Jurnal Elektro
p-ISSN: 1979-9780
e-ISSN: 2746-4288

Jurnal Elektro Vol. 19 No. 1 April 2026
journal homepage: https://ejournal.atmajaya.ac.id/index.php/JTE

EVALUATION OF THE C4.5 DECISION TREE AND RANDOM FOREST CLASSIFICATION ALGORITHMS IN PREDICTING DIABETES

Caecar Mikha Krisnanda[1], Kristiawan Dwi Usmanto*[2], Jonathan Jason Kristanto[3]

[1]Department of Informatics Engineering, Faculty of Science and Technology
Sanata Dharma University, Yogyakarta, Indonesia
[2]Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[3]Department of Electrical Engineering, College of Engineering
Southern Taiwan University of Science and Technology, Tainan, Taiwan

Article Info

Article history:

Received
10-04-2026

Accepted
24-04-2026

Keywords:
C4.5 Decision Tree, Random Forest, Data Balancing, Hyperparameter Tuning, Imbalanced Data, Diabetes Prediction.

Abstract

This study investigates diabetes prediction as a binary classification task using the C4.5 Decision Tree and Random Forest algorithms on the Pima Indians Diabetes dataset. The objective of this study is to compare the performance of both algorithms under three reported experimental settings: without data balancing, with data balancing, and with hyperparameter tuning without balancing. The dataset consists of 768 records, including 500 non-diabetes cases and 268 diabetes cases. The preprocessing stage included data cleaning, Box-Cox transformation, min-max normalization, feature selection, and data splitting into 80% training data and 20% test data. Model performance was evaluated using accuracy, precision, recall, and F1-score through 3-fold, 5-fold, and 9-fold cross-validation. The results show that Random Forest consistently outperformed the C4.5 Decision Tree across all reported settings. Under the non-balancing condition, Random Forest achieved the highest accuracy of 77.82%, while C4.5 achieved 69.65%. After applying data balancing, the performance of both models improved, with Random Forest achieving the best overall reported accuracy of 84.19%, compared with 75.68% for C4.5. Under hyperparameter tuning without balancing, Random Forest achieved 78.18%, while C4.5 achieved 74.18%. These findings indicate that Random Forest is more robust and effective than the C4.5 Decision Tree for diabetes prediction, and that data balancing contributes more significantly to performance improvement than hyperparameter tuning alone.

1. INTRODUCTION

Diabetes is a condition in which the body cannot produce or use insulin effectively, leading to high blood sugar levels that can potentially cause serious complications in vital organs [1]. The main challenge in managing diabetes is early detection and predicting its progression, as many patients only become aware of their condition after serious complications have already developed. From a computer science perspective, one way to address this issue is by utilizing data mining technology. Data mining is the process of analyzing data from various angles and previously unused datasets to gain new insights by identifying hidden patterns in the data and transforming them into useful information [2]. The C4.5 Decision Tree and Random Forest algorithms can serve as effective solutions for diagnosing and predicting diabetes. Therefore, this study employs the C4.5 Decision Tree and Random Forest algorithms to predict diabetes through a binary classification approach.

Previous research on the Pima Indians Diabetes Database has explored diabetes prediction using machine-learning methods and has shown that model performance is influenced by feature relevance and data characteristics [3]. In this context, C4.5 Decision Tree provides an interpretable rule-based classification model by splitting the data according to the most informative attributes, whereas Random Forest combines multiple decision trees to improve robustness and predictive performance. However, a focused comparison between these two algorithms under imbalanced-data handling and hyperparameter tuning settings remains limited. Therefore, this study evaluates diabetes prediction as a binary classification task by comparing the performance of C4.5 Decision Tree and Random Forest under three

*Corresponding author: Kristiawan Dwi Usmanto
Email address: 6022251013@student.its.ac.id

reported settings, namely without data balancing, with data balancing, and with hyperparameter tuning without balancing.

2. LITERATURE REVIEW

Previous studies on diabetes prediction using the Pima Indians Diabetes Database have explored a broad spectrum of machine-learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression, Gaussian Naive Bayes, Decision Tree C4.5, Random Forest, and deep-learning architectures such as LSTM and CNN. In general, the literature indicates that predictive performance is strongly influenced by feature selection, data preprocessing, class balancing, and hyperparameter optimization. Aziz Perdana et al. investigated diabetes prediction using KNN and reported the best accuracy of 83.12% at k = 22 [3]. Their study also highlighted the importance of variables such as glucose, age, and insulin, while suggesting that the diabetes pedigree function was less relevant for the KNN-based prediction setting. These findings emphasize that model quality is closely related to the discriminative value of the selected features.

