Cureus Journal of Computer Science

Open Access Original Article

Machine Learners' Performance Analysis for Meta-Learning Base Classifier Selection in Plant Disease Diagnosis

Received 04/08/2025
Review began 04/17/2025
Review ended 06/06/2025
Published 06/09/2025

DOI:
https://doi.org/10.7759/s44389-025-03667-5

Adebukola Onashoga [1] , Femi Johnson [2] , Adebayo Abayomi-Alli [3, 1] , Olorunjube Falana [4] , Aderolu Ismaila [5] , Afolabi Clement [5]

1. Computer Science, Federal University of Agriculture, Abeokuta, Abeokuta, NGA 2. Artificial Intelligence, Federal University of Agriculture, Abeokuta, Abeokuta, NGA 3. HumanISE, Institute for Systems and Computer Engineering, Technology and Science, Porto, PRT 4. Cybersecurity and Data Science, Federal University of Agriculture, Abeokuta, Abeokuta, NGA 5. Crop Science, Federal University of Agriculture, Abeokuta, Abeokuta, NGA

Corresponding author: Femi Johnson, femijohnson123@hotmail.com

Abstract

Efficient hyper-parameter tuning is crucial for high-performing machine learning in plant disease diagnosis. This study comparatively analyzes K-nearest neighbor, random forest, neural networks, and AdaBoost models under a developed meta-learning Machine Learners' Performance Analysis framework to optimize their hyper-parameters for cassava plant disease datasets. We evaluate their adaptability, efficiency, and robustness using metrics like accuracy and computational cost. Applying meta-learning techniques, including gradient-based optimization and neural architecture search, we identify significant performance variations. Findings underscore the importance of selecting the optimal model-meta-learning combination for effective and automated plant disease diagnosis in precision agriculture.

Categories: Artificial Intelligence and Machine Learning in the Cloud, Computational Science and Engineering, Meta Computing
Keywords: machine model, meta-learning, parameters, diagnosis, plant disease, performance

Introduction

Machine learning constitutes a subfield within the domain of artificial intelligence (AI) that has garnered considerable scholarly attention over recent decades [1-2] owing to its effective implementation across a multitude of life domains. As its global acceptance continues to gain relevance, it persistently propels innovations throughout various sectors, including, but not limited to, agriculture, healthcare, finance, transportation, security, and entertainment, thereby transforming the methodologies employed for problem-solving and solution deployment [3]. The machine learning methodologies that are implemented demonstrate superiority over traditional techniques [4], which frequently entail high costs due to reliance on trial-and-error approaches that are time-consuming and necessitate manual interventions for parameter tuning and associated tasks. Given the extensive range and complexity of contemporary tasks and challenges that require resolution, it is evident that there are few, if any, entirely novel problems that warrant a completely new solution approach; instead, advancements and enhanced techniques are continually introduced to facilitate more efficient and intelligent methods for addressing both existing and emerging issues [5-6].

Accurate identification of plant diseases is crucial in agriculture to ensure productivity and reduce crop losses [7-8]. Traditionally, this task is carried out by plant pathologists [9], but it can also be done using AI systems that apply ensemble machine learning techniques [10]. While some machine learning models have shown strong performance in this area [11], achieving high accuracy requires careful tuning of hyper-parameters [12]. This process is essential but time-consuming and complex. To address this challenge, meta-learning offers a promising solution. It allows systems to automatically adjust hyper-parameters for new tasks or datasets by using previously learned knowledge [13-14]. This knowledge is represented through meta-features [15], such as the number of attributes, classes, sample sizes, and performance scores, and stored in a knowledge base [16]. This information is then used by a learning classifier to solve similar future problems effectively [17].

An important part of a meta-learning system is the base learner, which provides the parametric features used in training [18]. Choosing the right base learner depends on the specific task and its requirements. In this paper, we apply popular ML algorithms to a pre-processed dataset of cassava plant disease images, evaluate their performance, and identify the most suitable base learners for disease diagnosis models. There is growing interest in using machine and deep learning for plant disease detection. For example, Ayu et al. [12] developed an intelligent graphical user interface based on the MobileNetV2 deep learning model. This system helps users identify four common cassava diseases - Cassava Brown Streak Disease, Cassava Mosaic Disease (CMD), Cassava Bacterial Blight, and Cassava Green Mite - and distinguish them

from healthy plants using image data. The system used 1,769 cassava leaf images from Kaggle and was capable of performing identification, classification, and segmentation tasks. It achieved 43.2% and 29.4% accuracy for image and video data with mild disease symptoms, and overall accuracies of 74.5%, 65.6%, and 67.3% on the training, testing, and validation datasets, respectively.

Image quality is a known concern in plant disease detection. To address this, researchers [19] explored the use of data augmentation techniques to train a neural network using low-quality cassava leaf images. They used a modified MobileNetV2 model combined with a unique color transformation method called the Chebyshev orthogonal and probability function. This approach achieved strong results: an F1-score of 0.9676, precision of 0.9772, accuracy of 0.977, and recall of 0.9634 on the validation set. However, it was also noted that lower image quality led to decreased model accuracy. Similarly, to detect CMD from 5,656 infected cassava leaf images obtained from Kaggle, Oyewola et al. [8] used a deep residual neural network instead of a plain convolutional neural network. To balance the dataset, images of other cassava diseases were increased to 2,700 per class using a block processing technique. This step helped reduce bias caused by low contrast and resolution in images of CMD and Cassava Brown Streak Disease. The balanced dataset improved model accuracy significantly, achieving a 96.75% accuracy rate with an improvement of 9.25% over similar models.

Meta-learning frameworks are designed to adapt to new situations [17], often requiring only a few learning examples. They do this by minimizing training loss and learning optimal parameters that help the model generalize better [20]. Many studies have applied this concept using various meta-learning structures [21-22]. For instance, researchers [23] proposed the automated relational meta-learning framework, which automatically identifies relationships between tasks and builds a meta-knowledge graph based on past data. They tested this on a toy regression problem and found that automated relational meta-learning outperformed similar models. In another study [24], a metric-based multimodal fusion network was used to predict facial expressions using three datasets (SMIC, CASME, and CASMEII). After combining them into one dataset, the model's accuracy improved from 71% to 75% when additional data was added. Zhang et al. [25] introduced a meta-learning approach that considers individual differences. Using a transformer-based image segmentation model, they captured both short- and long-range dependencies. This helped the system generalize better across tasks involving different individuals.

In education, Hildago et al. [26] used deep learning to predict student performance in a virtual learning environment. Their model, which involved optimizing hyper-parameters, achieved over 90% predictive accuracy. Another study [13] proposed the AMLBID model, a two-phase machine learning approach aimed at helping engineers and researchers with limited analytics skills. This model addressed the limitations of existing meta-learning solutions and offered better decision-making tools. It outperformed existing frameworks, including Auto-Sklearn, in both accuracy and computational efficiency. The research conducted by Tiwari et al. [27] introduced a novel Meta-Lottery Ticket Hypothesis method for few-shot learning. Using three datasets, namely FC100, MiniImageNet, and Omniglot, the aim was to create a self-learning model capable of extracting features from new, unseen tasks. They used a 4CONV model with convolution layers, batch normalization, ReLU activation, and max pooling. The model adjusted learning rates and thresholds during training. Interestingly, most important features were learned in the early layers, while the final layer showed the highest learning depth. The Meta-Lottery Ticket Hypothesis model showed faster convergence and needed about 50% fewer batch updates compared to other state-of-the-art models.

