Edu Komputika 12 (1) (2025)

Edu Komputika Journal

https://journal.unnes.ac.id/journals/edukom/

Performance Evaluation of Machine Learning Models for Soil Fertility Classification Based on the Indian Soil Fertility Dataset

Yoga Pristyanto[1)]✉, Ibrahim Aji Fajar Romadhon[1)], Anggit Ferdita Nugraha[2)], Atik Nurmasani[1)], and Irma Rofni Wulandari[1)]

[1]Department of Information System, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia.

[2]Department of Computer Engineering, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia.

Article Info

Article History:
Received: 23 July 2024
Revised: 17 August 2025
Accepted: 19 August 2025

Keywords:
Classification, Ensemble Classifier, Machine Learning, Single Classifier, Soil Fertility Classification

Abstract

Rice farming productivity worldwide has been declining due to improper soil management practices, including excessive chemical fertilizer use and irregular irrigation. The main challenge lies in accurately classifying soil fertility levels to support optimal land use and reduce resource waste, especially when dealing with imbalanced datasets. This study aims to compare the performance of single classifiers and ensemble classifiers in classifying soil fertility. The single classifiers used include K-Nearest Neighbor (KNN), Naive Bayes, Decision Tree, Support Vector Machine (SVM), and Artificial Neural Network (ANN), while the ensemble classifiers include Random Forest and XGBoost. The Indian Soil Fertility Dataset, obtained from Kaggle, contains 880 samples with 12 features and 1 output class. The research methodology involved data acquisition, preprocessing, data splitting, standardization, and classification, with performance evaluation conducted using a confusion matrix. The results show that ensemble classifiers, particularly Random Forest and XGBoost, outperform single classifiers in imbalanced datasets, achieving accuracy, precision, recall, and F1-score values exceeding 92%-95% across all split scenarios. The findings conclude that Random Forest and XGBoost can serve as reliable models for assisting farmers and agricultural experts in evaluating soil conditions, minimizing unnecessary fertilizer usage, and improving rice farming productivity globally.

✉ Correspondence Address:
Jl. Padjajaran, Ring Road Utara, Condongcatur,
Depok, Sleman, Daerah Istimewa Yogyakarta 55283
E-mail: yoga.pristyanto@amikom.ac.id

ISSN 2252-6811
E-ISSN 2599-297X

INTRODUCTION

Rice is one of the most important staple foods in the world, feeding over half of the global population. Maintaining its productivity largely depends on soil fertility, which plays a pivotal role in plant growth, yield stability, and long-term agricultural sustainability. However, soil fertility is often compromised worldwide due to improper management practices such as excessive chemical fertilizer application, unbalanced nutrient replenishment, and irregular irrigation. These practices not only reduce crop yields but also lead to soil degradation, threatening both food security and environmental health (Bouslihim et al., 2024).

Accurate soil fertility assessment is essential for guiding land management decisions, minimizing resource waste, and ensuring sustainable rice production. Traditional laboratory-based soil evaluation methods, although accurate, are often expensive and slow, making them impractical for large-scale implementation (Supriyanto & Atwa Magriyanti, 2022). Machine learning (ML) offers a powerful alternative, enabling automated, data-driven classification of soil fertility using measurable soil parameters (Blesslin Sheeba et al., 2022; Bouasria et al., 2023; Sarangi et al., 2024).

Several studies have explored ML for soil-related predictions: for example, (Pramoedyo et al., 2022) found that Random Forest outperformed Naive Bayes in classifying soil texture in East Java, achieving around 92.55 % accuracy in training and 87.5 % in testing, versus 89.98 % and 80.65 % for NB. (Reddy et al., 2024) employed Random Forest with conditioned Latin hypercube sampling and Boruta feature selection to predict soil pH and organic matter, achieving low RMSEs of 0.60 and 0.71. In the context of digital soil mapping (DSM), ML and remote sensing have significantly improved accuracy and scalability over conventional methods (Mallah et al., 2022). Other studies have demonstrated that combining Random Forest with resampling techniques substantially improves classification performance on imbalanced soil data (Wadoux et al., 2020). Yet, many investigations either focus on regression tasks—such as predicting organic carbon or pH—without addressing categorical fertility classification, or they lack standardized comparisons between single and ensemble models across similar datasets (Akula et al., 2023; Mallah et al., 2022). This indicates a clear research gap: existing work seldom offers head-to-head comparisons of single classifiers (like KNN, Naive Bayes, Decision Trees, SVM, and ANN) versus ensemble methods (Random Forest, XGBoost) under unified preprocessing and evaluation pipelines, particularly in handling class imbalance and using publicly available fertility datasets like the Indian Soil Fertility Dataset from Kaggle.

