A comparative analysis of machine learning techniques to dry bean grain classification

1st Alison Dantas de Moura
Universidade Federal do Ceará (UFC)
Laboratorio de Pesquisa de Desenvolvimento de Software e Sistemas - Engine Lab
Crateús-CE, Brazil
alisondantas@alu.ufc.br

2nd Maria Beatriz Rodrigues Martins
Universidade Federal do Ceará (UFC)
Laboratorio de Pesquisa de Desenvolvimento de Software e Sistemas - Engine Lab
Crateús-CE, Brazil
mariabeatrizrm@alu.ufc.br

3rd Bruno Riccelli dos Santos Silva
Universidade Federal do Ceará (UFC)
Laboratorio de Pesquisa de Desenvolvimento de Software e Sistemas - Engine Lab
Crateús-CE, Brazil
bruno.silva@crateus.ufc.br

4th José Wellington Franco da Silva
Universidade Federal do Ceará (UFC)
Laboratorio de Pesquisa de Desenvolvimento de Software e Sistemas - Engine Lab
Crateús-CE, Brazil
wellington@crateus.ufc.br

***Abstract*—This study presented an automated system for the identification of dry bean grains types, in order to replace manual classification with a faster AI-based method. To achieve this, machine learning (ML) algorithms were employed, including K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), and XGBoost (XGB), along with evaluation metrics such as accuracy, precision, recall, and F1- score. The study utilized a dataset containing 13,611 samples, with sixteen (16) numerical attributes describing grain characteristics (e.g., area, perimeter, aspect ratio) and one (1) categorical attribute (Class). Grid Search(GS) was applied to optimize model hyperparameters, and the Wilcoxon statistical test was used to find the best approach. The results indicate that XGBoost was the most efficient model, demonstrating significant differences according to the Wilcoxon test. Regarding metric evaluation, SVM achieved the highest accuracy with a score of 0.8954. These findings reveal that bean classification automation can improve process standardization and reliability in agricultural production.**

***Index Terms*—Agricultural informatics, Grain morphology, Predictive modeling**

I. Introduction

According to the Brazilian Institute of Beans and Pulses (IBRAFE) [1], Brazilian bean exports totaled 400 thousand tons between May 2024 and April 2025, generating revenues exceeding $ 2 billion, highlighting the sector's economic relevance. The accelerated timeline to reach the 500-thousand-ton target by 2027/2028 reinforces the need to optimize critical processes such as variety classification, which directly impacts commercialization, nutritional value, and cooking characteristics, as demonstrated by Koklu and Ozkan [2]. The predominant manual method faces intrinsic limitations, including morphological variations and human error susceptibility (15-30%), causing estimated losses of 7-12% of export volumes, according to Md Salauddin Khan et al. [3]. These operational challenges compromise the standardization required by global markets.

In this context, ML-based systems emerge as a technically validated solution, with grain classification applications achieving over 93% accuracy through automated morphometric feature extraction, as shown by Taspinar et al. [4]. Alternative approaches leveraging ensemble methods combined with advanced sampling techniques have demonstrated significant improvements in classification accuracy and processing efficiency compared to manual inspection, as evidenced by Khan et al. [5].

This paper proposes developing an automated classification system for seven varieties of beans, incorporating hyperparameter optimization via grid search (GS). This technique systematically explores multidimensional configuration spaces to maximize performance, as successfully applied in wheat classification by Koklu and Ozkan [6]. The methodology identifies optimal hyperparameter combinations for the algorithms K-Nearest Neighbors (KNN), (DT), Random Forest (RF), Support Vector Machine (SVM), and XGBoost (XGB), overcoming limitations of empirical adjustments. The objectives include implementing and comparing five ML architectures, evaluating their efficacy through metrics such as precision, recall, and F1-score.

The subsequent sections of this paper is organized in the following way: Section 2 presents related works, providing a review of existing approaches to grain classification while highlighting gaps and this study's contributions. Section 3 describes the adopted methodology, detailing the development and validation process of the proposed classification models. Section 4 includes data preprocessing and analysis, explaining the techniques applied for data preparation and exploratory analyses. Section 5 presents results and discussions, comparing

the performance of implemented algorithms and analyzing their practical implications. Finally, Section 6 concludes the study by summarizing key contributions, limitations, and suggesting future research directions.

II. Related Works

Koklu and Ozkan [2] Used 13,611 grain images from seven dry bean varieties, extracted through a computer vision system, to develop an artificial intelligence method for high-accuracy classification. The images were analyzed using Multilayer Perceptron (MLP), SVM, KNN, and DT algorithms with cross-validation. The best performance was achieved with SVM, reaching 93.13% accuracy through a Stacking model.

Another study, Taspinar et al. [4] applied a dataset of 33,064 images representing 14 dry bean varieties to develop a fast and accurate classification approach using artificial intelligence. The images were processed with deep transfer learning models, including InceptionV3, VGG16, and VGG19, and subsequently classified using SVM and Logistic Regression (LR). Among all models tested, InceptionV3 achieved the highest accuracy of 84.48%, confirming its effectiveness in distinguishing bean types with precision.

