IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 3, March, 2026, pp. 707-721
Journal homepage: http://iieta.org/journals/isi

Machine Learning and Deep Learning Approaches for Crop Yield Prediction: A Comprehensive Analysis of Model Performance and Computational Efficiency

S Jayanthi[1*], M. A. Josephine Sathya[2], B. Nathan[3], Karthik Karmakonda[4], Muthuvel Laxmikanthan[5], Manojkumar V[6]

[1] Department of Artificial Intelligence and Data Science, Faculty of Science and Technology (IcfaiTech), The ICFAI Foundation for Higher Education, Hyderabad 501203, Telangana, India
[2] Department of Computer Science and Applications, Christ Academy Institute for Advanced Studies, Bangalore 560083, Karnataka, India
[3] Department of Computer Science Engineering, Dhaanish Ahmed Institute of Technology, Coimbatore 641105, Tamil Nadu, India
[4] Department of Computer Science Engineering, CVR College of Engineering, Hyderabad 501510, Telangana, India
[5] Department of Artificial Intelligence and Data Science, Dhaanish Ahmed Institute of Technology, Coimbatore 641105, Tamil Nadu, India
[6] Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India

Corresponding Author Email: drsjayanthicse@gmail.com

https://doi.org/10.18280/isi.310305

Received: 23 September 2025
Revised: 10 December 2025
Accepted: 20 March 2026
Available online:31 March 2026

Keywords:
crop yield prediction, machine learning, deep learning, ensemble models, Random Forest, XGBoost, recurrent neural networks, computational efficiency, agricultural forecasting

ABSTRACT

Accurate crop yield prediction (CYP) is crucial for ensuring global food security and guiding agricultural planning. In this study, we explore the effects of environmental factors like temperature, rainfall, and pesticide use on crop yield by analyzing a comprehensive multiyear, multi-region dataset. We benchmarked seventeen machine learning (ML) and deep learning (DL) models, including Random Forest (RF), XGBoost, Bagging Regressor (BR), Gradient Boosting (GB), and various recurrent neural networks (RNNs). To ensure fairness, all models were evaluated under identical experimental conditions, including preprocessing, train-test splits, and standardized performance metrics. Model performance was assessed using 10-fold cross-validation and robustness analysis through bootstrap resampling. Our results showed that RF (R^2 = 0.9986, MSE = 1.08×10^7) and BR (R^2 = 0.9984, MSE = 1.32×10^7) achieved the best performance, with RF providing the highest accuracy. However, BR was more practical, with faster training times and lower memory usage (11.32 MB). While recurrent neural network (RNN) models performed similarly to ensemble methods in terms of accuracy, they incurred higher computational costs (e.g., 936 s training time). We also propose the AgroStackNet ensemble model, which combines the strengths of the top models for better efficiency. Our findings highlight the importance of balancing computational efficiency and accuracy in CYP, contributing to the development of scalable, data-driven frameworks for agricultural forecasting.

1. INTRODUCTION

Unpredictable climate conditions and population growth have exacerbated the global challenge of ensuring food security [1-4]. In this study, yield (hg/ha) is considered the target variable, representing crop output per unit land area and distinguishing it from total production. We utilized temperature, rainfall, and pesticide usage data for crop yield prediction (CYP) across 101 countries over 21 years. Traditionally, CYP relied on statistical models that assume linearity and rigorous mathematical distributions. Consequently, they are often inadequate for capturing the complex, nonlinear, and dynamic interdependencies common in agronomic systems [5]. However, this scenario has dramatically shifted toward data-driven approaches, particularly machine learning (ML) and deep learning (DL). For example, Random Forest (RF), Support Vector Machines (SVM), and Gradient Boosting (GB) algorithms in ML have provided encouraging results in recent studies [6-8]. These methods can handle high-dimensional, heterogeneous data and capture nonlinear feature interactions. DL models, particularly recurrent architectures like Long Short-Term Memory (LSTM) networks, are designed to capture temporal dependencies in sequential data; however, their effectiveness in tabular CYP datasets remains uncertain. These capabilities are crucial for processing diverse environmental and temporal data [9-13]. Despite these advancements, several limitations remain. Only a limited number of studies have benchmarked a comprehensive set of algorithms, particularly with respect to computational resource requirements. Though DL models are increasingly applied, their performance is rarely benchmarked across different architectural recurrent neural network (RNN)

variants (e.g., RNNs vs. LSTMs vs. attention-based models), and their computational implications are often overlooked.

To confront these gaps, we propose a multifaceted approach that includes carefully chosen preprocessing techniques, Exploratory Data Analysis (EDA) to gain more insights into crop yield, and systematic comparison of model efficiency. Unlike previous studies examining only a few models, we comprehensively evaluate 14 ML and 3 DL RNN variants for CYP using a multiyear, multiregional dataset. We selected these models to represent diverse learning paradigms, from traditional statistical methods to cutting-edge deep learning architectures. A key contribution of this study is the joint evaluation of predictive accuracy and computational efficiency under a unified experimental framework, ensuring a fair comparison across models.

Our study aims to address these challenges with the following objectives:

- To benchmark a broad set of ML and DL models for CYP, including ensemble methods and recurrent architectures.
- To evaluate models' performance based on prediction metrics (MSE, MAE, R^2) and deployment-oriented metrics (training time, inference time, and memory usage).
- To propose AgroStackNet and benchmark its performance against the other models.

The remainder of this paper is organized as follows: Section 2 presents the related work, Section 3 describes the materials and methods, Section 4 details the methodology and proposed AgroStackNet framework, Section 5 presents the results and discussion, and Section 6 concludes the study with future research directions.

2. RELATED WORK

Due to the agricultural dataset's nonlinear, complex relationships and high-dimensional nature, ML models have demonstrated improved capability in capturing such relationships compared to traditional statistical approaches. Consequently, many studies have focused on various ML models emphasizing accuracy, feature relevance, and environmental dependencies [5]. Models such as Random Forest (RF), GB, and Support Vector Machines (SVM) have been widely used as supervised learning models using weather and soil information [14-19].

Ensemble learning (EL) techniques have also emerged as promising tools in CYP. Zhang et al. [17] applied RF, bagging, and boosting methods to model climate-yield relationships, achieving high predictive performance. They also examined the effectiveness of integrating spatial and temporal data by applying GB for CYP. Bayesian probabilistic models, including Gaussian process regression, hold promise for CYP because they can quantify uncertainty. Convolutional networks have also proven beneficial for modeling seasonal patterns and long-term dependencies in agricultural time series data [18].

DL approaches have shown strong potential for CYP, particularly when integrated with remote sensing data. Zhang et al. implemented a CNN-RNN architecture for CYP with Sentinel-2 satellites [19]. Wang et al. used satellite images and weather and soil data for LSTM-based modeling and achieved better performance in wheat yield prediction results. Their study also incorporated more advanced techniques, such as attention mechanisms and processing space-time features [20]. Sagan et al. [21] performed CYP with 2D and 3D CNN models to enhance yield prediction in multi-temporal satellite images. Barbedo et al. utilized a deep multiscale residual network with crop growth indicators to reinforce the results of growth assimilation to achieve greater accuracy [22].

Bi et al. [6] created a Transformer-based model to address long-range dependencies in climatic data. They highlighted the advanced performance of those models compared to traditional models [23]. Similarly, Wang et al. introduced a hybrid model combining CNN and Transformers to exploit spatial and temporal features, often from satellite images [24]. However, these approaches predominantly rely on spatially rich data sources such as satellite imagery, and their applicability to structured tabular datasets remains less explored.

Several studies have conducted comparative analyses of ML models for CYP; however, these evaluations are often limited to a small subset of algorithms and focus primarily on accuracy-based metrics. Talaat et al. compared five ML models, including Gaussian Naive Bayes, GB, Decision Tree, RF, and Multimodal Naive Bayes. They reported that Extra Trees (ET) yielded the highest R^2 value of 0.9933 [25]. Badshah et al. [26] compared the efficacy of algorithms such as RF, SVM, ET, Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbour (KNN), and Gaussian Naive Bayes for CYP. In their study, RF attained the highest accuracy rate (99.7%) in CYP. Houque et al. [27] compared GB, KNN, and Multivariate Logistic Regression and evaluated the model performance with K-Fold CV.

These studies highlight the potential for high predictive accuracy but overlook the examination of deployment-oriented metrics. For example, Sharma et al. [28] explored RF, ET, and Artificial Neural Networks and reported the ET model as the most accurate, with an MAE of 5249.03 and an accuracy of 97.5%. Similar to other works, this study did not examine critical operational factors such as training time, inference time, and memory utilization. Our research directly addresses these gaps by systematically evaluating the performance of the models, with deployment-oriented metrics alongside other essential metrics. The results and discussion section provides a detailed comparison of our study with prior research [28-31] performed on the same dataset.

3. MATERIALS AND METHODS

3.1 Experimental setup

We conducted this study on a computing system with an Intel Core i9-12900 processor and an NVIDIA GeForce RTX 3060 GPU running on Windows 11. The details of the implementation platform are delineated in Table 1. The process flow of the proposed multifaceted CYP framework is depicted in Figure 1.

3.2 Dataset description

We utilized the CYP dataset from Kaggle, which compiles the Pesticides & Yield dataset from Food and Agriculture Organization (FAO) and the Rainfall & Temperature dataset from the World Data Bank [32-34]. It includes 28,242 records from 1990 to 2013 across 101 countries, including Russia, Ecuador, American Samoa, Gambia, China, India, and other geographically diverse countries. Table 2 presents the dataset attributes and their relevance to CYP.

