JSDSI
JOURNAL OF STATISTICS DATA SCIENCE AND INFORMATICS
vol. 1, No. 1, 2026, pp. 40-57

Classification of Antibiotic and Non-Antibiotic Compounds in *Moringa oleifera* Using a Boosting Technique with the Decision Tree Algorithm

Apriana Putri[1]*, Putri Indi Rahayu[2]

1 Statistics Program,Statistics, Faculty of Mathematics and Natural Sciences, University of West Sulawesi, Indonesia
2 Department of Statistics, Faculty of Mathematics and Natural Sciences, University of West Sulawesi,Indonesia
Corresponding Email: apriana09042000@gmail.com*

ARTICLE INFO

Article history

Received: 4 May, 2026
Revised: 30 June, 2026
Accepted: 30 June, 2026

Keywords
Antibiotic
Boosting
Decision Tree
Ensemble Learning
Moringa oleifera

ABSTRACT

Indonesia has high biodiversity, including medicinal plants with potential therapeutic properties such as Moringa oleifera. The identification of bioactive compounds as antibiotic candidates using conventional approaches generally requires substantial time and cost. Therefore, this study applies a machine learning approach to classify antibiotic and non-antibiotic compounds using a Boosting technique with the AdaBoost (Adaptive Boosting) algorithm and Decision Tree as the base estimator under an imbalanced class condition. The initial dataset consisted of 3,907 drug compounds; after the pre-processing stage, 3,904 compounds were obtained and used for model development, while 38 Moringa oleifera compounds were used as external prediction data. Model evaluation was performed using Accuracy, Precision, Recall, F1-Score, ROC-AUC, Specificity, and G-Mean metrics, along with Stratified 5-Fold Cross-Validation to evaluate model stability. The results showed that the Decision Tree model without Boosting on the validation set achieved a recall value of 0.6089 and a G-Mean value of 0.7728. After applying AdaBoost, the recall increased to 0.7542 and the G-Mean increased to 0.8636, indicating improved capability of the model in recognizing the minority class (Antibiotic Compound). The Stratified 5-Fold Cross-Validation result of the AdaBoost-Decision Tree model achieved a value of 0.9556 ± 0.0069, indicating stable model performance. Prediction results on 38 Moringa oleifera compounds identified three compounds predicted as Antibiotic Compound, namely Niazicin A, Niazinin A, and [(2S,3R,4S,5S,6R)-3,4,5-trihydroxy-6-(hydroxymethyl)oxan-2-yl] (1E)-N-sulfooxy-2-[4-[(2S,3R,4R,5R,6S)-3,4,5-trihydroxy-6-methyloxan-2-yl]oxyphenyl]ethanimidothioate. Thus, the application of AdaBoost with Decision Tree improves classification performance, particularly in identifying minority classes, and provides an initial approach for identifying potential antibiotic compounds from natural sources.

p-ISSN xxxx-xxxx
e-ISSN xxxx-xxxx

1. Introduction

Indonesia is one of the countries with the highest biodiversity in the world, possessing great potential as a source of natural resources for drug development [1]. Various medicinal plants are known to contain secondary phytochemical compounds such as flavonoids, terpenoids, tannins, saponins, and alkaloids that contribute to various biological activities. The presence of these bioactive compounds makes the exploration of medicinal plants an important approach in the discovery of new drug candidates [2].

One of the plants with potential as a natural source for drug development is *Moringa oleifera* (moringa). This plant is known to contain various secondary metabolites, such as flavonoids, alkaloids, tannins, saponins, and terpenoids/steroids, which have potential biological activities. These bioactive compounds make *Moringa oleifera* an interesting plant to be studied in the development of drug candidates, including as a potential source of natural antibiotic [3].

Antibiotics are compounds used to treat bacterial infections by inhibiting or eliminating disease-causing microorganisms. Although antibiotics have provided significant benefits in treating infections, inappropriate and excessive use has led to an increase in antibiotic resistance. Antibiotic resistance occurs when bacteria develop the ability to survive the effects of antibiotics, making treatment less effective and increasing the risk of therapeutic failure [4]. This issue highlights the need to discover new antibiotic candidates, particularly from natural sources containing potential bioactive compounds such as medicinal plants [5].

The identification process of bioactive compounds from natural plants is generally complex and requires considerable time when conducted through conventional approaches. Along with technological advancements, an effective solution for predicting the potential antibiotic activity of compounds based on their molecular characteristics or structures is the application of machine learning approaches. Machine learning has been widely applied in drug discovery to assist in the efficient selection of potential compound candidates [6].

However, one of the major challenges in applying this method is class imbalance, a condition where the number of samples in one class is significantly lower than the other class, which may reduce the performance of classification models [7]. To improve machine learning model performance, one appropriate approach is the use of Ensemble Learning. Ensemble Learning is a technique that combines several weak classifiers to produce a stronger and more comprehensive model [8].

One of the commonly used methods in Ensemble Learning is Boosting, which combines multiple classification models iteratively by assigning higher weights to samples that are incorrectly classified in previous stages. This mechanism allows subsequent models to focus more on correcting prediction errors, thereby improving classification performance [9]. In this study, the AdaBoost algorithm based on Decision Tree was applied. The weighting mechanism in AdaBoost enables the model to pay greater attention to difficult-to-classify samples, thereby improving the model's ability to recognize minority classes in imbalanced datasets [10]. Decision Tree was selected as the base learner because it can handle non-linear

relationships between features and provides an interpretable model structure, although it tends to be sensitive to data variations [10].

