BENCHMARKING OF MACHINE LEARNING ALGORITHMS FOR FERTILIZER RECOMMENDATION IN PRECISION AGRICULTURE

MILITARU FLORIN DANIEL[1], CIOLAC RAMONA[1], FIRU-NEGOESCU ADRIAN[1], MOISA SEBASTIAN[1], POPESCU GABRIELA*[1]
[1] University of Life Sciences "King Mihai I" from Timisoara, Faculty of Management and Rural Tourism, Timisoara, Romania
*Corresponding author's e-mail: gabrielapopescu@usvt.ro

Abstract: *This paper benchmarks fourteen machine learning algorithms for fertilizer recommendation in precision agriculture using soil, crop, and environmental data. Ensemble-based models—particularly Gradient Boosting, Random Forest, and LightGBM—achieved the highest accuracy and robustness, effectively modeling nonlinear agronomic relationships. In contrast, linear and distance-based methods showed limited adaptability. The results confirm that ensemble learning offers the most reliable and efficient framework for data-driven fertilizer recommendation, contributing to sustainable and resource-efficient crop management.*

Key words*: machine learning, fertilizer recommendation, benchmarking, precision agriculture*

INTRODUCTION

Precision agriculture has increasingly been recognized as a transformative paradigm in modern crop production, representing a shift from traditional uniform management toward data-driven and site-specific decision-making. By integrating advanced sensing technologies, geospatial analytics, and computational modeling, this approach enables the optimization of crop performance while minimizing resource waste and environmental impact [26,28]. Among the various components of precision agriculture, fertilizer management plays a pivotal role. Although fertilizers are indispensable for achieving optimal yields, their inefficient or excessive use can result in soil degradation, nutrient leaching, greenhouse gas emissions, and water contamination, posing major challenges to the sustainability of agroecosystems [28,47].

Traditional fertilizer application strategies, which often rely on fixed schedules or expert judgment, tend to disregard the spatial and temporal variability inherent in soil fertility, crop nutrient demand, and climatic conditions. Consequently, such practices frequently lead to suboptimal nutrient use efficiency and reduced economic and environmental sustainability [26,47]. Addressing these limitations requires adaptive, data-informed approaches capable of capturing complex interactions between environmental and agronomic factors.

In this context, machine learning (ML) has emerged as a powerful analytical framework capable of improving fertilizer recommendation systems through predictive modeling. By leveraging diverse datasets—including soil physicochemical attributes, weather parameters, crop phenology, and historical yield data—ML algorithms can identify patterns and model nonlinear relationships between multiple input variables and crop responses. Empirical studies have demonstrated that algorithms such as Random Forest, XGBoost, and neural networks provide superior performance in predicting optimal fertilizer rates compared to traditional statistical or rule-based methods [14,33,49].

A systematic review [14] highlighted the growing relevance of ML-based approaches for nutrient and fertilizer status estimation, underscoring their potential to support sustainable nutrient management. Similarly, Tanaka et al. [49] emphasized that yield prediction accuracy, while critical, does not automatically translate into reliable fertilizer recommendations, unless models are carefully calibrated and validated for

specific agroecological contexts. These findings point to the need for further research aimed at improving model interpretability, transferability, and robustness under varying environmental and management conditions.

Despite the promising results obtained so far, several challenges continue to hinder the large-scale adoption of ML-driven fertilizer recommendation systems. Chief among these are the integration of heterogeneous data sources, the scalability of predictive frameworks, and the interpretability of model outputs for practical decision-making. Furthermore, the significant heterogeneity in soil properties, climatic conditions, and cropping systems across regions underscores the necessity of developing generalizable models capable of adapting to diverse agroecosystems.

The present study addresses these research gaps by evaluating and comparing the performance of multiple ML algorithms in predicting fertilizer requirements based on a comprehensive dataset encompassing key physicochemical and agronomic variables. Through a systematic assessment of model accuracy, computational efficiency, and generalization capacity, this work aims to identify methodological strategies that enhance the reliability and applicability of ML-based fertilizer recommendation systems, thereby contributing to the broader objective of promoting sustainable and resource-efficient agricultural practices.

In order to provide a comprehensive overview of the methodological approaches employed in precision agriculture, this section reviews fifteen supervised machine learning algorithms that have been widely applied in data-driven fertilizer recommendation and crop management systems. These algorithms represent probabilistic, linear, ensemble, and neural network paradigms, each grounded in distinct theoretical principles and learning mechanisms.