A comparative study by Abdulazeez Mousa et al. evaluated LSTM, Random Forest, and CNN on the same dataset and found that LSTM achieved the highest accuracy, reaching 85%, whereas Random Forest and CNN also produced competitive results [4]. This result shows that advanced sequential models can capture meaningful patterns in medical data, but conventional ensemble methods remain strong baselines because of their robustness and practicality [4]. A broader comparison was presented by Anant Ram and Gohel, who evaluated KNN, Support Vector Classifier, Logistic Regression, Gaussian Naive Bayes, and Random Forest on the same dataset [5]. Their results showed that Random Forest achieved the best accuracy of 0.8084 when all relevant medical attributes were used. This finding suggests that Random Forest is well suited to handling nonlinear relationships and feature interactions in diabetes-prediction tasks [5].

Other studies have shown that a single decision-tree classifier may be less competitive than alternative classifiers or ensemble-based approaches. Rousyati et al. compared Naive Bayes, SVM, and Decision Tree C4.5 combined with AdaBoost and Bagging, and observed that SVM with Bagging achieved the highest accuracy of 77.47%, while Naive Bayes without ensemble support showed better precision [6]. In another study, Dwi Sri Rahayu et al. compared C4.5, SVM, and linear regression and reported that SVM achieved an accuracy of 82.01%, outperforming C4.5, which obtained 75.324% [7]. Several previous works also stressed the importance of improving the data preparation and model-tuning stages. Wahyu Nugraha and Setiawan demonstrated that Grid Search-based hyperparameter tuning can improve classification performance across multiple machine learning algorithms, even though the best result in their study was obtained by XGBoost and the lowest by Decision Tree [8]. Meanwhile, Dikan Ismafillah et al. showed that the application of SMOTE can improve diabetes prediction performance by reducing the effect of class imbalance and strengthening minority-class representation during training [9].

Based on the reviewed studies, Random Forest generally offers more stable and higher classification performance than a single C4.5 decision tree, while C4.5 remains valuable because of its transparent rule-based structure. Therefore, this study is relevant because it directly compares C4.5 and Random Forest under multiple experimental settings, namely without balancing, with balancing, and with hyperparameter tuning, in order to provide a more comprehensive evaluation of their performance for diabetes prediction. The C4.5 Decision Tree is a supervised classification algorithm that recursively partitions data into homogeneous subsets based on the most informative attributes [2], [7]. The model selects splitting variables by considering entropy, information gain, and gain ratio, so the resulting tree can be interpreted as a set of classification rules. This transparency makes decision trees attractive for healthcare problems; however, a single tree may become unstable when the dataset is noisy or imbalanced.

Random Forest is an ensemble-learning approach that constructs many decision trees from bootstrap samples of the training data and aggregates their predictions through majority voting [4], [5]. By injecting randomness into both data sampling and feature selection, Random Forest tends to reduce variance and improve generalization performance. For this reason, it is often more robust than a single decision tree when faced with complex feature interactions or class imbalance. A practical challenge in diabetes prediction is class imbalance, where the number of non diabetes cases is substantially larger than the number of diabetes cases. In such conditions, a classifier may achieve a high overall accuracy while still failing to identify minority-class cases correctly. SMOTE is a widely used oversampling technique that synthesizes new minority-class samples and helps produce a more balanced training distribution [10], [9]. In parallel, hyperparameter tuning is used to optimize model structure and improve predictive performance. Therefore, combining class balancing and parameter optimization is a relevant strategy for improving classification quality on medical datasets [8], [9].

3. METHODOLOGY

The process began with the acquisition of the Pima Indian Diabetes dataset, followed by a series of preprocessing steps including data cleaning, data transformation, normalization, feature selection, and data balancing [10]. After the data was preprocessed, it was split into training and testing sets. The training data was then used for modeling with K-fold cross-validation, using the C4.5 Decision Tree and Random Forest algorithms. Following the modeling phase, hyperparameter tuning was performed for the C4.5 Decision Tree and Random Forest models [8]. The tuned models were then evaluated using the test data. Finally, the results from the C4.5 Decision Tree and Random Forest models were analyzed, leading to the conclusions of this process.