To address this gap, this study proposes a comparative classification framework that evaluates multiple machine learning algorithms—both single and ensemble—using standardized preprocessing, stratified data splitting, and confusion-matrix-derived metrics. The proposed method includes feature standardization, model training with default hyperparameters, and performance evaluation across different train–test split ratios to analyze model robustness under varied conditions. The main purpose of this approach is to identify the most effective classification model for soil fertility prediction that can be applied in practical agricultural decision-support systems, thereby enabling better fertilizer management and improving rice productivity.

The contributions of this research are threefold: (1) delivering a reproducible benchmark comparing single and ensemble classifiers for soil fertility classification; (2) providing an empirical evaluation of classifier robustness under class imbalance, with comprehensive metric reporting across scenarios; and (3) offering actionable insights for global rice farming decision-support, highlighting that Random Forest and XGBoost consistently exceed 92% performance across all metrics, thus supporting effective soil assessment, optimized fertilizer use, and enhanced productivity.

RESEARCH METHODS

This study adopts a systematic machine learning workflow consisting of dataset acquisition, preprocessing, modeling, and evaluation, as illustrated in Figure 1. The experiments were implemented using Python with the scikit-learn library on a standard computing environment.

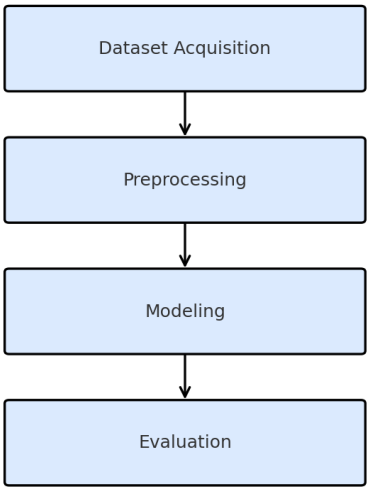

Figure 1. Research workflow

A. Data Acquisition

At this stage, a rice field soil fertility dataset was obtained from the Kaggle platform, which can be accessed at (Jaiswal, 2024). The dataset consists of 12 feature attributes and 1 class attribute representing the soil fertility category. Table 1 provides detailed information about the dataset used in this study.

Table 1. Dataset Feature Details

Attribute	Information
N	ratio of Nitrogen (NH4+) content in soil
P	ratio of Phosphorous (P) content in soil
K	ratio of Potassium (K) content in soil
pH	soil acidity (pH)
EC	electrical conductivity
OC	Organic carbon in land
S	sulfur (S)
Zn	Zinc (Zn)
Fe	Iron (Fe)
Cu	Copper (Cu)
MN	Manganese (Mn)
B	Boron (B)
Class	Class fertility (0 "Less Fertile", 1 "Fertile", 2 "Highly Fertile")

B. Data Preprocessing

In this stage, the raw dataset undergoes several data preparation steps to ensure it is ready for model training. First, data cleaning is performed to check for and handle missing values or inconsistent entries. Since the dataset is relatively small and balanced in terms of feature completeness, no records are removed, but numerical features are verified for validity (Hanif et al., 2022; Mukhtar et al., 2024; Siahaan et al., 2023). Next, categorical attributes (if any) are encoded into a numerical format using label encoding. To ensure a fair comparison among classifiers, feature standardization is applied using the StandardScaler method, which transforms all features to have a mean of 0 and a standard deviation of 1 (Fidiyanto & Izzati, 2024; Pradana et al., 2023). Finally, the dataset is split into training and testing sets using multiple split ratios (e.g., 70:30, 80:20, 90:10) to evaluate model performance under different data availability scenarios.

C. Modeling

In the modeling stage, various machine learning algorithms are implemented to classify soil fertility based on the given features. This stage involves selecting appropriate models, training them using the processed dataset, and comparing their predictive performance. The study applies both single classifiers and ensemble classifiers to determine the most effective approach for soil fertility classification.

The single classifiers used in this study include:

1. K-Nearest Neighbors (KNN) – A distance-based classification algorithm that assigns the class of a data point based on the majority class among its k nearest neighbors. Euclidean distance is used as the similarity measure.
2. Naive Bayes (NB) – A probabilistic classifier based on Bayes' theorem, assuming feature independence. It predicts the class with the highest posterior probability given the feature values.
3. Decision Tree (DT) – A tree-structured classifier that splits data into branches based on feature values, aiming to maximize class purity at each node.
4. Support Vector Machine (SVM) – A margin-based classifier that finds the optimal hyperplane separating different classes in the feature space.
5. Artificial Neural Network (ANN) – A computational model inspired by the human brain, consisting of interconnected layers of nodes that transform input data into classification outputs.

The ensemble classifiers used in this study include:

1. Random Forest (RF) – An ensemble method that builds multiple decision trees and combines their outputs through majority voting to improve accuracy and reduce overfitting.
2. Extreme Gradient Boosting (XGBoost) – A boosting algorithm that sequentially builds decision trees, where each new tree corrects the errors of the previous ones, resulting in high predictive performance.