The Naive Bayes classifier is a generative, probabilistic method that applies Bayes' theorem to compute posterior class probabilities from the class prior and the class-conditional likelihood [1,41]. The model adopts a conditional independence assumption so that the joint likelihood factorises as a product of marginal likelihoods for individual features; this reduces parameter complexity and enables efficient parameter estimation even in high-dimensional settings [36,41]. Despite the simplification entailed by the independence assumption, Naive Bayes often yields robust classification decisions in practice and serves as a computationally inexpensive baseline in many applied pipelines [36,41].

Random Forest constructs an ensemble of decision trees by training each tree on a bootstrap sample of the training data and by selecting, at each split, a random subset of features [6,29]. Predictions are obtained by aggregating individual tree outputs (majority vote for classification, average for regression). This procedure reduces variance relative to single trees while retaining the ability to capture nonlinear feature interactions; the built-in feature subsampling also renders the ensemble robust to correlated predictors and noisy inputs [29,35]. Random Forest is commonly applied in environmental and agricultural domains where heterogeneous inputs (remote sensing, soil, weather) require a model that is robust and relatively insensitive to hyperparameter tuning [29,48].

A decision tree models the mapping from features to class labels by recursively partitioning the feature space: at each internal node, a split (feature + threshold) is chosen to maximize a purity criterion (e.g., information gain, Gini decrease) and leaves assign class labels or predictive values [40,44]. The resulting tree encodes a set of explicit if–then rules that are straightforward to interpret. Unconstrained trees may overfit to idiosyncrasies of the training data; therefore, pruning, depth limits, or cost-complexity control are typically applied to improve generalization and stability [7,40]. Decision trees remain central both as interpretable standalone models and as base learners for ensembles [7,44].

ExtraTrees increases randomness relative to Random Forest by selecting split thresholds at random (in addition to randomly selecting features) rather than optimizing split points on the training sample [19]. Trees are typically grown on the whole sample (no bootstrap) but with randomized splits; the ensemble prediction is obtained by aggregation across trees. The additional randomness decreases correlation among trees and reduces variance of the aggregated predictor while offering fast training, although individual trees are biased relative to optimally split trees [19]. ExtraTrees is useful in settings where computational efficiency and variance reduction are priorities.

LightGBM is a gradient-boosting framework engineered for computational efficiency and scalability on large, sparse tabular datasets. The implementation applies leaf-wise tree growth (choosing the leaf with maximal loss reduction) and uses optimizations such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to reduce the effective data volume and feature dimension processed during training [27, 45]. These innovations accelerate training and often yield strong predictive performance; however, the leaf-wise strategy can lead to deeper, more complex trees and therefore demands careful regularization to avoid overfitting [2,45]. Comparative studies place LightGBM among the fastest and most practical GBDT implementations for production use [2].

XGBoost (Extreme Gradient Boosting) is a highly optimized implementation of gradient boosting that augments the standard boosting objective with regularization terms (L1/L2) and employs second-order (Newton) approximations for more accurate step updates [8]. Additional system-level optimizations (e.g., sparse-aware algorithms, block structure for parallelization, approximate split search and pruning) improve computational efficiency on structured/tabular data [8,25]. XGBoost frequently attains top performance in applied tabular tasks, balancing predictive power with means to control model complexity via explicit regularizers [8].

Gradient Boosting constructs a strong predictor as an additive ensemble of weak learners (commonly shallow trees) that are fit sequentially: each new learner is trained to approximate the negative gradient (residual) of the loss function with respect to the current ensemble prediction [13,18]. Model capacity and overfitting are controlled via hyperparameters such as learning rate, tree depth, and subsampling; appropriate regularization and early stopping are required to ensure generalization [13,18]. The method is effective at capturing complex nonlinearities and interactions in structured data and forms the conceptual basis for modern implementations such as XGBoost, LightGBM, and CatBoost [2,18].

Linear Discriminant Analysis (LDA) is a generative linear classifier that assumes class-conditional multivariate normal distributions with a common covariance matrix. Under these assumptions, LDA yields linear decision functions that maximize the ratio of between-class variance to within-class variance [16,31]. LDA performs well when Gaussianity and homoscedasticity approximately hold and remains useful as a low-variance, interpretable baseline; numerous modern variants introduce regularization and spectral corrections to address small-sample and high-dimensional regimes [31,39].