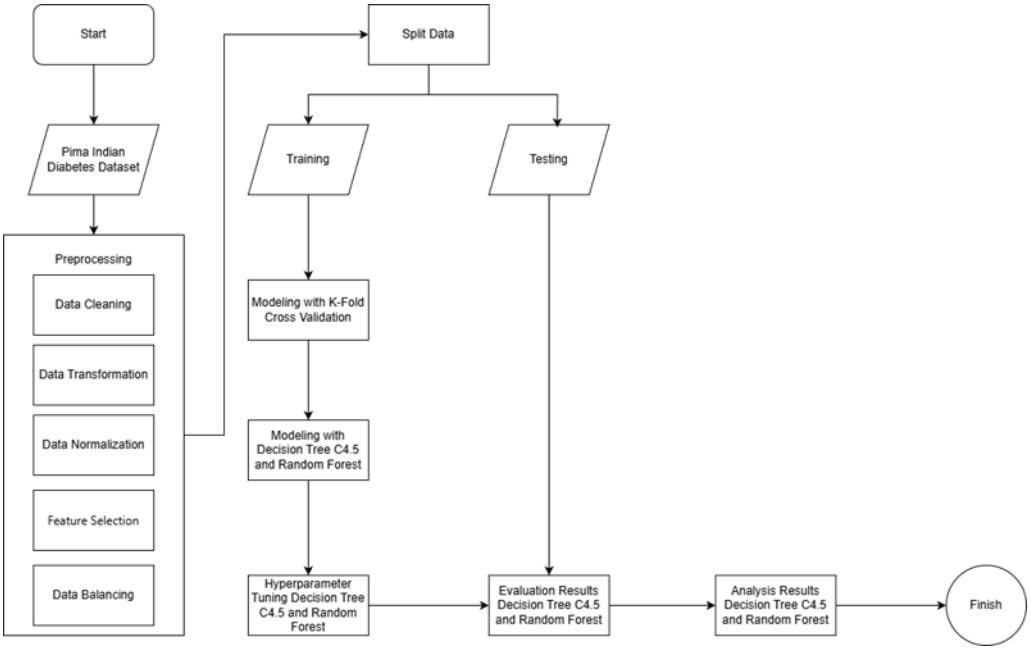

Figure 1. Research Method

3.1 Dataset

In this study, the Pima Indians Diabetes Database (PIDD) was obtained from Kaggle as a publicly available benchmark dataset for diabetes prediction research. The dataset consists of 768 records, including 500 non-diabetes cases and 268 diabetes cases, with 9 attributes including the class label. This dataset was selected because it is widely used in diabetes-related classification studies and contains relevant medical attributes, such as glucose level, blood pressure, body mass index, insulin, and age. Therefore, the dataset is appropriate for evaluating the comparative performance of the C4.5 Decision Tree and Random Forest algorithms in predicting diabetes.

3.2 Preprocessing

During this preprocessing stage, the collected data will be filtered to ensure it has a sound structure. This process involves several steps to ensure the data is ready for use in the next stage. The steps to be performed include data cleaning, transformation, normalization, feature selection, and data balancing. The preprocessing stage is essential because the quality of the input data greatly influences the performance of the classification model. Through proper preprocessing, irrelevant or inconsistent data can be minimized, while the important characteristics of the dataset can be preserved. As a result, the prepared data can improve the effectiveness of the training process and support more accurate prediction results in the classification stage.

3.2.1 Data Cleaning

At this stage, data cleaning is performed to identify invalid data, outliers, and duplicate entries that may affect the quality of the dataset [11]. This process is important because noisy or inconsistent data can reduce the accuracy of the classification model and lead to biased results. In this study, outlier detection is carried out using the Z-score method with a threshold value of 3, based on the following formula:

$$Z = \frac{x - \mu}{\sigma} \qquad (1)$$

Where Z is the Z-score of a value x, x represents an individual value in the dataset, μ is the mean of the entire dataset, and σ is the standard deviation of the entire dataset.

By applying this method, data points with extreme deviations from the mean can be identified and handled appropriately before further analysis. In addition, the removal of duplicate records and the correction of invalid values help ensure that the dataset is more reliable and representative of the actual data distribution. As a result, the cleaned dataset provides a stronger foundation for the subsequent preprocessing and classification stages.