Materials And Methods

This section describes the series of approaches applied to evaluating the performance of machine models using the developed Machine Learners' Performance Analysis (MALEPA) framework for cassava disease diagnosis. Four different machine learning algorithms are employed and critically evaluated with respect to their performance metrics. The dataset utilized, along with all pertinent preprocessing steps, is thoroughly detailed. This section shows the proposed methodology's stepwise architectures. Figure *1* illustrates the novel research workflow formulated for this investigation, and the stages are are carefully described below.

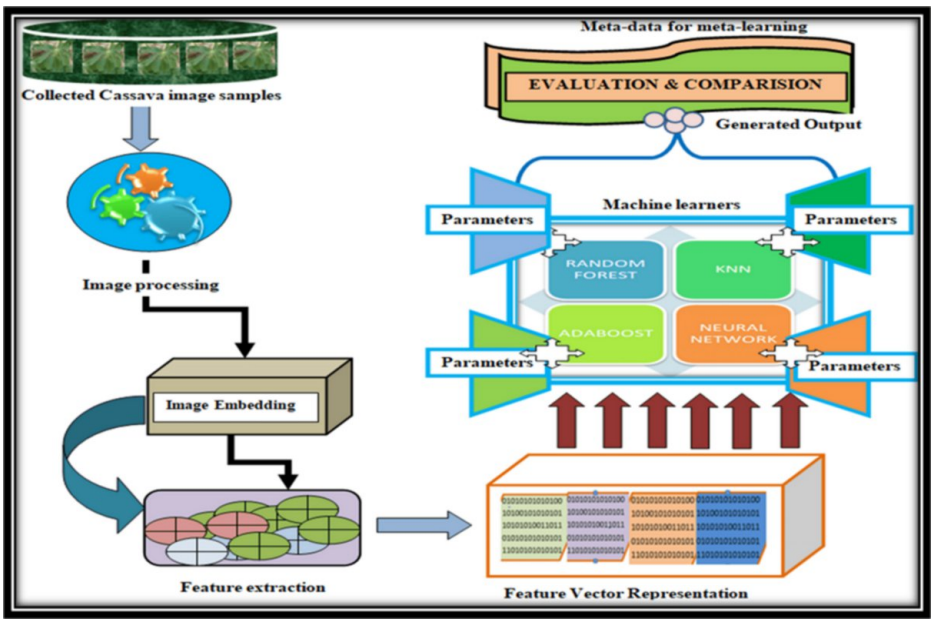


FIGURE 1: Architectural workflow of Machine Learners' Performance Analysis (MALEPA) framework

Dataset description

In this study, a comprehensive collection of 5,918 real-world images pertaining to cassava plant diseases were amassed from the Makerere University Artificial Intelligence Laboratory located in Uganda. The pre-processed and normalized form of this data is made available at https://www.kaggle.com/datasets/femijohnson/cassava-image-dataset. This dataset comprises a diverse assortment of two vegetative components (leaves and stems) encompassing five distinct classifications of cassava diseases, as delineated in Table *1*. It is noteworthy that the utilized dataset stands as one of the most coveted datasets employed for machine learning classification endeavors and concurrently addresses food security challenges within the Sub-Saharan African.

S/n	Plant	Vegetative Part	Diseases	Number of Images
	Cassava	Leaves stem	Cassava Mosaic Disease	2,658
			Cassava Green Mite	733
			Cassava Bacteria Blight	466
			Cassava Brown Spot	1,443
			Cassava Tuber Rot	302
			HEALTHY CASSAVA	316
Total				5,918

TABLE 1: Dataset description

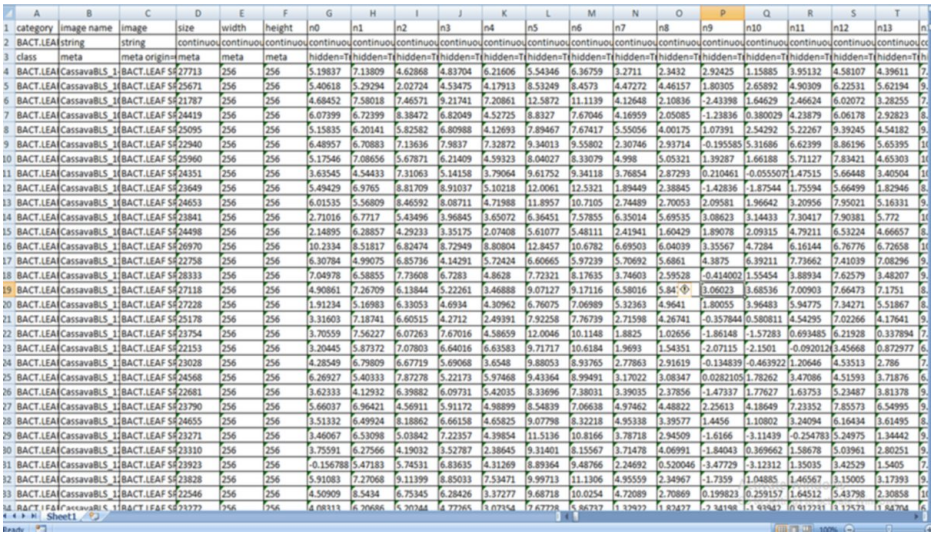
Image processing, embedding, and feature extraction

In order to ascertain equity in the comparative performance evaluation, the images undergo a preprocessing regimen that entails resizing them to a standardized dimension of 256 × 256 pixel JPEG format, as shown in Figure *2*. The heterogeneity evident in the images concerning background hues, resolution, and intensity levels is also duly considered during the preprocessing operations. These preprocessing measures are imperative to mitigate any conceivable bias or adverse influence on the classifiers performance metrics.

The embedding for all image samples was proficiently executed utilizing the SqueezeNet technique, which

necessitates a progressive multi-stage approach for its effective implementation. The results from the successful execution of all stages, as elucidated in Equations 1-5, culminate in the extraction of the most salient features from each sample image, thereby generating a list of numeric feature vectors as depicted in Figure 3 for subsequent processing.