Previous studies have demonstrated the effectiveness of Ensemble Learning techniques in various classification applications, including the prediction of biological activities of compounds. A study by [11] applied Ensemble Learning to predict anticancer activity from natural compounds and showed a significant improvement in prediction accuracy compared with single models. Furthermore, research by [12] implemented Ensemble Learning using the Random Forest algorithm to predict the potential of natural medicinal plants as antibiotics based on herbal medicine formulations, achieving an accuracy of 91.10% and identifying 14 potential plant candidates, with 10 candidates supported by scientific evidence of antibacterial activity. Another study [13] demonstrated that the LightGBM algorithm was highly effective for genomic selection in maize breeding, providing higher speed and accuracy compared with traditional methods. A recent study by [14] compared several Ensemble Learning algorithms, such as Random Forest and Boosting, for disease diagnosis classification and showed that ensemble methods improved model performance compared with single models.

Based on these studies, Ensemble Learning has been widely applied in various classification fields; however, research specifically focusing on the classification of antibiotic compounds in *Moringa oleifera* remains limited [11]-[14]. Research conducted by [12] investigated the prediction of medicinal plants as antibiotics, but the study applied a Random Forest approach and did not focus on compound classification based on molecular characteristics. Furthermore, the application of AdaBoost based on Decision Tree for compound classification under imbalanced class conditions, particularly to improve the model's ability to recognize minority classes, has not been extensively studied [11]-[14]. Therefore, this study focuses on implementing AdaBoost based on Decision Tree to improve the classification performance of antibiotic and non-antibiotic compounds from *Moringa oleifera*.

2. Literature Review

2.1. Ensemble Learning Techniques

Ensemble learning is a machine learning technique that combines two or more models to achieve superior performance compared to a single model. Although each algorithm may have limited performance on its own, combining multiple models can improve prediction accuracy and stability [15].

Conceptually, ensemble learning trains multiple learners to solve the same problem and then combines their prediction results. This approach aims to reduce bias and variance while improving the model's generalization ability. Additionally, this technique is effective in addressing various challenges such as imbalanced data, feature selection, and error correction [16].

p-ISSN xxxx-xxxx
e-ISSN xxxx-xxxx

The two most popular ensemble learning techniques are Bagging and Boosting, two algorithms frequently used in ensemble methods [17]. Bagging trains multiple base learners in parallel, and the final result is determined by majority voting, whereas Boosting trains models incrementally by correcting the errors of the previous model [17]. This study focuses on the Boosting technique using Decision Trees as the base estimator.

2.2 Boosting Techniques

Boosting is an ensemble learning method in machine learning designed to improve the performance of predictive models. This method works by combining multiple weak learners or classifiers into a single, stronger, and more accurate model. The core concept of this technique is to focus on data that is difficult to classify, thereby reducing prediction errors and improving overall accuracy [18].

The Boosting algorithm works iteratively by assigning different weights to the distribution of the training set data in each iteration. In each iteration, Boosting increases the weights of misclassified examples and decreases the weights of correctly classified examples. This process effectively alters the distribution of the training set data, allowing the model to focus on the more challenging cases [19].

One important application of Boosting is in addressing class imbalance issues. Boosting methods, such as selective costing ensemble boosting, have proven effective for this problem. This approach improves the identification of hard-to-detect minority classes while maintaining the classification performance for the majority class [19]. The working mechanism of the Boosting technique is illustrated in Figure 2.6 below:

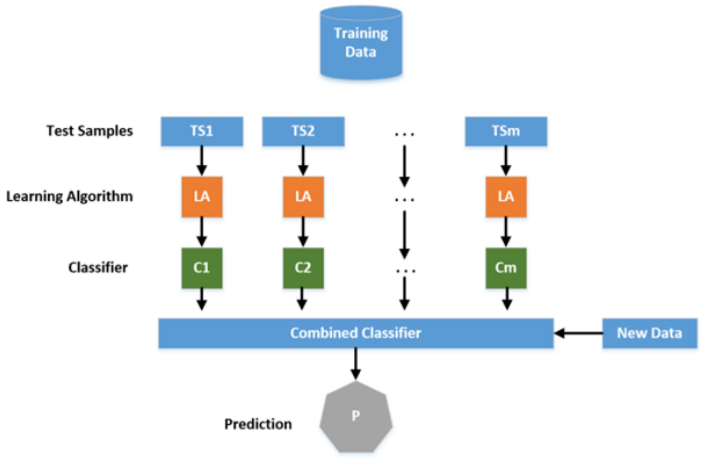


Figure 1. The mechanism of the boosting technique

2.3 Decision Tree Algorithm

In this study, the Decision Tree is used as the base estimator in the Boosting technique. A Decision Tree is a tree-structured model consisting of nodes that represent decisions and branches that describe the consequences of each decision made. Decision Tree is a predictive model categorized as Supervised Learning, which requires a training dataset to construct decision rules [20].

In a Decision Tree, several terms are used to describe its structure. A Decision Node refers to a feature used to make a decision, where the topmost node is called the Root Node. In addition, there is a Leaf Node, which represents the outcome of a decision, and a Subtree, which is a part or branch of the tree [20]. For a clearer understanding, the visualization of a Decision Tree can be seen in Figure 2.