3.2.2 Data Transformation

The Box-Cox transformation is a commonly used method in machine learning to make the data distribution more normal. Using the Box-Cox transformation can reduce skewness in the data, which can affect model performance [12]. Transforming the data before normalization is important for making the data distribution more even. In this dataset, some features have vastly different scales and ranges of values, which can affect the performance of the machine learning model. Therefore, the transformation stage is carried out to stabilize variance and improve the distributional characteristics of the features prior to model training. A more normalized data distribution allows the classification algorithms to capture relationships between variables more effectively and reduces the influence of highly skewed attributes. Consequently, this transformation step contributes to improving the overall quality of the input data and supports better predictive performance

3.2.3 Normalization

After the transformation process, the data in the diabetes study were normalized using the min-max method [13]. Min-max normalization is the process of transforming the values in a dataset so that their range falls between 0 and 1. This technique is useful for ensuring that all features contribute proportionally during the training process, especially when the dataset contains variables with different measurement scales. The normalization stage is particularly important because features with larger numerical ranges may dominate the learning process and negatively affect the performance of the model [13]. By scaling all attributes to the same range, the influence of each feature becomes more balanced, allowing the algorithms to process the data more consistently. As a result, normalization helps improve model stability, accelerates convergence during training, and supports more reliable classification outcomes.

3.2.4 Data Split

The data were split into 80% training data and 20% test data because this ratio provides sufficient data for both effective model training and reliable performance evaluation [14]. A larger training set helps the model learn data patterns more effectively, while a sufficiently large test set provides a more accurate assessment of the model's ability to predict unseen data. This ratio is also widely used in machine-learning studies because it offers a good balance between model development and performance assessment [14]. In this study, the training set was used for model development, including K-fold cross-validation and hyperparameter tuning using GridSearchCV, while the test set was reserved exclusively for final evaluation. A separate validation set was not used because model selection was carried out within the training set through cross-validation. This procedure was intended to avoid information leakage from the test set and to provide a fair final assessment of the comparative performance of the C4.5 Decision Tree and Random Forest algorithms.

3.2.5 Feature Selection

In the feature selection stage, information gain is used to select features based on the highest scores [15]. The following is a graph of the information gain scores:

```
                    Feature  Information Gain Score
0               Pregnancies                0.011160
1                   Glucose                0.124470
2             BloodPressure                0.009498
3             SkinThickness                0.007862
4                   Insulin                0.078215
5                       BMI                0.094231
6  DiabetesPedigreeFunction                0.000000
7                       Age                0.034020
Selected features: ['Pregnancies', 'Glucose', 'BloodPressure', 'Insulin', 'BMI', 'Age']
```

Figure 2. Information Gain Results

The figure above shows the results of the information gain measurement, which indicates how much information about the target class is gained from each feature. The following table presents the accuracy results of various models using the selected features, both with and without feature balancing:

Table 1. Feature Selection Results

Feature Selection	Without Balancing DC	Without Balancing RF	With Balancing DC	With Balancing RF
Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age	70.29%	72.46%	63.77%	74.64%
Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Age	68.84%	70.29%	63.04%	70.29%
Pregnancies, Glucose, Blood Pressure, Insulin, BMI, Age	**71.74%**	**71.74%**	**70.29%**	**73.91%**
Pregnancies, Glucose, Insulin, BMI, Age	69.57%	73.19%	61.59%	73.91%
Glucose, Insulin, BMI, Age	66.67%	70.29%	70.29%	72.46%
Glucose, Insulin, BMI	66.67%	66.67%	70.29%	67.39%
Glucose, BMI	63.04%	70.29%	68.84%	67.39%
Glucose	62.32%	62.32%	63.04%	62.32%

3.2.6 Data Balancing SMOTE

In a dataset containing a total of 768 data points, there are 268 diabetes cases and 500 non-diabetes cases. This indicates an imbalance between the diabetes and non-diabetes classes. To address this, the SMOTE (Synthetic Minority Over-sampling Technique) is used, which works by adding synthetic samples to the minority class (diabetes) to balance its size with that of the majority class (non-diabetes) [10].

In this study, three experimental settings were used to evaluate the performance of the C4.5 Decision Tree and Random Forest algorithms for diabetes prediction. The first setting used the original imbalanced dataset without any balancing treatment. The second setting applied SMOTE to balance the minority class before model training. The third setting applied hyperparameter tuning without balancing using GridSearchCV to optimize model performance. These three settings were designed to examine how class balancing and parameter optimization influence the classification performance of both algorithms.

