JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.6816

OPTIMIZING DECISION TREE PERFORMANCE WITH RECURSIVE FEATURE ELIMINATION FOR HIGH-DIMENSIONAL MUSHROOM CLASSIFICATION

Lili Tanti[1*]; Safrizal[2]; Yan Yang Thanri[3]

Computer Sience Program[1], Information System Program[3]
Universitas Potensi Utama, Medan, Indonesia[1,3]
www.potensi-utama.ac.id[1,3]
lilitanti82@gmail.com[1], ythanri@gmail.com[2]

Information System Program[2]
Universitas Muhammadyah Asahan, Kisaran, Indonesia[2]
www.ummas.ac.id[2]
rizalsyl75@gmail.com[2*]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Classifying mushroom species presents a significant challenge within biological data analysis because of the wide variety of species and their distinct attributes. This research investigates the effectiveness of the Decision Tree classifier for mushroom categorization by comparing two splitting criteria, the Gini Index and Entropy. Additionally, the study employs the Recursive Feature Elimination (RFE) method for dimensionality reduction to enhance model efficiency and performance. The dataset was collected, cleaned, and analyzed exploratorily before feature selection was conducted using RFE. The Decision Tree model was trained and evaluated using accuracy, precision, recall, and F1-score metrics. The results showed that applying RFE improved computational efficiency without compromising model accuracy. The Gini criterion provided more stable results across all metrics, while Entropy demonstrated higher precision in certain cases. Model optimization through parameter tuning produced the best parameter combination at max_depth = 5, min_samples_leaf = 5, and min_samples_split = 10. This study concludes that integrating RFE with the Decision Tree can significantly enhance the performance of high-dimensional dataset classification. The findings are expected to serve as a reference for developing efficient and accurate biological data classification models.*

Keywords*: Decision Tree, Hyperparameter Tuning, Mushroom Classification, Recursive Feature Elimination (RFE).*

Intisari— *Klasifikasi jamur merupakan salah satu tantangan penting dalam analisis data biologis, karena beragamnya jenis dan karakteristik unik yang dimiliki. Penelitian ini bertujuan untuk mengevaluasi kinerja algoritma Decision Tree dalam klasifikasi jamur dengan menggunakan dua kriteria pembelahan, yaitu Gini Index dan Entropy, serta menerapkan teknik reduksi dimensi Recursive Feature Elimination (RFE) untuk meningkatkan efisiensi model. Dataset dikumpulkan, dibersihkan, dan dianalisis secara eksploratif sebelum dilakukan seleksi fitur menggunakan RFE. Model Decision Tree dilatih dan dievaluasi dengan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa penerapan RFE mampu meningkatkan efisiensi komputasi tanpa mengorbankan akurasi model. Kriteria Gini memberikan hasil yang lebih stabil pada seluruh metrik, sementara Entropy menunjukkan presisi yang lebih tinggi dalam beberapa kasus. Optimisasi model melalui tuning parameter menghasilkan kombinasi parameter terbaik pada max_depth = 5, min_samples_leaf = 5, dan min_samples_split = 10. Penelitian ini menyimpulkan bahwa integrasi RFE dengan Decision Tree dapat meningkatkan kinerja klasifikasi dataset berdimensi tinggi secara signifikan. Hasil* *penelitian ini diharapkan dapat menjadi referensi bagi pengembangan model klasifikasi data biologis yang efisien dan akurat.*

***Kata Kunci**: Decision Tree, Hyperparameter Tuning, Klasifikasi Jamur, Recursive Feature Elimination (RFE).*

INTRODUCTION

The classification of mushrooms poses a complex challenge in machine learning, particularly when dealing with high-dimensional datasets [1]. Mushrooms are characterized by diverse morphological and biochemical features, making accurate classification essential for health, food safety, biotechnology, and environmental applications [2]. Misclassification can result in the consumption of toxic species or hinder scientific research. Decision Tree algorithms are commonly used for classification due to their interpretability and ability to handle categorical data efficiently [3]. These models construct hierarchical decision rules based on measures such as the Gini Index and Entropy, which help in determining the best attribute splits based on class impurity and information gain [4], [5]. However, when applied to datasets with a large number of features, Decision Trees can suffer from overfitting, reduced generalizability, and decreased model interpretability.

High-dimensional data increases computational complexity and may introduce redundant or irrelevant features that compromise model accuracy. To overcome this, Recursive Feature Elimination (RFE) offers a powerful technique for dimensionality reduction by recursively removing the least important features based on model performance [6], [7]. Despite its potential, integration of RFE with Decision Tree models in the context of mushroom classification remains underexplored.

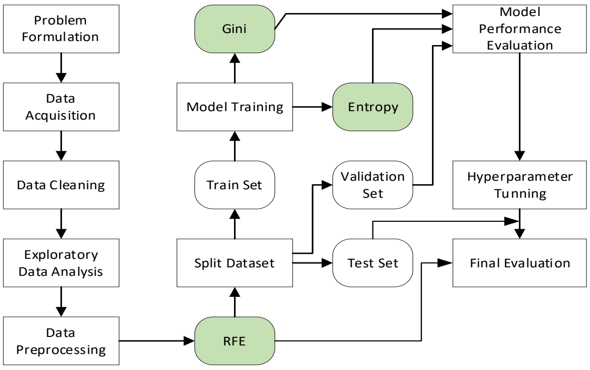
Several previous studies have applied machine learning algorithms for mushroom classification, but few have integrated RFE with Decision Tree or evaluated performance based on Gini Index and Entropy. Tutuncu et al. (2022) reported the success of Decision Tree with an accuracy of 96.82% for mushroom classification without employing RFE or specific criteria like Gini and Entropy [8]. Another study by Metlek and Cetiner (2023) reported precision and recall of 0.98 for Decision Tree, but it also did not incorporate RFE or Gini- and Entropy-based analyses [2]. A novel approach to Decision Tree was conducted by Lee et al. (2022)[9], who used the Rao-Stirling index to account for inter-class distances to improve accuracy; however, this study also did not integrate RFE or Gini- and Entropy-based analyses[10].