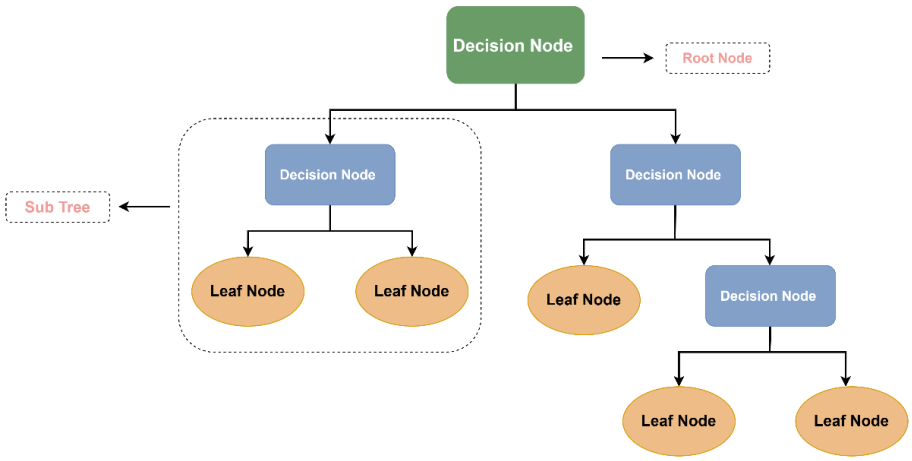


Figure 2. Illustration of the decision tree algorithm

Basically, the ID3 algorithm performs data splitting into groups based on the available attributes in the dataset by measuring a value called entropy. Entropy is defined as a measure of the randomness or impurity of a data group. The splitting process is carried out hierarchically from top to bottom [20].

The formula for Entropy is defined as follows:

$$Entropy(S) = -Entropy(t) = -\sum_{i=1}^{n} p_i \, log_2(p_i) \quad (1)$$

Description:

S : Set of cases

n : Number of partitions

p_i : Number of cases in the i-th partition

After the attribute selection process, the next step is processing using the Decision Tree algorithm in machine learning. The Decision Tree determines the root node by starting from the attribute with the highest weight, followed by the splitting process. The gain value is calculated after computing entropy, where k represents the number of classes and p_i represents the proportion of values belonging to class i at a certain level. Gain refers to the

information obtained from the modification of entropy over a set of data through dataset partitioning. Each attribute has entropy as its basis, and scientifically, the branches of the tree are expanded until the entire tree structure is formed.This process utilizes Information Gain to assign weights to each attribute, ensuring that only significant attributes are included in the machine learning process within the tree [20]. The formula for calculating Information Gain is defined as follows:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i) \quad (2)$$

Description:
S : Set of cases
n : Number of attribute partitions
$|S_i|$: Number of cases in the i-th partition
$|S|$: Total number of cases in S
A : Attribute

Based on Equations (1) and (2), the available data can be processed using these formulas to construct a decision tree using the Decision Tree algorithm.

2.4 Machine Learning Model Evaluation

Before calculating the evaluation metrics, the model is assessed using several classification metrics, namely accuracy, precision, recall, F1-score, specificity, G-Mean, and ROC-AUC. The model predictions are first summarized in a confusion matrix, which compares the actual class labels with the predicted results [21], [22].

Tabel 1. Confusion matrix

		Predicted Values	
		Positive	*Negative*
Actual Values	*Positive*	**TP (*True Positive*)**	**FP (*False Positive*)**
	Negative	**FN (*False Negative*)**	**TN (*True Negative*)**

In binary classification, the confusion matrix consists of four main components that represent different prediction outcomes. These components are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which serve as the basis for calculating the evaluation metrics. Based on these components, seven evaluation metrics are computed as follows.

a. Accuracy

Accuracy measures how often the model correctly predicts the class labels. However, in imbalanced datasets, this metric tends to be biased toward the majority class [23].

p-ISSN xxxx-xxxx
e-ISSN xxxx-xxxx

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

b. Precision

Precision measures how accurate the model is in predicting positive instances [23]. It does not take true negative values into account, and therefore is not affected by class imbalance (imbalanced classes) (Khan dkk., 2024).

$$\Pr = \frac{TP}{TP + FP} \tag{4}$$

c. Recall (Sensitivity)

Recall measures how well the model identifies actual positive instances [23]. Similar to precision, recall does not consider negative classes, and thus is not influenced by imbalanced class distributions [24].

$$\Pr = \frac{TP}{TP + FP} \tag{5}$$

d. *F1-Score*

The F1-Score is the harmonic mean of precision and recall, combining both into a single metric. It evaluates two types of errors simultaneously, namely False Positives and False Negatives [24].

$$F1 - Score = 2 \times \left(\frac{\Pr \times Rc}{\Pr + Rc} \right) \tag{6}$$

e. AUC/ROC

The Receiver Operating Characteristic (ROC) curve is a model evaluation method for classification tasks that illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different threshold values. The ROC curve is constructed based on the predicted probabilities generated by the model, where these probability values are compared against various thresholds to determine positive and negative class assignments.

At each threshold, corresponding TPR and FPR values are obtained and used to form the ROC curve. The TPR (also known as sensitivity) represents the proportion of actual positive instances that are correctly classified, while the FPR represents the proportion of actual negative instances that are incorrectly classified as positive.