By applying both single and ensemble classifiers, this study aims to identify the most accurate and robust algorithm for classifying soil fertility.

D. Model Evaluation

The final stage of this research involves evaluating the performance of the K-Nearest Neighbor (KNN), Naive Bayes, Decision Tree, Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest, and Extreme Gradient Boosting (XGBoost) algorithms using the Confusion Matrix. The Confusion Matrix provides four important metrics: True Positive (TP), which counts the

number of data instances correctly predicted as positive; True Negative (TN), which counts the number of data instances correctly predicted as negative; False Positive (FP), which counts the number of data instances incorrectly predicted as positive when they are actually negative; and False Negative (FN), which counts the number of data instances incorrectly predicted as negative when they are actually positive. These values serve as the basis for deriving evaluation metrics, including accuracy, precision, recall, and F1-score. Table 2 presents the Confusion Matrix used in this study.

Table 2. Confusion Matrix

Actual	Prediction	
	True	False
True	TP	FN
False	FP	TN

From the results of the confusion matrix, several performance metrics can be determined, including accuracy, precision, recall, and F-measure. Accuracy represents the overall correctness of the system in performing the classification process by measuring the proportion of correctly classified instances. Precision refers to the ratio of correctly classified positive instances to the total instances predicted as positive by the classification system. Recall represents the ratio of correctly classified positive instances to the total actual positive instances in the dataset. Meanwhile, the F-measure is a widely used evaluation metric for addressing class imbalance problems, as it combines precision and recall into a single value, providing a balanced measure of a model's ability to recover relevant information in imbalanced datasets.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (3)$$

$$\text{F1} - \text{Score} = \frac{\text{2x(precision x recall)}}{\text{precision + recall}} \quad (4)$$

RESULT AND DISCUSSION

This section presents the experimental results and discusses the performance of various machine learning algorithms applied to the soil fertility dataset. The evaluation is carried out by comparing single classifiers—such as K-Nearest Neighbor (KNN), Naive Bayes, Decision Tree, Support Vector Machine (SVM), and Artificial Neural Network (ANN)—with ensemble classifiers, including Random Forest and Extreme Gradient Boosting (XGBoost). Each model is assessed using evaluation metrics derived from the confusion matrix, namely accuracy, precision, recall, and F-measure. The results not only highlight the strengths and limitations of individual classifiers but also demonstrate the comparative advantages of ensemble methods in classifying soil fertility. Before presenting the experimental results, a sample of the dataset used in this study is provided in Table 3 to give an overview of the data characteristics.

Table 3. Sampling of the Dataset

Num	N	P	K	pH	EC	OC	S	Zn	Fe	Cu	M N	B	Class
1	138	8.6	560	7.46	0.62	0.70	5.90	0.24	0.31	0.77	8.71	0.11	0
2	213	7.5	338	7.62	0.75	1.06	25.40	0.30	0.86	1.54	2.89	2.29	0
3	163	9.6	718	7.59	0.51	1.11	14.30	0.30	0.86	1.57	2.70	2.03	2
4	157	6.8	475	7.64	0.58	0.94	26.00	0.34	0.54	1.53	2.65	1.82	0
5	270	9.9	444	7.63	0.40	0.86	11.80	0.25	0.76	1.69	2.43	2.26	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
876	351	10.7	623	7.96	0.51	0.29	7.24	0.36	4.69	0.69	11.03	0.69	1
877	264	9.0	486	7.24	0.47	0.10	3.92	0.35	8.26	0.45	7.98	0.40	1
878	276	9.2	370	7.62	0.62	0.49	6.64	0.42	3.57	0.63	6.48	0.32	1
879	320	13.8	391	7.38	0.65	1.07	5.43	0.58	4.58	1.02	13.25	0.53	2
880	264	10.3	475	7.49	0.74	0.88	10.56	0.45	7.36	1.87	10.63	0.63	0

The first preprocessing stage in this study was to examine the dataset for missing values. A complete and consistent dataset is essential to ensure that the machine learning algorithms can learn effectively without bias or distortion. Therefore, each column in the dataset was

carefully checked to identify whether there were incomplete records or empty entries. Based on the results of this examination, it was confirmed that no missing values were present in the dataset. Consequently, no additional handling methods such as imputation, replacement, or deletion of incomplete rows were required. This finding simplified the preprocessing process and guaranteed that the dataset was already in a clean and structured form before further analysis.

The second step of preprocessing focused on ensuring the relevance and quality of the attributes used. In this phase, any features or records that were not useful or did not provide added value for the classification process were removed. Eliminating irrelevant data is an important step in machine learning because redundant or noisy attributes can reduce model performance, increase computational complexity, and potentially lead to overfitting. By refining the dataset, the study ensured that only informative features were included, which helped improve both the efficiency and accuracy of the classification models.

