attention to generalization, reliability, or practical robustness. More specifically, heterogeneous stacking (with a probabilistic meta-learner), systematic evaluation in the presence of noisy sensor readings, statistical significance testing, and component-wise ablation are still little explored in many of the reported experiments. Even if a train–test split is used to make comparisons, the results could be data-partition sensitive.

To overcome these drawbacks, the present study proposes a reproducible framework for evaluating the proposed stacked classifier, its constituent learners and competing models. The evaluation takes into account predictive performance, probability quality, robustness to measurement noise, size sensitivity, statistical significance, and ablation results. Cross-validation is also applied to test the performance stability across different partitions. In particular, when the proposed ensemble under specific evaluation conditions does not perform better than the other methods, the analysis also reports negative results. This way, the emphasis is on methodological transparency, balanced interpretation and reproducibility, rather than publishing a single value of accuracy.

III. PROBLEM FORMULATION

Let $D = \{(xi, yi)\}_{i=1}^{N}$ denote a labelled dataset, where $xi \in R^7$ contains the seven soil and climate measurements of sample i and $yi \in \{1, \dots, C\}$ is the crop label, with C = 22 crops. The goal is to learn a classifier $f: R^7 \to 1, \dots, C$ that minimizes the expected misclassification rate and also returns a class-probability vector, so that the user can see the second-best and third-best crops as well as the top recommendation.

Dataset Description

It is a crop-recommendation study based on a sample of 2,200 observations, collected from 22 different crops, namely rice, maize, chickpea, kidney beans, pigeon pea, moth beans, mung bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute and coffee. Each of the sample measures seven features: the soil ratio of nitrogen, phosphorus and potassium, the temperature (°C), the relative humidity (%), the soil pH and the rainfall (mm). The data set is simulated. It was derived from the approximate ranges of agronomic requirements for each crop, and is laid out like public crop-recommendation benchmarks.

The typical requirement values are the center of a multivariate Gaussian distribution with a spread that represents the natural variation in the field for each crop. There are four elements which provide a realistic structure. (a) There is negative correlation between temperature with humidity, positive correlation between humidity with rainfall, and positive correlation between nitrogen with phosphorus and positive correlation between potassium with nitrogen and phosphorus. A mix of two agro-ecological regimes, such as two cultivars or regions, with equal proportions, is used for (b) each crop. (c) The distribution of rainfall is right skewed. (d) Values are kept within physically realistic limits. The spread was "scaled" so that neighbouring crops overlap, and it is hard to split into a trivial set of tasks. The feature statistics are summarized in **Table 1**. A copy of the generator and experiment code are available upon request from the authors.

Table 1. Statistical Summary of the Dataset Features

Feature	Min	Max	Mean	Std. dev.
Nitrogen (N)	0	140	50.48	37.19
Phosphorus (P)	5	145	52.48	33.02
Potassium (K)	5	205	46.91	47.69
Temperature	10.59	40.15	25.63	4.82
Humidity	14	100	71.15	22.49
pH	4.18	8.78	6.46	0.69
Rainfall	20	300	108.94	58.92

Pre-Processing
The dataset has no missing values. Because the features are measured on very different scales, from pH values near 6 to rainfall values above 200 mm, every feature is rescaled to the interval [0, 1] by min-max normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{min} and x_{max} are computed on the training data only and re-used for test samples, which prevents information leakage. The data are then split with stratification into 1,760 training and 440 test samples, so that each crop contributes 80 and 20 samples respectively.

Class Balance and Data Checks
The dataset is perfectly balanced, with 100 samples per crop, so accuracy is a meaningful summary metric and no re-sampling is needed. Stratified splitting preserves this balance in training (80 per crop) and test (20 per crop) data. Duplicate rows, impossible values (negative nutrient levels, pH outside 0–14, humidity above 100%) and missing entries were checked for and not found.

IV. PROPOSED METHODOLOGY

Architecture Overview
Proposed framework is a stacked ensemble of two levels, as illustrated in **Fig. 1**. Three base learners are used for the inputs that perform different types of modeling on the data, namely bagged forest of trees, boosted additive model of trees and kernel-based margin classifier. Each class probability computed out-of-fold serves as input into a SoftMax meta-learner to make the final recommendation. A permutation-important module is then used to explain which inputs are important.

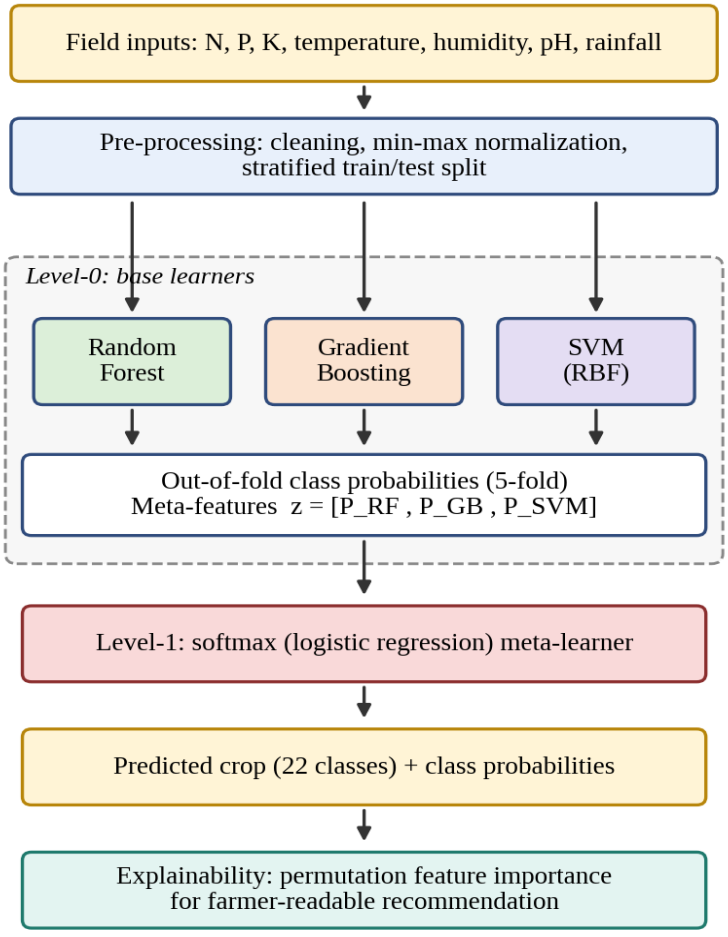


Fig 1. Architecture of the Proposed Explainable Stacked Ensemble for Crop Recommendation.

The selection of the base learners is based on the bias–variance paradigm on ensembles [17]. A Random Forest reduces variance by taking an average of a multitude of deep, decorrelated trees and does not scale down feature interactions.

Gradient Boosting is a technique that decreases bias by correcting errors made by shallow trees and often provides sharper decision boundaries on tabular data. The RBF-SVM learns a smooth, global boundary in the kernel space, and differs significantly from the two tree-based models. The models differ in their inductive biases so that errors made by the three models are expected to be only partially correlated, and as such stacking should work better than the individual models. To prevent over fitting the 66 meta-features, the SoftMax meta-learner is also maintained linear, and also so that their weights are interpretable.

Base Learner 1: Random Forest
A Random Forest grows *B* decision trees on bootstrap samples, choosing at each split the feature and threshold that most reduce the Gini impurity [6], [18]. For a node *t* with class proportions p_k,

$$\mathrm{G(t)} = 1 - \sum_{\mathrm{k=1}}^{\mathrm{C}} \mathrm{p_k^2} \tag{2}$$

where *C* is the number of classes. The forest prediction is the majority vote of the trees:

$$\hat{\mathrm{y}}(\mathrm{x}) = \arg\max_{\mathrm{c}} \sum_{\mathrm{b=1}}^{\mathrm{B}} \mathrm{I}[\mathrm{h_b}(\mathrm{x}) = \mathrm{k}] \tag{3}$$

where h_b is the *b*-th tree and $I[\cdot]$ is the indicator function (1 if true, 0 otherwise). The fraction of trees voting for each class is used as a probability estimate.