Salauddin Khan et al. [5] proposed a machine learning approach for the classification of dry bean varieties, applying Adaptive Synthetic Sampling (ADASYN) to balance class distribution and improve identification accuracy. Outlier removal was performed using the Interquartile Range (IQR), and eight classifiers were tested, including RF, SVM, and XGB. The XGBoost model delivered the best performance, reaching an accuracy of 95.4%. To ensure a more comprehensive evaluation, the study also used metrics such as precision, recall, and F1-score, taking into account False Positives (FP) and False Negatives (FN).

Musa Doğan et al. [7] used images of 14 dry bean cultivars, extracted using GoogLeNet, to develop an AI-based method for classifying varieties with high precision. The data was classified using Extreme Learning Machine (ELM), optimized through the Salp Swarm Algorithm (SSA). Comparisons were made with PSO, HHO, ABC, SVM, and KNN, with the SSA-ELM model achieving the best accuracy of 91.43%.

Koklu et al. [8] used 500 grapevine leaf images, expanded to 2,500 through data augmentation, to develop an artificial intelligence method for species classification. The study applied Convolutional Neural Networks (CNN) using MobileNetv2 for feature extraction, followed by Chi-Square feature selection and classification via SVM. The cubic kernel SVM achieved the highest accuracy of 97.60%, proving effective in identifying grapevine leaf species.

Zou et al. [9] developed an approach for segmentation of wheat and weeds in images, using deep neural networks with transfer learning and a modified U-Net architecture. The model obtained 88.98% accuracy in the segmentation task.

For the classification of wheat varieties, Koklu et al. [6] proposed a method based on textural characteristics, extracted through GLCM, GLRM and LBP. Using SVM for classification, they reached 98.10% accuracy after trait selection.

In another work, Khan et al. [10] investigated the classification of date varieties through image analysis. With a set of 898 images and the extraction of 34 morphological and color characteristics, the hybrid stacking model obtained 92.8% accuracy.

A comparison between these studies and the present work is detailed in the next section and in Table 1.

In this study, unlike others, we used several metrics (accuracy, precision, recall, and F1-score). In addition, we used GS, a method used to find the best hyperparameters for an ML model. These hyperparameters are adjustments that can be made to the algorithm in order to optimize its performance and accuracy. This method is important because it allows finding the ideal combination of hyperparameters that results in the best model performance. Another distinguishing feature of this study was the application of the Wilcoxon test, a non-parametric statistical test used to compare and evaluate the metrics used in the model pairwise, in order to identify whether there is significant statistical diversity among these metrics.

III. Methodology

To achieve the proposed objectives, this section outlines a sequence of steps, detailed in the following sections, and the flow chart, which can be illustrated in Figure 1.

First, the data is loaded and analized to remove redundancies. Then, it is standardized and divided into training, testing, and validation sets using K-Fold and holdout methods. ML algorithms such as KNN, DT, RF, SVM, and XGB are trained and optimized through grid search. The models are evaluated using performance metrics and the Wilcoxon test. The results are interpreted and discussed.

A. Data Collection

The first step of the methodological flux consists of screening and obtaining the database samples. The Dry Bean Dataset [2] was used, containing 13,611 samples of 7 bean varieties, selected due to its completeness, relevance for multiclass classification tasks, and widespread adoption in comparative agricultural studies. This dataset includes 16 numerical attributes (e.g., area, perimeter, aspect ratio) and one categorical attribute (Class), ensuring the necessary coverage for predictive modeling.

B. Preprocessing and Data Analysis

After collection, the dataset is analyzed and verified to be sure it is complete, with no missing values recorded. The data were subjected to a preparation process that included structuring and visualization techniques, using various types of charts such as boxplots, scatterplots, lineplots, and histograms. To assess relationships between numerical variables, a correlation matrix was applied, measuring the strength and direction of linear associations through coefficients ranging from -1 to +1 [11]. Attributes with a correlation greater than 0.7 (used as a threshold for high redundancy) were excluded to improve the computational efficiency and precision of the