FIGURE 2: Pre-processed images of cassava plants used for modeling



category	image name	image	size	width	height	n0	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11	n12	n13
BACT.LEA	string	string	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou	continuou
class	meta	meta origin	meta	meta	meta	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T	hidden=T
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	27713	256	256	5.19837	7.13809	4.62868	4.83704	6.21606	5.54346	6.36759	3.2711	2.3432	2.92425	1.15885	3.95132	4.58107	4.39611
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	25671	256	256	5.40618	5.29294	2.02724	4.53475	4.17913	8.53249	8.4573	4.47272	4.46157	1.80305	2.65892	4.90309	6.22531	5.62194
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	21787	256	256	4.68452	7.58018	7.46571	9.21741	7.20861	12.5872	11.1139	4.12648	2.10836	-2.43398	1.64629	2.46624	6.02072	3.28255
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24419	256	256	6.07399	6.72399	8.38472	6.82049	4.52725	8.8327	7.67046	4.16959	2.05085	-1.23836	0.380029	4.23879	6.06178	2.92823
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	25095	256	256	5.15835	6.20141	5.82582	6.80988	4.12693	7.89467	7.67417	5.55056	4.00175	1.07391	2.54292	5.22267	9.39245	4.54182
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	22940	256	256	6.48957	6.70883	7.13636	7.9837	7.32872	9.34013	9.55802	2.30746	2.93714	-0.195585	5.31686	6.62399	8.86196	5.65395
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	25960	256	256	5.17546	7.08656	5.67871	6.21409	4.59323	8.04027	8.33079	4.998	5.05321	1.39287	1.66188	5.71127	7.83421	4.65303
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24351	256	256	3.63545	4.54433	7.31063	5.14158	3.79064	9.61752	9.34118	3.76854	2.87293	0.210461	-0.055507	1.47515	5.66448	3.40504
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23649	256	256	5.49429	6.9765	8.81709	8.91037	5.10218	12.0061	12.5321	1.89449	2.38845	-1.42836	-1.87544	1.75594	5.66499	1.82946
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24653	256	256	6.01535	5.56809	8.46592	8.08711	4.71988	11.8957	10.7105	2.74489	2.70053	2.09581	1.96642	3.20956	7.95021	5.16331
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23841	256	256	2.71016	6.7717	5.43496	3.96845	3.65072	6.36451	7.57855	6.35014	5.69535	3.08623	3.14433	7.30417	7.90381	5.772
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24498	256	256	2.14895	6.28857	4.29233	3.35175	2.07408	5.61077	5.48111	2.41941	1.60429	1.89078	2.09315	4.79211	6.53224	4.66657
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	26970	256	256	10.2334	8.51817	6.82474	8.72949	8.80804	12.8457	10.6782	6.69503	6.04039	3.35567	4.7284	6.16144	6.76776	6.72658
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	22758	256	256	6.30784	4.99075	6.85736	4.14291	5.72424	6.60665	5.97239	5.70692	5.6861	4.3875	6.39211	7.73662	7.41039	7.08296
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	28333	256	256	7.04978	6.58855	7.73608	6.7283	4.8628	7.72321	8.17635	3.74603	2.59528	-0.414002	1.55454	3.88934	7.62579	3.48207
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	27118	256	256	4.90861	7.26709	6.13844	5.22261	3.46888	9.07127	9.17116	6.58016	5.84	1.06023	3.68536	7.00903	7.66473	7.1751
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	27228	256	256	1.91234	5.16983	6.33053	4.6934	4.30962	6.76075	7.06989	5.32363	4.9641	1.80055	3.96483	5.94775	7.34271	5.51867
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	25178	256	256	3.31603	7.18741	6.60515	4.2712	2.49391	7.92258	7.76739	2.71598	4.26741	-0.357844	0.580811	4.54295	7.02266	4.17641
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23754	256	256	3.70559	7.56227	6.07263	7.67016	4.58659	12.0046	10.1148	1.8825	1.02656	-1.86148	-1.57283	0.693485	6.21928	0.337894
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	22153	256	256	3.20445	5.87372	7.07803	6.64016	6.63583	9.71717	10.6184	1.9693	1.54351	-2.07115	-2.1501	-0.092012	3.45668	0.872977
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23028	256	256	4.28549	6.79809	6.67719	5.69068	3.6548	9.88053	8.93765	2.77863	2.91619	-0.134839	-0.463922	1.20646	4.53513	2.786
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24568	256	256	6.26927	5.40333	7.87278	5.22173	5.97468	9.43364	8.99491	3.17022	3.08347	0.0282105	1.78262	3.47086	4.51593	3.71876
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	22681	256	256	3.62333	4.12932	6.39882	6.09731	5.42035	8.33696	7.38031	3.39035	2.37856	-1.47337	1.77627	1.63753	5.23487	3.81378
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23790	256	256	5.66037	6.96421	4.56911	5.91172	4.98899	8.54839	7.06638	4.97462	4.48822	2.25613	4.18649	7.23352	7.85573	6.54995
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	24655	256	256	1.51332	6.49924	8.18862	6.66158	4.65825	9.07798	8.32218	4.95338	3.39577	1.4456	1.10802	3.24094	6.16434	3.61495
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23271	256	256	3.46067	6.53098	5.03842	7.22357	4.39854	11.5136	10.8166	5.78718	2.94509	-1.6166	-3.11439	-0.254783	5.24975	1.34442
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23310	256	256	3.75591	6.27566	4.19032	3.52787	2.38645	9.31401	8.15567	3.71478	4.06991	-1.84043	0.369662	1.58678	5.03961	2.80251
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23923	256	256	-0.156788	5.47183	5.74531	6.83635	4.31269	8.89364	9.48766	2.24692	0.520046	-3.47729	-3.12312	1.35035	3.42529	1.5405
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	23828	256	256	5.91083	7.27068	9.11399	8.85033	7.53471	9.99713	11.1306	4.95559	2.34967	-1.7359	1.04885	1.46567	3.15005	3.17393
BACT.LEA	CassavaBLS_1	BACT.LEAF SE	22546	256	256	4.50909	8.5434	6.75345	6.28426	3.37277	9.68718	10.0254	4.72089	2.70869	0.199823	0.259157	1.64512	5.43798	2.30858

FIGURE 3: CSV file of extracted cassava plant features before normalization

Furthermore, all samples are split into 80:20 and subjected to normalization via the min-max technique to guarantee that all features operate on a uniform rating scale, thereby contributing equitably and mitigating bias in predictive outcomes. The mathematical formulation of the normalization technique is represented in Equation 6. A substantial 80% of the dataset is utilized for the training of machine learning algorithms, whereas the remaining 20% is adopted for the assessment of the model's efficacy. The partitioning of the dataset is executed utilizing the train_test_split() function from the scikit-learn library, thereby addressing concerns related to overfitting and augmenting the model's performance with the application of k-fold cross-validation to segment the data for the purpose of validating the results.

$$\text{Image Input: } I \in \mathbb{R}^{H \times W \times N} \quad (1)$$

where
I = image input

H = height
W = width
N = number of channels for RGB images.

The Fire Module consists of two layers described in Equations 2 and 3:

1) Squeeze Layer: Applies 1×1 convolutions to reduce the number of channels/dimensions.

2) Expand Layer: Combines (1×1 and 3×3) convolutions on the output of the squeeze layer:

$$Z_{\text{squeeze}} = \text{ReLU}\left(\text{Conv}_{1\times1}(I)\right) \quad (2)$$

$$Z_{\text{expand}} = \text{ReLU}\left([\text{Conv}_{1\times1}(Z_{\text{squeeze}}),\ \text{Conv}_{3\times3}(Z_{\text{squeeze}})]\right) \quad (3)$$

Stacking Fire Modules: $P_{\text{module}_i} = \text{Fire}(P_{\text{module}_i}) - 1$ (4)

Global Average Pooling: $O_{\text{Embed}} = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} O[h, w, :]$ (5)

where O is the output feature map of the last Fire module, and H and W are its spatial dimensions.

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (6)$$

where
X = original value of the feature.
min(X) = minimum value of the feature in the dataset.
max(X) = maximum value of the feature in the dataset.
X' = normalized value of x, scaled to a range (0, 1).

Machine learners' and parameter description

Selected machine learners, as shown in Figure *1*, are used for the training. Each of these learners has shown positive result and high accuracy for similar classification problems. The random forest being one of the learners used represents a fusion of tree predictions, with each tree relying on the selected value of a vector. Upon receiving fresh input information, its algorithm creates a decision tree specific to the data and integrates it into a forest along with supplementary decision trees. K-nearest neighbors (KNN) utilize distance metrics to assess similarity between plant disease samples. It predicts the disease class of a new sample by identifying the most frequent label among its nearest neighbors, making it a simple yet effective method for disease classification. Also, considering the modus operandi of neural network with information passing through each layer and establishing connections within and between layers for accurate prediction has been substantiated in many scholarly publications. The last learner is an ensemble learning named Adaboost designed to improve the accuracy of weak classifiers by combining them into a strong classifier. Each weak classifier sequentially focuses on the mistakes (misclassifications) of the previous one. It relies on the adaptability of the boosting process, reducing the need for hyper parameter tuning for the base models. The models and their hyper-parameters are reflected in Table *2*. The resulting effects of the performance of these learners are further evaluated using evaluated using common machine learning metrics.