Mathematically, TPR and FPR are defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

The ROC curve is then obtained by plotting TPR against FPR across all threshold variations. The Area Under the Curve (AUC) represents the area under the ROC curve and

is used to measure the model's ability to distinguish between positive and negative classes. The AUC value ranges from 0 to 1, where a value closer to 1 indicates excellent model performance, while a value of 0.5 indicates performance equivalent to random classification. Mathematically, AUC can be expressed as:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \tag{9}$$

In this study, the ROC-AUC value is calculated using the predicted probabilities generated by the classification model, allowing evaluation of model performance across different threshold settings. This approach provides a more comprehensive assessment of the model's ability to discriminate between classes, particularly in imbalanced datasets [25].

f. Specificity

Specificity (SPEC) measures the model's ability to correctly identify negative cases. This metric is the counterpart of recall, but it focuses on the negative class [26].

$$SPEC = \frac{TN}{TN + FP} \tag{10}$$

g. G-*Mean*

G-Mean (Geometric Mean) is a comprehensive measure used to evaluate classification performance on imbalanced datasets. It is calculated as the geometric mean of sensitifity and specificity. The G-Mean value reaches one when all observations are correctly classified [27]. The formula for G-Mean is defined as follows:

$$G - mean = \sqrt{Sensitivity \times Specificity} \tag{11}$$

3. Methods

This study adopts a quantitative approach using a classification analysis method to predict the potential antibiotic activity of compounds found in the Moringa oleifera plant. The AdaBoost boosting technique with a Decision Tree as the base estimator is applied to address the issue of class imbalance in the training data.

3.1 Data

This study utilizes two types of secondary data obtained from different databases. Data on antibiotic and non-antibiotic compounds were collected from DrugBank (https://go.drugbank.com/), consisting of 3,907 compounds with 32 attributes, including the class label. Meanwhile, data on Moringa oleifera compounds were obtained from PubChem (https://pubchem.ncbi.nlm.nih.gov/) and pkCSM (https://biosig.lab.uq.edu.au/pkcsm/), consisting of 38 compounds with 31 attributes. The Moringa oleifera dataset is used as external prediction data to identify the potential antibiotic activity of compounds based on the best-performing model that has been trained and validated using labeled compound data.

Tabel 2. Data sources

Database	Data	Description
DrugBank	Antibiotic and non-antibiotic drug compounds	A database containing information on approved drugs that have undergone multiple stages of clinical testing. In DrugBank, drugs are classified into two categories: compounds with antibiotic activity and compounds without antibiotic activity.
PubChem	Moringa oleifera compounds	PubChem is a chemical compound database containing information on various chemical substances, including drugs. This dataset is used for prediction after the model has been validated.

3.2 Preprocessing

The preprocessing stage begins with exploratory data analysis and visualization to identify the distribution and relationships among variables. Next, a data cleaning step is performed to ensure high data quality. Based on the inspection results, three compounds were found to contain missing values, reducing the total dataset used in the modeling stage to 3,904 compounds. These missing values were handled by removing the corresponding rows. This is followed by the transformation of binary categorical variables ("Yes/No") into numerical values (1/0). The dataset is then split into training and validation sets using a 70:30 ratio.

3.3 Application of Boosting-Decision Tree

In this study, a Decision Tree is used as the base estimator within the AdaBoost algorithm. AdaBoost builds models sequentially by assigning higher weights to misclassified samples, allowing subsequent models to focus more on correcting previous errors. The model implementation is conducted using Python through Google Colab. Hyperparameter optimization is performed by testing several parameter combinations, including the number of estimators (n_estimators), learning rate, and Decision Tree parameters (max_depth, min_samples_split, and min_samples_leaf), resulting in the best configuration presented in Table 3.

Tabel 3. Optimal hyperparameters of the model

Parameter	Optimal Value
n_estimators	50
learning_rate	0,7
max_depth	3
min_samples_split	10
min_samples_leaf	5

The hyperparameter values obtained were used as the final configuration in the model training process. The max_depth parameter was applied to reduce model complexity and

p-ISSN xxxx-xxxx
e-ISSN xxxx-xxxx

minimize the risk of overfitting. Model evaluation was conducted using stratified k-fold cross-validation with 5 folds to preserve the class distribution in each fold, ensuring that the evaluation results are more representative and reliable.

3.4 Model Evaluation

The performance of the model was evaluated using seven metrics, namely Accuracy, Precision, Recall, F1-Score, ROC AUC, Specificity, and G-Mean. In imbalanced data conditions, the primary evaluation metrics used as the main reference are Recall, F1-Score, Specificity, and G-Mean, as these metrics are more representative in assessing the model's ability to correctly identify the minority class.

4. Results and Discussion

4.1 Data Description and Classification

Based on a dataset consisting of 3,904 compounds, several experiments were conducted to determine the appropriate proportion of training and validation sets. From these experiments, a 70:30 split ratio was selected for this study, with 70% of the data used as the training set and 30% as the validation set. The data splitting scheme is illustrated in Figure 3 below.

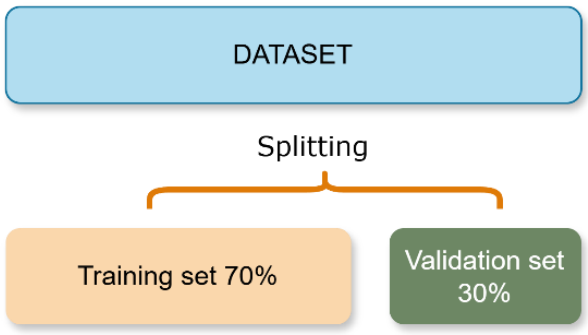


Figure 3. Illustration of training and validation set split

Based on the 70:30 data split, the training set accounts for 70% of the total dataset, while the validation set accounts for the remaining 30%. The class distribution within each subset maintains the proportion between antibiotic and non-antibiotic compounds. The visualization of class distribution in the training and validation sets is shown in the bar plot in Figure 4 below.