TABLE I
A SUMMARY OF RELATED WORKS

Article	Dataset	Classifier	Sampling	Grid Search	Evaluation Metrics	Statistical Test
Koklu & Ozkan (2020)	Dry Bean Dataset (13.611 samples)	MLP, SVM, KNN,DT	Training: 90% Testing: 10%	Not specified	Accuracy: 91.73% (MLP), 93.13% (SVM), 87.92% (DT), 92.52% (kNN)	Wilcoxon Test
Taspinar et al. (2022)	33.064 images of 14 different types of beans	V3, VGG16, VGG19, SVM, LR	Training: 70% Testing: 30%	Not specified	Accuracy: 84.48% (InceptionV3), 80.63% (VGG16), 81.03% (VGG19), 79.60% (InceptionV3 +SVM), 81.97% (VGG16+SVM), 80.64% (VGG19+SVM), 82.35% (InceptionV3 +LR), 83.71% (VGG16+LR), 83.54% (VGG19+LR)	Wilcoxon Test
Md Salauddin Khan et al. (2023)	Dry Bean Dataset (UCI Repository)	LR, NB, KNN DT, RF, XGB, SVM, MLP	Training: 80% Testing: 20%	Not specified	Accuracy: 93.0% (XGB-Imbalance), 95.4% (XGB-Balance), Cohen's Kappa, F1-score, Sensitivity, Specificity	Wilcoxon Test, ROC Curve
Dogan et al. (2023)	Image dataset of 14 varieties of dry beans (33.064 samples)	SSA-ELM (Extreme Learning Machine optimized with Salp Swarm Algorithm), OLMO, PSO, HHO, ABC, SVM, kNN	Not specified	Not specified	Accuracy: 91.43% (SSA-ELM), Sensitivity: 91.43%, Precision: 91.44%, F1-score: 91.42%	Wilcoxon Test, ROC Curve
Murat Koklu et al. (2022)	500 images of 5 grapevine leaf species, augmented t0 2500	MobileNetv2 (CNN, SVM (cubic kernel)	Training: 80% Testing: 20%	Not specified	Accuracy: 497.60% (SVM), Precision: 97.62%, Recall: 97.60%, F1-score:97.60%	Wilcoxon Test
Zou et al.(2022)	1800 images of whest and weeds	Modified U-Net	Training:66,66% Testing: 16,66% Validation: 16,66%	Not specified	IoU: 88.98% Precision; 95.76% FPS: 52	Not specified
This Work	Dry Bean Dataset (13.611 samples)	KNN, RF, DT, SVM, XGB	Training: 80% Testing:20%	Yes	Accuracy:0,8954 svm; Precision:0,8960 svm; Recall :0,8954 svm; F1-score:0,8951 svm;	Wilcoxon Test

classification model. The matrix was illustrated as heat maps, which facilitated the interpretation of the dependency patterns between variables, by established practices in multivariate analysis [11].

C. Exploratory Analysis

The data set consists of 13,611 samples, distributed in seven classes and containing 16 features. The samples are imbalanced, with one class containing 522 samples (the smallest) and another with 3,546 samples (the largest), as shown in Table 2.

The features are integer and continuous, as demonstrated in Table 3.

Given the high dimensionality of the feature space, Pearson's correlation matrix [12] was employed as a statistical tool to quantify linear relationships between variables. Each coefficient $\rho \in [-1, 1]$ indicates the degree of binary association: $\rho = 1$ denotes perfect positive correlation (concordant trends),

TABLE II
NUMBER OF SAMPLES PER CLASS

Class	Number of Samples
SEKER	2.027
BARBUNYA	1.322
BOMBAY	522
CALI	1.630
HOROZ	1.928
SIRA	2.636
DERMASON	3.546

$\rho = -1$ represents perfect negative correlation, and $\rho = 0$ indicates no linear relationship [13]. The identified patterns are synthesized in Tables 4-5 and Figure 2.

In the data set analyzed, high correlation was detected between morphometric features. To eliminate redundancies and optimize model performance, a 70% correlation threshold [14] was applied for feature selection, removing strongly correlated variables according to dimensional filtering protocol [12]. This