Base Learner 2: Gradient Boosting
Gradient boosting builds an additive model in stages [7]. At stage *m* a new regression tree h_m is added to the current model:

$$\mathrm{F_m(x)} = \mathrm{F_{m-1}(x)} + \nu\gamma_\mathrm{m}\mathrm{h_m(x)} \tag{4}$$

where *v* is the learning rate and γ_m is the step size. For multi-class problems with softmax probabilities, the tree at stage *m* is fitted to the pseudo-residuals, i.e., the negative gradient of the loss *L*,

$$\mathrm{r_{ik}^{(m)}} = -\frac{\partial \mathrm{L(y_i, F(x_i))}}{\partial \mathrm{F_k(x_i)}}\Big|_{\mathrm{F=F_{m-1}}} = \mathrm{y_{ik}} - \mathrm{p_{ik}^{(m-1)}(x_i)} \tag{5}$$

with y_{ik} the one-hot label and $p_{ik}^{(m-1)}$ the current probability of class *k*. A histogram-based implementation is used for speed.

Base Learner 3: Support Vector Machine
The SVM [12] separates classes with a maximum-margin boundary in a feature space induced by a kernel. Using the radial basis function (RBF) kernel, the decision function of a binary sub-problem is

where, $K(x_i, x) = \exp(-\gamma|x_i - x|^2)$, α_i are the dual coefficients, b is the bias and γ controls the kernel width. Multi-class prediction uses one-versus-one voting, and probabilities are obtained by Platt scaling.

$$\mathrm{f(x)} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \alpha_\mathrm{i} \mathrm{y_i K(x_i, x)} + \mathrm{b} \tag{6}$$

Stacking and Meta-Learner
In stacked generalization [10], [17] the second-level model must not see predictions that base learners made on their own training samples, otherwise it learns to trust over-fitted members. Therefore *K*-fold cross-validation ($K = 5$) is used: for each sample the probabilities come from base models that did not train on it. The meta-feature vector of sample i is

$$\mathrm{z_i} = \left[\mathrm{p_{RF}^{(-f)}(x_i)}, \mathrm{p_{GB}^{(-f)}(x_i)}, \mathrm{p_{SVM}^{(-f)}(x_i)}\right] \in \mathrm{R^{3C}} \tag{7}$$

where $\boldsymbol{p_h^{(-f)}}(x_i)$ is the probability vector of base learner h trained without the fold *f* that contains sample i, giving $3C = 66$ meta-features. A multinomial logistic-regression (SoftMax) meta-learner then estimates

$$\mathrm{P(\,y_i = c \mid z_i\,)} = \frac{\exp\left(\mathrm{w_c^T z_i + b_c}\right)}{\sum_{\mathrm{j=1}}^{\mathrm{C}} \exp\left(\mathrm{w_j^T z_i + b_j}\right)} \tag{8}$$

with weight vector $\boldsymbol{w}_c$ and bias b_c for crop *c*. The meta-learner is trained by minimizing the regularized cross-entropy

$$\mathcal{L}(\mathrm{W},\mathrm{b}) = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{y_{ic}} \log \mathrm{p_{ic}} + \frac{\lambda}{2}|\mathrm{W}|_{\mathrm{F}}^{2} \quad (9)$$

where y_{ic} is the one-hot label, $p_{ic} = P(y = c|\boldsymbol{z}_i)$, **W** collects all weight vectors and λ is the L2 regularization strength. A linear meta-learner is deliberately chosen: it has few parameters, is hard to over-fit, and its weights can be inspected.

Explainability by Permutation Importance
To explain the model, the importance of feature *j* is measured as the average drop in score when that feature is randomly shuffled [15]:

$$\mathrm{PI_j} = \mathrm{A(D)} - \frac{1}{\mathrm{R}}\sum_{\mathrm{r}=1}^{\mathrm{R}} \mathrm{A}\left(\mathrm{D}^{(\mathrm{j,r})}\right) \quad (10)$$

where $A(\cdot)$ is accuracy, D is the evaluation set, $D^{(j,r)}$ is D with column j permuted in repetition r, and $R = 30$ repetitions are used. A large drop means that the model relies strongly on the feature.

Algorithm and Flowchart
Algorithm 1 gives the training and prediction procedure of the proposed model, and **Fig. 2** shows the corresponding flowchart.

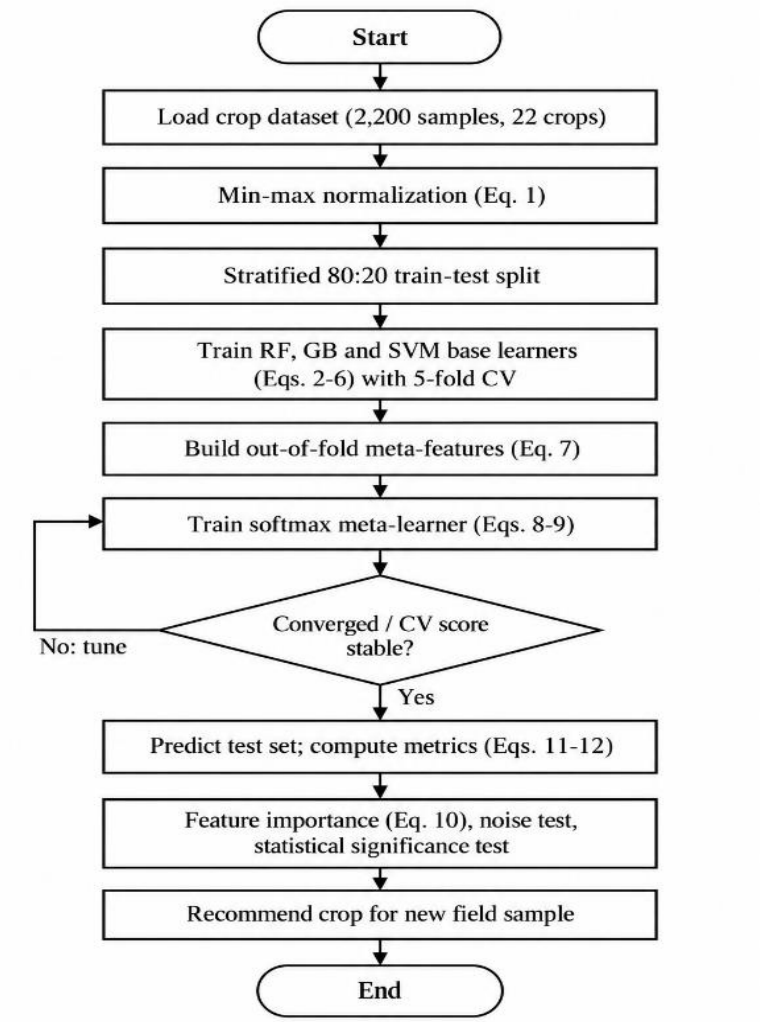


Fig 2. Flowchart of the Proposed Crop Recommendation Framework.

Choice of Meta-Learner and Hyper-Parameters
Several meta-learners are possible, including trees, neural networks and simple averaging. A regularized SoftMax regression is used for three reasons. It has $C(3C + 1) = 1{,}474$ parameters, which is not small relative to 1,760 training samples, but L2 regularization keeps over-fitting under control. It is trained by convex optimization and is therefore deterministic and fast. Its weights show how strongly each base model's class probabilities are used for each crop. Base-learner hyper-parameters were set to widely used values (**Table 2**) rather than searched, for two reasons: a search would

favour models with more tunable parameters, and the aim of the study is to compare model families under equal, modest effort.

Algorithm 1: Explainable stacked ensemble for crop recommendation
Input: training set D_{tr}, folds K = 5, base learners {RF, GB, SVM}
Output: trained stacked model; predicted crop for a new sample
1: Normalize D_{tr} with Eq. (1); store x_{min} and x_{max}
2: Split D_{tr} into K stratified folds F_1, …, F_K
3: for each base learner h in {RF, GB, SVM} do
4: for k = 1 to K do
5: Train h on D_{tr}/F_K
6: Predict probabilities $P_h(x_i)$ for x_i in F_k
7: end for
8: Re-train h on the whole D_{tr}
9: end for
10: Build meta-features z_i for all i using Eq. (7)
11: Train SoftMax meta-learner on (z_i, y_i) by minimizing Eq. (9)
12: for each new sample x do
13: Normalize x; obtain P_{RF}, P_{GB}, P_{SVM}; form z
14: Compute $P(y = c \mid z)$ by Eq. (8); return $argmax_c$
15: end for
16: Compute permutation importance on held-out data (Eq. 10)

Computational Complexity
Let T_{RF}, T_{GB} and T_{SVM} be the training costs of the base learners. Training the stacked model requires about $(K + 1)(T_{RF} + T_{GB} + T_{SVM})$ plus the cost of fitting a linear model on 66 meta-features, so training is roughly six times more expensive than training the three learners once. At prediction time the cost is the sum of three base predictions and one 66 × 22 matrix product, which remains small for a single field sample.