The final step in preprocessing was dividing the dataset into training and testing subsets. This step is crucial to evaluate how well the models generalize to unseen data. In this study, three different split ratios were applied: 90% training and 10% testing, 80% training and 20% testing, and 70% training and 30% testing. The variation in data division allowed the researchers to compare model performance under different proportions of training and testing data. Using multiple split ratios also provided a more reliable performance evaluation, as it helped identify whether the models remained stable and consistent across different training data sizes.

For the first scenario, with a 70% data split for training and 30% for testing, the confusion matrix results for each classification algorithm are presented in Table 4.

Table 4. Confusion Matrix on 70:30 Scenario

Model	TP	TN	FP	FN
K-Nearest Neighbor	91	118	21	23
Naive Bayes	109	48	3	93
Decision Trees	100	132	12	9
Support Vector Machine	103	136	9	5
Artificial Neural Network	98	128	14	13
Random Forest	106	134	6	7
Extreme Gradient Boosting	106	134	6	7

Based on the confusion matrix results under the 70%:30% data split, each classification algorithm shows different levels of performance. The K-Nearest Neighbor (KNN) model produced 91 True Positive (TP), 118 True Negative (TN), 21 False Positive (FP), and 23 False Negative (FN), which indicates balanced performance but still leaves room for improvement due to the number of misclassifications. In contrast, Naive Bayes achieved 109 TP, 48 TN, 3 FP, and 93 FN, showing high sensitivity toward positives but very weak performance in identifying negatives, reflected by the high FN count. The Decision Tree performed better with 100 TP, 132 TN, 12 FP, and 9 FN, demonstrating stable results and strong recall. Similarly, Support Vector Machine (SVM) yielded 103 TP, 136 TN, 9 FP, and 5 FN, making it one of the best-performing models with very few errors and strong decision boundary formation.

Meanwhile, the Artificial Neural Network (ANN) generated 98 TP, 128 TN, 14 FP, and 13 FN, reflecting competitive but slightly lower results than SVM and Decision Tree, possibly due to the limited tuning of its parameters. The ensemble-based methods outperformed most individual models, with Random Forest producing 106 TP, 134 TN, 6 FP, and 7 FN, while Extreme Gradient Boosting (XGBoost) achieved identical results. Both ensemble methods show high reliability with low error rates, confirming their robustness in handling diverse feature interactions. Overall, the comparison highlights that SVM, Random Forest, and XGBoost deliver the most effective performance under this data split, while Naive Bayes struggles significantly, and KNN, along with ANN, provide moderate but less optimal results compared to ensemble and margin-based classifiers.

In the second scenario, where the dataset was divided into 80% training and 20% testing data, the confusion matrix results for all applied algorithms are summarized in Table 5.

Table 5. Confusion Matrix on 80:20 Scenario

Model	TP	TN	FP	FN
K-Nearest Neighbor	64	78	11	16
Naive Bayes	74	14	1	80
Decision Trees	68	87	7	7
Support Vector Machine	68	87	7	7
Artificial Neural Network	68	84	7	10
Random Forest	70	87	5	7
Extreme Gradient Boosting	70	87	5	7

Based on the confusion matrix results for the 80%:20% data split, variations in model

performance can be clearly observed. The K-Nearest Neighbor (KNN) model produced 64 True Positive (TP), 78 True Negative (TN), 11 False Positive (FP), and 16 False Negative (FN), showing reasonably balanced performance but still prone to errors, particularly with false negatives. Naive Bayes, on the other hand, recorded 74 TP, 14 TN, 1 FP, and 80 FN, which reveals a strong bias towards positive classifications but extremely poor handling of negative cases, as shown by its very high FN values. Meanwhile, the Decision Tree achieved 68 TP, 87 TN, 7 FP, and 7 FN, indicating stable and consistent classification ability, while Support Vector Machine (SVM) produced identical results, reflecting strong performance and robust separation of classes.

Furthermore, the Artificial Neural Network (ANN) yielded 68 TP, 84 TN, 7 FP, and 10 FN, which is competitive but slightly less optimal compared to the Decision Tree and SVM due to a higher number of misclassifications. The ensemble-based algorithms again demonstrated superior performance, with Random Forest achieving 70 TP, 87 TN, 5 FP, and 7 FN, while Extreme Gradient Boosting (XGBoost) produced identical outcomes. These results reaffirm the advantage of ensemble methods, which combine multiple decision boundaries to improve robustness and reduce classification errors. Overall, the 80%:20% split shows that ensemble classifiers and margin-based models like SVM and Decision Tree remain consistently strong, while Naive Bayes again struggles with class balance, and KNN and ANN provide only moderate results.

Finally, for the third scenario with a 90%:10% data split, the confusion matrix outcomes for each model are shown in Table 6.