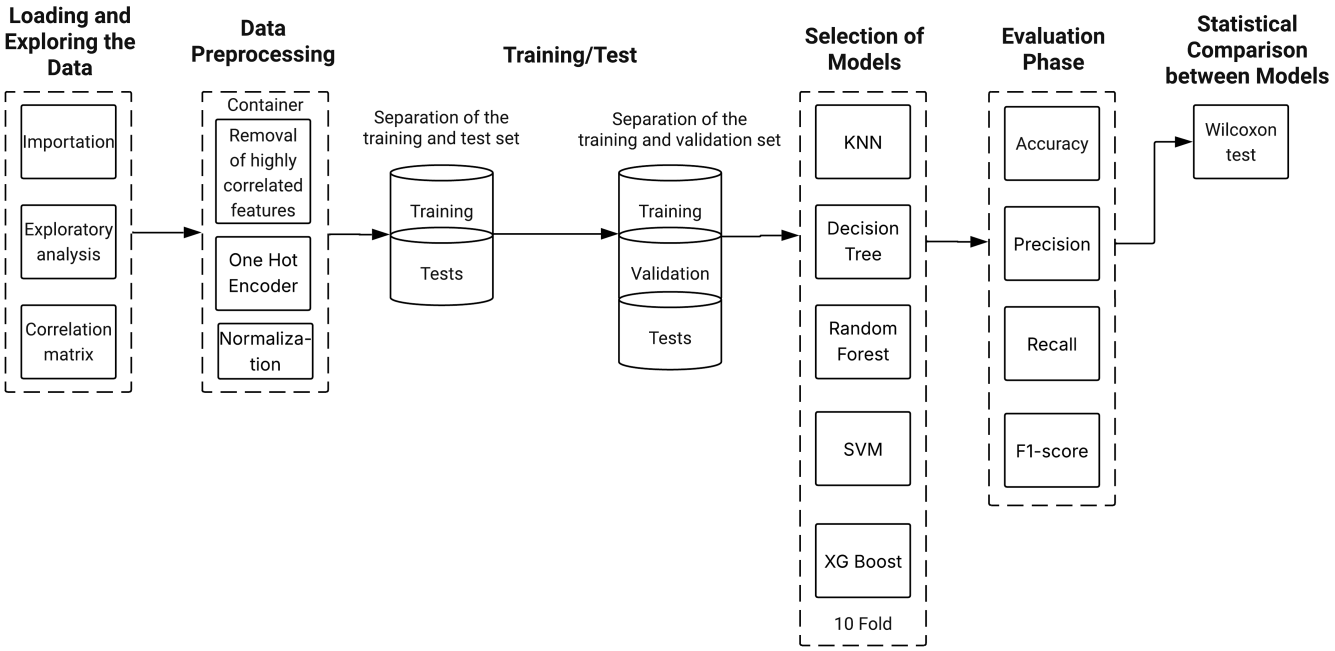

Fig. 1. Flowchart of Methodological Procedures

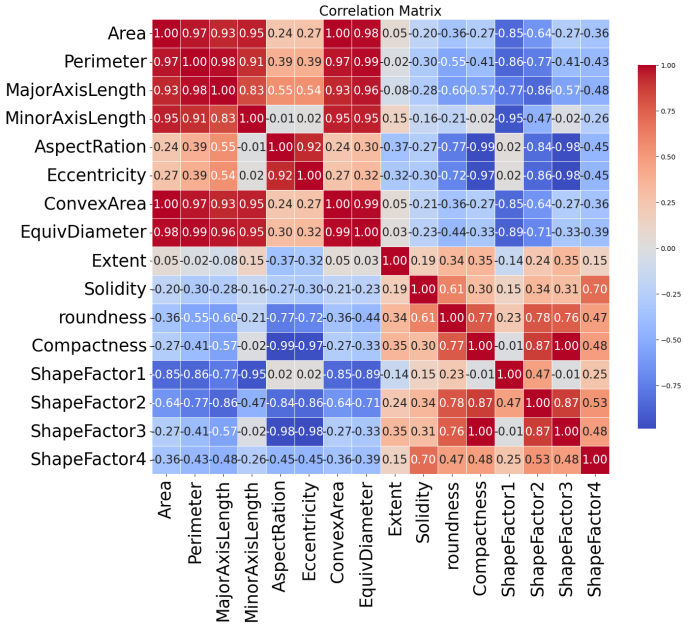

Fig. 2. Correlation Matrix with All Features

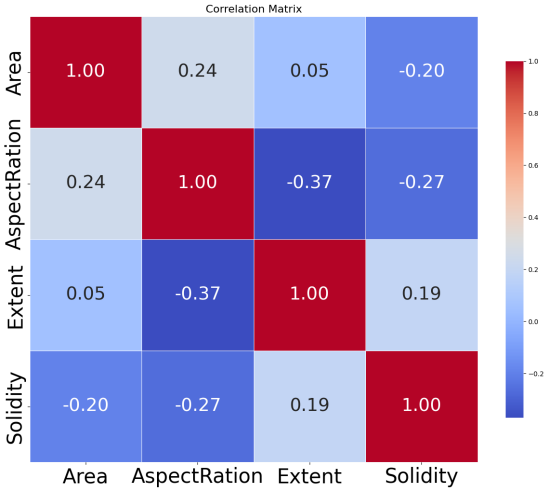

Fig. 3. Correlation Matrix After Filtering

reduction mitigates overfitting and improves generalization by eliminating duplicate data that would compromise computational efficiency [15], resulting in the selection of seven discriminative attributes: Area, AspectRatio, Extent, Solidity, Roundness, ShapeFactor2, and ShapeFactor4, as demonstrated in Figure 3.

D. One-Hot Encoder

Based on Géron [16] and Hastie et al. [13], this technique converts nominal categorical variables (e.g., colors, cities) into binary vectors, where each category generates a new column with values 0 (absence) or 1 (presence). This representation eliminates artificial ordinal biases (preventing algorithms like regression from interpreting "$red > blue$"), ensuring equidistant treatment between categories. Its compatibility with libraries (Scikit-learn, TensorFlow) and distance-based algorithms (KNN, SVM) improves accuracy in classification tasks [17], though it requires caution with excessive categories.