V. EXPERIMENTAL SETUP

Implementation
All experiments were conducted using Python (using the scikit-learn library version 1.8.0 [16] on a single-core CPU and with 4GB of RAM. Data splitting, model initialization and model cross validation were made reproducible by using the fixed random seed (42). The proposed model is compared with eight baselines namely: Logistic Regression (LR), Gaussian Naive Bayes (NB), K-nearest neighbours (KNN), Decision Tree (DT) [18], AdaBoost [9], SVM [12], Random Forest (RF) [6] and Gradient Boosting (GB) [7]. The same normalized inputs are used for all models. **Table 2** summarizes the key settings and no extensive hyper-parameter search was conducted for any model, to ensure a fair comparison between the models and enable the proposed model to stand out.

Table 2. Main Hyper-Parameters of the Models

Model	Settings
LR	Multinomial, L2 penalty, C = 10, L-BFGS
NB	Gaussian likelihood, variance smoothing 10^{-9}
KNN	k = 5, Euclidean distance, uniform weights
DT	Gini criterion, fully grown tree
AdaBoost	100 trees of depth 3
SVM	RBF kernel, C = 10, γ = scale, Platt scaling
RF	200 trees, Gini, √d features per split
GB	Histogram-based, 100 iterations, ν = 0.1, depth 4
Meta-learner	Multinomial LR, C = 1.0 (L2), max. 2,000 iterations
Stacking	Out-of-fold probabilities, K = 5 folds

Evaluation Protocol and Metrics
There are two complementary approaches taken to the evaluation of performance. These two models are independently trained from the 1,760 training samples and tested by the independent test set with 440 samples each. Secondly, 10-fold level evaluations are obtained from repeated stratified 5 × 2 cross validation. To evaluate predictive performance and

variability the mean and standard deviation of accuracy will be reported. The differences in performance between models are statistically evaluated using a paired t-test.

For crop class (c), precision and recall are calculated from the true-positive (TP_c), false-positive (FP_c), and false-negative (FN_c) counts as follows:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c+\text{FP}_c} \tag{11}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c+\text{FN}_c} \tag{12}$$

Precision, recall and F1 are macro-averaged over the 22 crops, so that every crop has equal weight. The area under the ROC curve (AUC) is macro-averaged in a one-versus-rest manner.

VI. RESULTS AND DISCUSSION

This section presents the results in the following order: exploratory analysis of the data, hold-out performance, per-crop errors, ROC analysis, cross-validation with significance testing, ablation, learning curves, noise robustness, feature importance, base-learner diversity, sensitivity analysis, a case study, probability quality, class overlap, per-crop comparison, computational cost, a summary of findings, and a discussion of limitations. Unless stated otherwise, the proposed model is the full stack of Random Forest, Gradient Boosting and SVM with the settings in **Table 2**.

Exploratory Analysis

Fig. 3 shows the Pearson correlation between features over the whole dataset. Most correlations are weak (absolute value below 0.2). The exception is the pair phosphorus and potassium ($r = 0.69$), which arises because crops such as grapes and apple have high requirements for both nutrients while most other crops have low requirements for both. This pair carries partly shared information, but because both features also differ across crops in other ways and tree and kernel models tolerate correlated inputs, all seven features are retained.

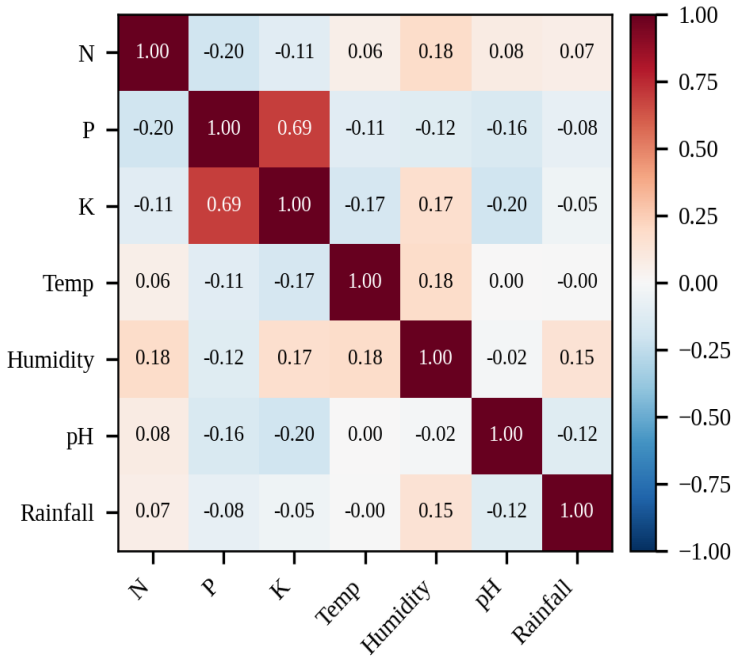


Fig 3. Pearson Correlation Matrix of the Seven Input Features.

Hold-Out Performance

Table 3 and **Fig. 4** compare all models on the 440 test samples. The proposed stacked model achieves the highest accuracy (94.55%), precision (94.75%), recall (94.55%), macro F1 (94.56%) and AUC (0.9994). It is 0.68 percentage points ahead of the best baseline, the SVM (93.86%), 1.36 points ahead of Random Forest and 2.27 points ahead of Gradient Boosting. The Decision Tree is weakest (85.23%), which confirms the high variance of a single tree, and AdaBoost with shallow trees (88.18%) struggles with 22 classes. A caution is needed: with 440 test samples the standard error of an accuracy near 94% is about 1.1 points, so the margin over SVM (three samples) lies within sampling noise. The cross-validation results in Section VII-E give a more reliable picture.

Table 3. Hold-Out Test Performance of All Models (%)

Model	Acc.	Prec.	Rec.	F1	AUC
LR	92.95	93.20	92.95	92.90	0.9979
NB	92.95	93.33	92.95	92.94	0.9984
KNN	93.18	93.49	93.18	93.15	0.9966
DT	85.23	85.93	85.23	85.10	0.9226
AdaBoost	88.18	89.02	88.18	88.30	0.9919
SVM	93.86	94.09	93.86	93.89	0.9993
RF	93.18	93.26	93.18	93.17	0.9989
GB	92.27	92.71	92.27	92.34	0.9985
Proposed	**94.55**	**94.75**	**94.55**	**94.56**	**0.9994**

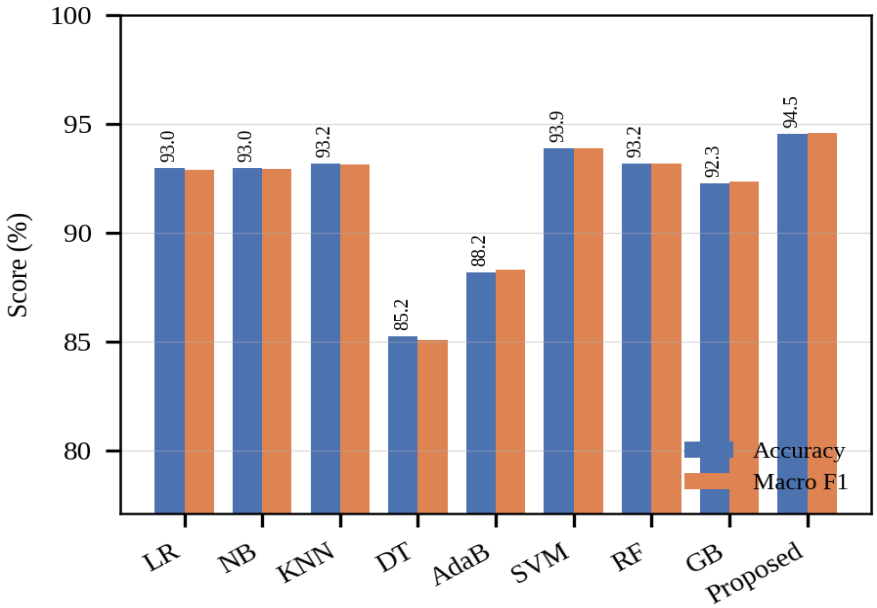


Fig 4. Accuracy and Macro F1-Score of All Models on the Test Set.