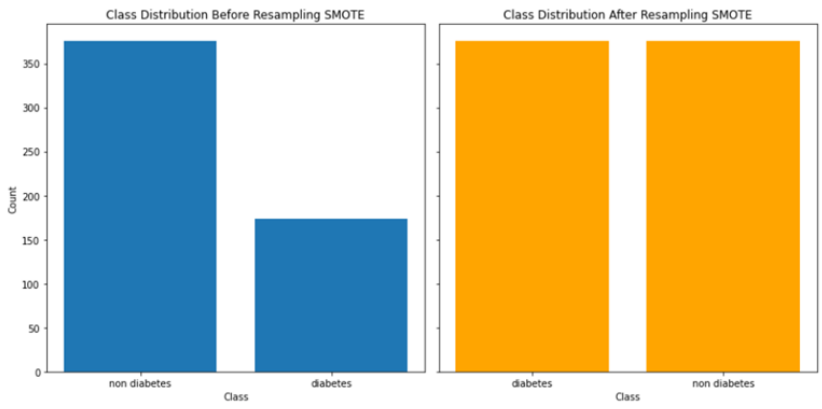

Figure 3. Class Distribution Before and After SMOTE

3.3 C4.5 Decision Tree Modeling with and without Balancing

C4.5 decision tree modeling in this study was conducted using two approaches, namely with balancing and without balancing. In the approach without balancing, the model was trained using the original dataset, which may contain an unequal class distribution between diabetes and non-diabetes cases. Meanwhile, in the balanced approach, the class distribution was adjusted before the modeling process in order to reduce bias toward the majority class. The use of these two approaches allows this study to compare the effect of class balancing on the performance of the C4.5 classification model.

The construction of the C4.5 decision tree begins with data preparation, followed by the calculation of entropy, information gain, split information, and gain ratio for the dataset and for each attribute used in the classification process [7]. These measures are used to determine the most appropriate attribute for splitting the data at each node of the tree. The selected attribute is the one that is considered most effective in reducing uncertainty while also maintaining a reasonable distribution of data among the generated partitions. Entropy is used to measure the level of impurity or disorder in a dataset. The entropy value is calculated using the following formula [16]:

$$Entropy(S) = \sum_{i=1}^{n} -\pi \cdot log_2 \pi \qquad (2)$$

In (2), S denotes the set of cases, n represents the number of partitions in S, and π refers to the proportion of cases contained in partition i. The notation $\sum_{i=1}^{n} =$ indicates summation from i = 1 to n, while $log_2 \pi$ denotes the base-2 logarithm of the probability of partition i. A higher entropy value indicates that the data are more heterogeneous, whereas a lower entropy value indicates that the data are more homogeneous.

After calculating entropy, the next step is to compute information gain, which is used to evaluate how much uncertainty can be reduced by splitting the dataset according to a particular attribute. The formula for information gain is given as follows [16]:

$$Gain\ (S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_I) \qquad (3)$$

In (3), S denotes the set of cases, A represents the attribute being evaluated, and n is the number of partitions generated by attribute A. The term $|S_i|$ refers to the number of cases in the i-th partition, while $|S|$ represents the total number of cases in S. Information gain measures the reduction in entropy achieved after splitting the data using attribute A. An attribute with a higher information gain is considered more effective in separating the data into distinct classes [17]. However, information gain alone may favor attributes with many distinct values. Therefore, the C4.5 algorithm also calculates split information, which measures how broadly the data are divided by a particular attribute. The formula for split information is shown as follows [7]:

$$Split\ Info(A) = -\sum_{i=1}^{n} \left(\frac{|D_i|}{|D|}\right) log_2 \left(\frac{|D_i|}{|D|}\right) \qquad (4)$$

In (4), $SplitInfo(A)$ denotes the split information value of attribute A, n represents the number of partitions of attribute A, $|D_i|$ is the number of data points in partition i of dataset D based on attribute A, and $|D|$ is the total number of data points in dataset D. This measure is used to evaluate the distribution of data after a split, so that attributes producing too many partitions are not automatically preferred. To improve attribute selection, the C4.5 algorithm uses gain ratio, which normalizes information gain by the split information [18]. The gain ratio is calculated using the following equation:

$$\text{Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Info}} \qquad (5)$$

In (5), *Information Gain* refers to the reduction in entropy obtained after splitting the dataset using a particular attribute, while *Split Info* represents the intrinsic information generated by the split. The gain ratio is used to select better attributes by considering not only the reduction in uncertainty but also the balance of data distribution among the partitions. In this way, the resulting tree is less likely to be biased toward attributes with many categories or values [19]. Once the gain ratio of each attribute has been calculated, the attribute with the highest gain ratio is selected as the root node or as the next branching node in the tree. The same process is then repeated recursively for each resulting partition until all cases are completely classified. Through this recursive mechanism, the C4.5 algorithm constructs a decision tree capable of identifying patterns in the dataset and classifying whether a patient belongs to the diabetes or non-diabetes class [19]. The partitioning process in the decision tree stops when one of the stopping criteria has been reached. First, the process stops when all cases in a node belong to the same class.