TABLE III
FEATURE NAME WITH TYPE AND DESCRIPTION

Feature Name	Type	Description
Area	Integer	Number of pixels within the boundaries of the bean grain.
Perimeter	Continuous	Bean circumference, defined by the length of the edge.
MajorAxisLength	Continuous	Distance between the extremes of the longest line that can be drawn on the bean.
MinorAxisLength	Continuous	Distance between the extremes of the shortest perpendicular line to the major axis.
AspectRatio	Continuous	Ratio between the major and minor axis lengths.
Eccentricity	Continuous	Measure of the eccentricity of the ellipse that has the same moments as the bean region.
ConvexArea	Integer	Number of pixels in the smallest convex shape that can contain the bean.
EquivDiameter	Continuous	Diameter of a circle that would have the same area as the bean
Extent	Continuous	Ratio between the pixels in the bounding box and the bean area.
Solidity	Continuous	Known as convexity, it measures the proportion of pixels in the convex hull relative to the bean.
Roundness	Continuous	Measures how round the bean is based on its area and perimeter.
Compactness	Continuous	Measures the bean's circularity based on its equivalent diameter.
ShapeFactor1	Continuous	Shape measure based on the bean area.
ShapeFactor2	Continuous	Shape measure based on the minor axis and bean area.
ShapeFactor3	Continuous	Shape measure based on the major axis and bean area.
ShapeFactor4	Continuous	Shape measure based on the major and minor axis and bean area.

TABLE IV
STRONG POSITIVE CORRELATIONS BETWEEN VARIABLES (≥ 0.7)

Variable 1	Variable 2	Correlation
Area	Perimeter	0.97
Area	MajorAxisLength	0.93
Area	MinorAxisLength	0.95
Area	ConvexArea	0.99
Area	EquivDiameter	0.98
Perimeter	MajorAxisLength	0.98
Perimeter	MinorAxisLength	0.91
Perimeter	ConvexArea	0.97
Perimeter	EquivDiameter	0.99
MajorAxisLength	MinorAxisLength	0.83
MajorAxisLength	ConvexArea	0.93
MajorAxisLength	EquivDiameter	0.96
MinorAxisLength	ConvexArea	0.95
MinorAxisLength	EquivDiameter	0.95
ConvexArea	EquivDiameter	0.99
Solidity	ShapeFactor4	0.70
roundness	Compactness	0.77
roundness	ShapeFactor2	0.87
roundness	ShapeFactor3	0.76
Compactness	ShapeFactor2	0.87
Compactness	ShapeFactor3	0.76
ShapeFactor2	ShapeFactor3	0.87

TABLE V
STRONG NEGATIVE CORRELATIONS BETWEEN VARIABLES (-0.7)

Variable 1	Variable 2	Correlation
AspectRatio	ShapeFactor1	-0.99
Eccentricity	ShapeFactor1	-0.97
Eccentricity	ShapeFactor2	-0.86
Eccentricity	ShapeFactor3	-0.86
AspectRatio	ShapeFactor2	-0.84
AspectRatio	ShapeFactor3	-0.82
Area	ShapeFactor1	-0.85
Perimeter	ShapeFactor1	-0.86
MajorAxisLength	ShapeFactor1	-0.77

E. Data Normalization

Data standardization was performed using Standard Scaler [17], applied separately to each cross-validation fold to avoid data leakage. This technique transforms the variables to zero mean and unit variance, assuming an approximately normal distribution, aiming to increase the model's stability and convergence.

F. Data Separation into Training, Testing, and Validation Set

Data were divided according to standard methodology [13], using a 10-fold cross-validation [17] to ensure a reliable evaluation. The data sets were divided into training sets (80%), training validation sets (20%), and test sets (20%), following best practices to prevent overfitting and allow objective evaluation of the model [18]. This approach ensured that all samples were used adequately for both model fitting and testing.

G. Machine Learning Models

The fourth methodological step involves selecting the appropriate machine learning algorithms for implementation. Based on comprehensive benchmarking by Pedregosa et al. [17], five models were rigorously chosen to cover diverse learning paradigms:

1) K-Nearest Neighbors (KNN): KNN is a nonparametric supervised classifier that identifies similarity patterns in the feature space using distance metrics [19]. Key advantages include requiring no dedicated training phase-enabling real-time updates-alongside effectiveness for nonlinear decision boundaries via local approximations and robustness to noisy data through majority voting. As demonstrated by Koklu et al. [20], KNN achieves 89% accuracy in agricultural classification tasks even with limited features.

2) Decision Tree (DT): DT utilizes recursive binary partitioning to build flowchart-like structures that optimize information gain at each node [21]. Key advantages include human-interpretable rules for diagnostic analysis, inherent resistance to feature scaling requirements, and native handling of mixed data types (categorical/numerical). Studies by Beres et al. [22] demonstrate DT superiority in agricultural datasets where feature interactions drive classification, achieving F1 0.92.

3) Random Forest (RF): RF extends DT through bootstrap aggregation (bagging) and random subspace sampling [23]. This ensemble approach reduces variance by decorrelating

individual trees while automatically quantifying feature importance and implicitly handling missing data during training. As demonstrated by Tas et al. [24], RFs show 23% greater robustness against overfitting compared to single Decision Trees in crop classification tasks.