S/n	Learners	Hyper-parameters
1	K-Nearest Neighbors	Number of neighbors = 5
		Base metric = Euclidean distance
		Weight = Uniform
2	Random Forest	Number of trees = 100
		Number of attributes at each epoch = 4
		Depth limit of trees = 10
3	AdaBoost	Base estimator = Tree
		Number of estimators = 50
		Learning rate = 1.0
		Loss function = linear
4	Neural Network	Number of neurons = 10
		Optimizer = Adam
		Activation function = ReLU
		Regression rate = 0.001
		Maximum number of iterations: 200

TABLE 2: Optimized hyper-parameters in used machine learning models

Implementation

The implementation of this study with interface represented in Figure *4* is conducted on a LENOVO T410 PC, which operates on an Intel Core™ i5-5020U CPU at a frequency of 2.20 GHz, equipped with a 250GB HDD, 4.0 GB of RAM, and a suite of installed software that includes the Microsoft Office Suite, Orange 3.8 data mining tool, and PyCharm software. Each of these components is utilized effectively for the execution of this investigation. The implementation procedure commenced with the augmentation of various classes of imbalanced data, achieving a standardized data size with equal quantities of 3,000 samples for each class, as illustrated in Figure *5*. Table *3* shows the 10 most significant characteristic features (components) of each image within the dataset are extracted and validated to ascertain their impact on the dataset as represented in Table *2* and Figure *6*.

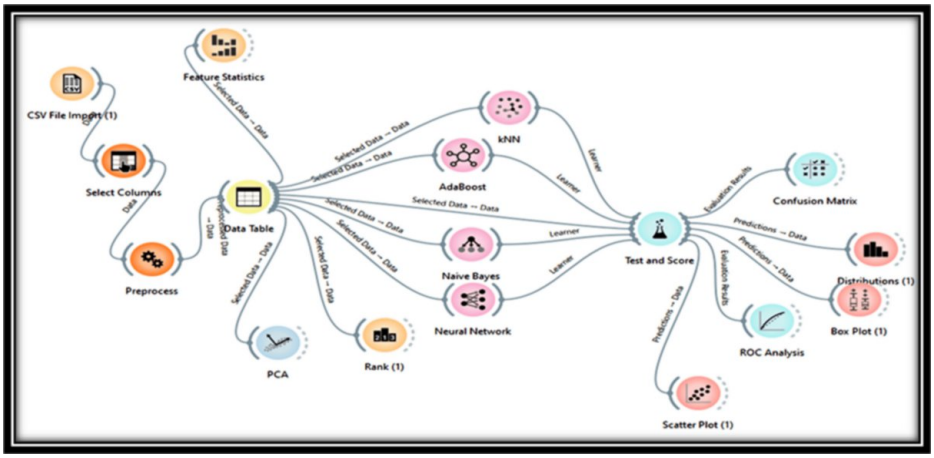


FIGURE 4: Implementation interface of Machine Learners' Performance Analysis

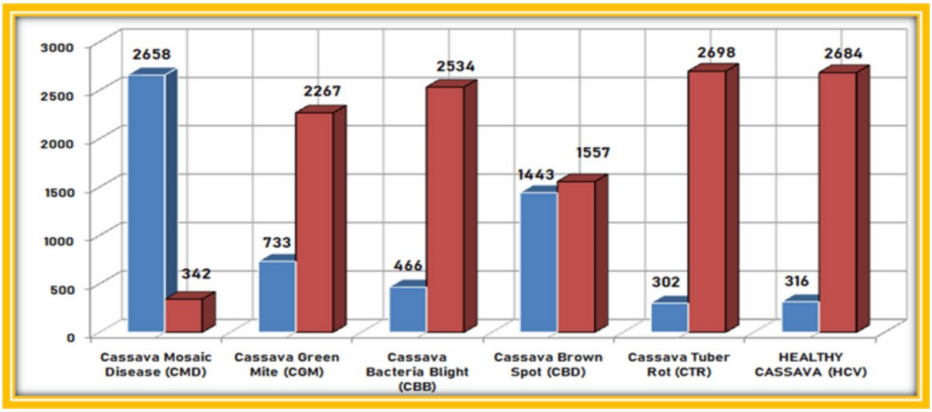


FIGURE 5: Sample size of real and augmented dataset

Component	Image Feature	Description
PC 1.	Area of infected regions	The total surface of plant tissues visibly affected by disease symptoms
PC 2.	Number of spots or lesions	The count of visible disease-affected areas
PC 3.	Shape irregularity index	A deviation of a lesion's shape from a perfect form
PC 4.	Colour Correlation (GLCM)	The changes in the pixel intensity patterns
PC 5.	Excess Red Index (ERI)	A highlights of red intensities in diseased plants
PC 6.	Edge density	The concentration of edges in diseased plant regions
PC 7.	Perimeter of infected regions	The quantifiers of the boundary length of diseased areas
PC 8.	Skewness score	A measure of the asymmetry of pixel intensity distribution in diseased plant regions
PC 9.	Aspect ratio	The value of the compared width to the height of infected regions in diseased plants
PC 10.	Color entropy	The measure of randomness in the colours within an image

TABLE 3: Extracted image feature components

Moreover, the comprehensive nature of the dataset is evaluated through the feature statistics collected, which are represented in Figure 7. Following a thorough assessment of the database and the image features it comprises, training datasets encompassing 70% of the total dataset are presented to the machine learners for iterative learning until the optimal set of hyper-parameter tuning is attained, thus yielding the best performance outcomes. Upon the identification of the optimal hyper-parameters, as delineated in Table 2, which were employed for training, a fresh set of undisclosed test datasets, constituting 30%, was subsequently introduced to validate the learning experiences accrued by the machine models, and the corresponding performance scores were compiled for comparative analysis and evaluation. The performance metrics employed for the comparative analysis of learners include accuracy, precision, recall, F-measure scores, and Matthews's correlation coefficient (MCC) for the various categories of cassava diseases. The mathematical formulations for these evaluation metrics are represented in Equations 7-11, respectively.

$$\text{Accuracy} = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \times 100 \quad (7)$$

$$\text{Precision}(P) = \left(\frac{TP}{TP + FP}\right) \times 100 \quad (8)$$

$$\text{Recall}(R) = \left(\frac{TP}{TP + FN}\right) \times 100 \quad (9)$$

$$\text{FMeasure}(FMeas.) = 2 \times \left(\frac{P \times R}{P + R}\right) \quad (10)$$

$$\text{Matthews Correlation Coefficient (MCC)} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (11)$$

where
TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

Results

The findings produced in this investigation are compelling and merit particular attention. Initiating with the outcomes obtained during the data preparation stage, as illustrated in Figure *4*, it is evident that there exist imbalanced classes of data pertaining to various cassava diseases. CMD is identified as the predominant disease impacting cassava, thereby exhibiting the most substantial sample size of 2,648 diseased images, in contrast to the minimal augmented image count of 342 samples. This is succeeded by cassava brown spot disease, which has a genuine image sample of 1,443, reflecting a difference of 114 between the real images and the augmented samples. Other augmented images within these categories consist of approximately 1,000-2,000 images to support the real images obtained. This clearly signifies a scarcity of genuine data samples for the diagnosis of cassava diseases. The results derived from the implementation are further depicted in Figures *6*-*13*.