Table 1. Implementation platform details

Name	Details
Processor & RAM	12th Gen Intel Core i9-12900, 64.0 GB RAM
Architecture & OS	64-bit operating system, x64-based processor, Windows 11 Pro (Version 24H2)
GPU	NVIDIA (RTX 3060) with CUDA 12.8
Python Version	3.12.9 in Jupyter Notebook
DL Framework	PyTorch & Modules: torch, torch.nn, torch.optim, torch.utils.data
RNN Architectures	SimpleRNN, LSTM, and Attention-LSTM (configured with 1-3 layers, 32-128 units, ReLU/Tanh activation for recurrent layers)
ML Libraries Libraries	xgboost, lightgbm, catboost, sklearn, scipy time, psutil, gc, numpy, pandas
ML Models	LinearRegression (LR), Ridge, Lasso, ElasticNet (ENet), BayesianRidge (BRidge), HuberRegressor (HR), RandomForestRegressor (RF), GradientBoostingRegressor (GB), AdaBoostRegressor (AdaBst), BaggingRegressor (BR), KNeighborsRegressor(KNN), XGBRegressor(XGBoost), LGBMRegressor(LGBM), CatBoostRegressor (CatBst)
Preprocessing	LabelEncoder, MinMaxScaler, StandardScaler, SimpleImputer
Model Selection	train_test_split, KFold, cross_val_score
Visualization	Matplotlib, Seaborn
Statistical Analysis	scipy.stats.zscore, scipy.stats.boxcox, scipy.stats.mstats.winsorize
Performance Metrics	MSE, MAE, R2 Score, and computational time (psutil), Memory usage

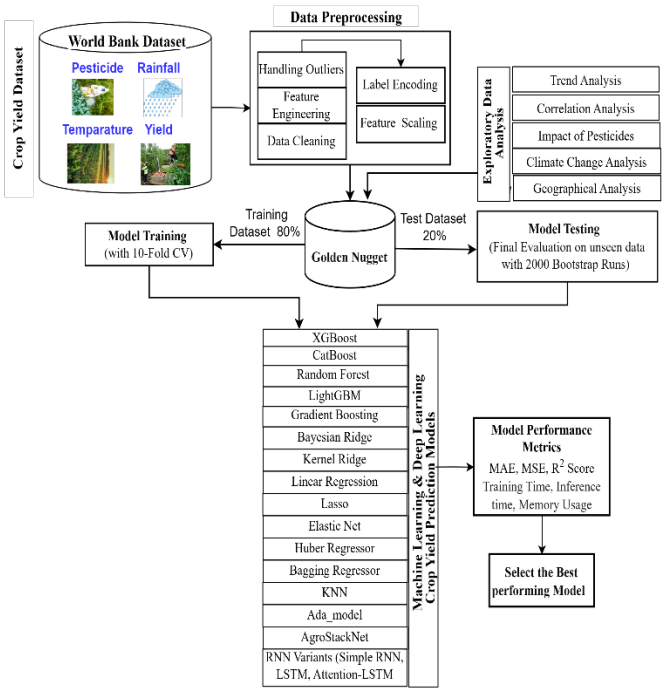

Figure 1. Multifaceted crop yield prediction (CYP) model

3.3 Data preprocessing

We began the preprocessing pipeline with the removal of irrelevant attributes (e.g., unnamed columns), followed by verification of missing values. No missing entries were observed. Outliers were handled using the Winsorization technique, where extreme values in numerical features were limited using a 5% threshold. The Interquartile Range (IQR) and Z-score methods were employed for exploratory identification of outliers; however, Winsorization was applied as the final transformation method. The effect of this transformation is illustrated in Figure 2(a)-(c).

Table 2. Crop yield prediction (CYP) dataset attributes

Features	Description
Area	Name of the country where the crop is cultivated.
Item	Type of crop cultivated (e.g., maize, wheat, rice).
Year	Calendar year in which data were gathered.
Pesticide Usage (tonnes/year)	Amount of pesticides applied per region.
Average Rainfall (mm/year)	Average precipitation level influencing soil moisture and irrigation.
Average Temperature (°C/year)	Average temperature impacting crop growth.
Crop Yield (hg/ha)	Target variable representing the productivity of crops in hectograms per hectare.

To address skewness in feature distributions, a logarithmic transformation of the form $ln(1 + x)$ was applied as shown in the Eq. (1).

$$\log\_value=\ln(1+value) \quad (1)$$

Prior to transformation, feature values were adjusted to ensure non-negativity. The distributions before and after transformation are shown in Figure 2(d)-(e).

Categorical features were encoded using label encoding to convert them into numerical representations suitable for machine learning models; however, no ordinal interpretation was assumed during analysis.

Feature scaling was performed in two sequential stages. First, Min-Max scaling was applied to normalize features within the range [0,1]. Subsequently, Z-score standardization was applied to ensure zero mean and unit variance across features.

All preprocessing steps, including transformations and scaling, were applied exclusively to the training data and subsequently applied to the test data to prevent data leakage.

These preprocessing steps collectively enhanced data quality and improved model robustness.

3.4 Exploratory crop yield data analysis

After careful data preprocessing, we performed correlation analysis as part of EDA to explore relationships between the agricultural features and crop yield, which is illustrated in Figure 3. The crop yield has a correlation of -0.11 with average temperature and -0.06 with rainfall, which implies that these two environmental factors, among others, are weakly correlated with crop yield. The negative correlation of Area with rainfall (-0.23) and pesticide use (-0.31) indicates an association where regions with lower rainfall exhibit relatively higher pesticide usage; however, no causal relationship can be inferred.

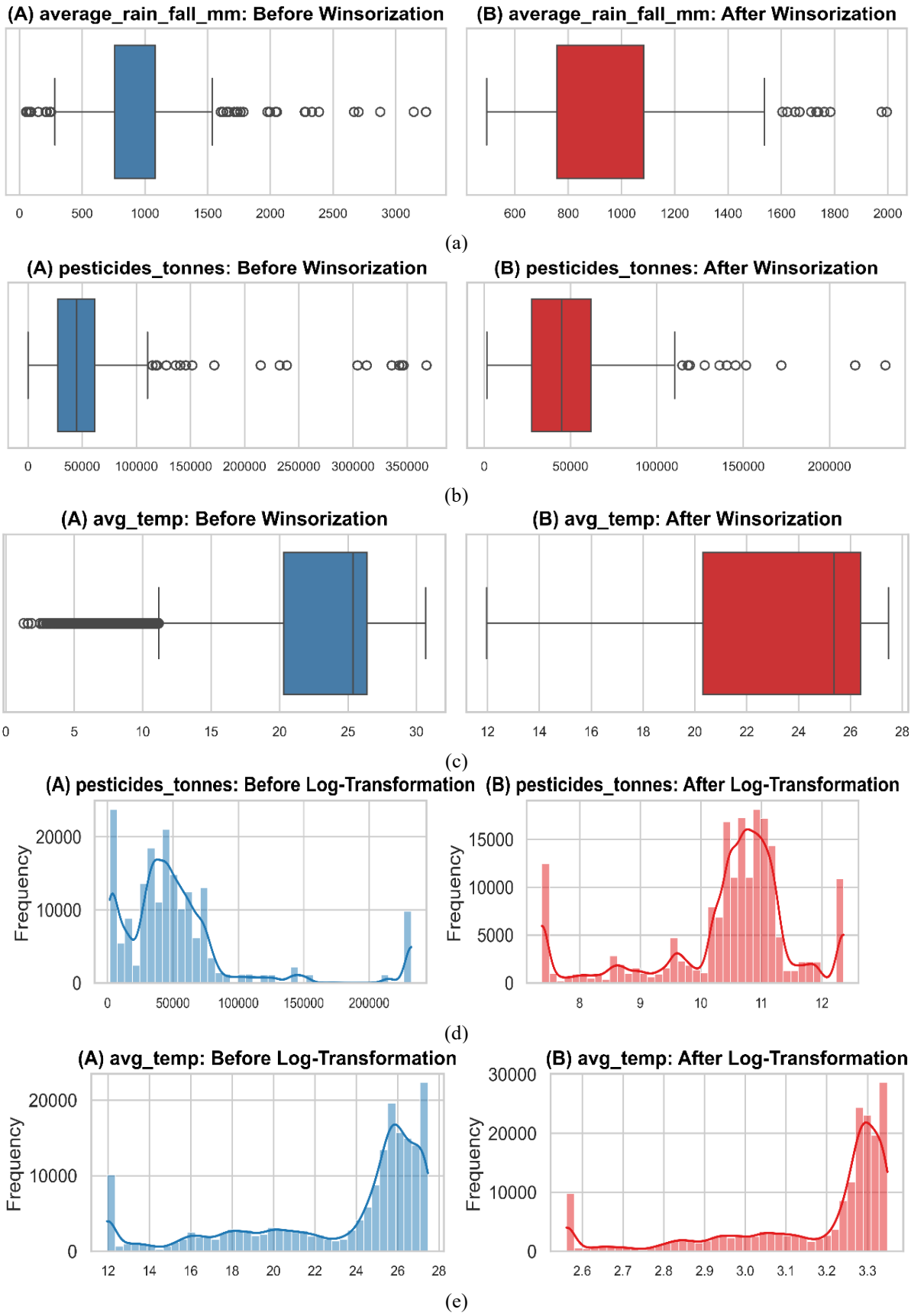

Figure 2. (a) Winsorization Before/After – Rainfall, (b) Winsorization Before/After – Pesticides, (c) Winsorization Before/After – Temperature, (d) Log Transformation Before/After – Pesticides, (e) Log Transformation Before/After – Temperature

A weak positive correlation (0.10) between pesticide usage and crop yield suggests a weak linear association with crop yield. Similarly, the negligible correlation (0.03) between temperature and pesticide usage indicates minimal linear association between temperature and pesticide usage. A moderate correlation (0.31) between temperature and rainfall reflects a climatic trend where higher temperatures are associated with increased precipitation.

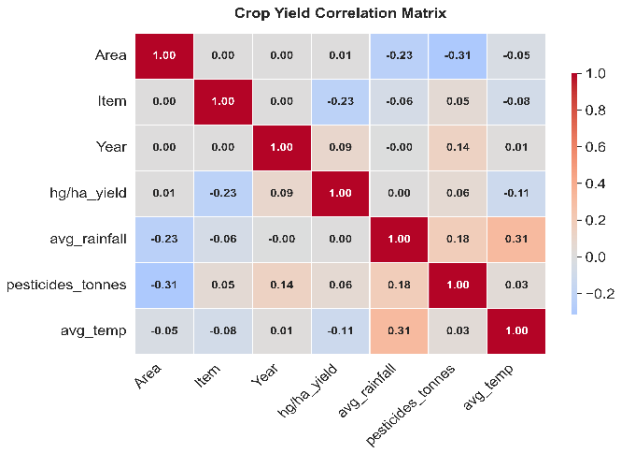

Figure 3. Heatmap of correlation coefficients for agricultural variables

Following the correlation analysis, we generated a density map (Figure 4) to better understand crop yield distribution across various countries concerning average annual rainfall. Most of the reported yields hover around the 500-1,500 mm average rainfall, and there are high yield outliers spread across various levels of rainfall. It is important to note that over 2,000 mm regions of annual rainfall have few recorded data points, hinting at possible issues like waterlogging and nutrient-deficient soils. Among all countries, Belgium has the highest yield record (~501,412 hg/ha), which may be attributed to advanced agricultural practices, efficient irrigation systems, and better resource management. These outcomes suggest that rainfall alone does not fully explain variations in crop yield.