Second, it stops when there are no remaining attributes available for further partitioning. Third, it also stops when a branch contains no cases. These stopping conditions ensure that the tree construction process ends at an appropriate stage and produces a classification model that can be used for prediction and performance evaluation [20].

3.4 Random Forest Modeling with and without Balancing

Random Forest modeling in this study was carried out using two approaches, namely with balancing and without balancing. In the unbalanced approach, the model was trained using the original dataset, which may contain an unequal class distribution between diabetes and non-diabetes cases. In the balanced approach, the class distribution was adjusted before the training process in order to reduce the dominance of the majority class and improve the model's ability to classify minority-class instances. By applying these two approaches, this study aims to evaluate the effect of class balancing on the performance of the Random Forest algorithm in diabetes prediction.

The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees and combines their outputs to produce a final prediction [21]. The first step in building a Random Forest model is preparing the training data through the bootstrapping technique, in which multiple training subsets are generated by repeatedly sampling from the original training dataset. Each bootstrap sample is then used to grow one decision tree in the forest. This approach increases diversity among the trees and helps reduce the risk of overfitting compared with a single decision tree model.

In the next step, each tree is constructed independently using a random subset of features selected at each node. Instead of evaluating all available features when determining the best split, the Random Forest algorithm randomly chooses only a subset of features at every node. This random feature selection process is important because it introduces additional variation among the trees and prevents the model from relying too heavily on a small number of dominant attributes. The number of randomly selected features at each node can also be adjusted in order to obtain better classification performance.

After a large number of trees have been generated, all trees in the forest are combined to produce the final prediction. In classification tasks, including diabetes prediction, the final class label is determined using a voting mechanism, where the class receiving the majority vote from all trees is selected as the final output. Through this ensemble process, Random Forest generally provides better stability, robustness, and predictive accuracy than a single decision tree, because the final decision is based on the collective performance of many trees rather than a single model. Therefore, Random Forest is considered a suitable method for classification problems involving complex data patterns and potential class imbalance [22].

3.5 Hyperparameter Tuning for the C4.5 Decision Tree with and without Data Balancing

At this stage, the modeling process using the C4.5 Decision Tree involves two approaches: with balancing and without balancing, where the model's hyperparameters are tuned using GridSearchCV to identify the optimal combination of hyperparameters based on accuracy [8].

Table 2. Parameter Grid Search Decision Tree C4.5

criterion	gini	entropy		
splitter	best	random		
Max features	sqrt	log2		
Max depth	none	10	20	
min samples split	2	5	10	
Min samples leaf	1	2	3	4

3.6 Hyperparameter Tuning for Random Forests with and without Balancing

At this stage, the modeling process using Random Forest involved hyperparameter tuning with GridSearchCV to identify the optimal combination of parameters based on accuracy [8].

Table 3. Parameter Grid Search Random Forest

N estimators	100	200	300	
Max features	sqrt	log2		
Max depth	none	10	20	
min samples split	2	5	10	
Min samples leaf	1	2	3	4

4. RESULT AND DISCUSSION

In this stage, the classification performance of the Pima Indians Diabetes dataset was evaluated under three reported experimental settings, namely without data balancing, with data balancing, and with hyperparameter tuning without balancing. The performance of the C4.5 Decision Tree and Random Forest models was assessed using accuracy, precision, recall, and F1-score obtained from 3-fold, 5-fold, and 9-fold cross-validation. Overall, the results show that Random Forest consistently achieved better and more stable performance than the C4.5 Decision Tree across all reported settings.