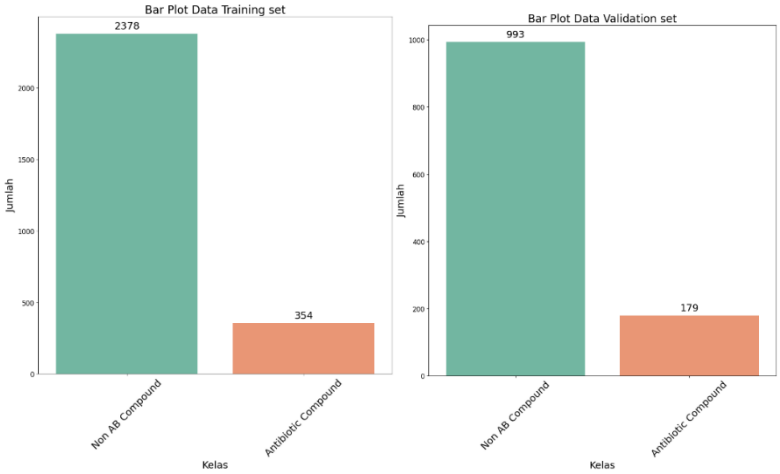


Figure 4. Class Distribution in Training and Validation Sets

Following the data splitting process, the training set consists of 2,732 compounds (354 antibiotic and 2,378 non-antibiotic), while the validation set consists of 1,172 compounds (179 antibiotic and 993 non-antibiotic). The consistent class imbalance across both subsets indicates that the data splitting was performed in a proportional and representative manner.

4.2 Results Without Boosting Implementation

The prediction results without applying boosting were evaluated using a confusion matrix and several evaluation metrics to assess the model's performance in classifying the Antibiotic Compound and Non-antibiotic Compound classes. The results are presented in Tables 4 and 5.

Table 4. Confusion matrix of the decision tree model on training and validation sets

Dataset	Actual Class	Predicted		Total Actual
		Antibiotic Compound	*Non-antibiotic Compound*	
Training Set	*Antibiotic Compound*	225	129	**354**
	Non-antibiotic Compound	27	2351	**2378**
	Total Prediction	**252**	**2480**	**2732**
Validation Set	*Decision Tree*	109	70	**179**
	Non-antibiotic Compound	19	974	**993**
	Total Prediction	**128**	**1044**	**1172**

Based on the confusion matrix in Table 4, the Decision Tree model without boosting correctly classified 225 out of 354 Antibiotic Compound samples and 2,351 out of 2,378 Non-antibiotic Compound samples in the training set. Meanwhile, in the validation set, the

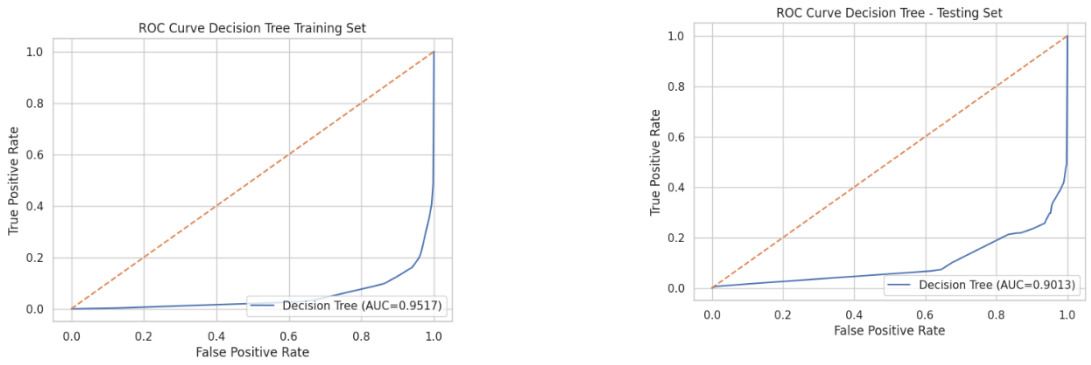
p-ISSN xxxx-xxxx
e-ISSN xxxx-xxxx

model correctly classified 109 out of 179 Antibiotic Compound samples and 974 out of 993 Non-antibiotic Compound samples.These results indicate that the model performs better in identifying the majority class (Non-antibiotic Compound) compared to the minority class (Antibiotic Compound), demonstrating a bias toward the dominant class in the imbalanced datase.

Table 5. Evaluation metrics of the decision tree model on training and validation Sets

Evaluation Metrics	Training Set	Validation Set
Accuracy	0,9429	0,9241
Precision	0,8929	0,8516
Recall	0,6356	0,6089
F1 Score	0,7426	0,7101
ROC AUC	0,9517	0,9013
Specificity	0,9886	0,9809
G-Mean	0,7927	0,7728
CV-Mean	0,9374	0,9223
CV-Std	0,0090	0,0050

Based on Table 5, the Decision Tree model without boosting achieved an accuracy of 0.9429 on the training set and 0.9241 on the validation set. However, the recall values of 0.6356 and 0.6089 indicate that the model is still not optimal in identifying the Antibiotic Compound class. The high specificity values (0.9886 on the training set and 0.9809 on the validation set) show that the model performs better in identifying the majority class (Non-antibiotic Compound). Meanwhile, the G-Mean values of 0.7927 and 0.7728 indicate that the balance between class-wise performance still needs improvement. The results of Stratified 5-Fold Cross-Validation show that the model performance is stable, with mean values of 0.9374 ± 0.0090 for the training set and 0.9223 ± 0.0050 for the validation set.



(a). Training set (b). Validation set

Figure 5. ROC/AUC curve of the decision tree on training and validation sets

Based on Figure 5, the ROC curve of the Decision Tree model without boosting shows an AUC value of 0.9517 on the training set and 0.9013 on the validation set. These

values indicate that the model has good classification capability in distinguishing between the Antibiotic Compound and Non-antibiotic Compound classes.