CatBoost is a gradient-boosting implementation explicitly designed to handle categorical features natively. It uses ordered target statistics and an ordered boosting procedure to prevent target leakage when computing categorical encodings and applies symmetric tree structures that improve robustness and inference efficiency [2,37]. These design choices reduce the need for ad-hoc preprocessing of categorical variables and often yield competitive or superior predictive performance on mixed tabular datasets [2,37].

The Multi-Layer Perceptron (MLP) is a feedforward neural network composed of one or more fully connected hidden layers with nonlinear activation functions [5,42].

Forward propagation computes successive affine transformations followed by activations, while training employs backpropagation to compute gradients of a loss function and a gradient-based optimizer to update parameters [5, 43]. With sufficient capacity and appropriate regularization, MLPs can approximate complex nonlinear mappings; recent benchmark work has demonstrated that carefully tuned MLP architecture remains highly competitive on tabular tasks [9,23].

Support Vector Machines (SVMs) formulate classification as the search for a hyperplane that maximizes the margin between classes, yielding a solution with strong theoretical generalization guarantees [10,46]. For nonlinearly separable data, kernel functions (e.g., RBF) implicitly map inputs into a higher-dimensional feature space where a linear separator may exist; the kernel trick permits this mapping without explicit computation [46]. SVM performance depends critically on kernel choice and hyperparameters (C, kernel scale), and SVM variants remain widely used as robust baselines in applied domains [20,46].

Logistic regression is a discriminative linear model that models the log-odds of class membership as a linear combination of input features and estimates parameters by maximizing the (regularized) likelihood [12,24]. The model yields interpretable coefficients and probabilistic outputs and is computationally efficient and statistically well understood. Regularization (L1/L2) and careful validation are used to control overfitting in higher-dimensional settings [24,50].

K-Nearest Neighbors (KNN) is a nonparametric, instance-based method that assigns a query point the class most common among its k nearest training examples according to a chosen distance metric [3,11]. The algorithm requires no explicit training phase beyond storing the dataset; classification incurs a nearest-neighbour search at inference time. KNN's performance is sensitive to feature scaling, the choice of k, and the curse of dimensionality [3,21].

The Perceptron is a simple linear classifier that updates a weight vector incrementally according to misclassification errors: when an example is misclassified, the weight vector is adjusted by a constant multiple of the input vector [42]. The perceptron algorithm converges in a finite number of updates if and only if the training data are linearly separable; otherwise the algorithm will not converge to a bounded solution [34,42]. Despite its simplicity and limitations (inability to represent nonlinearly separable functions), the Perceptron remains a fundamental building block in the theory of linear classifiers and online learning [34].

Quadratic Discriminant Analysis (QDA) is a generative classifier that models each class by a multivariate Gaussian with its own covariance matrix; the resulting decision surfaces are quadratic in the input features [22,31]. This flexibility allows QDA to capture heteroscedastic class structure but increases the number of parameters to estimate and sensitivity to limited sample sizes; modern variants therefore incorporate shrinkage or spectral regularization to stabilize covariance estimation [30,31].

MATERIALS AND METHODS

The research design was structured to facilitate a comparative evaluation of multiple machine learning (ML) algorithms with respect to their applicability in fertilizer recommendation for precision agriculture. The analysis was performed using a publicly available dataset obtained from the Kaggle platform.

The data set comprises 3,100 records, which represent the different combinations of agronomic and environmental variables. It contains a list of important physicochemical and agronomic parameters—Temperature (Temp), Moisture (Moist), Rainfall (Rain), pH, Nitrogen (N), Phosphorus (P), Potassium (K), Carbon (C), Soil type (Soil), and Crop type

(Crop)—all of which together work as important indicators for soil fertility evaluation and fertilizer optimization [28]. They were chosen by considering how well factors relate to estimating crop nutrient needs and affecting fertilizer response efficiency.

The database includes 31 crop varieties, ranging from staple crops to high-value crops like rice, wheat, maize, coffee, cotton, mango, orange, and papaya. Similarly, ten fertilizer categories are supported, both organic and inorganic fertilizers: Compost, Organic Fertilizer, Balanced NPK Fertilizer, Water Retaining Fertilizer, Gypsum, Lime, Diammonium Phosphate (DAP), Urea, Muriate of Potash, and General-Purpose Fertilizer. This heterogeneity offers a sound foundation on which to build and test ML algorithms across a broad spectrum of agronomic situations and management systems.