Confusion Matrix and Per-Crop Results

Fig. 5 shows the confusion matrix of the proposed model. Only 24 of 440 test samples are misclassified, and ten crops (banana, chickpea, coconut, cotton, kidney beans, mung bean, muskmelon, orange, papaya and pomegranate) are recognized without error. The largest confusion is between apple and grapes: four apple samples were labelled as grapes and three grape samples as apple. Both crops are grown with high phosphorus and potassium, and their temperature and humidity ranges overlap. Other errors occur between watermelon and muskmelon, lentil and moth beans, jute and rice, and black gram and maize, which are pairs with similar temperature or rainfall requirements. **Table 4** lists the weakest crops; the lowest F1-score is 82.1% (apple), and the mean F1 over all crops is 94.6%.

Table 4. Per-Crop Results of the Proposed Model: Eight Weakest Crops (%)

Crop	Precision	Recall	F1-score
apple	84.2	80.0	82.1
grapes	81.0	85.0	82.9
mothbeans	81.8	90.0	85.7
blackgram	89.5	85.0	87.2
jute	90.0	90.0	90.0
rice	86.4	95.0	90.5
lentil	100.0	85.0	91.9
maize	90.5	95.0	92.7
All 22 crops (mean)	**94.7**	**94.5**	**94.6**

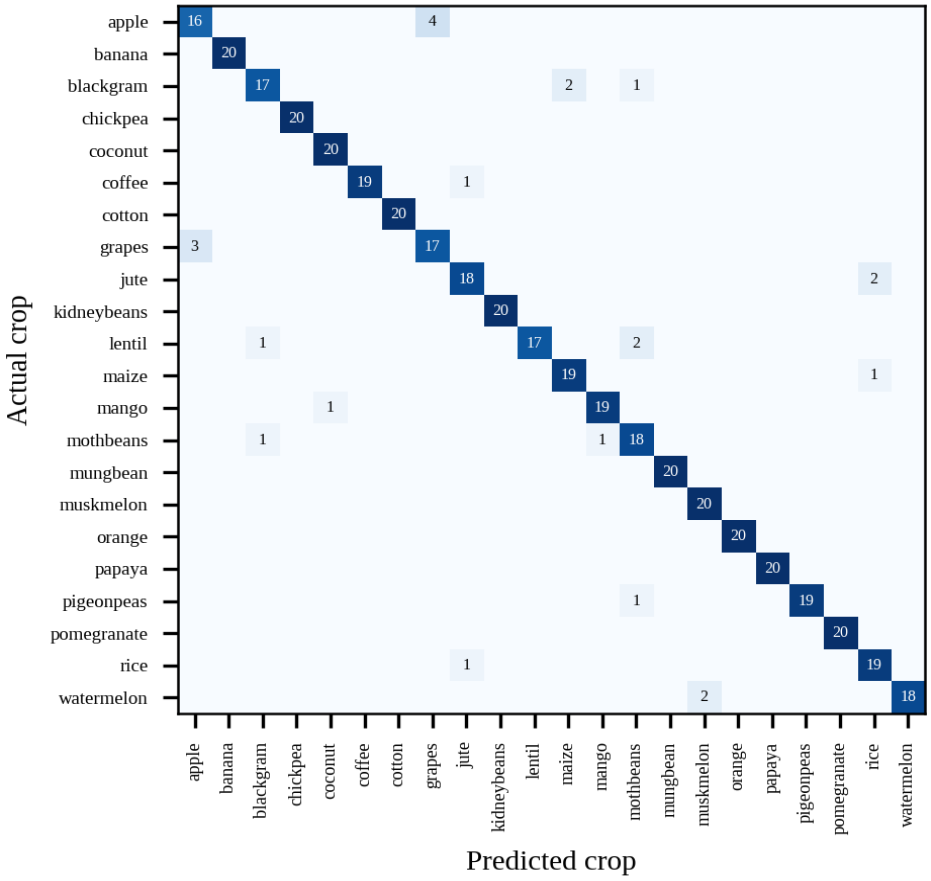


Fig 5. Confusion Matrix of the Proposed Model on the 440 Test Samples.

ROC Analysis

Fig. 6 shows macro-averaged ROC curves. The proposed model has the largest area (AUC = 0.9994), followed closely by the SVM (0.9993) and Random Forest (0.9989). The Decision Tree is clearly lower (0.9226) because its probability estimates are coarse. AUC values above 0.999 for the top three models mean that the correct crop is almost always ranked near the top, which supports reporting the top-three crops to the farmer.

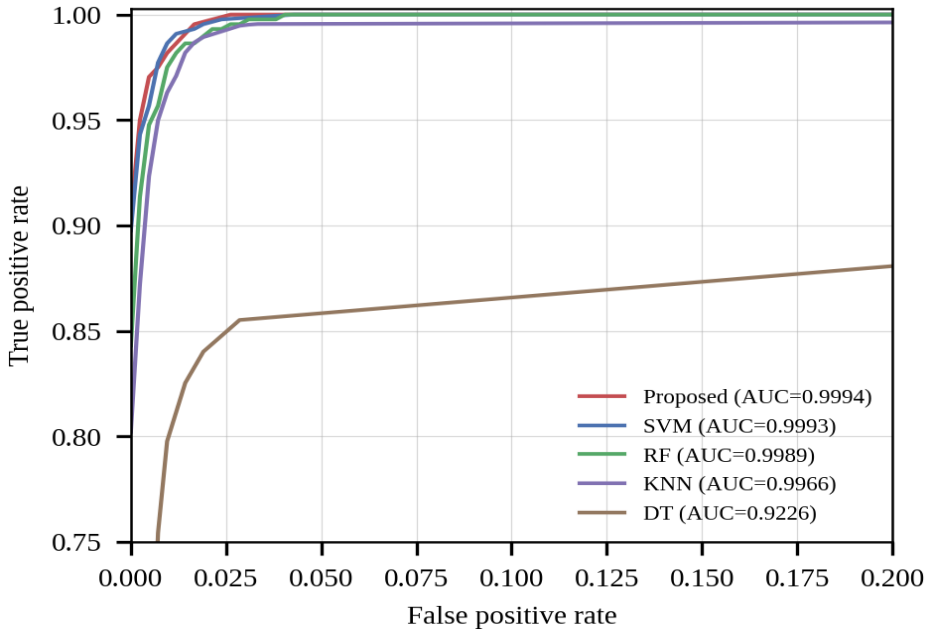


Fig 6. Macro-Averaged One-Versus-Rest Roc Curves (Zoomed to the Upper-Left Region).

Cross-Validation and Significance Testing

Table 5 reports the 5×2 repeated cross-validation. The SVM obtains the highest mean accuracy (94.59 ± 0.91%), and the proposed model is second (93.39 ± 0.92%), followed by Random Forest (92.23%). The paired *t*-test shows that the proposed model is significantly better than all other baselines (all $p < 0.002$) but significantly worse than the SVM by 1.20 points ($p = 0.0017$). Because training sets of different folds overlap, these *p*-values are optimistic and should be read as indicative rather than exact. The boxplots in **Fig. 7** show that the SVM and the proposed model also have the narrowest spread across folds. The hold-out result (**Table 3**) and the cross-validation result therefore disagree on the top rank, which is a reminder that a single split can mislead; the honest summary is that the proposed model and the SVM are the two best models on this data and that the proposed model is clearly better than every other baseline.