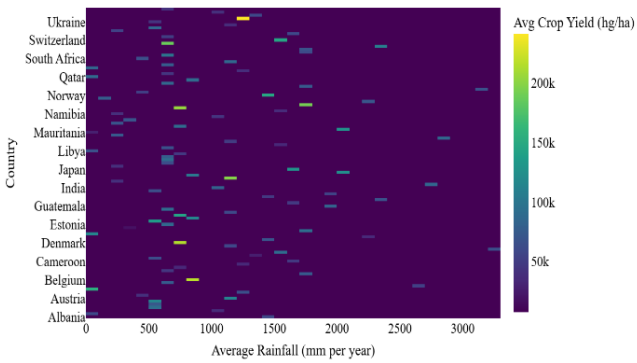

Figure 4. Crop yield distribution by rainfall and country

Table 3. High-yield records by crop item and country

Item	Country	Crop Yield
Cassava	India	142,810,624
Potatoes	India	92,122,514
	Brazil	49,602,168
	United Kingdom	46,705,145
	Australia	45,670,386
	Japan	42,918,726
	Mexico	42,053,880
Sweet Potatoes	India	44,439,538
	Mexico	35,808,592
	Australia	35,550,294

Table 3 presents high-yield records across different crop types and countries, highlighting dominant contributors to global production. We noticed India's significant role in producing substantial yields in cassava and sweet potatoes, constituting 25% and 18% of the recorded global share, respectively. This might be due to 14% of the dataset being of India.

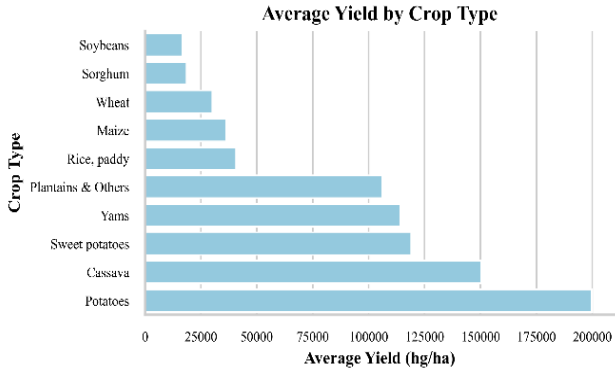

Figure 5. Average crop yield by crop type

Among the cultivated crops, potatoes exhibited the highest average yield (Figure 5), followed by cassava, sweet potatoes, and yams. The top ten potato yields and the corresponding features across different countries and years are illustrated in Table 4. As per the table analysis, Belgium topped in potato yield in 2011, 2004, and 2002. These instances were correlated with high pesticide usage (5,740-9,204 tonnes) and low rainfall (847 mm). During 2007-2013, New Zealand was able to sustain yields at a moderate level (~478,154-495,751 hg/ha) with moderate pesticide usage (~5,086 tonnes) and high rainfall (1,732 mm).

Table 4. Top ten potato yield by country, item, year, pesticides usage, and average rainfall

Country	Year	Pesticides Usage	Average Rainfall	Crop_yield
Belgium	2011	5,740.44	847.0	501,412.0
New Zealand	2010	5,086.00	1,732.0	495,751.0
New Zealand	2011	5,086.00	1,732.0	490,361.0
Switzerland	1996	1,746.30	1,537.0	487,219.0
New Zealand	2012	5,086.00	1,732.0	484,810.0
Belgium	2004	9,186.00	847.0	483,955.0
New Zealand	2013	5,086.00	1,732.0	482,926.0
New Zealand	2009	5,086.00	1,732.0	478,154.0
New Zealand	2007	4,939.00	1,732.0	477,612.0
Belgium	2002	9,204.00	847.0	471,475.0

To investigate variations in crop yield among different countries, we created a boxplot shown in Figure 6. It also illustrates that Belgium is at the top in crop yield, reinforcing its highly productive agricultural practices.

We visualized the crop yield trend over the years for the top fifteen countries by yield using the line plot shown in Figure 7. We noticed that countries like Denmark, Netherlands, and Germany have consistently high yields, while others like Egypt and Jamaica show lower values. Some countries display stable trends, while others experience significant fluctuations. These fluctuations may be influenced by climate change, rainfall, or pesticide usage.

We created a spline interpolation plot (Figure 8) and noticed a general upward trend in crop yield from 1990 to 2013.

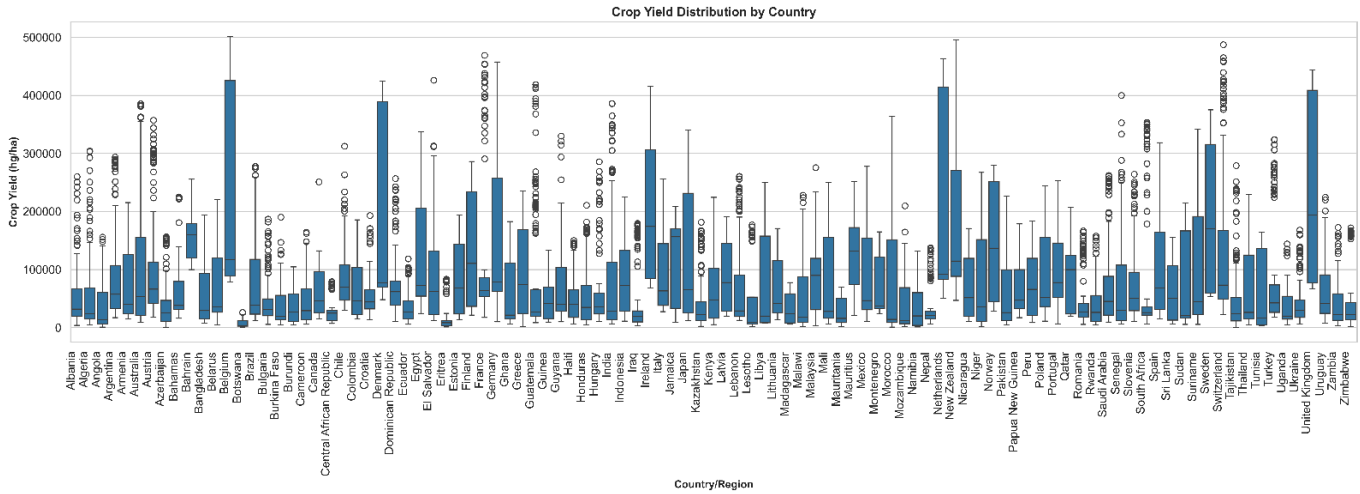

Figure 6. Crop yield distribution by country

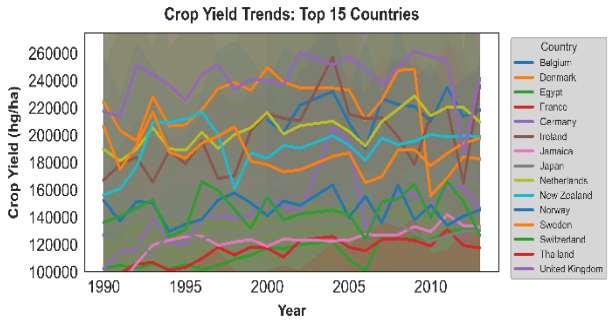

Figure 7. Temporal analysis of crop yield for the top 15 countries

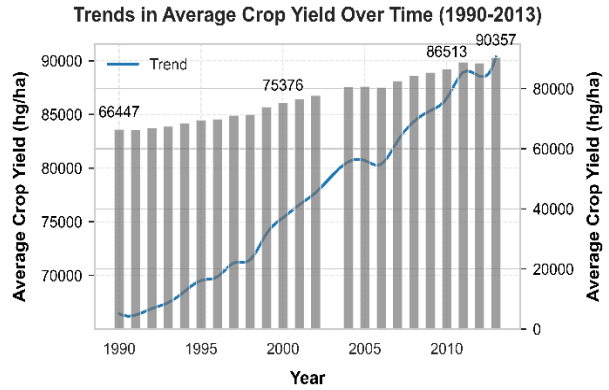

Figure 8. Average crop yield over time (1990-2013)

While these exploratory analyses provide valuable insights, the weak linear relationships observed indicate that CYP requires models capable of capturing complex nonlinear interactions. This motivates the use of ML to improve the accuracy of agricultural predictions.

4. METHODOLOGY

4.1 Multi-model exploration for crop yield prediction

To identify the most suitable predictive model for CYP, we analyzed a broad range of regression algorithms, including linear methods, ensemble techniques, instance-based learning, and RNN variants from DL. To ensure reproducibility, the random state was fixed at 42 for all applicable ML models, as specified in Table 5. To ensure a fair and unbiased evaluation, the dataset was divided into training and testing subsets using an 80:20 split. All preprocessing steps, including imputation, scaling, and transformation, were fitted exclusively on the training data and subsequently applied to the test data to prevent data leakage. Model performance was evaluated using 10-fold cross-validation on the training dataset to ensure robustness and reduce variance in performance estimation. The final model evaluation was performed on the unseen test dataset.

Hyperparameter tuning was conducted using a combination of grid search and predefined configurations based on prior studies and computational feasibility. For ensemble models such as RF and GB, key parameters, including the number of estimators and learning rates were tuned. For DL models, parameters including the number of layers, hidden units, and learning rate were systematically varied as shown in Table 6.

4.2 Model selection

4.2.1 Linear regression models

We examined several linear regression models, including LR, HR, Ridge, Lasso, ENet, and BRidge. These models heavily depend on the linear relationships between features, making them relatively interpretable.

4.2.2 Ensemble approaches

The ensemble approaches, such as RF, GBst, AdaBst, and BR, were included due to their popularity in providing more accurate results. RF takes advantage of the dataset's large number of random subsets to make predictions with the ensemble of decision trees. It reduces variance and improves generalization by averaging over many predictions. GB and AdaBst try to improve the generalization by sequentially fitting weak learners to correct the errors of previous models. BR uses bootstrapping to generate multiple models and aggregates their predictions to reduce variance. Advanced XGBst, LGBM, and CatBst algorithms can also capture the complicated feature interactions required for large-scale agricultural data.

4.2.3 Instance-based learning

We also analyzed KNN to determine how effectively it recognizes crop yield trends based on distance measures of

similarity between data points. Although it can monitor localized trends, its performance may degrade in high-dimensional datasets due to the increased computational complexity of distance calculations. The configuration of each model is listed in Table 5. The listed configurations represent baseline settings, which were further refined through cross-validation-based tuning for selected models to achieve optimal performance. For ensemble models such as GB and AdaBst, standard configurations were used as baseline settings and further evaluated using cross-validation. Key parameters such as the number of estimators and learning behavior were analyzed to ensure consistent comparison across models.