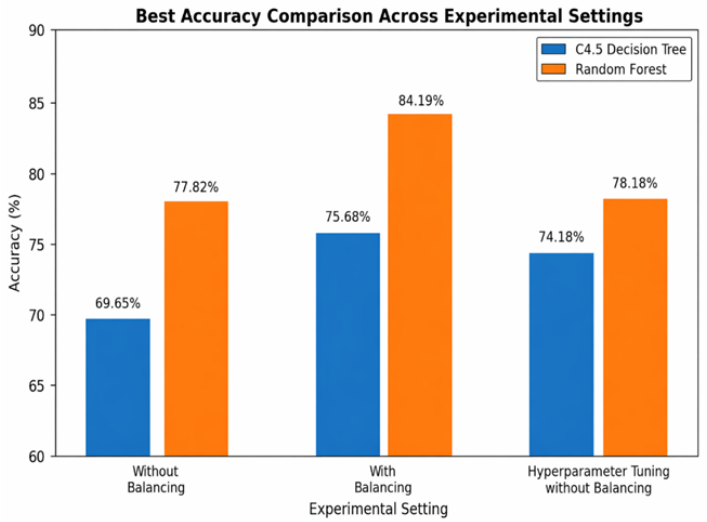

Figure 4. Accuracy Comparison Across Experimental Settings

Figure 4 shows the comparison of the best accuracy values achieved by the C4.5 Decision Tree and Random Forest models across the three reported experimental settings: without balancing, with balancing, and with hyperparameter tuning without balancing. Under the without balancing condition, Random Forest achieved a higher best accuracy of 77.82% compared with 69.65% for the C4.5 Decision Tree. After applying data balancing, the performance of both models improved, with Random Forest reaching the highest overall reported accuracy of 84.19%, while the C4.5 Decision Tree achieved 75.68%. Under hyperparameter tuning without balancing, both models also showed improvement compared with the baseline setting, with Random Forest achieving 78.18% and the C4.5 Decision Tree achieving 74.18%. Overall, Figure 4 confirms that Random Forest consistently outperformed the C4.5 Decision Tree across all reported experimental settings, while data balancing produced the greatest improvement in classification accuracy.

Table 4 shows the classification performance of both models under the non-balancing condition. In this setting, Random Forest consistently outperformed the C4.5 Decision Tree across all K-fold values. The highest accuracy achieved by Random Forest was 77.82% at 5-fold cross-validation, while the C4.5 Decision Tree achieved its highest accuracy of 69.65% at 9-fold cross-validation. Similar patterns can also be observed in precision, recall, and F1-score, where Random Forest produced higher values overall. These results indicate that Random Forest provided better baseline performance on the original imbalanced dataset.

Table 4. Data Classification Accuracy Without Data Balancing

Without Balancing								
K-Fold	Decision Tree C4.5				Random Forest			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
3	67.45%	48.67%	54.02%	51.08%	76.18%	64.58%	55.75%	59.77%
5	68.73%	51.29%	55.21%	53.04%	77.82%	69.27%	54.03%	60.58%
9	69.65%	52.92%	56.23%	54.01%	77.27%	69.56%	52.87%	59.70%

Table 5 presents the results after applying data balancing. In this setting, the performance of both models improved compared with the non-balancing condition. Random Forest achieved accuracies of 83.38%, 83.79%, and 84.19% for 3-fold, 5-fold, and 9-fold cross-validation, respectively, while the C4.5 Decision Tree achieved 71.55%, 75.68%, and 74.22%. In addition to accuracy improvement, both models also showed better recall and F1-score values after balancing. This result indicates that balancing the dataset helped the classifiers better identify diabetes cases in the minority class. Among all reported settings, this condition produced the best overall reported accuracy, which was achieved by Random Forest at 84.19%.

Table 5. Classification Performance with Data Balancing

Balancing								
K-Fold	Decision Tree C4.5				Random Forest			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
3	71.55%	70.08%	74.51%	72.01%	83.38%	81.64%	86.20%	83.71%
5	75.68%	74.84%	77.15%	75.69%	83.79%	82.14%	86.72%	84.19%
9	74.22%	73.01%	77.73%	74.79%	84.19%	82.88%	86.49%	84.51%

Table 6 shows the performance of both models under hyperparameter tuning without balancing. The results indicate that hyperparameter tuning also improved model performance compared with the baseline non-balancing condition. Random Forest achieved its highest accuracy of 78.18% at 5-fold cross-validation, while the C4.5 Decision Tree achieved 74.18% under the same setting. Although both models benefited from parameter optimization, the performance gain obtained from hyperparameter tuning was smaller than that obtained from data balancing. This suggests that, for the dataset used in this study, addressing class imbalance had a greater effect on classification performance than tuning model parameters alone.