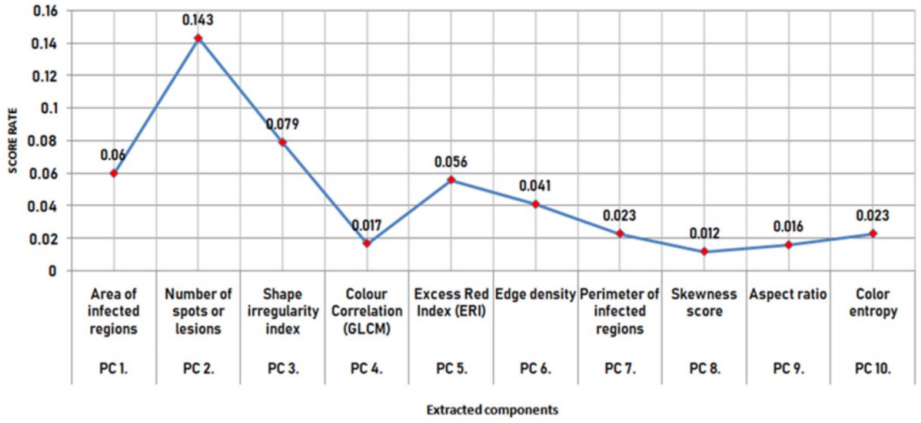
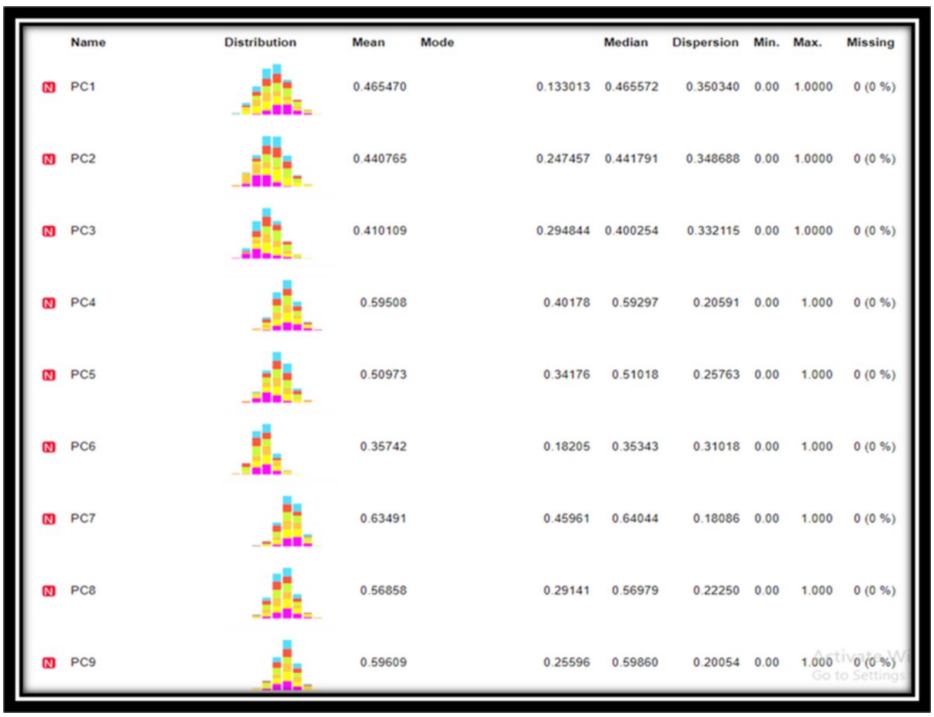


FIGURE 6: Gain ratio score of the image components



	Name	Distribution	Mean	Mode		Median	Dispersion	Min.	Max.	Missing
N	PC1		0.465470		0.133013	0.465572	0.350340	0.00	1.0000	0 (0 %)
N	PC2		0.440765		0.247457	0.441791	0.348688	0.00	1.0000	0 (0 %)
N	PC3		0.410109		0.294844	0.400254	0.332115	0.00	1.0000	0 (0 %)
N	PC4		0.59508		0.40178	0.59297	0.20591	0.00	1.000	0 (0 %)
N	PC5		0.50973		0.34176	0.51018	0.25763	0.00	1.000	0 (0 %)
N	PC6		0.35742		0.18205	0.35343	0.31018	0.00	1.000	0 (0 %)
N	PC7		0.63491		0.45961	0.64044	0.18086	0.00	1.000	0 (0 %)
N	PC8		0.56858		0.29141	0.56979	0.22250	0.00	1.000	0 (0 %)
N	PC9		0.59609		0.25596	0.59860	0.20054	0.00	1.000	0 (0 %)

FIGURE 7: Feature statistics of dataset principal components

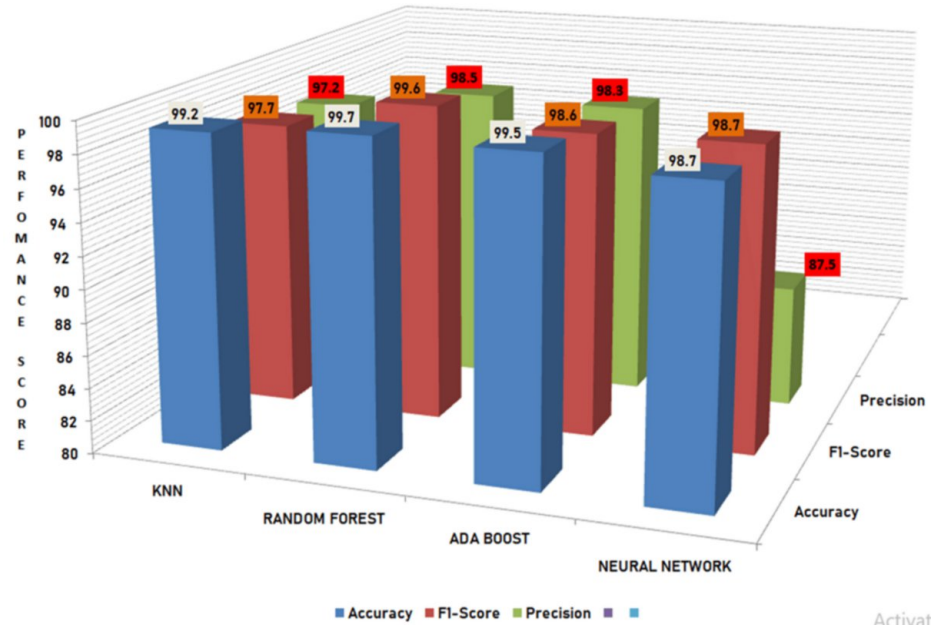


FIGURE 8: Average performance score of learners on accuracy, F1-score, and precision metrics