A records sample is shown in Table 1, which provides an overview of the design, types of variables, and general heterogeneity of the data. Such diversity guarantees suitability of the data to assess the performance of algorithms under various soil–crop–climate interactions, a requirement to create generalizable fertilizer recommendation models.

Table 1.

Sample Dataset

Temp	Moist	Rain	PH	N	P	K	C	Soil	Crop	Fertilizer
22.386	0.227	292.74	5.9026	78.814	60.471	66.060	1.518	Peaty Soil	rice	Lime
21.342	0.78	249.98	5.6922	72.082	42.591	68.035	2.410	Peaty Soil	rice	DAP
25.658	0.756	250.70	6.6146	75.032	118.00	142.004	0.280	Loamy Soil	rice	Compost

Categorical variables were label-encoded, and numerical ones standardized using *StandardScaler* to ensure consistent scaling and model convergence. The dataset was split into 80% training (2480 records) and 20% testing (620 records) subsets, stratified by fertilizer type. All experiments were conducted in Python 3.11, employing scikit-learn, XGBoost, LightGBM, CatBoost, and TensorFlow/Keras.

The classification algorithms evaluated in this study were organized according to their theoretical learning principles into five categories [32], as summarized in Table 2.

Table 2.

Algorithms clsification

Category	Algorithms Included	Number of Algorithms
Probabilistic and Statistical Models	Logistic Regression (LR), Naive Bayes (NB), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA)	4
Distance-Based and Instance-Based Methods	K-Nearest Neighbors (KNN), Perceptron (P)	2
Margin-Based and Boundary Optimization Methods	Support Vector Machine (SVM)	1
Tree-Based and Ensemble Learning Methods	Decision Tree (DT), Random Forest (RF), Extra Trees (ET), Gradient Boosting (GB), XGBoost (XGB), Light Gradient Boosting Machine (LGBM), CatBoost (CB)	7
Connectionist and Neural Network Models	Multilayer Perceptron (MLP)	1

Model performance was measured using Accuracy, Precision, Recall, F1-score, Balanced Accuracy, Cohen's Kappa, and Matthews Correlation Coefficient (MCC). Class-specific indicators (True Positive, False Positive, False Negative, True Negative — TP, FP, FN, TN) and confusion matrices were generated [38]. Additionally, training and prediction times were recorded, and a composite score—the mean of normalized metrics—was computed for integrated performance comparison [15].

All models were saved for reproducibility, and results, including detailed metrics, radar charts, and comparative plots, were exported to Excel for visualization and analysis, following standard reporting practices in precision agriculture ML research [26].

RESEARCH RESULTS

The results presented in Table 2 show that tree-based and ensemble learning methods achieved the highest overall performance across all evaluation metrics, with Gradient Boosting, Random Forest, and LightGBM exhibiting particularly strong results (Accuracy > 0.99 and F1 > 0.98).

In contrast, probabilistic and linear models (such as Naive Bayes, Logistic Regression, LDA, and QDA) displayed lower predictive accuracy and stability, confirming their limited capacity to model nonlinear relationships within the analyzed dataset. The neural network model (Multilayer Perceptron) yielded intermediate results, outperforming linear models but remaining below ensemble-based methods. Instance-based methods (KNN, Perceptron) also recorded moderate performance, likely due to their sensitivity to parameter tuning and high dimensionality.

Table 3.

Category results

Category	Representative Algorithms	Mean Accuracy	Mean F1-score
Probabilistic and Statistical Models	LR, NB, LDA, QDA	0.64	0.55
Distance-Based and Instance-Based Methods	KNN, P	0.66	0.52
Margin-Based and Boundary Optimization	SVM	0.80	0.69
Tree-Based and Ensemble Learning Methods	DT, RF, ET, GB, XGB, LGBM, CB	0.99	0.98
Connectionist and Neural Network Models	MLP	0.90	0.85

The detailed benchmarking results for all fifteen models are shown in Table 3. Gradient Boosting achieved the best overall classification performance with an accuracy of 0.9968 and an F1-score of 0.9842, followed closely by Random Forest (Acc. = 0.9952) and LightGBM (Acc. = 0.9919). Among classical models, Logistic Regression and SVM performed significantly better than Naive Bayes or QDA, yet remained below the ensemble methods.