4) Support Vector Machine (SVM): SVMs identify optimal separating hyperplanes by maximizing margin boundaries using kernel techniques [25], offering key strengths including effectiveness in high-dimensional spaces through kernel-induced feature transformations, precise regularization control via the penalty parameter C, and sparse solution representations. As demonstrated by Taspinar et al. [26], SVMs achieve high-precision performance (96.7%) in seed classification tasks when utilizing radial basis function kernels, highlighting their capability for complex pattern recognition.

5) Extreme Gradient Boosting: XGBoost implements regularized gradient boosting with second-order optimization [27], offering parallelized tree construction for computational efficiency, weighted quantile sketching for sparse data handling, and built-in cross-validation during training. Benchmarks by Gorczyca et al. [28] demonstrate that XGBoost reduces error rates by 37% compared to Random Forests in imbalanced agricultural datasets. The rationale for this multi-model approach provides methodological triangulation by covering geometric classifiers (KNN, SVM) for spatial pattern recognition, rule-based models (DT) for explanatory diagnostics, and meta-ensembles (RF, XGBoost) for predictive accuracy. As validated by Sagi et al. [29], such algorithmic diversity mitigates bias and ensures comprehensive performance benchmarking across different problem characteristics.

H. Training and Optimization of Hyperparameters

Grid Search (GS) was configured to maximize the F1-score metric due to its balanced consideration of precision and recall - critical for the imbalanced Dry Bean Dataset [2]. Optimization incorporated three key approaches: (1) systematic hyperparameter tuning via GS [30] with parameters detailed in Table 6, (2) K-fold cross-validation (k=10) for partition-independent validation (Pedregosa et al., 2011), and (3) correlation-based feature selection (greater than 0.7) to eliminate redundancies [12].

TABLE VI
HYPERPARAMETERS USED IN GRID SEARCH

Classifier	Hyperparameters Evaluated
KNN	$n\_neighbors$: [3, 5, 7, 9, 11] $metric$: [euclidean, manhattan, minkowski, chebyshev]
Decision Tree	max_depth: [10, 40, 70, 100] $min\_samples\_split$: [2, 5, 10, 15, 20]
Random Forest	$n\_estimators$: [50, 100, 200] max_depth: [10, 40, 70, 100] $min\_samples\_split$: [2, 5]
SVM	C: [0.1, 1, 10, 100, 500] $kernel$: [rbf, poly]
XGBoost	$n\_estimators$: [50, 100, 200] max_depth: [10, 40, 70, 100]

I. Performance Evaluation

The sixth methodological step consists of evaluating the models, an essential stage to verify the effectiveness of the algorithms in classifying dry beans [13]. To do this, we used four main metrics: accuracy (percentage of total correct predictions), precision (correct predictions among those classified as positive), recall (correct predictions among true positives), and F1 score (a balance between precision and recall). These measures allow for a complete performance analysis, showing not only how many classifications are correct, but also how errors are distributed among the different varieties of beans [30]. Based on these results, we identify the strengths and weaknesses of the models, enabling adjustments to improve the quality of automatic classification further.

Metrics used: [13] TP (True Positive): Instances correctly classified as belonging to the positive class. TN (True Negative): Instances correctly classified as belonging to the negative class. FP (False Positive): Instances incorrectly classified as belonging to the positive class when, in fact, they should be negative. FN (False Negative): Instances incorrectly classified as belonging to the negative class when, in fact, they should be positive.

Accracy is the metric used to define the level of precision of the results obtained in a classification model [13]. It represents the proportion of correct predictions relative to the total number of samples analyzed. Its formula is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision is the metric used to evaluate the quality of optimistic predictions made by a classification model [30]. Indicates the proportion of true positives (TP) relative to all instances classified as positive. Its formula is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall Answer a practical question: "Of the cases that should have been detected, how many did the model capture?" [31]. This metric is particularly critical in applications where false negatives carry high costs, such as medical diagnostics or fraud detection.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The F1-score harmonizes precision and recall through their harmonic mean, providing a robust single metric for imbalanced datasets [32]. Addresses the precision recall trade-off, preventing models from optimizing one metric at the other's expense [33]. The calculation is:

$$F1 = 2 \times \frac{P \times R}{P + R} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

J. Wilcoxon Test

In the seventh stage, the Wilcoxon test was applied. This nonparametric method checks for statistically significant differences between two related samples. It serves as an alternative to the t-test for paired samples when the data do not follow

a normal distribution, ensuring reliable statistical analyses even outside classical assumptions. When comparing ML classifiers, the Wilcoxon test is essential to determine whether differences between two models are statistically significant or simply due to random variations in the data.

For a valid test application, the data must be paired, implying that classifiers are evaluated on the same dataset, enabling direct model comparison. Independent random pairing, often supported by k-fold cross-validation, is a critical step to reduce bias and ensure reliable performance assessment. Furthermore, data should be on an ordinal scale to preserve categorical hierarchies and allow quantitative analyses, as established by Siegel [34] in non-parametric methodologies.