4.3 Results with Boosting Implementation

The prediction results with the application of the AdaBoost-Decision Tree boosting technique were evaluated using a confusion matrix and several evaluation metrics to assess the model's performance in classifying the Antibiotic Compound and Non-antibiotic Compound classes. The results are presented in Tables 6 and 7.

Table 6. Confusion matrix of the AdaBoost-Decision Tree Model on training and validation sets

Dataset	Actual	Predicted		Total Actual
		Antibiotic Compound	*Non-antibiotic Compound*	
Training Set	*Antibiotic Compound*	281	73	**354**
	Non-antibiotic Compound	18	2360	**2378**
	Total Predcted	**299**	**2433**	**2732**
Validation Set	*Decision Tree*	135	44	**179**
	Non-antibiotic Compound	11	982	**993**
	Total Predicted	**146**	**1026**	**1172**

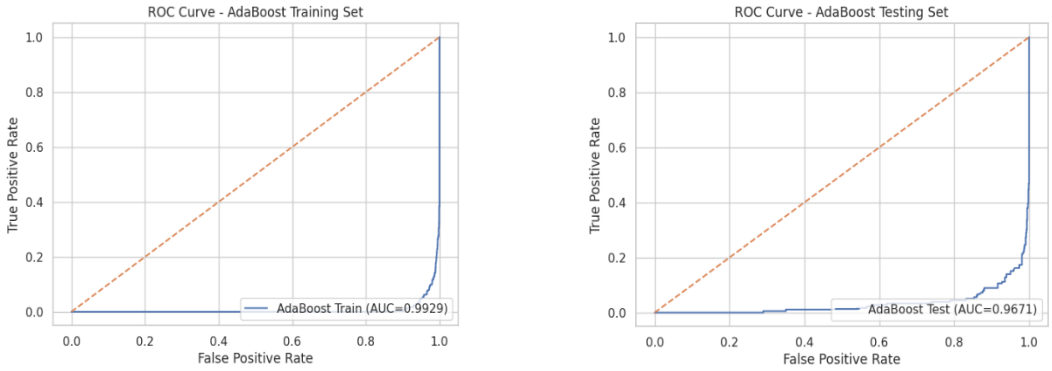
Based on the confusion matrix in Table 6, the AdaBoost-Decision Tree model correctly classified 281 out of 354 Antibiotic Compound samples and 2,360 out of 2,378 Non-antibiotic Compound samples in the training set. Meanwhile, in the validation set, the model correctly classified 135 out of 179 Antibiotic Compound samples and 982 out of 993 Non-antibiotic Compound samples. These results indicate that the application of boosting improves the model's ability to correctly identify the Antibiotic Compound class compared to the Decision Tree model without boosting.

Table 7. Evaluation Metrics of the AdaBoost-Decision Tree Model on training and validation sets

Evaluation Metric	AdaBoost-Decision Tree Training Set	AdaBoost-Decision Tree Validation Set
Accuracy	0,9667	0,9531
Precision	0,9398	0,9247
Recall	0,7938	0,7542
F1 Score	0,8606	0,8308
ROC AUC	0,9929	0,9671

Evaluation Metric	AdaBoost-Decision Tree Training Set	AdaBoost-Decision Tree Validation Set
Specificity	0,9924	0,9889
G-Mean	0,8876	0,8636
CV-Mean	0,9513	0,9556
CV-Std	0,0062	0,0069

Based on Table 6, the AdaBoost-Decision Tree model achieved an accuracy of 0.9667 on the training set and 0.9531 on the validation set. The recall values of 0.7938 and 0.7542 indicate that the application of boosting improves the model's ability to identify the Antibiotic Compound class compared to the model without boosting. The high specificity values of 0.9924 on the training set and 0.9889 on the validation set show that the model remains highly effective in recognizing the Non-antibiotic Compound class. The improvement in G-Mean values to 0.8876 on the training set and 0.8636 on the validation set indicates a better balance in class-wise performance. Furthermore, the Stratified 5-Fold Cross-Validation results of 0.9513 ± 0.0062 and 0.9556 ± 0.0069 demonstrate that the model maintains stable performance across different data folds.



(a). Training Set (b). Validation Set

Figure 7. ROC/AUC Curve of the AdaBoost-Decision Tree Model on training and validation sets

Figure 7 shows the ROC/AUC curve of the AdaBoost-Decision Tree model on the validation set, with an AUC value of 0.9671. This value indicates that the model has excellent classification ability in distinguishing between the Antibiotic Compound and Non-antibiotic Compound classes on the validation data.

4.4. Prediction Results of Compounds in Moringa oleifera

At this stage, the prediction results of compound classes in Moringa oleifera are presented using both the Decision Tree model and the Decision Tree model with the boosting technique. The prediction is conducted to identify whether each compound belongs to the Antibiotic Compound or Non-antibiotic Compound category based on the previously evaluated models. The prediction results of both models are presented in Table 12 below.