The lowest performance was recorded by Naive Bayes and QDA, both with an accuracy of 0.5403, suggesting strong limitations in capturing feature interactions within the data.

Table 4.

Algorithm Results

Model	Acc.	Prec.	Recall	F1	BalAcc	MCC	Kappa
Gradient Boosting	0.9968	0.9961	0.9742	0.9842	0.9742	0.9960	0.9960
Random Forest	0.9952	0.9911	0.9811	0.9851	0.9811	0.9940	0.9940
LightGBM	0.9919	0.9877	0.9851	0.9860	0.9851	0.9900	0.9899
Decision Tree	0.9919	0.9698	0.9809	0.9746	0.9809	0.9900	0.9899
CatBoost	0.9855	0.9782	0.9849	0.9811	0.9849	0.9820	0.9819
XGBoost	0.9839	0.9756	0.9739	0.9736	0.9739	0.9800	0.9799
MLP (two hidden layers)	0.8952	0.8470	0.8531	0.8491	0.8531	0.8698	0.8697
Extra Trees	0.8935	0.8591	0.7415	0.7728	0.7415	0.8668	0.8654
SVM	0.8048	0.8027	0.6661	0.6905	0.6661	0.7552	0.7498
Logistic Regression	0.7984	0.7087	0.6904	0.6833	0.6904	0.7468	0.7454
LDA	0.6645	0.5898	0.5506	0.5481	0.5506	0.5759	0.5672
KNN (k-NN)	0.6742	0.5778	0.5178	0.5346	0.5178	0.5882	0.5847
Perceptron	0.6500	0.5607	0.5605	0.5121	0.5605	0.5826	0.5725
Naive Bayes	0.5403	0.5827	0.5426	0.4833	0.5426	0.4537	0.4421
QDA	0.5403	0.5218	0.5465	0.4677	0.5465	0.4638	0.4510

Figure 1 illustrates the relative performance distribution among the main algorithmic families. The dominance of ensemble-based models is evident, with extremely high consistency across all performance indicators, whereas linear and probabilistic models exhibit substantial variance.

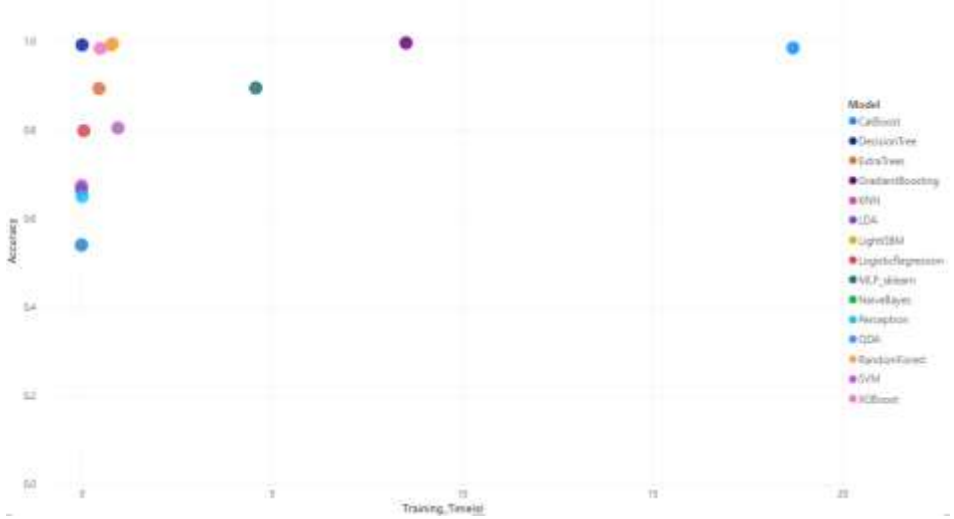

Figure 1. Algorithm performance

In addition to high predictive accuracy, ensemble-based models exhibited robust agreement metrics, with MCC and Kappa values exceeding 0.98. This indicates consistent classification reliability and minimal bias toward particular fertilizer classes. The strong Balanced Accuracy scores (≥ 0.97) further demonstrate their capacity to manage class imbalance effectively. Regarding efficiency, Gradient Boosting and LightGBM achieved superior Accuracy-to-training-time ratios, with training durations of 8.53 s and 0.77 s, respectively. Although CatBoost reached similar Accuracy levels, its training cost (18.70s) was considerably higher. Decision Tree and Random Forest provided a favorable trade-off between performance and computational cost, requiring less than one second to train while maintaining accuracies above 0.99. These findings are consistent with prior benchmarking studies, where gradient-boosted frameworks were recognized for delivering optimal performance-speed balance on structured agricultural data [2,4].