KNN, K-Nearest Neighbors

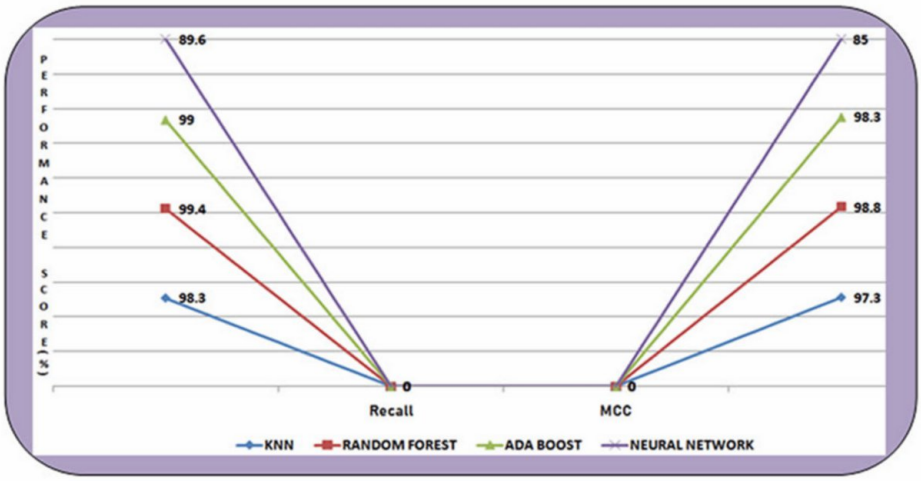


FIGURE 9: Average performance score of learners on recall and MCC metrics

KNN, K-Nearest Neighbors; MCC, Matthews's Correlation Coefficient

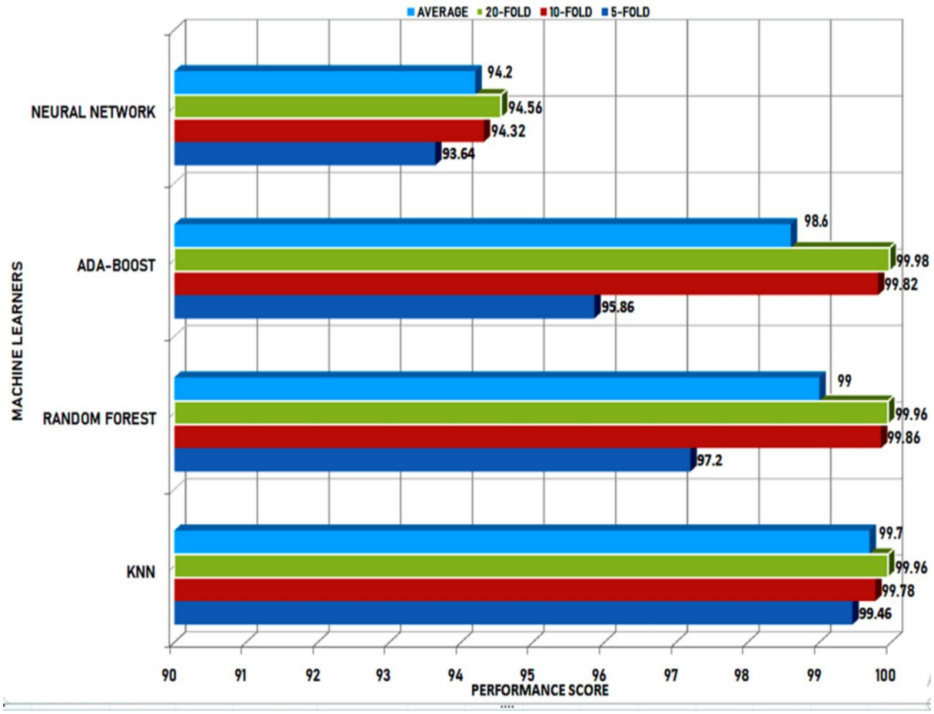


FIGURE 10: K-fold validation assessment score of machine learners

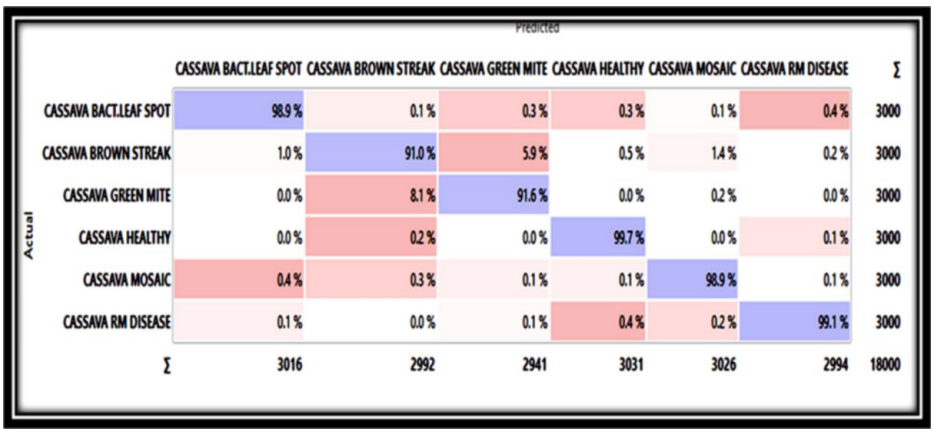


FIGURE 11: AdaBoost confusion matrix for cassava diagnosis

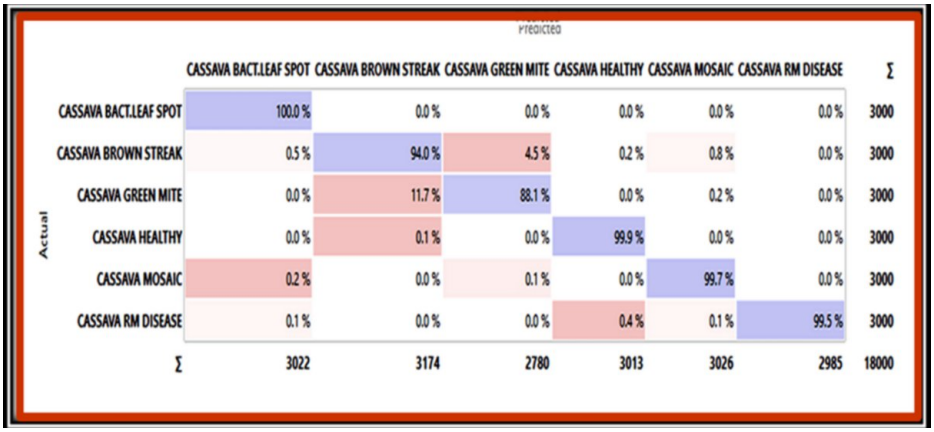


FIGURE 12: Random forest confusion matrix for cassava diagnosis

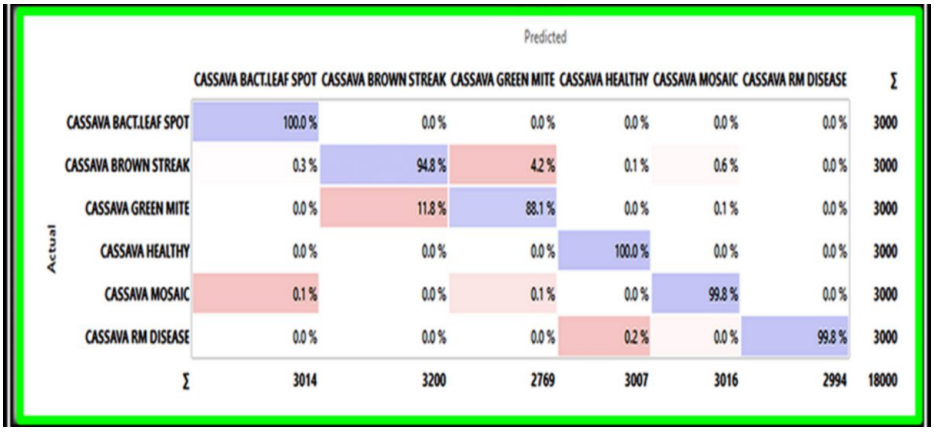


FIGURE 13: KNN confusion matrix for cassava diagnosis

Discussion

In the realm of image processing, various manipulation techniques are employed, including saturation adjustments, brightness modifications, and both linear and horizontal flips. Moreover, specific salient features pertinent to the diagnosis of diseased plants are ranked to illustrate their impact rating, as depicted in Figure 6. At the forefront of this list is the number of spots or lesions on plants. A positive correlation exists between the quantity of spots and the status of disease prediction in plants. An aggregate ranking of 0.14 among the selected features underscores the preeminence of this characteristic for the identification of diseased plant status. Additional features exerting significant influence include