Table 12. Predicted results of antibiotic compounds in Moringa oleifera

CID	*Compound Name*	Algorithm	
		Decision Tree	**Decision Tree Boosting**
5281672	*Myricetin*	*Non-Antibiotic*	*Antibiotic*
10068657	*Niazicin A*	*Antibiotic*	*Antibiotic*
44198238	*[(2S,3R,4S,5S,6R)-3,4,5-trihydroxy-6-(hydroxymethyl)oxan-2-yl] (1E)-N-sulfooxy-2-[4-[(2S,3R,4R,5R,6S)-3,4,5-trihydroxy-6-methyloxan-2-yl]oxyphenyl]ethanimidothioate*	*Non-Antibiotic*	*Antibiotic*
89812211	*Niazinin A*	*Non-Antibiotic*	*Antibiotic*

Based on the prediction results obtained using the Decision Tree and AdaBoost-Decision Tree models, out of 38 compounds from the Moringa oleifera plant, several compounds were predicted as antibiotic compounds by the AdaBoost-Decision Tree model. However, only Niazicin A (CID 10068657) was consistently classified as an antibiotic compound by both models. This finding indicates that Niazicin A is one of the most promising candidate compounds with potential antibiotic activity based on a machine learning predictive approach.

This result is in line with previous studies reporting that Moringa oleifera contains various bioactive compounds with antimicrobial activity. Niazicin A has significant potential for drug development due to its strong antibacterial properties and important implications in addressing antibiotic resistance. This further supports the potential of Niazicin A as a valuable innovative antibiotic [28].

Myricetin, which is also found in Moringa oleifera, shows strong indications of antimicrobial activity, supported by literature reporting its ability to inhibit bacterial growth through interactions with cell membranes and inhibition of essential enzymes [29].

Meanwhile, the newly identified compound trihydroxy-6-(hydroxymethyl)oxan-2-yl] (1E)-N-sulfooxy-2-[4-[(2S,3R,4R,5R,6S)-3,4,5-trihydroxy-6-methyloxan-2-yl]oxyphenyl]ethanimidothioate requires further laboratory testing to validate its antibiotic activity through experimental studies. Therefore, the prediction results of this study can serve as an initial screening step for identifying potential antibiotic compounds from Moringa oleifera.

5. Conclusion

Based on the results of this study, the application of AdaBoost-Decision Tree significantly improves model performance compared to the Decision Tree model without ensemble implementation. This improvement is particularly evident in the model's ability to detect the minority class (antibiotic compound) in an imbalanced dataset. The weighting mechanism in AdaBoost enables the model to focus more on difficult-to-classify samples, resulting in a more balanced classification performance between the antibiotic compound and non-antibiotic compound classes.

The prediction results of compounds from Moringa oleifera indicate that one compound was consistently classified as an antibiotic compound by both models, namely Niazicin A (CID 10068657). This finding demonstrates that the AdaBoost-Decision Tree model can be used as a predictive approach for the initial screening process of candidate compounds with potential antibiotic activity.

Therefore, the AdaBoost-based Decision Tree approach shows potential as a method for addressing class imbalance problems and supporting the exploration of antibiotic candidate compounds from natural sources. However, the classification results in this study are still predictive and based on molecular characteristics; therefore, further experimental validation is required to confirm the biological activity of the predicted compounds.

References

[1] A. S. Syam, N. O. Ramadhani, I. Inayah, and E. Nuraeni, "Kajian literatur: Inventarisasi dan penggunaan tumbuhan obat tradisional di Indonesia," Jurnal Biosense, vol. 9, no. 1, pp. 109–122, 2026, doi: 10.36526/biosense.v9i1.6814.

[2] W. Wafizo, I. Artanti, A. D. Lestari, and A. Fadhila, "Review tanaman obat tradisional Indonesia: Kandungan fitokimia dan potensi farmakologis," *Jurnal Kesehatan Unggul Gemilang*, vol. 10, no. 2, pp. 41–47, 2026.

[3] Yulia, M. Idris, and Rahmadina, "Skrining fitokimia dan penentuan kadar flavonoid daun kelor (*Moringa oleifera* L.) Desa Dolok Sinumbah dan Raja Maligas Kecamatan Hutabayu Raja," *KLOROFIL: Jurnal Ilmu Biologi dan Terapan*, vol. 6, no. 1, pp. 49–56, 2022, doi: 10.30821/kfl:jibt.v6i1.11678.

[4] V. Arianti, "Gambaran tingkat pengetahuan penggunaan obat tradisional mahasiswa farmasi Politeknik Kesehatan Hermina," *Indonesian Journal of Health Science*, vol. 2, no. 2, pp. 73–76, 2022, doi: 10.54957/ijhs.v2i2.296.

[5] Y. Yang, M. G. C. Kessler, M. R. Marchán-Rivadeneira, and Y. Han, "Combating antimicrobial resistance in the post-genomic era: Rapid antibiotic discovery," *Molecules*, vol. 28, no. 10, Art. no. 4183, 2023, doi: 10.3390/molecules28104183.

[6] M. Jiang *et al.*, "Drug-target affinity prediction using graph neural network and contact maps," *RSC Advances*, vol. 10, no. 35, pp. 20701-20712, 2020, doi: 10.1039/d0ra02297g.

[7] N. Istiana and A. Mustafiril, "Perbandingan metode klasifikasi pada data dengan imbalance class dan missing value," *Jurnal Informatika*, vol. 10, no. 2, pp. 101-108, 2023, doi: 10.31294/inf.v10i2.15540.

[8] L. Liu, X. Wu, S. Li, Y. Li, S. Tan, and Y. Bai, "Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection," *BMC*

Medical Informatics and Decision Making, vol. 22, no. 1, pp. 1-16, 2022, doi: 10.1186/s12911-022-01821-w.

[9] N. D. Saputri, K. Khalid, and D. Rolliawati, "Komparasi penerapan metode bagging dan Adaboost pada algoritma C4.5 untuk prediksi penyakit stroke," *SISTEMASI: Jurnal Sistem Informasi*, vol. 11, no. 3, pp. 567–577, 2022.