Meanwhile, Rianasari *et al.* (2022) applied PCA for mushroom classification using Naïve Bayes, achieving good accuracy levels, but they did not combine RFE with Decision Tree [11]. Other studies, such as Wati et al. (2022), compared the C4.5 and C5.0 algorithms, reporting high accuracy for identifying poisonous mushrooms, but without addressing RFE or Gini and Entropy criteria [12]. Siddique et al. (2023) highlighted the use of genetic algorithms for feature selection and random forest algorithms for classification, reporting that genetic algorithms outperformed correlation-based approaches in terms of accuracy and precision [13]. However, this research did not consider Decision Tree with RFE using Gini and Entropy. Additionally, Morshed et al. (2023) focused on evaluating nine machine learning methods for mushroom classification, highlighting k-NN as the best-performing method with 96% accuracy and the highest F1-score [14]. However, this study did not include an analysis of Decision Tree performance with RFE or criteria such as Gini and Entropy.

These gaps highlight the need for a comprehensive study integrating Recursive Feature Elimination with Decision Tree models and assessing the effect of Gini Index and Entropy on classification performance. This research aims to address that gap by developing an optimized Decision Tree model for high-dimensional mushroom classification using RFE and hyperparameter tuning. Performance evaluation will focus on the impact of Gini and Entropy as splitting criteria. This study not only contributes to improving classification performance in practical domains such as agriculture and health but also advances theoretical insights on applying dimensionality reduction to interpretable models. The proposed approach leverages RFE for feature selection, employs Decision Trees for classification, and systematically compares splitting criteria, thus offering a robust framework for high-dimensional biological data classification and contributing to the development of explainable AI-based systems.

MATERIALS AND METHODS

In this study, we evaluate the performance of the Decision Tree algorithm for mushroom classification by applying dimensionality reduction techniques using Recursive Feature Elimination (RFE). The research also compares the effectiveness of two splitting criteria, Gini Index and Entropy, in producing an optimal classification model. To achieve these objectives, a systematic and structured research methodology was implemented, as outlined in the flow diagram shown in Figure 1.



Source : (Research Results, 2025)

Figure 1. Research Methodology for Interpretable Mushroom Classification using RFE and Decision Tree

The study begins with problem formulation, emphasizing the importance of mushroom classification to support food safety and biotechnological research. This is followed by data acquisition and data cleaning to ensure dataset quality before modeling. An Exploratory Data Analysis (EDA) is then performed to understand feature distributions and relationships, as well as to identify potential irrelevant or redundant features.

To address the key issue highlighted in the background high dimensionality of features this study applies Recursive Feature Elimination (RFE) for feature selection. This process aims to reduce model complexity while maintaining performance and interpretability. The reduced dataset is then split into training, validation, and test sets to ensure balanced model development and generalization. The classification model is built using the Decision Tree algorithm, selected for its ability to handle categorical data and for its interpretability an essential element of explainable AI. Two splitting criteria, Gini Index and Entropy, are used separately during training to observe their impact on classifying biological data. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, providing a comprehensive understanding of its classification effectiveness. To further optimize the model, hyperparameter tuning is conducted using Grid Search with Cross-Validation. The final stage involves evaluating the model on the test set to confirm its reliability and performance stability.

Through this approach, the research methodology not only aligns with the challenges of high-dimensional mushroom classification but also directly implements solutions outlined in the background. The integration of RFE, Decision Tree, and a comparative analysis of Gini and Entropy criteria provides a meaningful contribution to the development of accurate and interpretable biological classification systems based on Explainable AI.

Research Dataset

This dataset originates from the UCI Machine Learning Repository and describes the physical and environmental characteristics of mushrooms [8], as shown in Table 1.

Table 1. Attributes Description

Attribute	Description
class	Edible (e) or Poisonous (p) mushrooms
cap-shape	Shape of mushroom cap (e.g., bell, conical, convex, flat)
cap-surface	Surface texture of cap (fibrous, grooves, scaly, smooth)
cap-color	Color of cap (brown, buff, cinnamon, gray, green, etc.)
bruises	Presence of bruises (yes or no)
odor	Smell of mushroom (almond, anise, creosote, fishy, etc.)
gill-attachment	Attachment of gills to stalk (attached, descending, free)
gill-spacing	Distance between gills (close, crowded, distant)
gill-size	Size of gills (broad or narrow)
gill-color	Color of gills (black, brown, buff, chocolate, etc.)
stalk-shape	Shape of stalk (enlarging or tapering)
stalk-root	Root type of stalk (bulbous, club, cup, equal, etc.)
stalk-surface-above-ring	Surface texture above ring on stalk (fibrous, scaly, silky)
stalk-surface-below-ring	Surface texture below ring on stalk (same as above)
stalk-color-above-ring	Color above ring on stalk (brown, buff, cinnamon, etc.)
stalk-color-below-ring	Color below ring on stalk (same as above)
veil-type	Type of veil (partial or universal)
veil-color	Color of veil (brown, orange, white, yellow)
ring-number	Number of rings (none, one, two)
ring-type	Type of ring (cobwebby, evanescent, flaring, large, etc.)
spore-print-color	Color of spore print (black, brown, buff, chocolate, etc.)
population	Population size (abundant, clustered, numerous, scattered)
habitat	Habitat where mushroom grows (grasses, leaves, meadows, etc.)