Table 6. Confusion Matrix on 90:10 Scenario

Model	TP	TN	FP	FN
K-Nearest Neighbor	33	38	8	6
Naive Bayes	41	9	0	35
Decision Trees	37	40	4	4
Support Vector Machine	37	43	4	1
Artificial Neural Network	35	41	6	3
Random Forest	40	41	1	3
Extreme Gradient Boosting	39	41	2	3

Based on the confusion matrix results for the 90%:10% data split, each algorithm exhibited varying levels of effectiveness in classifying data. The K-Nearest Neighbor (KNN) model produced 33 True Positive (TP), 38 True Negative (TN), 8 False Positive (FP), and 6 False Negative (FN), indicating moderate performance but still vulnerable to misclassification, particularly in positive instances. Naive Bayes achieved 41 TP, 9 TN, 0 FP, and 35 FN, showing a tendency to classify many samples as positive while poorly identifying negative samples, which resulted in very high false negatives. The Decision Tree demonstrated a stronger balance, yielding 37 TP, 40 TN, 4 FP, and 4 FN, while Support Vector Machine (SVM) further improved upon this with 37 TP, 43 TN, 4 FP, and only 1 FN, indicating more reliable separation of the two classes.

Meanwhile, the Artificial Neural Network (ANN) produced 35 TP, 41 TN, 6 FP, and 3 FN, showing competitive results but slightly less effective compared to SVM due to a higher FP rate. The ensemble-based approaches again provided strong performance, with Random Forest generating 40 TP, 41 TN, 1 FP, and 3 FN, while Extreme Gradient Boosting (XGBoost) produced similar results with 39 TP, 41 TN, 2 FP, and 3 FN. These findings reinforce the consistent strength of ensemble models in achieving higher stability and minimizing misclassification errors, even with a relatively small testing portion in the 90%:10% split scenario. Overall, SVM, Random Forest, and XGBoost emerge as the most effective classifiers, while Naive Bayes once again demonstrates limitations in handling class balance.

A comparative analysis across the three data split scenarios (70%:30%, 80%:20%, and 90%:10%) shows that the distribution of training and testing data has a notable effect on model performance. In the 70%:30% scenario, ensemble methods such as Random Forest and XGBoost, along with SVM, consistently achieved the highest accuracy and balanced classification results, demonstrating strong generalization when tested with a larger proportion of unseen data. The 80%:20% split produced similar trends, with Random Forest and XGBoost again dominating performance, while KNN, ANN, and Decision Tree showed moderate results, and Naive Bayes continued to struggle with recall, misclassifying a significant number of positive instances. In the 90%:10% split, although the smaller test size led to slightly higher performance variance, Random Forest, XGBoost, and SVM still maintained superior results, while Naive Bayes remained the weakest across all scenarios. These findings indicate that ensemble-based models and SVM are the most reliable and robust classifiers in handling the dataset used, regardless of the data split ratio, whereas Naive Bayes is highly sensitive to data distribution and class imbalance.

In order to evaluate the effectiveness of the applied classification models, this study employs

four commonly used performance metrics, namely accuracy, precision, recall, and F-measure. These metrics are derived from the confusion matrix and provide a more detailed perspective on model performance beyond simple classification outcomes. The evaluation is conducted across three data-splitting scenarios (70%:30%, 80%:20%, and 90%:10%), enabling a comprehensive comparison of the relative strengths and limitations of each algorithm under different proportions of training and testing data.

Tables 7, 8, and 9 present a comprehensive summary of the performance evaluation results for all classification algorithms across the three data-splitting scenarios (70%:30%, 80%:20%, and 90%:10%). This table consolidates the values of accuracy, precision, recall, and F-measure, allowing for a clearer comparison of model performance and highlighting which algorithms consistently deliver optimal results under varying experimental conditions.

Table 7. Comparison of Model Performance on 70:30 Scenario

Model/Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor	82.61%	82.65%	82.61%	82.62%
Naive Bayes	62.06%	76.34%	62.06%	58.60%
Decision Trees	91.70%	91.70%	91.70%	91.69%
Support Vector Machine	94.47%	94.49%	94.47%	94.45%
Artificial Neural Network	89.33%	89.32%	89.33%	89.32%
Random Forest	94.86%	94.87%	94.86%	94.86%
Extreme Gradient Boosting	94.86%	94.87%	94.86%	94.86%

Table 8. Comparison of Model Performance on 80:20 Scenario

Model/Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor	84.02%	84.25%	84.02%	84.06%
Naive Bayes	52.07%	73.24%	52.07%	42.97%
Decision Trees	91.72%	91.72%	91.72%	91.72%
Support Vector Machine	91.72%	91.72%	91.72%	91.72%
Artificial Neural Network	89.94%	90.03%	89.94%	89.96%
Random Forest	92.90%	92.94%	92.90%	92.91%
Extreme Gradient Boosting	92.90%	92.94%	92.90%	92.91%