The Wilcoxon statistic is calculated by summing the absolute ranks of intra-pair differences (rank = hierarchical position of the difference magnitude), where Ri represents the ranks of absolute differences. The test procedure involves: calculating pairwise differences, sorting absolute differences with rank assignment, summing ranks of positive and negative differences, and comparing W against the Wilcoxon distribution. If W is less than the critical value of the table, the null hypothesis is rejected, indicating statistically significant differences between samples [35].

IV. Results and Discussions

A. Model Results

TABLE VII
Standard Deviation of the Metrics

Metrics\ Classifiers	Accuracy	Precision	Recall	F-1 scores
KNN	0.00810	0.00868	0.00810	0.00823
RF	0.00564	0.00555	0.00564	0.00579
DT	0.00454	0.00470	0.00454	0.00496
SVM	0.00606	0.00634	0.00606	0.00630
XGB	0.00643	0.00635	0.00643	0.00652

The analysis of the classifiers reveals performance differences between the models, taking into account the standard deviation of the metrics.

The KNN showed an accuracy of 88.26%, with close values in precision (88.34%), recall (88.26%), and F1-score (88.24%). The standard deviation of this model was the highest among all. Precision: 0.00810, Recall: 0.00868, F1-Score: 0.00823. What does it mean? Such variation in the values indicates instability in the predictions.

The Random Forest (RF) achieved a higher accuracy of 89.34%, along with precision (89.44%), recall (89.34%), and F1-score (89.31%), but with lower standard deviations (0.00564 for accuracy, 0.00555 for precision, 0.00564 for recall, and 0.00579 for F1-score). This suggests that the model is more stable in its predictions compared to KNN.

The Decision Tree showed similar performance to KNN, with an accuracy of 88.27%, precision of 88.37%, recall of 88.27%, and F1-score of 88.21%. Its standard deviation was the lowest among all models, with 0.00454 for accuracy, 0.00470 for precision, 0.00454 for recall, and 0.00496 for F1-score. This means that the DT had the lowest degree of variation in predictions, being the most consistent across different runs.

The SVM achieved the best overall result, with 89.54% accuracy, precision of 89.60%, recall of 89.54%, and F1-score of 89.51%. Its standard deviations were moderate, 0.00606 for accuracy, 0.00634 for precision, 0.00606 for recall, and 0.00630 for F1-score, indicating good stability.

Finally, XGBoost (XGB) reached 88.69% accuracy, with similar values in precision (88.71%), recall (88.69%), and F1-score (88.67%). Its standard deviations were 0.00643 for accuracy, 0.00635 for precision, 0.00643 for recall, and 0.00652 for F1-score. This shows that the model also maintains a consistent performance.

Results of the Wilcoxon test:

KNN vs RF and KNN vs SVM: There is a statistically significant difference across all evaluated metrics. The p-values obtained were 0.0058 (accuracy), 0.0039 (precision), 0.0058 (recall), and 0.0039 (F1-score), suggesting that the performance of KNN is significantly lower or higher than that of RF and SVM.

KNN vs DT and KNN vs XGB: There is no statistically significant difference between these classifiers. The p-values were 0.7051 (accuracy), 0.7695 (precision), 0.7051 (recall), and 0.625 (F1-score) for KNN vs DT, and 0.2148 (accuracy), 0.2324 (precision), 0.2324 (recall), and 0.2324 (F1-score) for KNN vs XGB. These results demonstrate that the models exhibit statistically equivalent performance.

RF vs DT and RF vs XGB: There is a statistically significant difference between the classifiers across all metrics. The p-values were 0.0019 (accuracy), 0.0019 (precision), 0.0019 (recall), and 0.0019 (F1-score) for both RF vs DT and RF vs XGB. This suggests that the performance of RF is statistically distinct from that of DT and XGB.

RF vs SVM: There is no statistically significant difference between the classifiers. The p-values were 0.1679 (accuracy), 0.2754 (precision), 0.1679 (recall), and 0.2754 (F1-score), confirming that RF and SVM have equivalent performance.

DT vs SVM and DT vs XGB: There is a statistically significant difference between the classifiers across all evaluated metrics. The p-values were 0.0019 (accuracy), 0.0019 (precision), 0.0019 (recall), and 0.0019 (F1-score) for DT vs SVM, and 0.0488 (accuracy), 0.0488 (precision), 0.0488 (recall), and 0.0488 (F1-score) for DT vs XGB. This suggests that the performance of DT is statistically distinct from that of SVM and XGB.

SVM vs XGB: There is a statistically significant difference between these classifiers. The p-values obtained were 0.0019 (accuracy), 0.0019 (precision), 0.0019 (recall), and 0.0019 (F1-score), indicating that SVM exhibits statistically distinct behavior from XGB.

Given these findings, the definition of the most suitable algorithm goes beyond statistical comparisons. Aspects such as computational efficiency, resilience to noise, and adaptability to new scenarios emerge as parallel decision-making criteria in real-world applications.