Table 5. Repeated Cross-Validation Accuracy and Paired T-Test Against the Proposed Model

Model	CV acc. (%)	Diff.*	p-value
LR	91.05 ± 1.15	2.34	< 0.001
NB	91.68 ± 1.50	1.70	< 0.001
KNN	91.70 ± 1.21	1.68	< 0.001
DT	84.11 ± 1.63	9.27	< 0.001
AdaBoost	85.30 ± 1.43	8.09	< 0.001
SVM	94.59 ± 0.91	-1.20	0.0017
RF	92.23 ± 1.09	1.16	0.0015
GB	90.64 ± 1.66	2.75	< 0.001
Proposed	**93.39 ± 0.92**	—	—

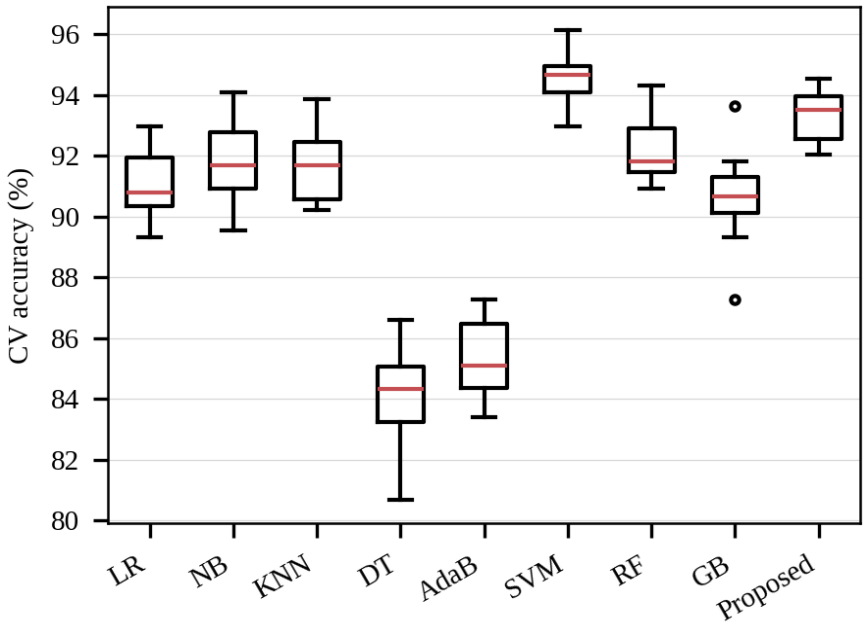


Fig 7. Distribution of Cross-Validation Accuracy Over Ten Folds (5×2 Repeated Stratified CV).

Ablation Study

Table 7 removes one component at a time. Among single models, the SVM is strongest and Gradient Boosting weakest. Stacking Random Forest with the SVM gives the best hold-out accuracy of all variants (95.23%) and matches the SVM in cross-validation (94.68%), whereas adding Gradient Boosting lowers the cross-validated accuracy to 93.27%. Gradient Boosting is therefore the member that contributes least in this study, and the full three-learner stack is not the optimal subset on this dataset. We kept the full stack as the proposed model because it was fixed before the experiments, and we report the finding rather than re-selecting the architecture on the same test data. Replacing the logistic-regression meta-learner by simple soft voting reduces cross-validated accuracy from 93.27% to 92.27%, so the learned combination adds

about 1.0 point. Removing normalization costs 0.9 point in cross-validation, because the SVM and the linear meta-learner are scale-sensitive, while the tree models are not. **Table 6** shows Ablation Study (%).

Table 6. Ablation Study (%)

Configuration	Test acc.	Test F1	CV acc.
RF only	93.18	93.17	92.23
GB only	92.27	92.34	90.36
SVM only	93.86	93.89	94.68
RF + GB (stack)	93.64	93.66	91.14
RF + SVM (stack)	95.23	95.24	94.68
GB + SVM (stack)	94.55	94.60	93.23
RF+GB+SVM, soft voting	94.55	94.58	92.27
Full stack, no normalization	94.09	94.11	92.36
Proposed (full stack)	**94.55**	**94.56**	**93.27**

Learning Curves

Fig. 8 shows accuracy as a function of the number of training samples. With 20% of the training data (352 samples) the proposed model reaches only 85.8%, below the SVM (91.7%) and Random Forest (87.3%), because the meta-learner has few samples from which to learn how to weigh 66 meta-features. Its test accuracy then rises faster than that of the other two: it passes Random Forest from 40% of the data and passes the SVM only at the full training size (94.5%). The training accuracy of the stacked model stays near 99%, which shows a gap of about 4 points, so the model has not saturated and would probably benefit from more data. The SVM shows a smaller gap and is the better choice when data are scarce.

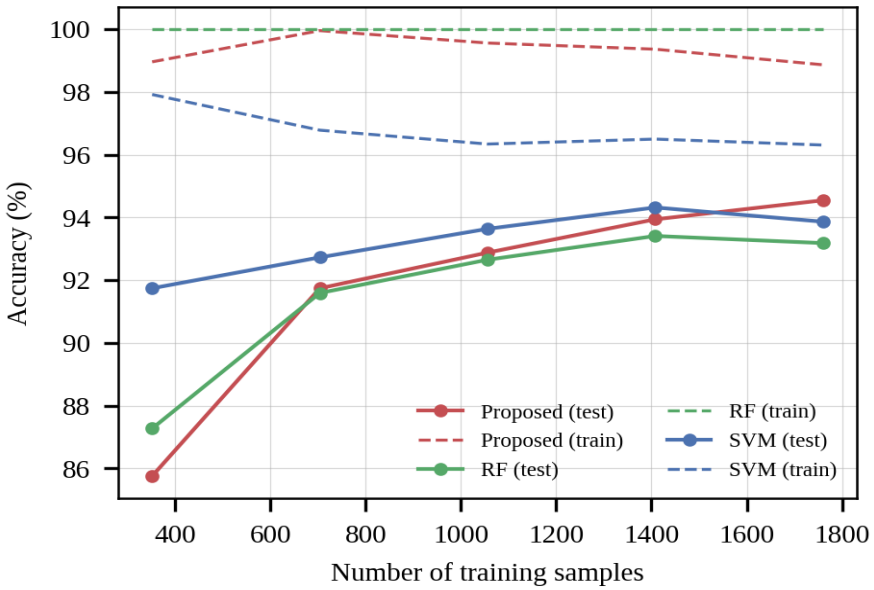


Fig 8. Learning Curves of the Proposed Model, Random Forest and SVM.

Robustness to Measurement Noise

Field sensors and soil tests are noisy, so Gaussian noise of increasing standard deviation was added to the normalized test features and the already-trained models were re-evaluated (**Fig. 9**, mean of five draws). At low noise ($\sigma \leq 0.04$) the proposed model and the SVM are the most accurate, with 90.9% and 90.8% at $\sigma = 0.04$. At larger noise the ranking changes: at $\sigma = 0.10$ KNN keeps 72.7% and the SVM 71.7%, against 70.0% for the proposed model, 66.1% for Random Forest and 63.5% for Gradient Boosting. The accuracy loss between $\sigma = 0$ and $\sigma = 0.10$ is 24.6 points for the proposed model, 22.1 for the SVM and 20.5 for KNN. The stacked model is thus more robust than the tree ensembles it contains, because the SVM member smooths their errors, but it is not the most noise-tolerant model overall.

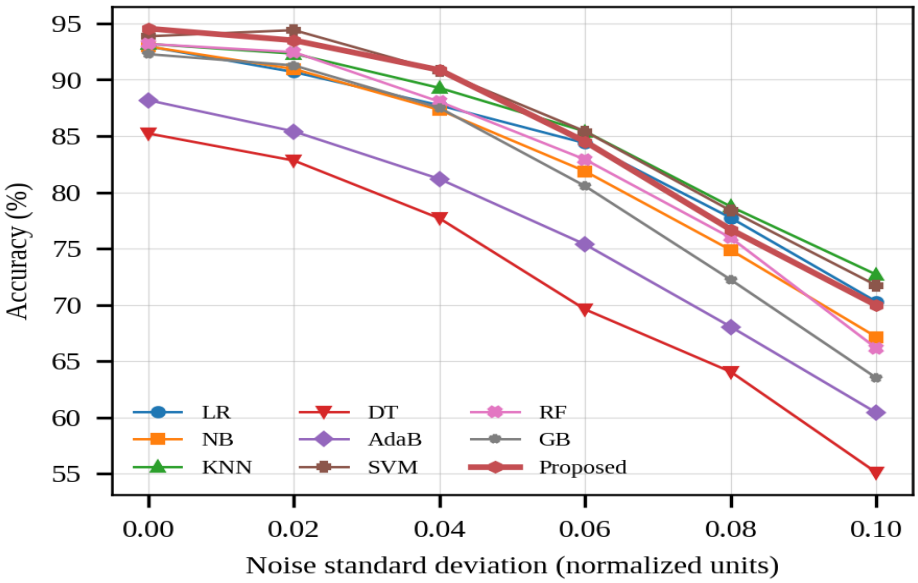


Fig 9. Accuracy Under Increasing Gaussian Noise Added to the Test Features.