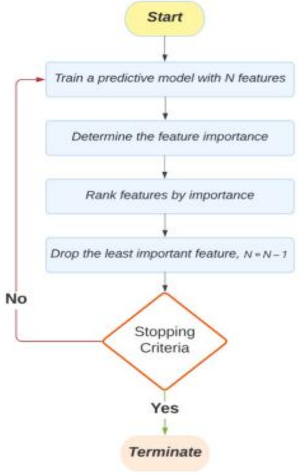
Source : (Research Results, 2025)

The dataset consists of 8,124 entries and 23 columns, all of which are of object (categorical) data type. Each column contains no missing values (nonnull) and includes features such as cap shape, color, odor, gill type, ring presence, and habitat. The first column, labeled class, serves as the target label indicating whether the mushroom is edible or poisonous.

Recursive Feature Elimination (RFE)

The Recursive Feature Elimination (RFE) algorithm was initially introduced by Guyon et al. [15] in the context of gene selection for disease classification. The approach begins by training a machine learning (ML) model using all available features, followed by assessing feature relevance based on the characteristics of the model such as regression coefficients in linear models or feature importance scores in tree-based classifiers. Features with the lowest contribution to the model are iteratively removed, and the model is retrained using the remaining features. This elimination cycle continues until a predefined stopping criterion is reached, such as achieving the target number of features or observing no further performance improvement after feature removal [16].

This iterative procedure represents a form of backward feature elimination [17]. RFE begins by training a model with the complete set of features and progressively removes the least useful ones, retraining the model after each elimination step. This iterative refinement enables a more reliable assessment of feature relevance compared to single-step selection techniques. RFE employs a greedy search approach, choosing the most beneficial subset of features locally at each stage in pursuit of the best overall feature set [18]. Additionally, this greedy strategy offers greater computational efficiency than exhaustive search methods, which become increasingly expensive as the dimensionality of the feature space grows [19].


Start
Train a predictive model with N features
Determine the feature importance
Rank features by importance
Drop the least important feature, N = N − 1
No
Stopping Criteria
Yes
Terminate

Source : (Okan, 2025)

Figure 2. Stages of the Recursive Feature Elimination (RFE) Algorithm Process

The flowchart in Figure 2 illustrates the process of the Recursive Feature Elimination (RFE) algorithm [20]. The process begins by training a predictive model using all available features in the dataset (N features). Once the model is built, the next step is to determine the importance score of each feature. Based on these importance scores, the features are ranked from the most important to the least important. The feature with the lowest importance score is then removed, reducing the number of features by one (N = N - 1).

After removing the feature, the system checks whether the stopping criteria have been met, such as the remaining number of features reaching the desired minimum threshold or no further improvement in model performance. If the criteria are not met, the process repeats starting again with the remaining features. This iterative process continues until the stopping criteria are satisfied, at which point the algorithm stops and produces the most relevant subset of features for the predictive model.

Decision Tree

Decision Tree is a machine learning algorithm used for classification and regression tasks [21]. In classification, this algorithm constructs a decision tree by selecting the best attribute at each node based on certain criteria [22]. Two common criteria used to split data at nodes are the Gini Index and Entropy (Information Gain) [23].

Gini Index

The Gini Index measures the impurity of a node using the formula (1)[24].

$$Gini = 1 - \sum_{i=1}^{k} p_i^2 \qquad (1)$$

where p_i is the probability of occurrence of the i class at a particular node, and k is the total number of classes in the dataset.

$$p_i = \frac{N_i}{N} \qquad (2)$$

Where N_i is the number of data points belonging to the i class, and N is the total number of data points at that node [25].

Total Impurity at a Split:
If a node is split into m subsets$(D_1, D_2, \dots, D_m)$ the total impurity is calculated as the weighted average[26].

$$Gini_{Split} = \sum_{j=1}^{m} \frac{|D_j|}{|D|} . Gini(D_j) \qquad (3)$$

Entropy and Information Gain

Entropy measures the uncertainty (impurity) of the data using formula (4)[27]

$$Entropy = -\sum_{i=1}^{k} p_i . \log_2(p_i) \qquad (4)$$

Where k is the number of classes in the dataset, and p_i is the probability of a data point belonging to the i class[28].

$$IG = Entropy_{parent} - \sum_{j=1}^{n} \frac{|D_j|}{|D|} . entropy\ (D_j) \qquad (5)$$

$Entropy_{parent}$ is the entropy value of the data before the split. The data is then divided into several subsets D_j, each with $|D_j|$. data points. Meanwhile, $|D_j|$ is the total number of data points before the split. This information is used to calculate Information Gain in the Decision Tree[29], [30].

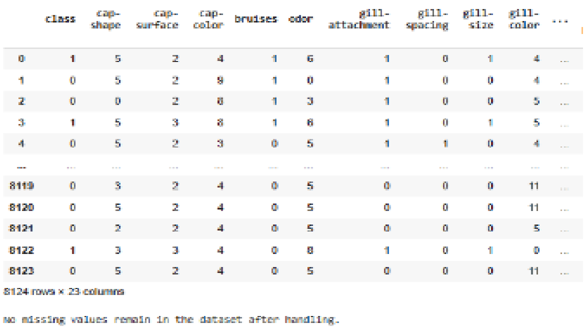
Hyperparameter tuning

Hyperparameter tuning in a Decision Tree refers to the process of determining the optimal configuration of parameters such as maximum tree depth (*max_depth*), the minimum number of samples needed to split an internal node (*min_samples_split*), the minimum number of samples required at a leaf node (*min_samples_leaf*), and the splitting criterion (*criterion*). Adjusting these parameters helps control model complexity and prevents issues such as overfitting or underfitting [31], [32], [33]. The primary aim is to enhance the model's predictive performance and its ability to generalize effectively to unseen data [31].