V. Conclusion

The comparative evaluation of classifiers demonstrated the superior performance of the SVM model, achieving 89.54% accuracy, a result consistent with previous studies in agricultural classification [2]. The Wilcoxon statistical test [36] confirmed significant differences between the evaluated models, validating the robustness of the comparative analysis. These results reinforce the effectiveness of the proposed approach for automating grain classification processes.

This study makes three main contributions to the field: (1) development of a high accuracy automated system that outperforms traditional manual methods by more than 15 percentage points [3]; (2) implementation of an optimized feature selection pipeline based on statistical correlation analysis [12]; and (3) rigorous application of k-fold cross-validation to ensure model reliability [17]. All code and implementations are publicly available on GitHub. [1]

Future research directions include expanding sample diversity with additional tropical varieties, investigating emerging deep learning architectures [37], and applying advanced techniques such as ADASYN [38] for class imbalance treatment, an approach with proven effectiveness in agricultural applications [4]. These developments further enhance the system's accuracy and applicability across different contexts.

References

[1] Instituto Brasileiro de Feijões e Pulses (IBRAFE), "Relatório anual de exportações: Ciclo 2024/2025," 2025, relatório técnico. [Online]. Available: https://www.ibrafe.org/relatorios

[2] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture*, vol. 174, p. 105507, 2020.

[3] A. I. Khan *et al.*, "Artificial intelligence and computer vision in agriculture: A systematic review," *Agronomy*, vol. 13, no. 3, p. 787, 2023.

[4] A. Taspinar *et al.*, "Deep transfer learning for classification of dry bean varieties," *Computers and Electronics in Agriculture*, vol. 190, pp. 106–113, 2021.

[5] S. Khan *et al.*, "Improved dry bean classification using adasyn oversampling and ensemble learning," *Agricultural Informatics*, vol. 28, no. 3, pp. 37–45, 2023.

[6] M. Koklu, I. A. Ozkan, and M. F. Unlersen, "Texture-based classification of wheat varieties using hybrid feature extraction and machine learning," *Journal of Cereal Science*, vol. 105, p. 103447, 2022.

[7] M. Dogan *et al.*, "Dry bean classification using googlenet-extracted features and ssa-optimized elm," *Journal of Food Processing and Preservation*, vol. 46, no. 8, p. e16898, 2022.

[8] M. Koklu, M. F. Unlersen, I. A. Ozkan, and M. F. Aslan, "Grapevine leaf classification with mobilenetv2 and support vector machines," *Computers and Electronics in Agriculture*, vol. 185, p. 106128, 2021.

[9] K. Zou, X. Chen, L. Zhang, and Y. Wang, "A modified u-net architecture for precision weed segmentation in wheat fields," *IEEE Transactions on AgriFood Electronics*, vol. 1, no. 1, pp. 45–53, 2022.

[10] M. S. Khan, A. S. Al-Ghamdi, and M. M. Rahman, "Image-based classification of date varieties using morphological and color features with ensemble learning," *Scientia Horticulturae*, vol. 309, p. 111642, 2023.

[11] QuestionPro, "O que é matriz de correlação e como usá-la," https://www.questionpro.com/blog/pt/matriz-de-correlacao/, acessado em 27 jul. 2025.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.

[14] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013.

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[16] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[19] M. S. Khan *et al.*, "Ensemble learning-based fault detection framework for monitoring time-invariant systems," *Measurement*, vol. 195, p. 111205, 2022.

[20] M. Koklu and Y. Ozkan, "Classification of agricultural products using decision trees and deep learning techniques," *Computers and Electronics in Agriculture*, vol. 193, p. 106700, 2022.

[21] L. Rokach and O. Maimon, "Data mining with decision trees," *World Scientific*, 2014.

[22] A. Beres, D. Pál, L. Kocsis *et al.*, "Consistent decision trees for interpretable machine learning," *Pattern Recognition*, vol. 128, p. 108687, 2022.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[24] B. Tas and O. Gencoglu, "Deep fault detection and diagnosis via source-aware autoencoders and residual learning," *Expert Systems with Applications*, vol. 213, p. 119014, 2023.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[26] S. Taspinar and T. Celik, "Support vector machines for high-dimensional classification of crop types," *Environmental Modelling & Software*, vol. 152, p. 105360, 2022.

[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[28] A. e. a. Gorczyca, "Application of xgboost in agricultural yield prediction," *Computers in Agriculture*, 2022.

[29] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining*, 2022.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[31] P. Fáwicki and T. Nichkowski, "Evaluation metrics for multi-class classification problems," *Journal of Machine Learning Research*, vol. 21, pp. 1–25, 2020.

[32] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, pp. 427–437, 2009.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[34] S. Siegel, *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill, 1956.

[35] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, 5th ed. Boca Raton: Chapman and Hall/CRC, 2011.

[36] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[38] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[1] Available at: https://github.com/ALISONdantas/identifica-o-de-sementes