Table 9. Comparison of Model Performance on 90:10 Scenario

Model/Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor	83.53%	83.58%	83.53%	83.51%
Naive Bayes	58.82%	77.79%	58.82%	51.39%
Decision Trees	90.59%	90.59%	90.59%	90.59%
Support Vector Machine	94.12%	94.33%	94.12%	94.10%
Artificial Neural Network	89.41%	89.58%	89.41%	89.39%
Random Forest	95.29%	95.40%	95.29%	95.30%
Extreme Gradient Boosting	94.12%	94.15%	94.12%	94.12%

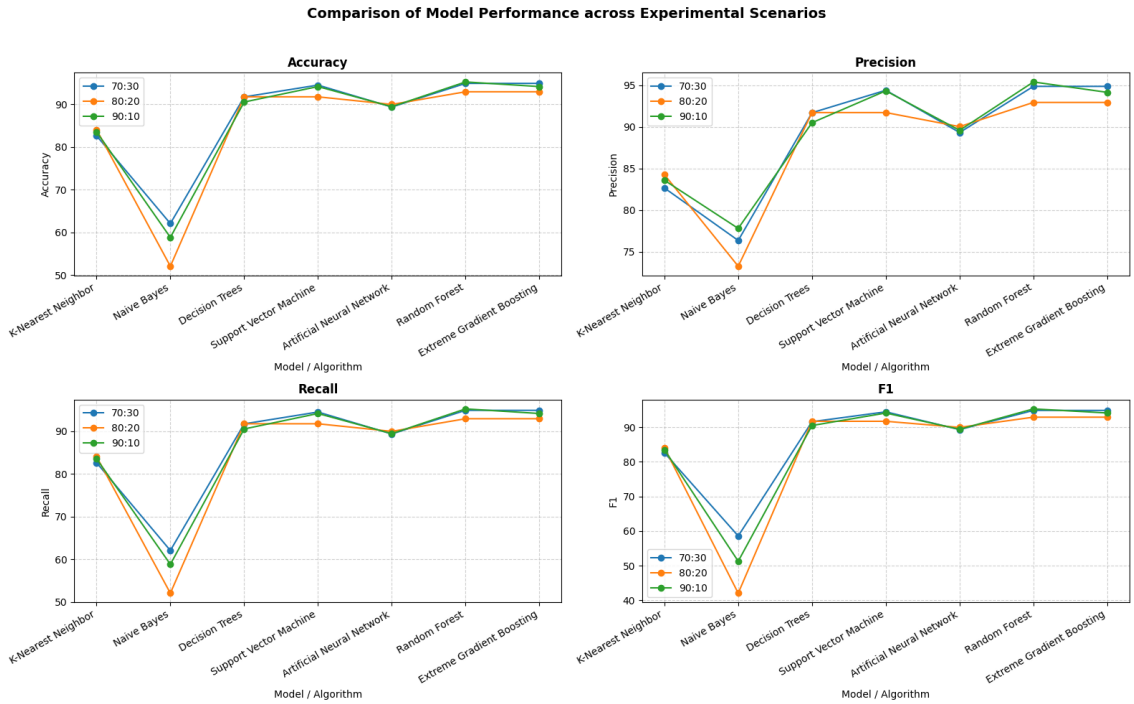

Figure 2. Comparison of model performance across experimental scenarios

Based on Table 7, Table 8, Table 9, and Figure 2, the evaluation of model performance with a 90% training and 10% testing data split shows that Random Forest achieved the highest accuracy of 95.29%, followed by XGBoost and SVM with values of 94.12%. KNN obtained an accuracy of 83.53%, while Naive Bayes recorded the lowest accuracy of 58.82%. Precision, recall, and f1-score were consistent with accuracy, with Random Forest achieving the highest score across all metrics. These results indicate that Random Forest is the most effective model for classification in this scenario, followed by XGBoost and SVM.

In the 80% training and 20% testing data split, Random Forest and XGBoost both achieved the highest accuracy of 92.90%, followed by SVM and Decision Tree, each with an accuracy of 91.72%. KNN achieved an accuracy of 84.02%, while Naive Bayes again showed the weakest performance with an accuracy of 52.07%. Precision, recall, and f1-score followed a consistent pattern with accuracy, confirming that Random Forest and XGBoost are the most effective models, followed by SVM and Decision Tree.

For the 70% training and 30% testing data split, Random Forest and XGBoost once again achieved the highest accuracy, each with 94.86%, followed by SVM with 94.47%. The Decision Tree also performed well, with an accuracy of 91.70%, while the ANN achieved 89.33%. KNN recorded an accuracy of 82.61%, and Naive Bayes remained the weakest with 62.06%. In terms of precision, recall, and f1-score, Random Forest and XGBoost consistently delivered the best performance, followed by SVM and Decision Tree.