RESULTS AND DISCUSSION

Data Cleaning

In the data cleaning stage, the dataset was cleaned by removing rows containing missing values to ensure the quality of the data used. Next, all categorical features in the dataset were converted into numerical form using the Label Encoding method so they could be processed by the Decision Tree algorithm, as shown in Figure 2. In this way, variables such as the shape, surface, and color of the mushroom cap, as well as the class label (edible or poisonous), were converted into appropriate numerical representations, making the data ready for training classification models using either the Gini or Entropy criteria.



	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...
0	1	5	2	4	1	6	1	0	1	4	...
1	0	5	2	9	1	0	1	0	0	4	...
2	0	0	2	8	1	3	1	0	0	5	...
3	1	5	3	8	1	6	1	0	1	5	...
4	0	5	2	3	0	5	1	1	0	4	...
...	...	...	...	...	...	...	...	...	...	...	...
8119	0	3	2	4	0	5	0	0	0	11	...
8120	0	5	2	4	0	5	0	0	0	11	...
8121	0	2	2	4	0	5	0	0	0	5	...
8122	1	3	3	4	0	8	1	0	1	0	...
8123	0	5	2	4	0	5	0	0	0	11	...

8124 rows × 23 columns

```
No missing values remain in the dataset after handling.
```

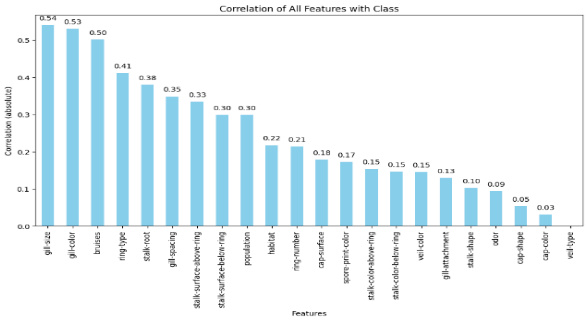
Source : (Research Results, 2025)
Figure 3. Result of Label Encoding Process on the Mushroom Dataset

Figure 3 displays the mushroom dataset after undergoing data cleaning and transformation of categorical features into numerical values using the Label Encoding method. This dataset consists of 8,124 rows and 23 columns, including various features such as the shape and color of the mushroom cap, odor, gill color, as well as the mushroom class indicating whether the mushroom is edible or poisonous. All missing values have been successfully addressed, as indicated by the absence of any remaining missing values. The dataset is now ready to be used in the training process of classification models based on algorithms like Decision Tree.

Exploratory Data Analysis (EDA)

At the Exploratory Data Analysis (EDA) stage, a correlation analysis was conducted between the features in the dataset and the target class (edible or poisonous). Since all features had been converted into numerical form using Label Encoding, the correlations were calculated using a direct numerical approach.

The results of the correlation analysis were then visualized in the form of a bar plot to illustrate the extent of influence each feature has on the classification target, as shown in Figure 4.



Source : (Research Results, 2025)
Figure 4. Correlation of All Features with Class

Figure 4 shows the absolute correlation values between each feature in the dataset and the classification target class, indicating whether a mushroom is edible or poisonous. It is observed that the feature *gill-size* has the highest correlation at 0.54, followed by *gill-color* at 0.53, and *bruises* at 0.50. These three features exhibit the strongest influence on mushroom classification. In contrast, features such as *veil-type*, *cap-color*, and *odor* have low correlation values, indicating relatively little impact on predicting the mushroom class. This result demonstrates that not all features contribute equally in the classification process, and features with higher correlation may be prioritized in the feature selection process to improve model performance.

Recursive Feature Elimination (RFE)

In this implementation, the Decision Tree is used as the base estimator in RFE due to its ability to assess feature importance based on data splits. The parameter random state=42 ensures consistent and reproducible results. RFE is then configured to select the top 10 features out of the initial 22 features, as shown in Table 1. The fitting process is performed only on the training data (X train and y train) to prevent data leakage from the test or validation sets, ensuring that the model evaluation remains objective and valid.

Table2. Number and List of Features Before and After RFE

Stage	Number of Features	Feature List
Before RFE	22	cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, spore-print-color, population, habitat
After RFE	10	cap-surface, cap-color, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-below-ring, stalk-color-below-ring, spore-print-color, population

Source : (Research Results, 2025)

Table 2 shows the number and list of features used before and after the RFE process. Before RFE, there were 22 initial features covering various mushroom characteristics such as shape, color, odor, as well as stem and spore structures. After the RFE process, only the 10 most relevant features were selected, such as cap-surface, gill-size, and spore-print-color. This selection aims to simplify the model without sacrificing performance, while also reducing complexity and the risk of overfitting.

Split Data

The original data was split into two parts: 80% for the combined training and validation set, and 20% for the test set using train test split with random state=42 to ensure consistent results. Then, from the train validation data, 25% was allocated for validation and the remaining 75% for training, resulting in an overall data split of 60% training, 20% validation, and 20% test. This splitting ensures that the test data remains completely separate for final evaluation, while the training and validation sets are used for model training and tuning.

Decision Tree Model

In the model training process, Decision Tree was applied using two main splitting criteria: Gini Index and Entropy. The Gini Index aims to minimize impurity to produce more homogeneous nodes, while Entropy focuses on maximizing the reduction of uncertainty (information gain) during data splitting. Using both criteria allows the evaluation of the model's performance based on different splitting methods in mushroom classification. To evaluate the performance of the Decision Tree model, two different splitting criteria, Gini Index and Entropy, were used. Table 2 presents the classification results based on four main evaluation metrics: Accuracy, Precision, Recall, and F1-Score.