Table 5.

Algorithm training and predict time

Model	Train time (s)	Predict time (s)	Notes
Gradient Boosting	8.53	0.000	Highest Accuracy/F1; moderate training cost; very fast inference.
Random Forest	0.82	0.0167	Near-best performance with short training; stable across metrics.
LightGBM	0.77	0.0188	Top-tier performance; compact runtime; excellent class balance.
Decision Tree	**0.016**	0.000	Competitive accuracy with *minimal* training cost; simpler model.
CatBoost	**18.70**	0.0161	Strong performance, but the highest training cost among boosters.
XGBoost	0.49	**0.0040**	Slightly below top boosters; very fast inference.
MLP	4.58	0.000	Best non-ensemble; markedly below boosters in accuracy/F1.
SVM	0.96	**0.107**	Moderate accuracy; slowest inference among tested models.

The computational results indicate clear efficiency differences among the tested models. LightGBM and Random Forest achieved an optimal balance between accuracy and speed, both training in under one second while exceeding 0.99 in Accuracy and F1-score. Gradient Boosting offered the highest precision (0.9968) at a moderate cost, whereas CatBoost delivered comparable performance but required longer training (18.7 s). Decision Tree proved exceptionally fast (0.016 s) with acceptable accuracy, while XGBoost combined high accuracy with the fastest overall inference time (0.004 s).

Overall, ensemble-based methods showed both computational efficiency and predictive robustness, confirming their suitability for real-time fertilizer recommendation systems. In contrast, linear and kernel models such as SVM were slower and less accurate, making them less practical for operational deployment in precision agriculture.

CONCLUSIONS

The relative contrast, however, definitively shows that ensemble-based methods are clearly superior to all other forms of models in prediction quality as well as stability. Such consistent superiority is corroborated by earlier work in the agricultural machine learning community, which has underlined the versatility of ensemble tree-based learners to varied and nonlinear agronomic information [26,28,49]. In particular, Random Forest and Gradient Boosting worked best of all, as was also the case in findings set out by [17] and [2] who indicated the same advantages when comparing gradient-boosted decision tree models for classifying and predicting tasks.

LightGBM and CatBoost also yielded competitive performance, but their trends had different computational behaviors. These agree with the efficiency-oriented architecture of LightGBM [27] and with categorical feature improvement inherent in CatBoost [37]. The Multilayer Perceptron (MLP) network yielded good performance, which aligns with trends found in recent neural network research [23], but its predictability lagged behind boosting-based methods by a small margin.

On the other hand, linear and distance classifiers like Logistic Regression, Support Vector Machine, Linear and Quadratic Discriminant Analysis, and K-Nearest Neighbors were found to have limited capacity to model the multivariate and nonlinear relationships inherent in fertilizer recommendation tasks. Such a finding supports the hypothesis that such a task intrinsically demands algorithms that can model complex multivariate dependencies among soil, crop, and environmental variables.

Lastly, the findings confirm that ensemble learning frameworks, including Gradient Boosting, Random Forest, and LightGBM, are the most precise and computationally intelligent data-driven fertilizer suggestion algorithms for precision agriculture. Their capacity to integrate various input variables, handle nonlinear patterns, and be resilient on other datasets explains their promise as base technologies for sustainable and intelligent nutrition management platforms.

REFERENCES

[1]. ABBAS M., MEMON K. A., JAMALI A. A., MEMON S., AHMED A., 2019, Multinomial Naive Bayes classification model for sentiment analysis. IJCSNS Int. J. Comput. Sci. Netw. Secur, 19(3), 62. DOI:10.13140/RG.2.2.30021.40169

[2]. ALSHBOUL O., ALMASABHA G., SHEHADEH A., AL-SHBOUL K., 2024, A comparative study of LightGBM, XGBoost, and GEP models in shear strength management of SFRC-SBWS. Structures, 61, 106009. https://doi.org/10.1016/j.istruc.2024.106009