Overall, across all three scenarios (90%:10%, 80%:20%, and 70%:30% data splits), Random Forest and XGBoost consistently demonstrated the best performance in terms of accuracy, precision, recall, and f1-score. In the 90%:10% split, Random Forest achieved the highest accuracy of 95.29%, while in the 80%:20% and 70%:30% splits, both Random Forest and XGBoost produced equally high accuracies of 92.90% and 94.86%, respectively. SVM and Decision Tree also showed stable performance, consistently ranking third and fourth after Random Forest and XGBoost. On the other hand, KNN and particularly Naive Bayes showed lower performance, with Naive Bayes consistently obtaining the lowest accuracy across all scenarios. In conclusion, Random Forest and XGBoost are the most effective and reliable models for data classification across all tested data split scenarios, followed by SVM and Decision Tree.

The comparative evaluation of model performance across three different data split scenarios (90%:10%, 80%:20%, and 70%:30%) highlights the consistent superiority of ensemble-based algorithms, particularly Random Forest and XGBoost, which achieved the highest accuracies and balanced metric scores in all cases. Random Forest attained the single highest accuracy of 95.29% in the 90%:10% split, while in the 80%:20% and 70%:30% splits, both Random Forest and XGBoost maintained equally strong performance

with accuracies exceeding 92%. This stability can be attributed to their ensemble mechanisms, where Random Forest reduces variance by aggregating multiple decision trees and XGBoost minimizes error iteratively through gradient boosting and regularization, thereby ensuring resilience against overfitting and improving generalization. SVM also demonstrated robust and stable performance, consistently ranking just below the ensemble methods due to its ability to maximize class separation in high-dimensional spaces, while Decision Tree emerged as a moderately strong performer, benefitting from its interpretability but showing some sensitivity to data partitioning. Conversely, KNN exhibited lower stability and accuracy, reflecting its reliance on local data density and sensitivity to feature scaling, while Naive Bayes consistently underperformed, likely due to its conditional independence assumption being misaligned with the correlated nature of the features. ANN, although capable of capturing nonlinear patterns, delivered only moderate results, possibly constrained by the absence of advanced architectural tuning. Importantly, the alignment of precision, recall, and f1-scores with accuracy across all models confirms that performance differences were not driven by class imbalance, but by the intrinsic capacity of the algorithms to model complex decision boundaries. Taken together, these findings provide strong empirical evidence that Random Forest and XGBoost are the most reliable and effective models across varying data availability conditions, with SVM and Decision Tree serving as viable secondary alternatives, whereas KNN and Naive Bayes are less suited for this classification task.

CONCLUSION

In conclusion, this study demonstrates that ensemble-based algorithms, particularly Random Forest and XGBoost, consistently deliver the highest performance across all data split scenarios, with stable accuracy, precision, recall, and f1-scores. Support Vector Machine and Decision Tree also show competitive results, although slightly lower, while KNN and especially Naive Bayes exhibit relatively weak performance, indicating their limited suitability for the classification task in this context. Despite these findings, the study has several limitations. First, the evaluation was conducted without hyperparameter tuning, which may have constrained the optimal potential of certain algorithms, such as ANN and SVM. Second, the dataset used in this study, although sufficient for comparative evaluation, may not fully represent more complex or large-scale real-world scenarios. Finally, the study focused exclusively on traditional machine learning algorithms without incorporating deep learning approaches, which could provide additional insights, albeit at higher computational costs. Future research is therefore recommended to explore advanced optimization strategies, including hyperparameter tuning, feature engineering, and hybrid feature selection methods, to enhance model performance further. Moreover, extending the analysis to larger and more diverse datasets, as well as integrating deep learning and hybrid ensemble techniques, would provide a broader perspective on the generalizability and scalability of classification models in real-world applications.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to the Department of Information Systems and the Department of Research and Community Service at Universitas Amikom Yogyakarta for their continuous support and facilitation in completing this research. Their encouragement and provision of resources have been instrumental in enabling the successful execution of this study.

REFERENCES

Akula, B., Reddy, Dr. K. I., N, D., & RS, P. (2023). Advances in soil fertility classification: Data mining approach. *International Journal of Statistics and Applied Mathematics*, *8*(5S), 475–481. https://doi.org/10.22271/maths.2023.v8.i5sg.1240

Blesslin Sheeba, T., Anand, L. D. V., Manohar, G., Selvan, S., Wilfred, C. B., Muthukumar, K., Padmavathy, S., Ramesh Kumar, P., & Asfaw, B. T. (2022). Machine Learning Algorithm for Soil Analysis and Classification of Micronutrients in IoT-Enabled Automated Farms. *Journal of Nanomaterials*, *2022*(1), 5343965. https://doi.org/https://doi.org/10.1155/2022/5343965