Table 3. Decision Tree Performance Using Gini and Entropy Splitting Criteria

Splitting Criterion	Accuracy	Precision	Recall	F1-Score
Gini	0.79	0.80	0.79	0.79
Entropy	0.72	0.82	0.72	0.70

Source : (Research Results, 2025)

Table 3 shows a performance comparison of the Decision Tree model based on two splitting criteria, namely Gini and Entropy. The model using the Gini criterion achieved an accuracy of 0.79, with precision, recall, and F1-Score balanced around 0.79–0.80. This indicates that the model is fairly stable in classification. On the other hand, the model using the Entropy criterion produced a higher precision of 0.82, but its accuracy, recall, and F1-Score were lower, at 0.72, 0.72, and 0.70 respectively. This suggests that although Entropy can identify positive predictions more accurately, the model is less balanced in recognizing the overall data. Overall, the Gini criterion provides more consistent and balanced performance compared to Entropy.

The model performance evaluation was also conducted based on the combination of splitting methods (Gini and Entropy) and the application of the feature selection technique (Recursive Feature Elimination). Table 4 summarizes the test results based on the number of features used, training time, as well as four main evaluation metrics: accuracy, precision, recall, and F1-Score.

Table 4. Decision Tree Performance: Gini vs Entropy with/without RFE

Model	Criterion	#Features	Train Time (s)	Acc	Prec	Rec	F1
DT	Gini	22	0.0099	0.79	0.80	0.79	0.79
DT	Entropy	22	0.0213	0.72	0.82	0.72	0.70
RFE + DT	Gini	10	0.0056	0.79	0.80	0.79	0.79
RFE + DT	Entropy	10	0.0086	0.72	0.82	0.72	0.70

Source : (Research Results, 2025)

The results show that the model using the Gini criterion delivers more stable and balanced performance across all evaluation metrics, with accuracy, precision, recall, and F1-Score ranging between 0.79 and 0.80. In contrast, the model with the Entropy criterion records a higher precision value (0.82), but its accuracy, recall, and F1-Score are lower, at 0.72, 0.72, and 0.70 respectively. This indicates that although Entropy is better at specifically identifying the positive class, its overall ability to recognize the entire dataset is less balanced.

The application of RFE demonstrates that reducing the number of features from 22 to 10 does not significantly decrease the model's performance. Training time becomes shorter, especially for the model using the Gini criterion, without sacrificing accuracy or prediction quality. This shows that feature selection can improve computational efficiency while maintaining model performance.

Hyperparameter Tuning with Grid Search Cross-Validation

To obtain the best performance from the Decision Tree model, hyperparameter optimization is conducted using the Grid Search with Cross-Validation (GridSearchCV) approach. This optimization is carried out on the dataset after the feature selection stage using Recursive Feature Elimination (RFE). The objective of this procedure is to identify the optimal configuration of three key hyperparameters: the maximum depth of the tree (*max_depth*), the minimum number of samples allowed in a leaf node (*min_samples_leaf*), and the minimum number of samples necessary to split a node (*min_samples_split*). Adjustments are made for two splitting criteria, namely Gini and Entropy, to evaluate the effectiveness of each in producing an accurate and balanced classification model. The details of the parameter combination scenarios tested are shown in table 5.

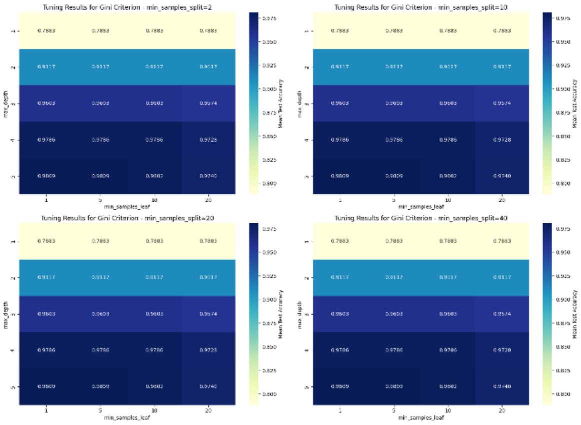
Table 5. Hyperparameter Tuning Scenarios for Decision Tree with RFE

Criteria	max_depth	min_samples leaf	min_samples_ split	Number of Combinations
Gini	1–5	1, 5, 10, 20	2, 10, 20, 40	5 × 4 × 4 = 80
Entropy	1–5	1, 5, 10, 20	2, 10, 20, 40	5 × 4 × 4 = 80

Source : (Research Results, 2025)

Table 5 presents the hyperparameter tuning scenarios applied to the Decision Tree model after feature selection using the Recursive Feature Elimination (RFE) method. The tuning process was conducted for two splitting criteria, Gini and Entropy, by testing combinations of three main hyperparameters: *max_depth* (1–5), *min_samples_leaf* (1, 5, 10, 20), and *min_samples_split* (2, 10, 20, 40). Each criterion resulted in a total of 80 parameter combinations, evaluated using 5-fold cross-validation.

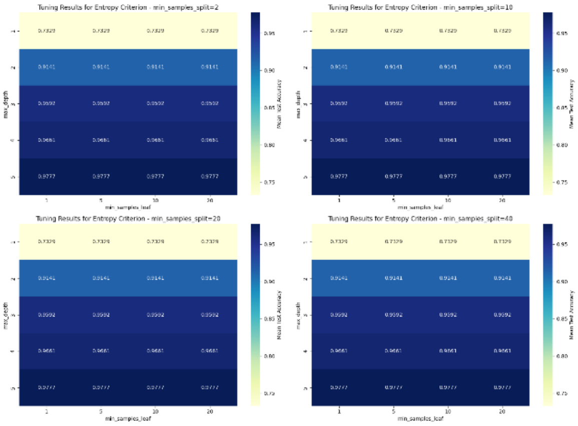
The results of the tuning process, including the best model performance for each criterion, are visualized in Figure 5 for the Decision Tree with the Gini criterion and Figure 6 for the Decision Tree with the Entropy criterion. These visualizations aim to facilitate the analysis and comparison of model performance.