the shape irregularity index, the area of infected regions, and the Excess Red Index, which are assigned values of 0.079, 0.06, and 0.056, respectively. The integration and comparison of the cumulative ranking influence of these features for plant disease diagnosis demonstrate that these features can be utilized autonomously or in conjunction with one or more of the remaining features to achieve precise disease diagnosis.

Furthermore, it is noteworthy that the similarity score of 0.023 achieved by colour entropy and the perimeter of infected regions in diseased plants indicates that both features may be interchangeable in scenarios necessitating a reduced number of features. The lowest score of 0.017 among the features pertains to colour correlation (GLCM), which reflects alterations in colour intensity patterns of diseased plants and influences the efficacy of plant disease diagnosis. In Figure *7*, the quality and validity of the database employed in this study are assessed through statistical metrics, and the dataset is devoid of any missing values across all features, with the highest mean, mode, and median ratings recorded for the perimeter of infected regions, yielding scores of 0.63, 0.46, and 0.64, respectively. Additionally, the dispersion scores range from 0.18 to 0.35. Subjecting this meticulously analyzed dataset and its features to machine learning algorithms, utilizing optimal parameters as delineated in Table *2*, produced outcomes as illustrated in Figures *8*-*10*. In Figure *8*, three common metrics (accuracy, F1-score, and precision) are documented, revealing that the highest scores of 99.7%, 99.6%, and 98.5% are, respectively, attained by the random forest algorithm. This indicates that the random forest model emerges as the superior classifier within the ensemble of algorithms. The AdaBoost classifier secures the position of the first runner-up, achieving values of 99.5%, 98.6%, and 98.3%, thereby exceeding the performance of KNN, which recorded values of 99.2%, 97.7%, and 97.2% for accuracy, F1-score, and precision, respectively. Despite the neural network achieving commendable scores of 98.7% for both accuracy and F1-score, its precision rate, quantified at 87.5%, renders it comparatively less effective than the other models assessed.

Moreover, the metrics pertaining to recall and MCC illustrate that random forest outperformed the other classifiers, attaining values of 99.4% and 98.8%. AdaBoost similarly demonstrates a performance level closely aligned with that of the random forest model. The lowest performance scores of 89.6% and 85.5% were recorded by the neural network. A critical hyper-parameter examined in this study pertains to the number of folds utilized by the learners. Each algorithm was evaluated under three distinct fold configurations (5, 10, and 20) to assess performance concerning the diagnosis of cassava plant diseases. The results, as illustrated in Figure *10*, indicate that all algorithms exhibit optimal performance at the 20-fold validation state, with scores ranging from 94.56% to 99.98% for the neural network and AdaBoost. Notably, KNN attained the highest average rating score of 99.7% across the various fold validation states, consistently maintaining scores exceeding 99% across each validation scenario. A comparative analysis of the average performance ratings across the folds reveals that KNN, with a score of 99.7%, surpasses the other learners. Random forest, AdaBoost, and the neural network achieved average scores of 99%, 98.6%, and 94.2%, respectively, across the folds.

In addition, the performance results of the learners are depicted through confusion matrices, as presented in Figures *11*-*13*. Utilizing the AdaBoost learner for diagnostic purposes indicates that the classifications for healthy plants and tuber rot disease of cassava were achieved with a prediction value exceeding 99%, while bacterial leaf spot and mosaic disease exhibited equal prediction values of 98.9%. In contrast, brown streak and green mite diseases demonstrated scores ranging from 91% to 91.6%. Both random forest and KNN attained a perfect diagnostic score of 100% for bacterial leaf spot, as illustrated in Figures *12* and 13. Among all the learning models assessed, the KNN model provided the most accurate diagnosis for brown streak, achieving an accuracy score of 94.8%. AdaBoost recorded the highest diagnostic score of 98.1% for green mite disease, whereas random forest and KNN achieved identical scores of 88.1%. The healthy plants were diagnosed with accuracy rates ranging from 99.1% to 100% across the learners. AdaBoost exhibited the lowest accuracy scores of 98.9% and 99.1% for mosaic and tuber rot diseases, respectively. Remarkably, KNN surpassed its counterparts with a classification score of 99.8% for tuber rot disease, while random forest achieved the highest rating score of 99.7% for the diagnosis of mosaic disease.

Conclusions

Implications for research and practice

This study provides key insights that are valuable for both the research community and practical implementation in agriculture-based AI systems:

1. Feature Optimization for Accurate Diagnosis: The identification of high-impact features such as the number of lesions, shape irregularity, infected area, and Excess Red Index demonstrates that effective plant disease diagnosis can be achieved with a minimal yet optimized set of image features. This supports the development of lightweight, efficient models suitable for real-world applications, especially in low-resource settings.

2. Superior Performance of Ensemble Learning Models: Random forest and AdaBoost consistently outperformed other classifiers across all evaluation metrics, including accuracy, precision, F1-score, and

MCC. Their robustness makes them ideal candidates for deployment in automated plant disease diagnostic systems, offering reliable, scalable solutions for farmers and agricultural stakeholders.

3. Scalability Through Model Generalization and Data Quality: The consistent performance improvements at higher cross-validation folds and the use of a clean, well-distributed dataset affirm the importance of data quality and validation strategy. These findings provide practical guidelines for building generalizable models and designing scalable, high-accuracy diagnostic tools.

4. Disease-Specific Model Adaptation: Variation in model performance across different disease types suggests that customized models tailored to specific cassava diseases (e.g., Brown Streak, Green Mite) can enhance diagnostic precision. This paves the way for more targeted and reliable disease management interventions.

Conclusively, the objectives of this research endeavor have been successfully attained through the systematic analysis conducted by various machine learning algorithms aimed at accurately diagnosing diseases in cassava plants to facilitate effective selection of base classifiers. This domain of machine learning has been meticulously directed towards agricultural applications, particularly focusing on disease diagnosis in the cassava plant, which is a staple crop widely consumed across Africa. An adequately curated dataset of cassava plant images served as the foundational resource from which the derived results were obtained, in conjunction with four distinct machine learning algorithms whose hyper-parameters were meticulously optimized to achieve maximal performance. Through an array of parameters and training sequences, complex yet comprehensive insights have emerged regarding the efficacy of the machine learners. This further reinforces the premise that no singular model is universally applicable across all contexts of plant disease diagnosis. The comparative analyses of the performances elucidate the most versatile set of features, meta-parameters, and learning algorithms to be utilized as base learners, with the primary objective of minimizing computational expenses for any proposed meta-learning framework aimed at diagnosing cassava diseases in agricultural settings. Future investigations in this area will endeavor to leverage automated classifier evaluation of selected features and hyper-parameters for disease diagnosis, while substantially reducing computational costs, optimizing memory usage, and enhancing accuracy.