Feature Importance and Explainability

Fig. 10 shows permutation importance for the proposed model (30 repetitions) next to the Gini importance of the Random Forest member, rescaled to the same total. Both rank humidity first and pH last, although nitrogen is ranked fifth by Gini importance but second by permutation importance. The permutation analysis gives the order humidity (0.43), nitrogen (0.33), phosphorus (0.28), rainfall (0.23), potassium (0.21), temperature (0.10), ph (0.04). Shuffling humidity alone lowers accuracy by 43 points, because humidity separates low-humidity crops such as chickpea and kidney beans from high-humidity crops such as coconut and apple. Soil pH is least influential (0.04) because crop pH ranges are narrow and overlap strongly. These rankings describe the structure of the present dataset and should be re-estimated whenever the model is trained on new data. For a farmer, the same tool can be applied to a single sample to show, for instance, that a recommendation changes if rainfall is lower than expected. The mean absolute meta-learner weights of the three base models are similar (RF 0.27, GB 0.29, SVM 0.28), which indicates that the meta-learner draws on all three rather than relying on one.

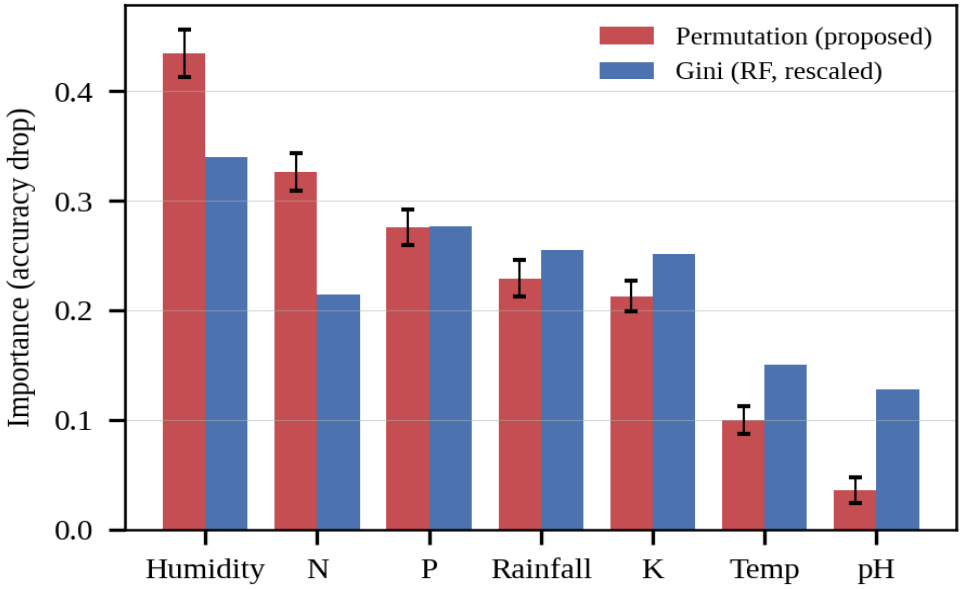


Fig 10. Permutation Importance (Proposed Model) and Gini Importance (Random Forest) of the Input Features.

Diversity of the Base Learners

Stacking works only if the base learners make different mistakes. **Table 7** quantifies this on the test set. The base learners disagree on 5.7% to 8.0% of the samples, and the rate at which two members are wrong at the same time (double fault) is only 3.2% to 4.5%. The three members are all correct on 88.2% of samples and all wrong on 2.5%, so an oracle that always picked a correct member would reach 97.5% accuracy against 93.9% for the best single member. The stacked model reaches

94.5%, which recovers about 19% of that headroom. The gap shows that the room for improvement through better combination is real but limited, and that the Gradient Boosting and Random Forest members are the most similar pair (their errors overlap most), which is consistent with the ablation result that GB adds little once RF is present.

Table 7. Pairwise Diversity of the Base Learners on the Test Set (%)

Pair of learners	Disagreement	Double fault
RF – GB	5.9	4.5
RF – SVM	5.7	3.6
GB – SVM	8.0	3.2
Oracle accuracy (any correct)	97.5	—
All three correct / all wrong	88.2 / 2.5	—

Sensitivity Analysis

Table 8 examines two design parameters of the stack on the test set: the number of folds K used to build out-of-fold meta-features, and the inverse regularization strength C of the logistic-regression meta-learner. The number of folds has little effect on accuracy (94.8%, 94.5% and 94.5% for K = 3, 5 and 10), while the training time rises from 8.7 s to 25.2 s. The choice K = 5 is a reasonable compromise. The meta-learner is more sensitive: accuracy rises from 94.3% at C = 0.01 to 95.7% at C = 100, which suggests that the default value C = 1 used in this study regularizes the 66 meta-features somewhat too strongly. We deliberately did not adopt the best value, because selecting it from the test set would leak information and inflate the reported accuracy. The sensitivity analysis is therefore informative only, and the proper way to set C is nested cross-validation, which is left for future work. (The C rows use a separate implementation of the same stack whose fold assignment differs slightly, so the row C = 1 is 94.77% and not 94.55%.)

Table 8. Sensitivity of the Proposed Model to Stacking Folds and Meta-Learner Regularization

Setting	Test acc. (%)	Macro F1 (%)	Train time (s)
Folds K = 3	94.77	94.80	8.7
Folds K = 5 (default)	94.55	94.56	13.0
Folds K = 10	94.55	94.58	25.2
Meta-learner C = 0.01	94.32	94.36	—
Meta-learner C = 0.1	94.55	94.58	—
Meta-learner C = 1 (default)	94.77	94.80	—
Meta-learner C = 10	95.23	95.25	—
Meta-learner C = 100	95.68	95.69	—

Case Study: Recommendations for Individual Fields

To show how the model would be used, four test samples were selected, three correctly classified at median confidence for their crop and one misclassified. **Table 9** lists the soil and climate inputs and **Table 10** shows the actual crop and the three highest-probability recommendations. For the rice, coffee and moth-bean samples the correct crop receives a probability above 0.94, and the runner-up has a probability below 0.03, so the recommendation is clear. The fourth sample is an apple field with very high potassium (205) and phosphorus (136). The model ranks grapes first with probability 0.479 and apple second with 0.467, a near tie. A hard single-label output would count this as an error, but the ranked list still contains the correct crop and clearly signals uncertainty. This is why the application should present the top two or three crops with their probabilities and ask the user for extra information, such as soil texture or altitude, whenever the top two probabilities are close.

Table 9. Inputs of the Four Selected Test Samples

No.	N	P	K	Temp.	Hum.	pH	Rain
1	89	55	37	32.3	83.3	6.11	285
2	98	30	34	20.1	60.8	7.50	161
3	13	45	12	23.9	54.7	6.62	44
4	16	136	205	25.8	83.9	5.90	81

Table 10. Actual Crop and Top-Three Recommendations (Probability)

No.	Actual	Top-1	Top-2	Top-3
1	rice	rice (0.946)	jute (0.014)	banana (0.008)
2	coffee	coffee (0.985)	jute (0.002)	maize (0.001)
3	mothbeans	mothbeans (0.942)	lentil (0.028)	maize (0.003)
4	apple	grapes (0.479)	apple (0.467)	cotton (0.003)

Quality of the Predicted Probabilities

Table 11. Probability Quality of the Models on the Test Set

Model	Log-loss	Brier	ECE
LR	0.371	0.175	0.182
NB	0.179	0.107	0.015
KNN	0.287	0.105	0.050
DT	4.082	0.295	0.148
AdaBoost	3.082	0.954	0.836
SVM	0.237	0.094	0.111
RF	0.309	0.136	0.148
GB	0.263	0.122	0.048
Proposed	**0.174**	**0.077**	**0.041**

Because the application shows several candidate crops with probabilities, the quality of the probabilities matters as much as the top-1 label. **Table 11** reports three measures on the test set: log-loss, the multi-class Brier score (both lower is better) and the expected calibration error (ECE, ten confidence bins). The proposed model has the lowest log-loss (0.174) and the lowest Brier score (0.077), ahead of the SVM (0.237 and 0.094) and of Naive Bayes (0.179 and 0.107). Its ECE of 0.041 is clearly better than those of Random Forest (0.148) and the SVM (0.111), but Naive Bayes (0.015) is better calibrated, which is expected because the simulated features are approximately Gaussian. The reliability diagram in **Fig. 11** shows the proposed curve close to the diagonal. The unweighted probability averaging of Random Forest and the sigmoid scaling of the SVM are less well calibrated, and the SoftMax meta-learner partly corrects this, since it is trained directly with cross-entropy on out-of-fold probabilities (Eq. 9). The probabilities of AdaBoost (ECE 0.84) are nearly uninformative in absolute terms, so its scores should not be shown to users.