Source : (Research Results, 2025)

Figure 5. Hyperparameter Tuning Results for Decision Tree (Gini)

The hyperparameter tuning results in Figure 5 for the Decision Tree model with the Gini criterion are visualized as a heatmap. This visualization illustrates the impact of the parameter combinations of *max_depth* and *min_samples_leaf* on the average validation accuracy across four different values of *min_samples_split* (2, 10, 20, and 40). Overall, the model's performance tends to improve with increasing tree depth (*max_depth*) up to a certain point, with the optimal configuration achieved at *max_depth* = 5, *min_samples_leaf* = 5, and *min_samples_split* = 10 or 20, resulting in the highest validation accuracy, approaching 0.98.



Source : (Research Results, 2025)

Figure 6. Hyperparameter Tuning Results for Decision Tree (Entropy)

The hyperparameter tuning results shown in Figure 6 pertain to the Decision Tree model with the Entropy criterion. Each heatmap depicts the average validation accuracy for combinations of *max_depth* and *min_samples_leaf* values across four different *min_samples_split* values (2, 10, 20, and 40).

In general, the performance of the model with the Entropy criterion tends to be lower compared to the Gini criterion (see Figure 7). The highest accuracy achieved is approximately 0.97, which remains stable at configurations of *max_depth* = 5 and *min_samples_leaf* = 5–10. This indicates that while the model using Entropy can identify patterns, its classification effectiveness is inferior to the Gini-based model, particularly in terms of average validation accuracy.

After hyperparameter tuning was performed using the GridSearchCV method for the Decision Tree model with two splitting criteria approaches—Gini and Entropy—the best parameters yielding optimal performance were obtained. Further evaluation was conducted on the test data to measure the model's effectiveness in real-world predictions. Table 6 presents a comprehensive comparison between the two models based on the best parameters, cross-validation accuracy, and key evaluation metrics such as precision, recall, and F1-score.

Table 6. Comparison of Tuning Results and Evaluation of Decision Tree Models

Criteria	Gini	Entropy

Best Parameters	max_depth = 5
	min_samples_leaf = 1
	min_samples_split = 2
		max_depth = 5
		min_samples_leaf = 1
		min_samples_split = 2

CV Accuracy	0.9809	0.9777
Test Accuracy	0.9766	0.9711
Precision (0 / 1)	0.99 / 0.96	1.00 / 0.94
Recall (0 / 1)	0.97 / 0.99	0.94 / 1.00
F1-Score (0 / 1)	0.98 / 0.98	0.97 / 0.97
Accuracy	0.98	0.97

Source : (Research Results, 2025)

Based on Table 6, the Decision Tree model using the Gini criterion demonstrates slightly better performance compared to the Entropy criterion. Although both models utilize the same optimal parameters, the Gini-based model achieves a higher cross-validation accuracy (0.9809) and test accuracy (0.9766) than the Entropy-based model (0.9777 and 0.9711, respectively).

Furthermore, the evaluation metrics — precision, recall, and F1-score — are more balanced across the classes in the Gini model. Precision represents the proportion of correctly identified positive instances among all instances predicted as positive. A high precision value indicates a low number of false positives, which is essential in contexts where incorrect classification of negative instances as positive may lead to harmful consequences. Recall, in contrast, measures the model's ability to correctly identify all actual positive instances. A high recall value means fewer false negatives, which is crucial when missing positive cases (e.g., poisonous mushrooms) could be dangerous.

In this case, the Gini-based model not only delivers high values of precision and recall but also maintains a good balance between them, as reflected in its F1-score. This balance suggests that the model is capable of making accurate predictions without significantly favoring one class over another. Consequently, the Gini-based Decision Tree is recommended for use, as it provides more consistent and reliable classification performance across all evaluation metrics.

The performance metrics of various Decision Tree models were evaluated with different configurations. The models used include the basic Decision Tree (DT), Decision Tree combined with Recursive Feature Elimination (DT+RFE), and Decision Tree with RFE and hyperparameter tuning (DT+RFE+Tuning). Two splitting criteria, Gini and Entropy, were employed to observe their impact on model performance. The reported evaluation

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

metrics include accuracy, precision, recall, and F1-score for class 0 and class 1, providing a comprehensive view of each model's effectiveness, as shown in Table 7.

Table 7. Performance Metrics of Decision Tree Models with Feature Selection and Hyperparameter Tuning

Model Category	Criterion	Acc	Prec (0 / 1)	Rec (0 / 1)	F1 (0 / 1)
DT	Gini	0.7914	0.76 / 0.84	0.87 / 0.71	0.81 / 0.77
DT	Entropy	0.7231	0.65 / 1.00	1.00 / 0.43	0.79 / 0.60
DT+RFE	Gini	0.7914	0.76 / 0.84	0.87 / 0.71	0.81 / 0.77
DT+RFE	Entropy	0.7231	0.65 / 1.00	1.00 / 0.43	0.79 / 0.60
DT+RFE+ Tuning	Gini	0.9766	0.99 / 0.96	0.97 / 0.99	0.98 / 0.98
DT+RFE+ Tuning	Entropy	0.9711	1.00 / 0.94	0.94 / 1.00	0.97 / 0.97

Source : (Research Results, 2025)

From Table 7, it can be observed that applying Recursive Feature Elimination (RFE) alone does not significantly improve the model's performance compared to the basic Decision Tree. The accuracy, precision, recall, and F1-scores of DT and DT+RFE under both Gini and Entropy criteria remain similar, indicating that feature selection by itself does not sufficiently enhance predictive ability.