Appendices

Data Source: Cassava Image Dataset for Meta-Learning.

Available at: https://www.kaggle.com/datasets/femijohnson/cassava-image-dataset

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Femi Johnson, Adebukola Onashoga , Adebayo Abayomi-Alli, Olorunjube Falana, Aderolu Ismaila, Afolabi Clement

Acquisition, analysis, or interpretation of data: Femi Johnson, Adebukola Onashoga , Adebayo Abayomi-Alli, Olorunjube Falana, Aderolu Ismaila, Afolabi Clement

Drafting of the manuscript: Femi Johnson, Adebukola Onashoga , Adebayo Abayomi-Alli, Aderolu Ismaila, Afolabi Clement

Critical review of the manuscript for important intellectual content: Femi Johnson, Adebukola Onashoga , Adebayo Abayomi-Alli, Olorunjube Falana, Aderolu Ismaila, Afolabi Clement

Supervision: Femi Johnson, Adebukola Onashoga , Adebayo Abayomi-Alli, Olorunjube Falana, Aderolu Ismaila, Afolabi Clement

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there

are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Johnson FT, Onashoga A, Thomas I, Victor O, Cecilia A: WORK-PERF: An intelligent predictive model for work performance rating. Proceedings of Third Emerging Trends and Technologies on Intelligent Systems. Noor A, Saroha K, Pricop E, Sen A, Trivedi G (ed): Springer, Singapore; 2023. 730:11-20. 10.1007/978-981-99-3963-3_2
2. Musinat B, Johnson F, Folorunso O, Ezinne I: Genetic algorithm-based multi-objective optimization model for software bugs prediction. Annual Journal of Technical University of Varna, Bulgaria. 2022, 6:34-48. 10.29114/ajtuv.vol6.iss1.245
3. Onashoga A, Ojesanmi O, Johnson F, Ayo FE: A fuzzy-based decision support system for soil selection in olericulture. Journal of Agricultural Informatics. 2018, 93:65-77. 10.17700/jai.2018.9.3.480
4. Johnson FT, Akerele J: Artificial intelligence for weed detection - A techno-efficient approach. ICTACT Journal on Image and Video Processing. 2020, 11:2297-2305. 10.21917/ijivp.2020.0326
5. Ahmad A, Saraswat D, El Gamal A: A survey on using deep learning techniques for plant disease diagnosis recommendations for development of appropriate tools. Smart Agricultural Technology. 2023, 3:100083. 10.1016/j.atech.2022.100083
6. Onashoga A, AbdulAzeez A, Johnson FT, Rele A, Vivian N: EAAICS: A technological driven system for improving crop productivity. Annals. Computer Science Series. 2021, 19:37-45.
7. Albattah W, Nawaz M, Javed A, Masood M, Albahli S: A novel deep learning method for detection and classification of plant diseases. Complex Intelligent Systems. 2022, 8:507-524. 10.1007/s40747-021-00536-1
8. Oyewola DO, Dada EG, Misra S, Damaševičius R: Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. PeerJ Computer Science. 2021, 7:e352. 10.7717/peerj-cs.352
9. Ngugi HN, Akinyelu AA, Ezugwu AE: Machine learning and deep learning for crop disease diagnosis: performance analysis and review. Agronomy. 2024, 14:3001. 10.3390/agronomy14123001
10. Onashoga AS, Johnson FT, Ojo OE, Aderolu IA: A critical review of machine learning models for intelligent plant disease diagnosis. International Conference on Communication and E-Systems For Economic Stability. 2023, 359-368.
11. Ireri D, Belal E, Okinda C, Makange N, Ji C: A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. Artificial Intelligence in Agriculture. 2019, 2:28-37. 10.1016/j.aiia.2019.06.001
12. Ayu HR, Surtono A, Apriyanto DK: Deep learning for detection cassava leaf disease. Journal of Physics: Conference Series. 2021, 1751:012072. 10.1088/1742-6596/1751/1/012072
13. Garouani M, Ahmad A, Bouneffa M, Hamlich M, Bourguin G, Lewandowski A: Using meta-learning for automated algorithms selection and configuration: An experimental framework for industrial big data. Journal of Big Data. 2022, 9:57. 10.1186/s40537-022-00612-4
14. Brazdil P, van Rijn JN, Soares C, Vanschoren J: Metalearning: Applications to Automated Machine Learning and Data Mining. Springer, Cham; 2022. 10.1007/978-3-030-67024-5
15. Golshanrad P, Rahmani H, Karimian B, Karimkhani F, Weiss G: MEGA: Predicting the best classifier combination using meta-learning and a genetic algorithm. Intelligent Data Analysis. 2021, 25:1547-1563. 10.3233/IDA-205494
16. Li L, Wang Y, Xu Y, Lin K-Y: Meta-learning based industrial intelligence of feature nearest algorithm selection framework for classification problems. Journal of Manufacturing Systems. 2022, 62:767-776. 10.1016/j.jmsy.2021.03.007
17. Vettoruzzo A, Bouguelia M-R, Vanschoren J, Rögnvaldsson T, Santosh K: Advances and challenges in meta-learning: a technical review. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024, 46:4763-4779. 10.1109/TPAMI.2024.3357847
18. Gong W, Zhang Y, Wang W, Cheng P, González J: Meta-MMFNet: Meta-learning based multi-model fusion network for micro-expression recognition. ACM Transactions on Multimedia Computing, Communications and Applications. 2023, 20:1-20. 10.1145/3539576
19. Abayomi-Alli OO, Damaševičius R, Misra S, Maskeliūnas R: Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. Expert Systems. 2021, 38:e12746. 10.1111/exsy.12746
20. Işık G, Paçal İ: Few-shot classification of ultrasound breast cancer images using meta-learning algorithms. Neural Computing and Applications. 2024, 36:12047-12059. 10.1007/s00521-024-09767-y
21. Brazdil P, van Rijn JN, Soares C, Vanschoren J: Metalearning approaches for algorithm selection II. Metalearning. Springer, Cham; 2022. 77-102. 10.1007/978-3-030-67024-5_5
22. Dissanayake K, Md Johar MG: Diagnostics using meta-learning-based hybrid feature selection. Applied Computational Intelligence and Soft Computing. 2024, 2024:8800497. 10.1155/2024/8800497
23. Yao H, Wu X, Tao Z, Li Y, Ding B, Li R, Li Z: Automated relational meta-learning. arXiv:2001.00745. 10.48550/arXiv.2001.00745
24. Zhang D, Bashar MA, Nayak R: A novel multi-modal fusion method based on uncertainty-guided meta-learning. Pattern Recognition. 2025, 158:110993. 10.1016/j.patcog.2024.110993
25. Zhang Z, Yin G, Ma Z, Tan Y, Zhang B, Zhuang Y: IDA-NET: Individual difference aware medical image segmentation with meta-learning. Pattern Recognition Letters. 2025, 187:21-27. 10.1016/j.patrec.2024.11.012
26. Hidalgo ÁC, Ger PM, Valentín LDLF: Using meta-learning to predict student performance in virtual learning environments. Applied Intelligence. 2022, 52:3352-3365. 10.1007/s10489-021-02613-x
27. Tiwari S, Gogoi M, Verma S, Singh KP: Learning to learn with indispensable connections. arXiv:2304.02862. 10.48550/arXiv.2304.02862