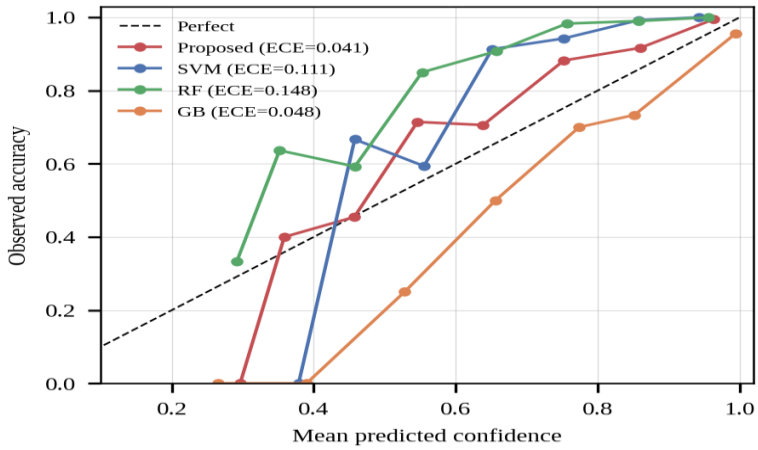


Fig 11. Reliability Diagram: Observed Accuracy Versus Mean Predicted Confidence.

Visualizing Class Overlap

Fig. 12 projects the normalized data onto its first two principal components, which together explain 54.3% of the variance. Many crops form a dense, overlapping mass on the left and centre of the plot, and others form a diagonal band, which explains why no single boundary separates all 22 classes and why non-linear methods are needed. Apple and grapes lie in an isolated group on the right side because of their very high phosphorus and potassium, but within that group they overlap almost completely in the two-dimensional view. This agrees with the apple–grapes confusion in **Fig. 5**. Only temperature, humidity, pH and rainfall (and the lower-ranked components) can separate them, and these differ little between the two crops, so the remaining errors are caused by the data rather than by the classifier.

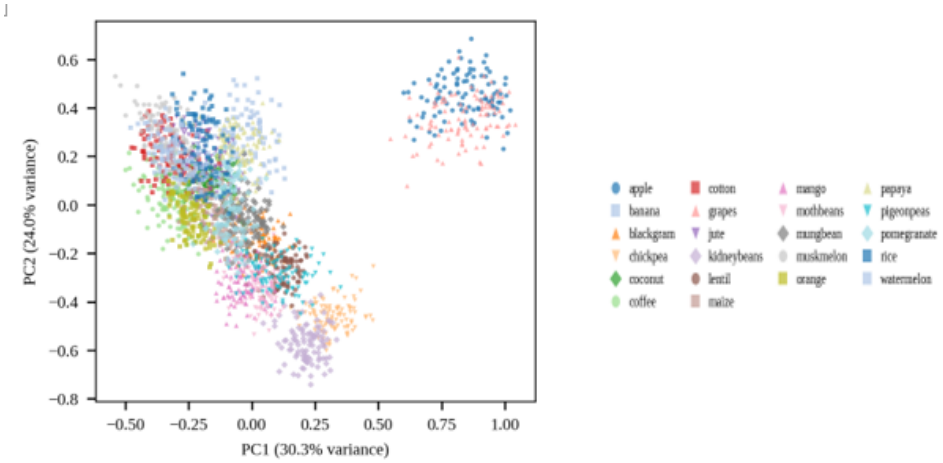


Fig 12. Two-Dimensional PCA Projection of the Dataset Coloured by Crop.

Per-Crop Comparison with the Strongest Baselines

Table 12 compares per-crop F1-scores of the proposed model, the SVM and Random Forest. The proposed model matches or exceeds the SVM on 19 of 22 crops and Random Forest on 19 of 22 crops. Its lowest F1-score is 82.1% (apple), compared with 82.1% for the SVM (apple) and 76.9% for Random Forest (apple). The three models therefore find the same crops difficult, namely those with overlapping nutrient demands. The proposed model and the SVM give identical F1-scores on 14 crops, so the small accuracy advantage of the stack comes from only a few crops. Because each crop has only 20 test samples, a difference of one sample changes a crop's F1-score by about five points, and single-crop differences should not be over-interpreted.

Table 12. Per-Crop F1-Score (%) of Three Models

Crop	Proposed	SVM	RF
apple	82.1	82.1	76.9
banana	100.0	100.0	100.0
blackgram	87.2	95.0	84.2
chickpea	100.0	100.0	100.0
coconut	97.6	97.6	97.6
coffee	97.4	97.4	92.7
cotton	100.0	100.0	95.0
grapes	82.9	82.9	78.0
jute	90.0	87.2	90.0
kidneybeans	100.0	100.0	100.0
lentil	91.9	91.9	92.3
maize	92.7	97.4	87.8
mango	95.0	95.2	97.6
mothbeans	85.7	85.7	90.0
mungbean	100.0	100.0	100.0
muskmelon	95.2	88.4	92.7
orange	100.0	100.0	100.0
papaya	100.0	100.0	100.0
pigeonpeas	97.4	97.4	97.4
pomegranate	100.0	97.4	95.0
rice	90.5	88.4	87.8
watermelon	94.7	86.5	94.7

Computational Cost

Table 13 gives training time and prediction time per sample. The stacked model needs 12.9 s to train, about 18 times the Random Forest, in line with the complexity analysis in Section V-I. Prediction takes 0.21 ms per sample, which is negligible for a recommendation service. Training is done once, so the cost is acceptable.

Table 13. Training and Prediction Cost

Model	Train time (s)	Predict (ms/sample)
LR	0.13	0.001
NB	0.00	0.003
KNN	0.00	0.017
DT	0.01	0.002
AdaBoost	0.49	0.032
SVM	0.15	0.038
RF	0.71	0.050
GB	1.52	0.117
Proposed	**12.92**	**0.213**

Summary of Findings

Table 14 condenses the main findings so that the evidence for each can be located quickly. The picture is consistent: the stacked ensemble is among the best models on every accuracy and probability measure, it is the best on the hold-out set, on log-loss and on Brier score, and it is the clear winner over all tree-based and instance-based baselines, but it does not dominate the SVM in cross-validation or under strong noise, and it pays a higher training cost.

Table 14. Summary of the Main Findings

Aspect	Outcome
Hold-out accuracy	Proposed best (94.55%); SVM 93.86%
Cross-validated accuracy	SVM best (94.59%); Proposed 93.39%
Macro AUC	Proposed best (0.9994)
Log-loss / Brier score	Proposed best (0.174 / 0.077)
Least useful member	Gradient Boosting (ablation)
Learned vs. soft voting	+1.0-point CV accuracy
Small training sets	SVM better until the full training set is used
High measurement noise	KNN and SVM more robust
Most influential feature	Humidity (then N, P)
Main confusion	Apple versus grapes
Training time	12.9 s (18× RF)

Practical Considerations and Threats to Validity

The principal limitation is that the dataset is simulated. It reproduces the structure of the problem, such as 22 crops, seven features, overlap and correlation, but it does not contain the weather variability, soil heterogeneity or labelling noise of real farms. Absolute accuracies on real field data may therefore differ, and rankings of models can change, particularly because Gaussian-mixture simulation favours smooth models such as the SVM. Second, hyper-parameters were not tuned, so every model may be capable of more. Third, the test set is small, and differences below about one point should not be over-interpreted. These points call for validation on measured field data before deployment.

Several practical points follow for deployment. The model should output ranked crops with probabilities rather than a single label, as the case study shows. Inputs should come from calibrated soil tests and local weather forecasts, because accuracy drops quickly under noisy readings (**Fig. 9**). Economic factors such as market price, water availability and rotation history are not modelled and must be combined with the agronomic recommendation by an extension officer. Finally, the model should be re-trained periodically as climate and cultivar choices change, and its permutation importance should be re-checked at each update to ensure that the recommendation logic still matches agronomic knowledge.