In contrast, combining RFE with hyperparameter tuning significantly improves all evaluation metrics. The Gini-based model achieves the highest accuracy (97.66%), followed closely by the Entropy-based model (97.11%). Notably, precision and recall are well-balanced across both classes, which indicates that the models are not biased toward either class. For instance, the Gini model achieves precision scores of 0.99 and 0.96 for class 0 and class 1, with corresponding recall scores of 0.97 and 0.99. This balance reduces the risk of misclassifying minority classes and ensures both sensitivity and specificity in predictions.

However, the consistently high performance of the DT+RFE+Tuning models raises potential concerns of overfitting, especially since training and test accuracies are both near-perfect. This could mean the models are capturing noise along with the patterns in the training data. Future work should include validation on external datasets or through k-fold cross-validation with more folds to confirm the model's generalizability. Additionally, since Decision Tree models are prone to overfitting by design, especially with deep trees and numerous features, regularization techniques or ensemble methods (e.g., Random Forests) could be considered.

Overall, the findings emphasize the importance of combining feature selection with thorough hyperparameter optimization to construct a well-performing and generalizable Decision Tree model, while also remaining cautious of potential overfitting and class imbalance bias.

CONCLUSION

This study was motivated by the need for an accurate, reliable, and interpretable classification model for high-dimensional biological data, particularly in distinguishing between edible and poisonous mushrooms based on categorical features. Considering the importance of transparency and interpretability, the Decision Tree algorithm was chosen as the foundation for model development, in line with the growing demand for explainable AI in critical domains such as food safety. The findings indicate that the Decision Tree algorithm, when combined with proper data preprocessing techniques (such as data cleaning and label encoding), feature selection using Recursive Feature Elimination (RFE), and hyperparameter tuning via Grid Search Cross-Validation, can produce a highly accurate and well-balanced classification model. The use of the Gini Index as the splitting criterion yielded slightly better performance stability than Entropy, confirming its suitability for categorical biological data classification.

The final model achieved an accuracy above 97%, demonstrating strong potential for practical applications. These results confirm that integrating dimensionality reduction techniques such as RFE with interpretable models like Decision Trees is an effective approach to handling high-dimensional data challenges, while also supporting the development of explainable classification systems. Although the findings are encouraging, this study is not without limitations. The work relies on a single classification model and does not incorporate validation using an external dataset. Future studies should consider combining RFE with ensemble learning techniques, such as Random Forest or XGBoost, to improve performance and model stability. Moreover, evaluating the method on additional biological datasets and including external validation would be valuable for assessing the model's generalizability and reducing the likelihood of overfitting.

Overall, the methods and findings presented in this study provide not only a robust approach for mushroom classification but also contribute both theoretically and practically to the development of transparent and reliable AI models in the biological domain. As such, this study serves as a relevant foundation for advancing explainable AI-based classification systems, particularly for high-dimensional categorical data such as mushrooms.

REFERENCE

[1] A. Rianti, T. Ridwan, S. Widodo, and R. Andrian, 'Application of Decision Tree Algorithm for Edible Mushroom Classification', *J. Appl. Informatics Comput.*, vol. 6, no. 1, 2022, doi: 10.30871/jaic.v6i1.4087.

[2] S. Metlek and H. Çetiner, 'Classification of Poisonous and Edible Mushrooms with Optimized Classification Algorithms', *Int. Conf. Appl. Eng. Nat. Sci.*, vol. 1, no. 1, 2023, doi: 10.59287/icaens.1030.

[3] O. Günlük, J. Kalagnanam, M. Li, M. Menickelly, and K. Scheinberg, 'Optimal decision trees for categorical data via integer programming', *J. Glob. Optim.*, vol. 81, no. 1, 2021, doi: 10.1007/s10898-021-01009-y.

[4] A. S. Saud, S. Shakya, and B. Neupane, 'Analysis of Depth of Entropy and GINI Index Based Decision Trees for Predicting Diabetes', *Indian J. Comput. Sci.*, vol. 6, no. 6, 2021, doi: 10.17010/ijcs/2021/v6/i6/167641.

[5] O. Rahmati, M. Avand, P. Yariyan, J. P. Tiefenbacher, A. Azareh, and D. T. Bui, 'Assessment of Gini-, entropy- and ratio-based classification trees for groundwater potential modelling and prediction', *Geocarto Int.*, vol. 37, no. 12, 2022, doi: 10.1080/10106049.2020.1861664.

[6] A. M. Priyatno and T. Widiyaningtyas, 'A SYSTEMATIC LITERATURE REVIEW: RECURSIVE FEATURE ELIMINATION ALGORITHMS', *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 9, no. 2, 2024, doi: 10.33480/jitk.v9i2.5015.

[7] H. A. Al Essa and W. S. Bhaya, 'Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system', *Bull. Electr. Eng. Informatics*, vol. 13, no. 1, 2024, doi: 10.11591/eei.v13i1.5844.

[8] K. Tutuncu, I. Cinar, R. Kursun, and M. Koklu, 'Edible and Poisonous Mushrooms Classification by Machine Learning Algorithms', 2022, doi: 10.1109/MECO55406.2022.9797212.

[9] S. Lee, C. Lee, K. G. Mun, and D. Kim, 'Decision Tree Algorithm Considering Distances between Classes', *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3187172.

[10] M. A. Bouke, A. Abdullah, J. Frnda, K. Cengiz, and B. Salah, 'BukaGini: A Stability-Aware Gini Index Feature Selection Algorithm for Robust Model Performance', *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3284975.

[11] D. Rianasari, M. N. Triana, M. R. Dewi, Y. Astutik, and R. Wirawan, 'The Classification of Mushroom Types Using Naïve Bayes and Principal Component Analysis', *JISA(Jurnal Inform. dan Sains)*, vol. 5, no. 2, 2022, doi: 10.31326/jisa.v5i2.1380.