Table 5. Configurations and parameters of machine learning (ML) models

Model	Configuration
XGBoost	XGBRegressor(tree_method='hist', device='cpu', random_state=42)
AdaBoost (AdaBst)	AdaBoostRegressor(random_state=42)
Random Forest (RF)	RandomForestRegressor(n_estimators=100, random_state=42)
Gradient Boosting (GB)	GradientBoostingRegressor(random_state=42)
LightGBM (LGBM)	LGBMRegressor(device='cpu', random_state=42)
CatBoost (CatBst)	CatBoostRegressor(task_type='cpu', random_state=42, verbose=0)
Linear Regression (LR)	LinearRegression()
Ridge	Ridge()
Lasso	Lasso()
Elastic Net (ENet)	ElasticNet(random_state=42)
KNN	KNeighborsRegressor()
Bayesian Ridge (BRidge)	BayesianRidge()
Huber Regressor (HR)	HuberRegressor()
Bagging Regressor (BR)	BaggingRegressor(random_state=42)

4.2.4 RNN-based deep learning model configurations

After a comprehensive assessment of ML models, we investigated the efficacy of RNN variants, including simple RNNs, LSTM models, and Attention-enhanced LSTMs to explore their ability to model temporal patterns represented through yearly observations in the dataset. We specifically applied these deeper recurrent architectures to address the vanishing gradient issue and preserve long-term temporal correlations. Hyperparameter tuning for the RNN-based models was performed through a structured experimental search over key architectural parameters. Multiple configurations were evaluated by varying model depth, hidden representation capacity, and architectural design, as summarized in Table 6.

Each configuration was trained using the same training dataset and evaluated on a held-out test set to ensure consistency. Model performance was assessed using MSE, MAE, and R^2 metrics. The optimal configuration was selected based on the best predictive performance while considering computational efficiency.

To improve training stability and mitigate overfitting, early stopping was employed based on the convergence of training loss. This systematic evaluation ensures that the reported comparative results are derived from controlled and reproducible experimentation.

Table 6. Configurations and parameters of simple RNN, LSTM and attention-LSTM

Parameter	Details
Model Architectures	**Simple RNN, LSTM, and Attention-LSTM**
Number of Layers	{1, 2, 3}
Units per Layer	{32, 64, 128}
Activation Function	{ReLU, Tanh} – RNN only
Dropout Rate	0.2
Optimizer	Adam (learning rate = 0.001)
Loss Function	MSE
Batch Size	32
Epochs	100 (with early stopping if loss change < 1e-4)

4.3 AgroStackNet hybrid ensemble

Building upon individual model assessments, we developed AgroStackNet, a hybrid ensemble framework designed to advance CYP performance. This framework was created by combining the strengths of top-performing base learners like RF, XGBst, BR, CatBst, and LGBM. Their predictions are then meticulously fused by a meta-learner, such as Ridge or BRidge regression. We performed this study to observe how ensemble diversity and meta-learner selection impact both prediction accuracy and computational efficiency. To prevent overfitting in the stacking framework, the meta-learner was trained using out-of-fold predictions generated during cross-validation. The findings from this study are provided in Section 6.

4.4 Evaluation metrics

Mean Squared Error (MSE) assesses the average squared difference between predicted and actual values. Lower MSE values generally indicate better model performance. The MSE formula is given below in Eq. (2).

$$MSE = (\frac{1}{n}) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2)$$

R^2 Score measures the percentage of the dependent variable's variance that the independent variables account for. In general, higher R^2 values indicate a better model fit. The formula for calculating the R^2 Score is given below in Eq. (3).

$$R^2\ Score = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3)$$

Mean Absolute Error (MAE) is a metric used to assess how close predictions are to actual values on average. Lower MAE signifies higher accuracy, indicating predictions are generally closer to the real targets. The formula for calculating the MAE is given below in Eq. (4).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (4)$$

In the above formulas, y_i represents the actual yield, $\hat{y}_i$ represents the predicted yield, $\bar{y}_i$ is the mean of the actual values, and n is the number of observations.

4.5 Computational efficiency assessment

Training time is the total duration required for a model to learn from the training dataset. Its computation is shown in Eq. (5).

$$\text{Training Time} = t_{end} - t_{start} \quad (5)$$

where, t_{start} and t_{end} represent the timestamps at the beginning and end of model training, respectively.

Inference time measures the duration taken for a trained model to make predictions on unseen test data or validation datasets. Its computation is shown in Eq. (6).

$$\text{Inference time} = T_{pred\_end} - T_{pred\_start} \quad (6)$$

t_{pred_start} and t_{pred_end} denote the timestamps recorded before and after executing the model's predict() function.

Memory usage quantifies the peak memory consumption during model training. The formula for memory usage computation is shown in Eq. (7).

$$\text{Memory usage} = M_{peak} - M_{base} \quad (7)$$

M_{peak} and M_{base} indicate maximum memory consumption during model training and baseline memory usage before training starts.

5. RESULTS AND DISCUSSION

We scrupulously evaluated the ML models using 10-fold CV on 80% of the training data and unseen test sets. The mean performance results of the 10-fold CV and test dataset performance are summarized in Tables 7 and 8, respectively. All models were evaluated under identical experimental conditions to ensure fair comparison. This includes the same train-test split, consistent preprocessing pipeline, identical cross-validation strategy, and standardized evaluation metrics.

5.1 Performance evaluation using CV

To ensure robustness and generalization, 10-fold cross-validation was performed. This evaluation aimed to evaluate the stability of the models across several partitions of the data and detect early signs of overfitting before test set evaluation.

Table 7. Performance comparison of machine learning models (10-fold CV)

Model	Mean MSE ($\times 10^7$)	Mean MAE	Mean R^2	Train Time (s)	Inference Time (s)	Memory (MB)
Random Forest	1.17	456.19	0.9985	13.40	0.166	12.30
Bagging Regressor	1.30	483.38	0.9984	1.36	0.020	1.29
XGBoost	4.14	2817.83	0.9948	0.20	0.005	1.94
CatBoost	4.70	3265.35	0.9941	5.17	0.007	2.77
KNN	7.67	1635.86	0.9904	0.16	0.160	1.54
LightGBM	8.39	4652.56	0.9895	0.15	0.007	1.60
Gradient Boosting	51.7	11846.55	0.9353	5.95	0.015	0.00
AdaBoost	263.0	41784.58	0.6709	3.06	0.025	0.15
Linear Regressor	666.0	66438.89	0.1663	0.01	0.001	0.32
Ridge	666.0	66438.60	0.1663	0.01	0.000	0.03
Lasso	666.0	66438.12	0.1663	0.03	0.001	0.01
Bayesian Ridge	666.0	66438.15	0.1663	0.01	0.002	0.02
Huber Regressor	752.0	62107.90	0.0590	0.42	0.002	0.47
Elastic Net	765.0	69557.35	0.0423	0.01	0.002	0.00

Table 8. Performance comparison of machine learning models with 2000 bootstrap-based uncertainty estimates (Test dataset)

Model	Test MSE Mean ($\times 10^7$)	MSE CI (Lower)	MSE CI (Upper)	Test MAE Mean	MAE CI (Lower)	MAE CI (Upper)	Test R^2 Mean	R^2 CI (Lower)	R^2 CI (Upper)	Train Time (s)	Inference Time (s)	Memory (MB)
Random Forest	1.08	0.89	1.31	432.46	399.88	468.16	0.9986	0.9984	0.9989	14.73	0.3832	127.55
Bagging Regressor	3.98	3.63	4.38	471.35	436.10	508.15	0.9984	0.9980	0.9986	1.47	0.0501	12.05
XGBoost	8.12	7.51	8.79	2817.22	2760.72	2874.25	0.9950	0.9945	0.9955	0.15	0.0041	5.89
CatBoost	4.59	4.15	5.09	3252.11	3192.97	3312.49	0.9943	0.9936	0.9948	4.54	0.0084	20.05
KNN	1.32	1.08	1.57	1447.70	1364.13	1531.24	0.9918	0.9908	0.9928	0.17	0.3586	11.23
LightGBM	51.37	49.25	53.62	4626.88	4549.06	4708.31	0.9899	0.9890	0.9906	0.17	0.0143	0.00
Gradient Boosting	221.51	217.54	225.30	11841.42	11644.40	12035.69	0.9358	0.9332	0.9383	6.65	0.0338	0.00
AdaBoost	65.43	57.82	73.54	37703.39	37414.16	37981.93	0.7230	0.7176	0.7278	2.54	0.0411	0.99
Linear Regression	668.72	657.92	679.42	66567.75	66063.03	67076.13	0.1642	0.1571	0.1707	0.01	0.0033	0.00
Ridge	668.79	658.78	679.28	66565.32	66118.74	67028.73	0.1642	0.1574	0.1712	0.01	0.0040	0.02
Lasso	668.53	658.08	679.05	66560.86	66071.68	67040.94	0.1642	0.1575	0.1709	0.08	0.0000	0.00
Bayesian Ridge	766.53	751.41	780.62	66564.65	66082.86	67049.70	0.1643	0.1570	0.1712	0.03	0.0000	0.65
Huber Regressor	668.75	658.30	679.90	62086.50	61454.76	62691.75	0.0583	0.0543	0.0623	0.43	0.0038	0.56
Elastic Net	753.55	737.44	769.75	69506.91	68964.51	70025.79	0.0417	0.0409	0.0426	0.02	0.0052	0.00

RF, with MSE of 1.17×10^7 and R^2 of 0.9985, exhibited the best performance during CV and was followed closely by BR (MSE = 1.30×10^7, R^2 = 0.9984). These two models maintained their high predictive power throughout the different validation folds. XGBst and CatBst also had similarly competitive R^2 values (0.994) but had comparatively higher MAE values. This suggests that these models make slightly less accurate predictions than ensemble-based models.

Conversely, linear models like LR, Ridge, and Lasso showed the lowest predictive performance, with an R^2 value of 0.166. This implies that linear methods are unable to handle complex nonlinear data. In addition, AdaBst and GBst had a mediocre performance with R^2 of 0.67 and 0.93, respectively. However, their increased MAE values exposed poor accuracy.

After CV, we did not eliminate underperforming models outright. Instead, we retained all models for further assessment on the test set to validate their generalization ability on unseen datasets and verify whether their CV results were consistent.