[10] A. Bisri, "Penerapan Adaboost untuk Penyelesaian ketidakseimbangan kelas pada penentuan kelulusan mahasiswa dengan metode Decision Tree," *Journal of Intelligent Systems*, vol. 1, no. 1, 2015.

[11] Y. Wang, S. Zhang, F. Li, Y. Zhou, Y. Zhang, Z. Wang, R. Zhang, J. Zhu, Y. Ren, Y. Tan, C. Qin, Y. Li, X. Li, Y. Chen, and F. Zhu, "Therapeutic target database 2020: Enriched resource for facilitating research and early development of targeted therapeutics," *Nucleic Acids Research*, vol. 48, no. D1, pp. D1031–D1041, 2020, doi: 10.1093/nar/gkz981.

[12] A. K. Nasution, S. H. Wijaya, P. Gao, R. M. Islam, M. Huang, N. Ono, S. Kanaya, and M. Altaf-Ul-Amin, "Prediction of potential natural antibiotics plants based on jamu formula using random forest classifier," *Antibiotics*, vol. 11, no. 9, Art. no. 1199, 2022, doi: 10.3390/antibiotics11091199.

[13] J. Yan, Y. Xu, Q. Cheng, *et al*., "LightGBM: Accelerated genomically designed crop breeding through ensemble learning," *Genome Biology*, vol. 22, Art. no. 271, 2021, doi: 10.1186/s13059-021-02492-y.

[14] O. J. Harmaja, I. Prasetia, Y. V. Hutagalung, and H. A. Sirait, "Comparison of ensemble learning algorithm in classifying early diagnostic of Diabetes," *Jurnal Sistem Informasi dan Ilmu Komputer Prima(JUSIKOM PRIMA)*, vol. 7, no. 1, pp. 218-231, 2023, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v7i1.4054.

[15] F. Churniansyah and D. W. Utomo, "Teknik bagging pada ensemble learning untuk kategorisasi produk E-Commerce," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 10, no. 1, pp. 92-99, 2024, doi: 10.25077/teknosi.v10i1.2024.92-99.

[16] J. Melvin and A. Soraya, "Analisis perbandingan algoritma XGBoost dan algoritma Random Forest Ensemble Learning pada klasifikasi keputusan kredit," vol. 2, no. 2, 2023.

[17] Y. Pristyanto and A. A. Zein, "Model balanced bagging berbasis decision tree pada dataset imbalanced class," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 9-15, 2023, doi: 10.32736/sisfokom.v12i1.1399.

[18] Jeffry, S. Usman, and F. Aziz, "Analisis perilaku pelanggan menggunakan metode ensemble logistic regression," *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia (UNPRI) Medan*, vol. 6, no. 2, pp. 90–97, 2023.

[19] A. R. Arrahimi, M. K. Ihsan, D. Kartini, M. R. Faisal, and F. Indriani, "Teknik bagging dan boosting pada algoritma CART untuk klasifikasi masa studi mahasiswa," *Jurnal Sains dan Informatika*, vol. 5, no. 1, pp. 21-30, 2019, doi: 10.34128/jsi.v5i1.171.

[20] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, and U. Firdaus, "Implementasi algoritma decision tree untuk klasifikasi pelanggan aktif atau tidak aktif pada data bank," *Karimah Tauhid*, vol. 3, no. 2, pp. 1860-1874, 2024, doi: 10.30997/karimahtauhid.v3i2.11952.

[21] K. Handayani and E. Erni, "Penerapan light gradient boosting dalam prediksi rasio klik tayang," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 13-18, 2023, doi: 10.36040/jati.v7i1.6010.

[22] K. Kristiawan and A. Widjaja, "Perbandingan algoritma machine learning dalam

menilai sebuah lokasi toko ritel," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 35-46, 2021, doi: 10.28932/jutisi.v7i1.3182.

[23] B. Sunarko *et al.*, "Penerapan Stacking ensemble learning untuk klasifikasi efek kesehatan akibat pencemaran udara," *Edu Komputika Journal*, vol. 10, no. 1, pp. 55-63, 2023, doi: 10.15294/edukomputika.v10i1.72080.

[24] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," *Expert Systems with Applications*, vol. 244, no. May 2023, p. 122778, 2024, doi: 10.1016/j.eswa.2023.122778.

[25] A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, B. Wernly, C. Bennett, S. Hawken, O. Magwood, Y. Sheikh, M. McInnes, and A. Holzinger, "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329–341, 2023, doi: 10.1109/TPAMI.2022.3145392.

[26] S. A. Hicks, I. Strümke, V. Thambawita, *et al*., "On evaluation metrics for medical applications of artificial intelligence," *Scientific Reports*, vol. 12, Art. no. 5979, 2022, doi: 10.1038/s41598-022-09954-8.

[27] R. F. W. Pratama, S. W. Purnami, and S. P. Rahayu, "Boosting support vector machines for imbalanced microarray data," *Procedia Computer Science*, vol. 144, pp. 174–183, 2018, doi: 10.1016/j.procs.2018.10.517.

[28] J. Kannan, K. L. Pang, Y. N. Ho, P. H. Hsu, and L. L. Chen, "A comparison of the antioxidant potential and metabolite analysis of marine fungi associated with the Red Algae Pterocladiella capillacea from Northern Taiwan," *Antioxidants*, vol. 13, no. 3, 2024, doi: 10.3390/antiox13030336.

[29] B. R. Cahyani, F. D. Anziani, N. P. Nurizha, and H. Hidayah, "Literature Review article : myricetin activity as an anti-cancer," *Journal Of Social Science Research*, vol. 3, no. 2, pp. 12131-12138, 2023.