[12] C. M. Wati, A. C. Fauzan, and H. Harliana, 'PERFORMANCE COMPARISON OF MUSHROOM TYPE CLASSIFICATION BASED ON MULTI-SCENARIO DATASET USING DECISION TREE C4.5 AND C5.0', *J. Ris. Inform.*, vol. 4, no. 3, 2022, doi: 10.34288/jri.v4i3.383.

[13] A. B. Siddique *et al.*, 'Studying the effects of feature selection approaches on machine learning techniques for Mushroom classification problem', 2023, doi: 10.1109/ICIT59216.2023.10335842.

[14] M. S. Morshed, F. Bin Ashraf, M. U. Islam, and M. S. R. Shafi, 'Predicting Mushroom Edibility with Effective Classification and Efficient Feature Selection Techniques', in *International Conference on Robotics, Electrical and Signal Processing Techniques*, 2023, vol. 2023-January, doi: 10.1109/ICREST57604.2023.10070049.

[15] Y.-H. Hu, R.-Y. Wu, Y.-C. Lin, and T.-Y. Lin, 'A novel MissForest-based missing values imputation approach with recursive feature elimination in medical applications', *BMC Med. Res. Methodol.*, vol. 24, no. 1, p. 269, 2024, doi: 10.1186/s12874-024-02392-2.

[16] R. C. Chen, W. E. Manongga, and C. Dewi, 'Recursive Feature Elimination for Improving Learning Points on Hand-Sign Recognition', *Futur. Internet*, vol. 14, no. 12, 2022, doi: 10.3390/fi14120352.

[17] N. M. Abdelwahed, G. S. El-Tawel, and M. A. Makhlouf, 'Effective hybrid feature selection using different bootstrap enhances cancers classification performance', *BioData Min.*, vol. 15, no. 1, 2022, doi: 10.1186/s13040-022-00304-y.

[18] M. Awad and S. Fraihat, 'Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for

Machine Learning-Based Intrusion Detection Systems', *J. Sens. Actuator Networks*, vol. 12, no. 5, 2023, doi: 10.3390/jsan12050067.

[19] F. Camattari, S. Guastavino, F. Marchetti, M. Piana, and E. Perracchione, 'Classifier-dependent feature selection via greedy methods', *Stat. Comput.*, vol. 34, no. 5, Oct. 2024, doi: 10.1007/s11222-024-10460-2.

[20] O. Bulut, B. Tan, E. Mazzullo, and A. Syed, 'Benchmarking Variants of Recursive Feature Elimination: Insights from Predictive Tasks in Education and Healthcare', *Information*, vol. 16, no. 6, p. 476, Jun. 2025, doi: 10.3390/info16060476.

[21] M. Morgan, C. Blank, and R. Seetan, 'Plant disease prediction using classification algorithms', *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, 2021, doi: 10.11591/ijai.v10.i1.pp257-264.

[22] I. Johri, M. K. Nallakaruppan, B. Balusamy, G. V, and V. Grover, 'Application of Neural Networks and Genetic Algorithms in Establishing Logical Rules for Evaluating the Edibility of Mushroom Data', in *Communications in Computer and Information Science*, 2023, vol. 1920, doi: 10.1007/978-3-031-45121-8_18.

[23] A. Al-majali, 'The Effect of Democracy and Income Inequality (Gini Index)', *Jordanian J. Law Polit. Sci.*, vol. 15, no. 4, 2024, doi: 10.35682/jjlps.v15i4.501.

[24] Y. Lu, T. Ye, and J. Zheng, 'Decision Tree Algorithm in Machine Learning', 2022, doi: 10.1109/AEECA55500.2022.9918857.

[25] G. Sahbeni, J. B. Pleynet, and K. Jarocki, 'A spatiotemporal analysis of precipitation anomalies using rainfall Gini index between 1980 and 2022', *Atmos. Sci. Lett.*, vol. 24, no. 7, 2023, doi: 10.1002/asl.1161.

[26] E. Ketzaki and N. Farmakis, 'A matrix based computational method of the Gini index', *Communications in Statistics - Theory and Methods*, vol. 52, no. 17. 2023, doi: 10.1080/03610926.2021.2024233.

[27] Y. Miao, J. Wang, B. Zhang, and H. Li, 'Practical framework of Gini index in the application of machinery fault feature extraction', *Mech. Syst. Signal Process.*, vol. 165, 2022, doi: 10.1016/j.ymssp.2021.108333.

[28] B. Shao, 'Decomposition of the Gini index by income source for aggregated data and its applications', *Comput. Stat.*, vol. 36, no. 3, 2021, doi: 10.1007/s00180-021-01069-4.

[29] B. Chen *et al.*, 'A full generalization of the Gini index for bearing condition monitoring', *Mech. Syst. Signal Process.*, vol. 188, 2023, doi: 10.1016/j.ymssp.2022.109998.

[30] S. Settepanella, A. Terni, M. Franciosi, and L. Li, 'The robustness of the generalized Gini index', *Decis. Econ. Financ.*, vol. 45, no. 2, 2022, doi: 10.1007/s10203-022-00378-7.

[31] W. Nugraha and A. Sasongko, 'Hyperparameter Tuning on Classification Algorithm with Grid Search', *SISTEMASI*, vol. 11, no. 2, 2022, doi: 10.32520/stmsi.v11i2.1750.

[32] M. Koeshardianto, K. E. Permana, D. S. Y. Kartika, and W. Setiawan, 'BEANS CLASSIFICATION USING DECISION TREE AND RANDOM FOREST WITH RANDOMIZED SEARCH HYPERPARAMETER TUNING', *Commun. Math. Biol. Neurosci.*, vol. 2023, 2023, doi: 10.28919/cmbn/8225.

[33] R. Gomes Mantovani *et al.*, 'Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms', *Data Min. Knowl. Discov.*, vol. 38, no. 3, 2024, doi: 10.1007/s10618-024-01002-5.