5.2 Model evaluation on the test set

The models' performances on the test dataset largely mirrored CV results, with RF providing the best accuracy with R^2 of 0.9986 and MSE of 1.08×10^7. BR performed similarly with R^2 of 0.9984 and MSE of 3.98×10^7, which is additional evidence for its stability. Of the boosting-based models, near the top was XGBst (R^2 = 0.995, MSE = 8.12×10^7). LGBM and CatBst, despite their high validation scores, underperformed on the test set, indicating potential overfitting caused by dataset distribution shifts. A key observation was the failure of linear models to generalize, as reflected in their poor performance on the test set (R^2 = 0.16, MSE = 668.72×10^7 for LR). Although AdaBoost achieved moderate performance in both CV (R^2 = 0.67) and test evaluation (R^2 = 0.72), it remained significantly inferior to other ensemble models.

Generally, the similarity between CV and test results for top-performing models (RF, BR, and XGBst) indicates reliability. On the other hand, all the linear models underperformed during both tests, confirming their incompatibility with the complex structure of the dataset.

5.3 Uncertainty quantification

In addition to generating point predictions from models, it is crucial, especially for critical applications in CYP, to quantify the confidence and reliability of these outputs. To achieve this, we performed uncertainty quantification using two methods: Gaussian Process Regression (GPR) and bootstrap resampling.

5.3.1 Predictive uncertainty with gaussian process regression

GPR quantifies prediction uncertainty by providing mean predictions and associated predictive variances used to construct confidence intervals. A Gaussian Process (GP) is a collection of random variables such that any finite number has a joint Gaussian distribution. It is fully specified by its mean function m(x) and covariance (kernel) function $k(x, x')$ as specified in Eq. (8):

$$f(x) \sim \mathrm{GP}\big(\mathrm{m}(\mathrm{x}), \mathrm{k}(\mathrm{x}, \mathrm{x}')\big) \tag{8}$$

where, $m(x) = \mathrm{E}[\mathrm{f}(\mathrm{x})]$ and $\mathrm{k}(\mathrm{x}, \mathrm{x}') = \mathrm{Cov}\big(\mathrm{f}(\mathrm{x}), \mathrm{f}(\mathrm{x}')\big)$.

5.3.2 Posterior predictive distribution for uncertainty quantification

Given training data $D = \{(x_i, y_i)\}_{i=1}^{N}$ (inputs x_i, observed outputs y_i), and assuming a GP prior, the posterior predictive distribution for a new test point x_* is also Gaussian as given in Eq. (9):

$$\mathrm{p}(\mathrm{f}(x_*) \mid \mathrm{X}, \mathrm{y}, x_*) = \mathrm{N}(\mu(x_*), \sigma^2(x_*)) \tag{9}$$

where the posterior mean $\mu(x_*)$ and variance $\sigma^2(x_*)$ are given in Eqs. (10) and (11):

$$\mu(x_*) = k_{*f}^T (K_{ff+}\sigma_n^2 I)^{-1} y \tag{10}$$

$$\sigma^2(x_*) = \mathrm{k}_{**} - k_{*f}^T (K_{ff+\sigma_n^2} I)^{-1} \mathrm{k}_{*f} \tag{11}$$

Here, K_{ff} is the training data covariance matrix, k_{*f} is the covariance vector between the test point and training data, k_{**} is the test point variance, and σ_n^2 is the noise variance.

We observed that 95.37% of test points fell within our study's 95% confidence intervals, as illustrated in Figure 9 This outcome suggests that the model's uncertainty estimates are well-calibrated.

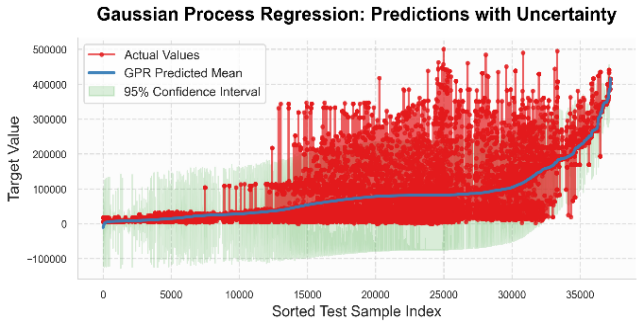

Figure 9. Gaussian process regression predictions with 95% confidence intervals

5.3.3 Performance uncertainty via bootstrap resampling

In bootstrap resampling, we repeatedly resampled the test set for 2,000 bootstrap iterations to re-evaluate the models on these resamples. Confidence intervals were calculated based on the models' performance in all the iterations to quantify their variability and robustness. The results of this bootstrap-based uncertainty analysis on the test dataset are presented in Table 8.

5.4 Computational efficiency and deployment considerations

One of the notable objectives of this study is to determine the efficacy of the models not only based on performance evaluation metrics but also on their computational efficiency, such as training time, inference time, and memory usage. The results highlight a trade-off between the accuracy and computational efficiency of the models.

5.4.1 Training time

RF took the longest training period, with 15.46 s in cross-validation and 18.92 s when trained on the entire training dataset before the test evaluation. Its high computational cost was justified by its superior accuracy. On the other hand, the boosting models trained significantly faster, with XGBst and

LGBM completing their training within 0.28 and 0.27 s during CV, while training time before final testing took 0.34 s and 0.31 s, respectively. These results indicate that boosting models are better suited for situations where frequent retraining is needed, whereas RF is optimal in situations where accuracy comes first. Though BR models have slightly lower accuracy, these models are viable for deployment due to minimal computational requirements. The linear models like Ridge, Lasso, and LR were trained simultaneously, each taking under 0.05 s. However, these models are not viable for deployment due to their much lower accuracy.

5.4.2 Inference time

Model inference speed is imperative, especially in scenarios that require real-time solutions. XGBst and LGBM provided the fastest inference times during CV, requiring 0.010 and 0.016 s, respectively, while on the test dataset, they needed 0.011 and 0.017 s. These metrics definitively demonstrate their efficiency in time-constrained scenarios.

RF showed slower inference times, needing 0.411 s during CV and 0.437 s during test evaluation. BR did almost as well but was better in terms of a lower inference time of 0.193 s in CV and 0.206 s in test evaluation, and thus, it stands out as a good balance.

Linear models had negligible inference times, taking under 0.002 s, but had no predictive ability. These results further confirm that XGBst and LGBM are optimal in speed and accuracy. RF has been proven useful in cases where more latency is acceptable in return for better performance.

5.4.3 Memory consumption

Memory efficiency is another important factor when deploying a model, especially when working with limited resources. RF used the most memory, at 126.93 MB for CV and 132.11 MB for test evaluation, which could be problematic for deployment in low-resource environments. However, BR reduced memory usage to 76.42 MB for CV and 81.87 MB for test evaluation with good performance.

Boosting models showed greater efficiency as XGBst required 104.34 MB for CV and 107.82 MB for test evaluation. Nonetheless, the memory efficiency of LGBM made it the most computationally efficient, using only 1.71 MB and 1.96 MB, for CV and test evaluation, respectively. These findings show that LGBM is optimal for memory-limited deployments.

Lastly, while linear models consumed less than 1 MB, their poor performance limited their practicality. In sum, these results point to LGBM outperforming other models for memory efficiency, followed by XGBst, while RF and BR models had considerably higher memory requirements to achieve accuracy. Bar plots illustrating the performance comparison of models in 10-fold CV and test dataset are shown in Figures 10 and 11 respectively.

5.4.4 Trade-offs and deployment suitability

Balancing accuracy and computational efficiency is crucial for model selection. RF remains the best choice for high-accuracy applications where computational cost is secondary. However, BR provides a viable alternative with reduced resource demands. XGBst and LGBM offer lower training times, minimal inference latency, and efficient memory usage for real-time applications. Among them, LGBM is the most memory-efficient. In environments with strict resource constraints, LGBM is ideal due to its balance of efficiency and accuracy. Evaluating computational efficiency across both CV and test datasets ensures a well-rounded understanding of each model's feasibility for deployment. These results confirm that tree ensemble models consistently outperform linear models when predicting crop yield. This emphasizes the need for nonlinear modeling approaches for large-scale agricultural data.