Bouasria, A., Bouslihim, Y., Gupta, S., Taghizadeh-Mehrjardi, R., & Hengl, T. (2023). Predictive performance of machine learning model with varying sampling designs, sample sizes, and spatial extents. *Ecological Informatics*, *78*. https://doi.org/10.1016/j.ecoinf.2023.102294

Bouslihim, Y., John, K., Miftah, A., Azmi, R., Aboutayeb, R., Bouasria, A., Razouk, R., & Hssaini, L. (2024). The effect of covariates on Soil Organic Matter and pH variability: a digital soil mapping approach using random forest model. *Annals of GIS*, *30*(2), 215–232. https://doi.org/10.1080/19475683.2024.2309868

Fidiyanto, N., & Izzati, A. N. (2024). Penerapan Data Mining Klasifikasi Lahan Tanam Buah Alpukat dengan Algoritma Naïve Bayes. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, *5*(2), 95–103. https://doi.org/10.37148/bios.v5i2.125

Hanif, N. A., Hannats, M., Ichsan, H., & Budi, A. S. (2022). *Rancangan Sistem Klasifikasi Kesuburan Tanah pada Tanaman Pangan berdasarkan PH dan Kelembapan berbasis Arduino Nano menggunakan Metode K-NN dan Aplikasi Android* (Vol. 6, Issue 8). http://j-ptiik.ub.ac.id

Jaiswal, R. (2024, August 17). *Soil Fertility Dataset*. Kaggle: Https://Www.Kaggle.Com/Datasets/Rahuljaiswalonkaggle/Soil-Fertility-Dataset.

Mallah, S., Delsouz Khaki, B., Davatgar, N., Scholten, T., Amirian-Chakan, A., Emadi, M., Kerry, R., Mosavi, A. H., & Taghizadeh-Mehrjardi, R. (2022). Predicting Soil Textural Classes Using Random Forest Models: Learning from Imbalanced Dataset. *Agronomy*, *12*(11). https://doi.org/10.3390/agronomy12112613

Mukhtar, H., Maulina Syafutri, T., Aulia Rahman, R., Putra, A., Hafsari, R., Ilmu Komputer, F., & Muhammadiyah Riau, U. (2024). *Analisis Kesuburan Pertanian Melalui Irigasi Dengan Menggunakan Metode K-Means Clustering* (Vol. 4, Issue 2).

Pradana, M. R., Hannats, M., Ichsan, H., & Akbar, S. R. (2023). *Klasifikasi Kesuburan dan Daya Ukur Cakupan Kelembaban Tanah pada Tanaman Jambu Merah berbasis Arduino* (Vol. 7, Issue 4). http://j-ptiik.ub.ac.id

Pramoedyo, H., Ariyanto, D., & Aini, N. N. (2022). Comparison of Random Forest and Naive Bayes Methods for Classifying and Forecasting Soil Texture in The Area Around DAS Kalikonto East Java. *Barekeng: Journal of Mathematics and Its Application*, *16*(4), 1411–1422. https://doi.org/10.30598/barekengvol16iss4pp1411-1422

Reddy, B. B., Maragatham, S., Santhi, R., Balachandar, D., Vijayalakshmi, D., Davamani, V., Vasu, D., & Gopalakrishnan, M. (2024). Predictive soil mapping using random forest models: Applications in pH and soil organic matter assessment. *Plant Science Today*, *11*(4), 463–474. https://doi.org/10.14719/pst.3865

Sarangi, A., Raula, S. K., Ghoshal, S., Kumar, S., Kumar, C. S., & Padhy, N. (2024). Enhancing Process Control in Agriculture: Leveraging Machine Learning for Soil Fertility Assessment †. *Engineering Proceedings*, *67*(1). https://doi.org/10.3390/engproc2024067031

Siahaan, A., Hannats, M., Ichsan, H., & Fitriyah, H. (2023). *Implementasi Fuzzy K-Nearest Neighbor dalam Sistem Klasifikasi Kualitas Tanah pada Tanaman Kedelai berdasarkan Kelembapan dan pH Tanah menggunakan Arduino* (Vol. 7, Issue 5). http://j-ptiik.ub.ac.id

Supriyanto, S., & Atwa Magriyanti, A. (2022). Perancangan Sistem Monitoring Kualitas Tanah Sawah Dengan Parameter Suhu Dan Kelembaban Tanah Menggunakan Arduino Berbasis Internet Of Things (IoT). *JURNAL ILMIAH ELEKTRONIKA DAN KOMPUTER*, *15*(2), 234–241. http://journal.stekom.ac.id/index.php/elkom□page234

Wadoux, A. M. J.-C., Samuel-Rosa, A., Poggio, L., & Mulder, V. L. (2020). A note on knowledge discovery and machine learning in digital soil mapping. *European Journal of Soil Science*, *71*(2), 133–136. https://doi.org/https://doi.org/10.1111/ejss.12909