[3]. ALTMAN N. S., 1992, An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, 46(3), 175–185. https://doi.org/10.1080/00031305.1992.10475879
[4]. BENTÉJAC C., CSÖRGŐ A., MARTÍNEZ-MUÑOZ G., 2020, A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5
[5]. BISHOP C. M., 1995, Neural networks for pattern recognition. Oxford university press
[6]. BREIMAN L., 2001, Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
[7]. BREIMAN L., FRIEDMAN J. H., OLSHEN R. A., STONE C. J., 2017, Classification and Regression Trees. Routledge. https://doi.org/10.1201/9781315139470
[8]. CHEN T., GUESTRIN C., 2016, XGBoost : A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. https://doi.org/10.1145/2939672.2939785
[9]. CHERNOV A., 2025, (GG) MoE vs. MLP on Tabular Data, Version 1, arXiv. https://doi.org/10.48550/ARXIV.2502.03608
[10]. CORTES C., VAPNIK V., 1995, Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/bf00994018
[11]. COVER T., HART P., 1967, Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/tit.1967.1053964
[12]. COX D.R., 1958, The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society: Series B, 20, 215-242
[13]. DENG Y., LIU Y., ZHANG D., CAO Z., 2025, A Hybrid Gradient Boosting Model for Predicting Longitudinal Dispersion Coefficient in Natural Rivers. Water Resources Management, 39(5), 2111–2131. https://doi.org/10.1007/s11269-024-04058-6
[14]. ENNAJI O., BELGAID A., EL ALLALI A., 2023, Machine learning in nutrient management: A review. ScienceDirect. DOI:10.1016/j.aiia.2023.06.001
[15]. FERNÁNDEZ-DELGADO M., CERNADAS E., BARRO S., AMORIM D., 2014, Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research, 15, 3133–3181. https://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf
[16]. FISHER R. A., 1936, THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. Annals of Eugenics, 7(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
[17]. FLOREK P., ZAGDAŃSKI, A., 2023, Benchmarking state-of-the-art gradient boosting algorithms for classification, Version 1, arXiv. https://doi.org/10.48550/ARXIV.2305.17094
[18]. FRIEDMAN J. H., 2001, Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5, https://doi.org/10.1214/aos/1013203451
[19]. GEURTS P., ERNST D., WEHENKEL, L., 2006, Extremely randomized trees. Machine Learning, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1
[20]. GUIDO R., FERRISI S., LOFARO D., CONFORTI D., 2024, An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review. Information, 15(4), 235. https://doi.org/10.3390/info15040235
[21]. HALDER R. K., UDDIN M. N., UDDIN MD. A., ARYAL S., KHRAISAT, A., 2024, Enhancing K-nearest neighbor algorithm: a comprehensive review and performance

analysis of modifications. Journal of Big Data, 11(1, https://doi.org/10.1186/s40537-024-00973-y

[22]. HASTIE T., TIBSHIRANI R., FRIEDMAN J., 2009, The Elements of Statistical Learning. In Springer Series in Statistics. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

[23]. HOLZMÜLLER D., GRINSZTAJN L., STEINWART, I., 2024, Better by default: Strong pre-tuned mlps and boosted trees on tabular data. Advances in Neural Information Processing Systems, 37, 26577-26658

[24]. HOSMER D. W., LEMESHOW, S., 2000, Applied Logistic Regression. Wiley. https://doi.org/10.1002/0471722146

[25]. ILERI K., 2025, Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks. International Journal of Machine Learning and Cybernetics, 16(9), 6937–6956. https://doi.org/10.1007/s13042-025-02654-5

[26]. KAMILARIS A., PRENAFETA-BOLDÚ F. X., 2018, Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

[27]. KE G., MENG Q., FINLEY T., WANG T., CHEN W., MA W., Y, Q., LIU T.-Y., 2017, LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, December 2017, 3149-3157

[28]. LIAKOS K. G., BUSATO P., MOSHOU D., PEARSON S., BOCHTIS, D., 2018, Machine learning in agriculture: A review. Sensors, Basel, Switzerland), 18(8), 2674. https://doi.org/10.3390/s18082674

[29]. LIU Z., LUONG P., BOLEY M., SCHMIDT D. F., 2025, Improving Random Forests by Smoothing, Version 1, arXiv. https://doi.org/10.48550/ARXIV.2505.06852

[30]. LUO W., LI H., BAI Z., LIU Z., 2025, Spectrally-Corrected and Regularized QDA Classifier for Spiked Covariance Model, Version 1, arXiv. https://doi.org/10.48550/ARXIV.2503.13582

[31]. MCLACHLAN G.J., 2004, Discriminant Analysis and Statistical Pattern Recognition. Wiley, New York

[32]. MURPHY K. P., 2022, Probabilistic machine learning: an introduction. MIT press.