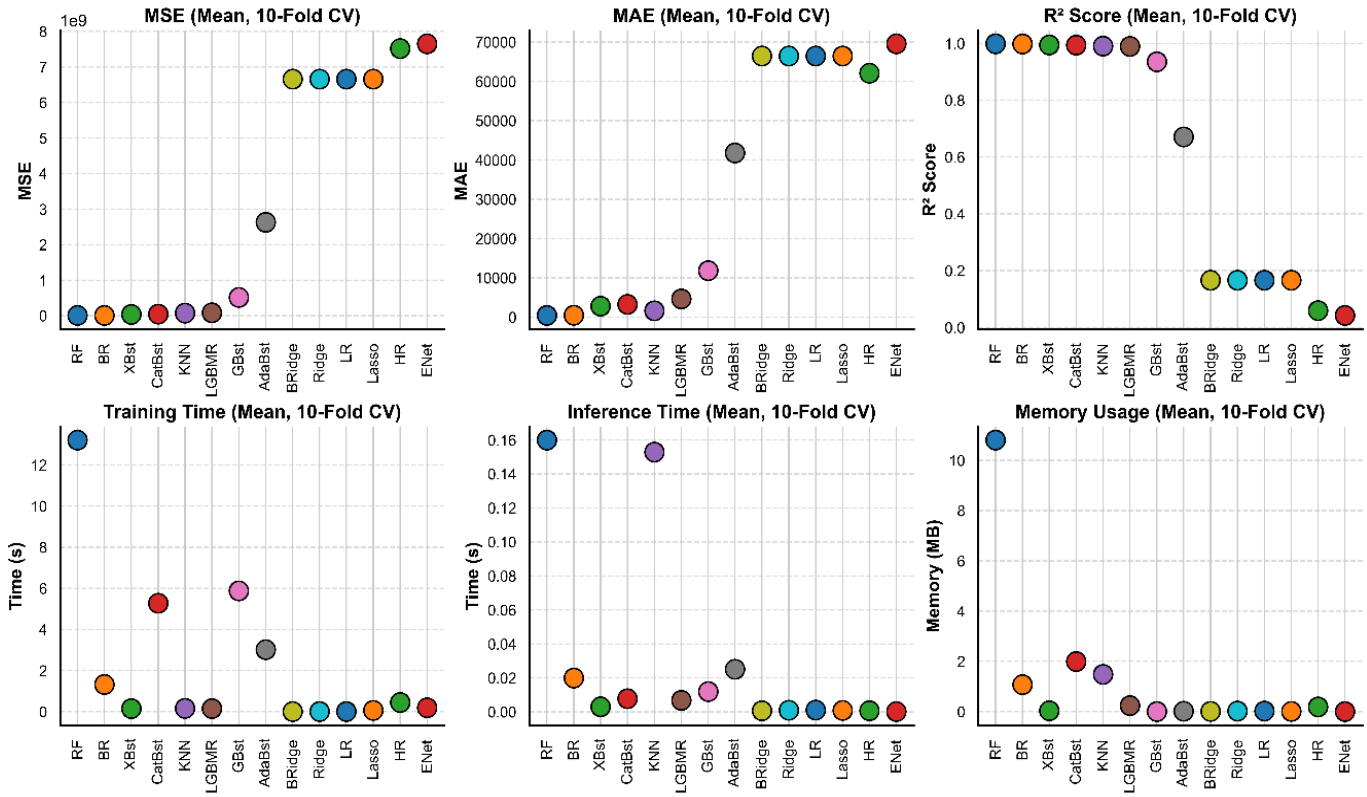

Figure 10. Performance comparison of models in 10-fold CV

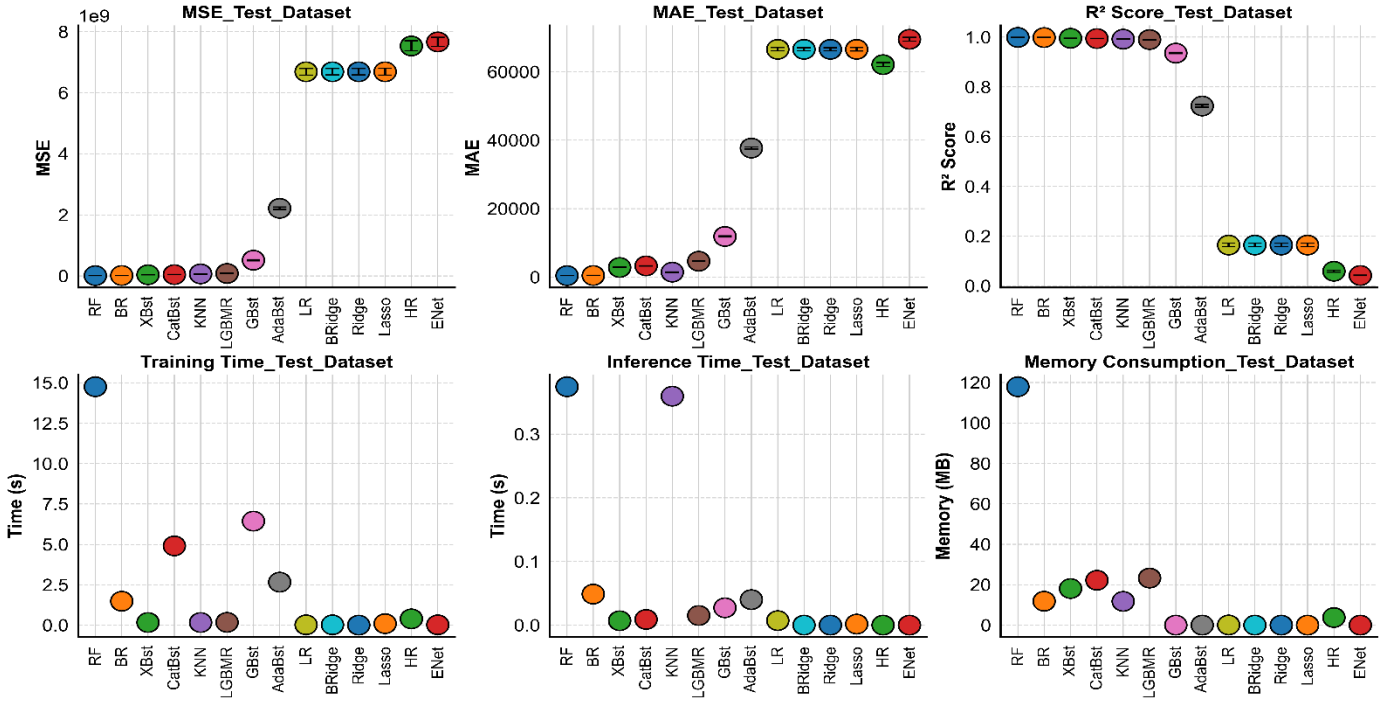

Figure 11. Performance comparison of models in test dataset

Moreover, the gap between CV and test performance for some boosting models (like CatBst and LGBM) raises concerns about overfitting, which indicates that their robustness could be enhanced through specific dataset tuning in future efforts. This study evaluates all the candidate models to choose the most suitable ones for real-world deployment based on predictive accuracy, generalization ability, and computational efficiency rather than prematurely removing weaker candidates.

5.5 Residual analysis of top performing machine learning models

We created residual plots to interpret the nature and distribution of errors in prediction for the top 6 models (RF, BR, XGBst, CatBst, KNN, and LGBM). These plots illustrated the differences between predicted and actual values (residuals) and predicted values. For an unbiased model, residuals should exhibit a random scatter around zero with consistent variance. The model has not fully captured underlying nonlinear relationships if the residual plot has persistent curved patterns. An inconsistent spread of residuals suggests varying error variance with predicted values. A systematic shift of residuals consistently above or below the zero line for specific predictions indicates model bias.

We can observe a broadly consistent and uniform scatter around the zero line in the residual plots of the models in Figure 12. This indicates the efficacy of these models in capturing underlying relationships in crop yield data with minimized systematic biases and improved modeling of nonlinear relationships. These plots reinforce confidence in models' reliability and generalization ability.

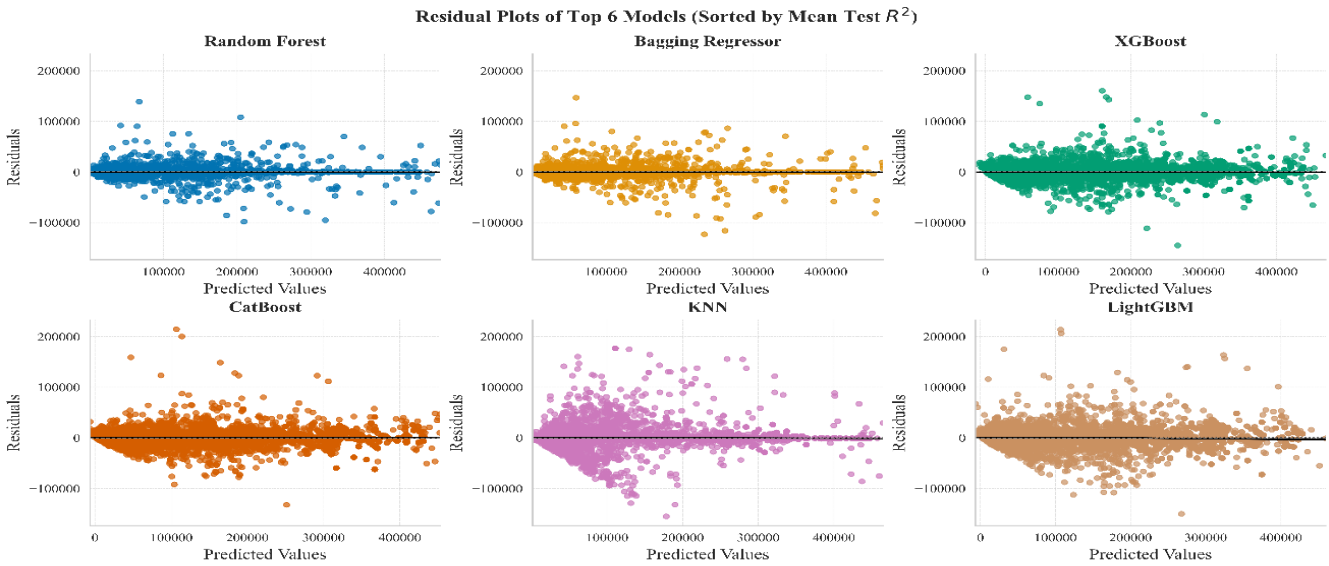

Figure 12. Residual plots of top 6 models by mean R^2

5.6 Performance analysis of recurrent neural network variants

This study was conducted to probe how depth, hidden unit size, and attention mechanisms influence predictive performance, training cost, and resource utilization of all the RNN variants (simple RNN, LSTM, and Attention-LSTM models). The performance of the best-performing RNN variants within each variant category is presented in Table 9 for a broader context. Among the simple RNN configurations, the three-layer RNN with 64 neurons and Tanh activation (R^2 = 0.9916, MSE = 6.73 × 10^7) demonstrated the best performance. This performance was followed closely by the two-layer RNN with 128 neurons and Tanh activation (R^2 = 0.9896, MSE = 8.30 × 10^7). We also noticed that the predictive performance and computational efficiency of a two-layer RNN with 128 neurons and ReLU were lower than those of RNN_2L_128N_tanh. These results emphasize that the optimal performance of any configuration depends on layers, neurons, and activation functions rather than just the depth of the network. This also indicates that Tanh activation was more effective in capturing long-term dependencies, as evidenced by the superior performance of Tanh-activated RNNs compared to their ReLU counterparts. Furthermore, smaller RNNs came with competitive inference speeds and a noticeable trade-off in accuracy.

Table 9. Top 5 models per architecture category based on R^2 Score

Model Name	MSE (× 10^7)	MAE	R^2	Train Time (s)	Inference Time (s)	Memory Usage
RNN Models						
RNN_3L_64N_tanh	6.73	4088.47	**0.9916**	901.57	0.7495	1.13 MB
RNN_2L_128N_tanh	8.30	4924.56	0.9896	859.97	0.8457	1.26 MB
RNN_2L_128N_relu	1.17	5471.47	0.9853	867.39	0.8728	1.56 MB
RNN_2L_64N_tanh	1.19	6246.91	0.9852	849.36	0.6617	1.09 MB
RNN_3L_32N_tanh	1.11	5587.14	0.9862	912.24	0.6990	1.06 MB
LSTM Models						
LSTM_3L_128N	2.29	1955.63	**0.9971**	1039.78	0.9367	5.89 MB
LSTM_3L_64N	3.03	2574.54	0.9962	1078.63	0.7937	1.88 MB
LSTM_2L_128N	3.15	2591.45	0.9961	961.51	0.8233	3.76 MB
LSTM_2L_64N	4.98	3589.23	0.9938	958.26	0.7905	1.55 MB
LSTM_3L_32N	5.30	3807.92	0.9934	1053.34	0.8922	1.13 MB
Attention-LSTM Models						
AttentionLSTM_3L_64N	2.92	2493.62	**0.9964**	1166.57	0.8683	1.89 MB
AttentionLSTM_3L_128N	2.77	2255.42	0.9965	1134.72	0.8744	5.89 MB
AttentionLSTM_2L_128N	3.26	2657.28	0.9959	1064.21	0.7899	3.76 MB
AttentionLSTM_2L_64N	4.79	3506.41	0.9940	1067.88	0.9693	1.56 MB
AttentionLSTM_3L_32N	5.32	3939.43	0.9933	1181.01	1.0122	1.13 MB

Among all the RNN variants, the LSTM models significantly outperformed their basic RNN counterparts. Even smaller LSTM configurations, such as LSTM_2L_128N and LSTM_3L_64N, maintained high accuracy (R^2 > 0.996) and reasonable inference times around ~0.79–0.93s.