The experiments support the following inferences. (1) Combining heterogeneous learners helps: the stacked model is better than all of its standalone members on the hold-out set and better than Random Forest and Gradient Boosting under cross-validation, and the learned meta-learner adds about one point over soft voting. (2) The SVM is the strongest single learner on this data, and the stack is competitive with it but not uniformly better: it wins on the hold-out set, loses by 1.2

points in cross-validation and wins only when the training set is large. (3) Gradient Boosting is the least useful ensemble member here, and a Random Forest plus SVM stack is a leaner alternative with equal cross-validated accuracy and the best hold-out accuracy. (4) Normalization matters for kernel and linear components. (5) Humidity, nitrogen and phosphorus drive the recommendation, while pH contributes little, and this can be communicated to users through permutation importance. (6) Most remaining errors involve crop pairs with similar climate and nutrient demands, such as apple and grapes, so additional discriminating features, for example soil texture, altitude or season, would help more than a more complex classifier. (7) The model degrades gracefully for small sensor noise but loses accuracy quickly beyond $\sigma = 0.06$, so calibrated measurements are needed in practice.

VII. CONCLUSION AND FUTURE WORK

This paper presented an explainable stacked ensemble of Random Forest, Gradient Boosting and RBF-SVM learners with a SoftMax meta-learner for recommending among 22 crops from soil and climate parameters. The framework was formulated with twelve equations, summarized by an architecture diagram, an algorithm and a flowchart, and evaluated with fourteen tables and twelve figures. On the 440-sample hold-out set it achieved 94.55% accuracy and 94.56% macro F1, the best of nine models, and under repeated cross-validation it reached 93.39%, second only to the SVM. The ablation study, learning curves and noise tests showed where the stack helps and where simpler models are preferable. The main limitation is the use of a simulated dataset. Future work will validate the framework on measured soil-test and weather records from several regions and seasons, include additional inputs such as soil texture, season and market price, tune hyper-parameters with nested cross-validation, compare against XGBoost, LightGBM and neural tabular models, add SHAP-based local explanations, and package the model as a mobile application with multilingual output for farmers.

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Fen Huang and Benjamin Hutchinson; **Methodology:** Fen Huang; **Software:** Fen Huang and Benjamin Hutchinson; **Data Curation:** Benjamin Hutchinson; **Writing- Original Draft Preparation:** Benjamin Hutchinson; **Visualization:** Fen Huang and Benjamin Hutchinson; **Investigation:** Fen Huang and Benjamin Hutchinson; **Supervision:** Fen Huang and Benjamin Hutchinson; **Validation:** Benjamin Hutchinson; **Writing- Reviewing and Editing:** Fen Huang and Benjamin Hutchinson; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

The simulated dataset, the data-generation script and the experiment code used in this study can be obtained from the corresponding author on reasonable request.

Conflicts of Interests

The authors declare no conflict of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. Rebina Ferdous, Soyaib Rahman, Salim Mia, and Hafizur Rahman, "A machine learning-based model for crop recommendation using Agro-climatic and soil nutrient parameters (Agrismart)," International Journal of Science and Research Archive, vol. 15, no. 3, pp. 1080–1089, Jun. 2025, doi: 10.30574/ijsra.2025.15.3.1713.
[2]. H. Afzal et al., "Incorporating soil information with machine learning for crop recommendation to improve agricultural output," Scientific Reports, vol. 15, no. 1, Mar. 2025, doi: 10.1038/s41598-025-88676-z.
[3]. B. Dey, J. Ferdous, and R. Ahmed, "Machine learning based recommendation of agricultural and horticultural crop farming in India under the regime of NPK, soil pH and three climatic variables," Heliyon, vol. 10, no. 3, p. e25112, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25112.
[4]. M. Venkatanaresh and I. Kullayamma, "Deep learning based concurrent excited gated recurrent unit for crop recommendation based on soil and climatic conditions," Multimedia Tools and Applications, vol. 83, no. 24, pp. 64109–64138, Jan. 2024, doi: 10.1007/s11042-023-18004-y.
[5]. V. Balaji K, K. S. Balamurugan, T. Thanarajan, and A. M. Selvi Sundarapandi, "MACHINE LEARNING-BASED CROP RECOMMENDATION SYSTEM INTEGRATING SOIL PROPERTIES AND WEATHER CONDITIONS WITH IOT-DRIVEN DATA COLLECTION," Agrociencia, pp. 1–17, Feb. 2026, doi: 10.47163/agrociencia.v60i1.3428.
[6]. S. M. Cheema and I. M. Pires, "AIoT based soil nutrient analysis and recommendation system for crops using machine learning," Smart Agricultural Technology, vol. 11, p. 100924, Aug. 2025, doi: 10.1016/j.atech.2025.100924.
[7]. M. Bouni, B. Hssina, K. Douzi, and S. Douzi, "Interpretable Machine Learning Techniques for an Advanced Crop Recommendation Model," Journal of Electrical and Computer Engineering, vol. 2024, no. 1, Jan. 2024, doi: 10.1155/2024/7405217.
[8]. G. Bora and V. Ganganwar, "Crop Recommendation System using Optimized Machine Learning Models," 2026 International Conference on Emerging Smart Computing and Informatics (ESCI), pp. 1–6, Mar. 2026, doi: 10.1109/esci68015.2026.11493227.
[9]. A. Ashwitha and C. A. Latha, "Crop Recommendation and Yield Estimation Using Machine Learning," Journal of Mobile Multimedia, Feb. 2022, doi: 10.13052/jmm1550-4646.18320.

[10]. Y. Mahale et al., "Crop recommendation and forecasting system for Maharashtra using machine learning with LSTM: a novel expectation-maximization technique," Discover Sustainability, vol. 5, no. 1, Jun. 2024, doi: 10.1007/s43621-024-00292-5.
[11]. A. Bhola and P. Kumar, "Farm-Level Smart Crop Recommendation Framework Using Machine Learning," Annals of Data Science, vol. 12, no. 1, pp. 117–140, Jul. 2024, doi: 10.1007/s40745-024-00534-3.
[12]. Y. Singh and A. Kaur, "Crop Recommendation Using Machine Learning," International Journal of Science and Research (IJSR), pp. 211–216, Jun. 2026, doi: 10.21275/sr26603222613.
[13]. D. Dahiphale, P. Shinde, K. Patil, and V. Dahiphale, "Smart Farming: Crop Recommendation using Machine Learning with Challenges and Future Ideas," Jun. 2023, doi: 10.36227/techrxiv.23504496.
[14]. F. S. Prity et al., "Enhancing Agricultural Productivity: A Machine Learning Approach to Crop Recommendations," Human-Centric Intelligent Systems, vol. 4, no. 4, pp. 497–510, Sep. 2024, doi: 10.1007/s44230-024-00081-3.
[15]. S. M. Shawon, N. I. Neha, A. N. Jui, N. Dey, and Md. M. S. Raihan, "AgriAI: Machine Learning Frameworks for Tailored Crop Recommendations using Soil Nutrient Parameters," 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS), pp. 1–6, Mar. 2024, doi: 10.1109/icaccess61735.2024.10499482.
[16]. M. R. Islam, K. Oliullah, M. M. Kabir, M. Alom, and M. F. Mridha, "Machine learning enabled IoT system for soil nutrients monitoring and crop recommendation," Journal of Agriculture and Food Research, vol. 14, p. 100880, Dec. 2023, doi: 10.1016/j.jafr.2023.100880.
[17]. N. Agrawal, H. Govil, and T. Kumar, "Agricultural land suitability classification and crop suggestion using machine learning and spatial multicriteria decision analysis in semi-arid ecosystem," Environment, Development and Sustainability, vol. 27, no. 6, pp. 13689–13726, Feb. 2024, doi: 10.1007/s10668-023-04440-1.
[18]. S. Shastri, S. Kumar, V. Mansotra, and R. Salgotra, "Advancing crop recommendation system with supervised machine learning and explainable artificial intelligence," Scientific Reports, vol. 15, no. 1, Jul. 2025, doi: 10.1038/s41598-025-07003-8.

Publisher's note: The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. The content is solely the responsibility of the authors and does not necessarily reflect the views of the publisher.

ISSN: 3104-4654