[33]. MUSANASE C., VODACEK A., HANYURWIMFURA D., UWITONZE A. KABANDANA I., 2023, Data-driven analysis and machine learning-based crop and fertilizer recommendation system for revolutionizing farming practices. Agriculture, 13(11), 2141. https://doi.org/10.3390/agriculture13112141

[34]. NOVIKOFF A., 1962, On convergence proofs on perceptrons. Proceedings of the Symposium on the Mathematical Theory of Automata, 12, 615–622

[35]. PARMAR A., KATARIYA R., PATEL V., 2018, A Review on Random Forest: An Ensemble Classifier. In Lecture Notes on Data Engineering and Communications Technologies, pp. 758–763, Springer International Publishing. https://doi.org/10.1007/978-3-030-03146-6_86

[36]. PERETZ O., 2024, Naive Bayes classifier – An ensemble procedure for recall. Expert Systems with Applications, 239, 122559. https://doi.org/10.1016/j.engappai.2024.108972

[37]. PROKHORENKOVA L., GUSEV G., VOROBEV A., DOROGUSH A.V. GULIN A., 2018, Catboost: Unbiased Boosting with Categorical Features. Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, 3-8 December 2018, 6639-6649

[38]. PROVOST F. FAWCETT T., 2013, Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media

[39]. QU L., PEI Y., 2024, A Comprehensive Review on Discriminant Analysis for Addressing Challenges of Class-Level Limitations, Small Sample Size, and Robustness. Processes, 12(7), 1382. https://doi.org/10.3390/pr12071382

[40]. QUINLAN J. R., 1986, Induction of decision trees. Machine Learning, 1(1), 81–106. https://doi.org/10.1007/bf00116251

[41]. RISH I., 2001, An empirical study of the naive Bayes classifier. IJCAI 2001 Workshop on Empirical Methods in AI, 3(22), 41–46.

[42]. ROSENBLATT F., 1958, The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408. https://doi.org/10.1037/h0042519

[43]. RUMELHART D. E., HINTON G. E., WILLIAMS R. J., 1986, Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0

[44]. SAFAVIAN S. R., LANDGREBE D., 1991, A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660–674. https://doi.org/10.1109/21.97458

[45]. SANKARI C., VITHYAVIGASINI S. P., VISHVAJIT S., ANTO JEBA INFANT M., RETHANISHA J., 2025, June, AI-Driven IoT-Enabled Soil Nutrient and Moisture Monitoring with XGBoost and LightGBM-Based Predictive Irrigation and Fertilization Optimization for Sustainable Precision Agriculture. In 2025 11th International Conference on Communication and Signal Processing, ICCSP) (pp. 795-800, IEEE. 10.1109/ICCSP64183.2025.11089358

[46]. SCHOLKOPF B., SMOLA A. J., 2002, Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT Press.

[47]. SHAMSHIRI R. R., JONES J. W., THORP K. R., AHMAD D., MAN H. C. TAHERI, S., 2018, Review of optimum temperature, humidity, and vapour pressure deficit for microclimate evaluation and control in greenhouse cultivation of tomato: a review. International agrophysics, 32(2), 287-302. doi: 10.1515/intag-2017-0005

[48]. SHOKATI H., MASHAL M., NOROOZI A., ABKAR A. A., MIRZAEI S., MOHAMMADI-DOQOZLOO Z., TAGHIZADEH-MEHRJARDI R., KHOSRAVANI P., NABIOLLAHI K., SCHOLTEN, T., 2024, Random Forest-Based Soil Moisture Estimation Using Sentinel-2, Landsat-8/9, and UAV-Based Hyperspectral Data. Remote Sensing, 16(11), 1962. https://doi.org/10.3390/rs16111962

[49]. TANAKA T. S., HEUVELINK G. B. M., MIENO T., 2024, Can machine learning models provide accurate fertilizer recommendations? Precision Agriculture, 25, 1839–1856. https://doi.org/10.1007/s11119-024-10136-x

[50]. WEIGARD A., SPENCER, R. J., 2022, Benefits and challenges of using logistic regression to assess neuropsychological performance validity: Evidence from a simulation study. The Clinical Neuropsychologist, 37(1), 34–59. https://doi.org/10.1080/13854046.2021.2023650