We also witnessed that integrating attention mechanisms into LSTMs can improve the performance of LSTM. The Attention-LSTM with 3 layers and 128 neurons achieved high accuracy, R^2=0.9965, MSE = 2.77 × 107, and MAE = 2255.42 with a moderate inference time (0.8744 s).

5.7 Comparison with traditional machine learning models

Though the predictive performance of LSTM and Attention-LSTM is commendable, ML models, such as RF (R^2 = 0.9986) and BR (R^2 = 0.9984), outperformed them in accuracy and significantly in training time.

The LSTM, with three layers and 128 neurons, achieved the best performance across all architectures. However, its training time (1039.78 s) is significantly higher than that of RF (14.73 s) and XGBst (0.15 s).

In terms of resource consumption, deeper RNNs were computationally more expensive. Interestingly, the RNN architectures utilized memory on par with other ML models and considerably less than RF. For instance, the highest memory usage of RNN is 5.89 MB, but for RF, it is 127.55 MB despite using GPU acceleration. The longer training times and larger memory footprints remained a limitation of all the RNN variants compared with ML models.

However, inference times across DL and ML models were relatively comparable. For example, the two-layer RNN with 64 neurons and Tanh achieved an inference time of 0.6617 s, close to that of RF and KNN, having 0.3832s and 0.3586s, respectively. It suggests that once RNNs are trained, they can perform at speeds comparable to traditional ML models in time-sensitive applications.

Although RNN variants required more time and memory to train, optimized configurations achieved performance close to top ML models while maintaining real-time inference capabilities. These observations can also be drawn from the scatter plots shown in Figure 13.

The results from this study confirm that:

- LSTM and Attention-LSTM architectures outperform simple RNN variants.
- Tanh activation is superior to ReLU in deeper RNNs in this study.
- Model depth, neuron count, and activation functions substantially impact performance and resource requirements.

Attention mechanisms can enhance the robustness and performance of the models.

5.8 Analysis of AgroStackNet hybrid ensemble models

Out of all the evaluated configurations, the five best-performing AgroStackNet hybrids are presented in Table 10. The model combining XGBst, CatBst, RF, and BR with Ridge as the meta-learner achieved the lowest test MSE (1.0777 × 10^7) and the highest R^2 score (0.998653), indicating excellent

generalization.

Other configurations that used Ridge or BRidge as meta-learners also performed comparably well. However, they had a minimal trade-off in training time and memory usage. Overall, the Ridge consistently performed effectively across the best ensembles.

These results demonstrate that carefully assembled hybrid ensembles can significantly increase prediction accuracy and maintain efficiency. This approach eliminates the need to rely on overly complex model stacks.

Table 10. Top AgroStackNet hybrid models and their performance metrics

Rank	Model Configuration	Test MSE Mean ($\times 10^7$)	Test MAE Mean	R^2 Score Mean	Train Time (s)	Inference Time (s)	Base Memory (MB)
1	Stacking: (XGBoost + CatBoost + RF + Bagging) + Ridge	**1.0777**	533.49	**0.998653**	113.09	0.471	10.50
2	Stacking: (XGBoost + CatBoost + RF) + BRidge	1.0794	534.08	0.998651	104.68	0.383	10.50
3	Stacking: (XGBoost + RF + Bagging Regressor) + Ridge	1.0794	519.80	0.998651	81.75	0.442	10.50
4	Stacking: (RF + Bagging Regressor) + Ridge	1.0797	464.75	0.998650	80.90	0.428	10.50
5	Stacking: (CatBoost + RF + Bagging Regressor) + Ridge	1.0798	470.95	0.998650	111.10	0.465	10.50

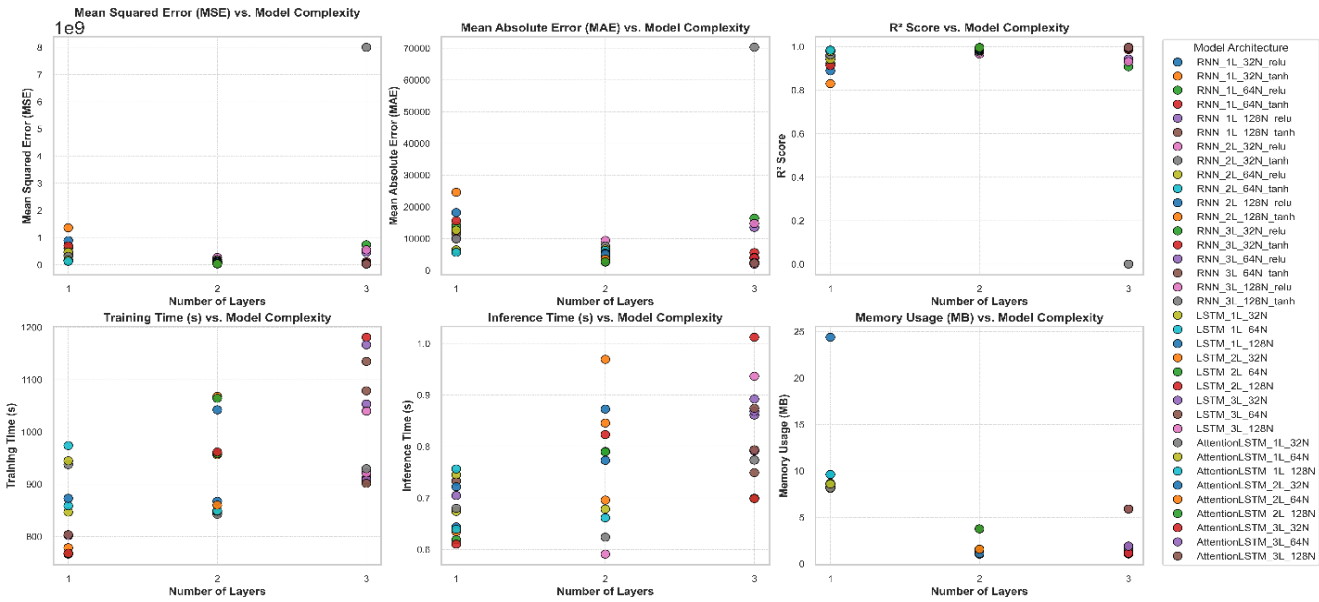

Figure 13. Performance evaluation of recurrent neural network (RNN) models

Table 11. Comparison with the prior studies

Model	MSE ($\times 10^7$)	MAE	R^2	Model Summary
RF [28]	18.74	5662.69	0.9744	Performed outlier removal using 1.5×IQR, one-hot encoding, Min-Max normalization for feature scaling, and random search for hyperparameter tuning.
KNN [29]	not reported	4650.21	0.9573	Performed one-hot encoding, Country-specific filtering (Saudi Arabia), Min-Max normalization for feature scaling, and manual hyperparameter tuning.
BR [29]	not reported	3294.23	0.9738	
RF [29]	not reported	3076.19	0.9740	
XGBst [29]	not reported	2681.33	0.9745	
RF [30]	**3.94**	4298.65	0.9674	Performed general cleaning, data transformation, and grid search for hyperparameter tuning. They deployed the model as an Android application.
XGBst [30]	1.84	2113.05	0.9848	
BR [30]	1.79	2177.05	0.9852	
RF [31]	677.25	3999.17	0.9837	Performed feature selection and standardized features using training data's mean and standard deviation.
LR [31]	892.43	57669.24	0.08628	
XGBst [31]	677.25	62779.32	0.9732	
RF [Our Model]	**1.08**	**433.02**	**0.9986**	Our Models: Rigorous preprocessing with advanced outlier handling (Winsorization, Log Transformation), 10-fold cross-validation, 2000 Bootstrap runs, GPR, and comprehensive accuracy and computational efficiency evaluation.
LR[Our Model]	669.0	66565.73	0.1642	
XGBst [Our Model]	3.98	2817.55	0.9950	
BR[Our Model]	**1.32**	**471.53**	**0.9984**	
KNN[Our Model]	6.55	1448.44	0.9918	

5.9 Comparative insights with prior research

To contextualize our findings, we benchmarked our models' performance against prior studies conducted on the same FAO-based crop yield dataset [28-31]. Table 11 summarizes the results of selected models from these studies, provides concise methodological descriptions, and presents the results of our corresponding implementations. We can observe from the table that our approach consistently achieves superior performance compared to prior studies and were made possible by rigorous preprocessing and advanced outlier handling techniques. Also, it can be noticed that these earlier works did not investigate the computational efficiency of the models, leaving a critical gap in CYP. This study evaluates computational efficiency as a core objective while achieving better predictions to address this gap. For example, our RF model achieved an R^2 of 0.9986 and an MAE of 433.02. This performance not only significantly surpasses prior RF implementations (with R^2 values ranging from 0.9674 to 0.9837 in prior studies) but also outperforms the overall best result from any prior model (R^2 = 0.9852, MAE = 2113.05 [30]). The slight improvements in R^2 score may seem modest, but they can significantly impact decisions in large-scale agricultural forecasting, such as those related to resource allocation, food security, and crop insurance schemes.

6. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This research thoroughly evaluated 17 models for CYP, including established methods such as RF, XGBst, GBst, and RNN variants of different depths and structures. The results demonstrate that model selection must be made carefully considering both accuracy and computational constraints. Among the models evaluated, RF achieved the highest performance with predictive power (R^2 = 0.9986, MSE = 1.08 × 10^7, MAE = 433.02). It balanced accuracy and generalization with relatively high memory consumption (126.93 MB) and relatively longer inference (0.411 s). On the other hand, BR was a viable contender (R^2 = 0.9984), which was less computationally demanding, using only 11.32 MB memory and taking 1.59 s of training time, which is suitable for resource-constrained environments. XGBst (R^2 = 0.9950) is another option if computational efficiency is essential. In contrast, linear models performed poorly (R^2 = 0.16), confirming their inability to capture complex nonlinear dependencies, whereas RNN models achieved R^2 scores of 0.9946 and 0.9923. In comparison, RNNs with two and three layers (128 neurons, tanh activation) performed on par with the best ensemble models. However, their longer training time and higher memory usage make them less suitable for large-scale deployment. Also, the results from RNN architectures confirm that RNNs with deeper architectures and Tanh activation are good options for sequential learning tasks.

6.1 Future research directions

Future studies should investigate hybrid modeling methods, combining various DL architectures with EL models to achieve structured feature representation and sequential dependencies.

- Transformer models and attention mechanisms can improve predictability with dynamically weighted influential features. Further, examining explainability methods, like SHAP values and attention visualizations, can make the models more interpretable for agricultural stakeholders.
- For greater applicability to the real world, subsequent research should emphasize scalability and deployment, tuning models for cloud-based or edge-computing inference to enable precision agriculture. Incorporation of multimodal data, such as satellite imagery, weather variables, and soil conditions, into this work would make the model even more robust. Model tuning for varied geographical locations with contrasting climatic conditions will also be essential for broad acceptance.
- Future developments can pave the way for more precise, scalable, and actionable CYP systems for global food security and sustainable agriculture by improving computational effectiveness, incorporating multimodal inputs, and enhancing model interpretability.

REFERENCES

[1] Oikonomidis, A., Catal, C., Kassahun, A. (2023). Deep learning for crop yield prediction: A systematic literature review. New Zealand Journal of Crop and Horticultural Science, 51(1): 1-26. https://doi.org/10.1080/01140671.2022.2032213

[2] Liu, Z., Di, L., Yang, R., Guo, L., Zhang, C., Li, H., Shao, B. (2026). In-season crop yield prediction: State of the art and future research direction. International Journal of Applied Earth Observation and Geoinformation, 146: 105129. https://doi.org/10.1016/j.jag.2026.105129

[3] Jayanthi, S., Rajkumar, K., Shaheen, Shrivastava, S., Herman, I.A. (2022). Design and development of framework for big data based smart farming system. In Innovations in Computer Science and Engineering, pp. 263-269. https://doi.org/10.1007/978-981-16-8987-1_27

[4] Leukel, J., Zimpel, T., Stumpe, C. (2023). Machine learning technology for early prediction of grain yield at the field scale: A systematic review. Computers and Electronics in Agriculture, 207: 107721. https://doi.org/10.1016/j.compag.2023.107721

[5] Obeidat, M.A., Abdallah, J., Hamadneh, T., Qawaqneh, H., Mansour, A.M. (2024). Enhancing agricultural operations through AI-driven agent communication in smart farming systems. Ingénierie des Systèmes d'Information, 29(3): 917-928. https://doi.org/10.18280/isi.290312

[6] Bi, L., Wally, O., Hu, G., Tenuta, A.U., Kandel, Y.R., Mueller, D.S. (2023). A transformer-based approach for early prediction of soybean yield using time-series images. Frontiers in Plant Science, 14: 1173036. https://doi.org/10.3389/fpls.2023.1173036

[7] Han, D., Wang, P.X, Tansey, K., Liu, J., Zhang, Y., Tian, H., Zhang, S.Y. (2022). Integrating an attention-based deep learning framework and the SAFY-V model for winter wheat yield estimation using time series SAR and optical data. Computers and Electronics in Agriculture, 201: 107334. https://doi.org/10.1016/j.compag.2022.107334

[8] Liu, N.T., Zhao, Q.S., Williams, R., Barrett, B. (2023). Enhanced crop classification through integrated optical and SAR data: A deep learning approach for multi-source image fusion. International Journal of Remote Sensing,

45(19-20): 7605-7633. https://doi.org/10.1080/01431161.2023.2232552
[9] Dong, G., Tang, M.Y., Wang, Z.Y., Gao, J.C., et al. (2023). Graph neural networks in IoT: A survey. ACM Transactions on Sensor Networks, 19(2): 1-50. https://doi.org/10.1145/3565973
[10] Wang, Y., Zhang, Q., Yu, F., Zhang, N., Zhang, X., Li, Y., Wang, M., Zhang, J. (2024). Progress in research on deep learning-based crop yield prediction. Agronomy, 14(10): 2264. https://doi.org/10.3390/agronomy14102264
[11] Iniyan, S., Varma, V.A., Naidu, C.T. (2023). Crop yield prediction using machine learning techniques. Advances in Engineering Software, 175: 103326. https://doi.org/10.1016/j.advengsoft.2022.103326
[12] Shams, M.Y., Gamel, S.A., Talaat, F.M. (2024). Enhancing crop recommendation systems with explainable artificial intelligence: A study on agricultural decision-making. Neural Computing and Applications, 36: 5695-5714. https://doi.org/10.1007/s00521-023-09391-2
[13] Wang, G., Li, B., Zhang, T., Zhang, S. (2022). A network combining a transformer and a convolutional neural network for remote sensing image change detection. Remote Sensing, 14(9): 2228. https://doi.org/10.3390/rs14092228
[14] Abdel-salam, M., Kumar, N., Mahajan, S. (2024). A proposed framework for crop yield prediction using hybrid feature selection approach and optimized machine learning. Neural Computing and Applications, 36: 20723-20750. https://doi.org/10.1007/s00521-024-10226-x
[15] Bouguettaya, A., Zarzour, H., Kechida, A. (2022). Deep learning techniques to classify agricultural crops through UAV imagery: A review. Neural Computing and Applications, 34: 9511-9536. https://doi.org/10.1007/s00521-022-07104-9
[16] Jhajharia, K., Mathur, P., Jain, S., Nijhawan, S. (2023). Crop yield prediction using machine learning and deep learning techniques. Procedia Computer Science, 218: 406-417. https://doi.org/10.1016/j.procs.2023.01.023
[17] Zhang, Y.F., Wang, L.C., Chen, X.X., Liu, Y.T., Wang, S.Q., Wang, L.Z. (2022). Prediction of winter wheat yield at county level in China using ensemble learning. Progress in Physical Geography: Earth and Environment, 46(5): 676-696. https://doi.org/10.1177/03091333221088018
[18] Alebele, Y., Wang, W., Yu, W., Zhang, X., Yao, X., Tian, Y. (2021). Estimation of crop yield from combined optical and SAR imagery using gaussian kernel regression. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 10520-10534. https://doi.org/10.1109/JSTARS.2021.3118707
[19] Zhang, T.X., Su, J.Y., Xu, Z.Y., Luo, Y.L., Li, J.Y. (2021). Sentinel-2 satellite imagery for urban land cover classification by optimized random forest classifier. Applied Sciences, 11(2): 543. https://doi.org/10.3390/app11020543
[20] Wang, J., Si, H.P., Gao, Z., Shi, L. (2022). Winter wheat yield prediction using an LSTM model from MODIS LAI products. Agriculture, 12(10): 1707. https://doi.org/10.3390/agriculture12101707
[21] Sagan, V., Maimaitijiang, M., Bhadra, S., Maimaitiyiming, M., Brown, D.R., Sidike, P., Fritschi, F.B. (2021). Field-scale crop yield prediction using multi-temporal WorldView-3 and PlanetScope satellite data and deep learning. ISPRS Journal of Photogrammetry and Remote Sensing, 174: 265-281. https://doi.org/10.1016/j.isprsjprs.2021.02.008
[22] Barbedo, J.G.A. (2023). A review on the combination of deep learning techniques with proximal hyperspectral images in agriculture. Computers and Electronics in Agriculture, 210: 107920. https://doi.org/10.1016/j.compag.2023.107920
[23] Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Tian, Q. (2022). Pangu-weather: A 3D high-resolution model for fast and accurate global weather forecast. arXiv preprint arXiv:2211.02556. https://doi.org/10.48550/arXiv.2211.02556
[24] Wang, D.S., Cao, W.J., Zhang, F., Li, Z.L., Xu, S., Wu, X.Y. (2022). A review of deep learning in multiscale agricultural sensing. Remote Sensing, 14(3): 559. https://doi.org/10.3390/rs14030559
[25] Talaat, F.M. (2023). Crop yield prediction algorithm (CYPA) in precision agriculture based on IoT techniques and climate changes. Neural Computing and Applications, 35: 17281-17292. https://doi.org/10.1007/s00521-023-08619-5
[26] Badshah, A., Alkazemi, B.Y., Din, F., Zamli, K.Z., Haris, M. (2024). Crop classification and yield prediction using robust machine learning models for agricultural sustainability. IEEE Access, 12: 162799-162813. https://doi.org/10.1109/ACCESS.2024.348665
[27] Hoque, M.J., Islam, M.S., Uddin, J., Samad, M.A., Abajo, B.S.D., Vargas, D.L.R. (2024). Incorporating meteorological data and pesticide information to forecast crop yields using machine learning. IEEE Access, 12: 47768-47786. https://doi.org/10.1109/ACCESS.2024.3383309
[28] Sharma, S., Walia, G.K., Singh, K., Batra, V., Sekhon, A.K., Kumar, A., Rawal, K., Ghai, D. (2024). Comparative analysis on crop yield forecasting using machine learning techniques. Rural Sustainability Research, 52(347): 63-77. https://doi.org/10.2478/plua-2024-0015
[29] Islam, M., Alharthi, M., Alkadi, R., Islam, R., Masum, A. (2024). Crop yield prediction through machine learning: A path towards sustainable agriculture and climate resilience in Saudi Arabia. AIMS Agriculture and Food, 9(4): 980-1003. https://doi.org/10.3934/agrfood.2024053
[30] Ismail, M., Muhammad, F.S., Ibrahim, M.M. (2024). Development and validation of an ensemble machine learning model for enhanced crop yield prediction. International Journal of Scientific Research and Modern Technology, 3(12): 25-32. https://doi.org/10.5281/zenodo.14557299
[31] Manjunath, M.C., Palayyan, B.P. (2023). An efficient crop yield prediction framework using hybrid machine learning model. Revue d'Intelligence Artificielle, 37(4): 1057-1067. https://doi.org/10.18280/ria.370428
[32] Crop Yield Prediction Dataset. https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset.
[33] FAO. https://www.fao.org/home/en/.
[34] World Bank Open Data. https://data.worldbank.org/.