Table 6. Classification Performance under Hyperparameter Tuning without Data Balancing

Hyperparameter Without Balancing								
K-Fold	Decision Tree C4.5				Random Forest			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
3	70.37%	70.42%	70.37%	70.28%	75.82%	63.76%	54.60%	58.82%
5	74.18%	73.48%	74.18%	73.33%	78.18%	69.85%	54.60%	61.29%
9	69.46%	68.23%	69.46%	68.48%	76.17%	65.69%	51.72%	57.88%

Across all reported experimental settings, Random Forest demonstrated more stable and superior performance than the C4.5 Decision Tree. This may be explained by the ensemble-based mechanism of Random Forest, which combines the predictions of multiple decision trees and thereby reduces overfitting while improving generalization performance. In contrast, the C4.5 Decision Tree appeared to be more sensitive to data imbalance and model configuration. Although its performance improved after balancing and hyperparameter tuning, it still did not reach the same level of consistency as Random Forest.

Overall, the findings of this study indicate that Random Forest is more effective than the C4.5 Decision Tree for diabetes prediction when formulated as a binary classification task on imbalanced data. The results also emphasize that appropriate preprocessing, particularly data balancing, plays an important role in improving model performance on medical datasets with unequal class distribution. Therefore, combining a robust classifier with suitable preprocessing can provide more reliable prediction results in diabetes classification problems [23].

5. CONCLUSION

This study evaluated the performance of the C4.5 Decision Tree and Random Forest algorithms for predicting diabetes as a binary classification task on the Pima Indians Diabetes dataset. The experiments were conducted under three reported settings, namely without data balancing, with data balancing, and with hyperparameter tuning without balancing. The results showed that Random Forest consistently outperformed the C4.5 Decision Tree across all reported settings. Under the non-balancing condition, Random Forest achieved better baseline performance on the original imbalanced dataset. After applying data balancing, the performance of both models improved, with Random Forest achieving the best overall reported accuracy of 84.19%. Hyperparameter tuning without balancing also improved the performance of both models, although the improvement was less substantial than that achieved through data balancing.

Overall, these findings indicate that Random Forest is more robust and effective than the C4.5 Decision Tree for diabetes prediction on imbalanced data. Its ensemble-based mechanism enables the model to reduce overfitting and achieve more stable classification performance across different evaluation settings. In contrast, the C4.5 Decision Tree was more sensitive to data imbalance and model configuration, even though its performance improved after balancing and hyperparameter tuning. Therefore, Random Forest can be considered the more reliable method for diabetes classification when predictive performance, stability, and robustness are the main considerations. These findings also highlight the importance of appropriate preprocessing, especially data balancing, in improving classification performance on imbalanced medical datasets. Future studies may extend this work by including additional evaluation metrics such as ROC-AUC and confusion matrix analysis, as well as other machine-learning models for comparison.

REFERENCE

[1] E. Indriyani, L. and T. K. Dewi, "Penerapan Senam Kaki Diabetes Melitus Terhadap Kadar Glukosa Darah Pada Penderita Diabetes Melitus di Puskesmas Yosomulyo," *Jurnal Cendikia Muda,* pp. 252-259, 2023.

[2] Z. D. R. Sari, J. Jasmir and Y. Arvita, "Penerapan Data Mining Untuk Prediksi Penyakit Diabetes Menggunakan Algoritma C4.5," *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM),* vol. 4, no. 1, pp. 827-834, 2024.

[3] A. Perdana, A. Hermawan and D. Avianto, "Analyze Important Features of PIMA Indian Database For Diabetes Prediction Using KNN," *Jurnal SISFOKOM (Sistem Informasi dan Komputer),* vol. 12, no. 1, pp. 70-75, 2023.

[4] A. Mousa, W. Mustafa, R. B. Marqas and S. H. M. Mohammed, "A Comparative Study of Diabetes Detection Using The Pima Indian Diabetes Database," *Journal of University of Duhok,* vol. 26, no. 2, pp. 277-288, 2023.

[5] A. Ram and H. Vishwakarma, "Diabetes Prediction using Machine learning and Data Mining Methods," in *IOP Conference Series: Materials Science and Engineering*, 2020.

[6] R. Rousyati, A. N. Rais, E. Rahmawati and R. F. Amir, "Prediksi Pima Indians Diabetes Database Dengan Ensemble Adaboost Dan Bagging," *Evolusi: Jurnal Sains dan Manajemen,* vol. 9, no. 2, pp. 36-42, 2021.

[7] D. S. Rahayu, N. J. Afifah and S. Intan, "Klasifikasi Penyakit Diabetes Melitus Menggunakan Algoritma C4.5, Support Vector Machine (SVM) dan Regresi Linear," in *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, 2023.

[8] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search," *SISTEMASI: Jurnal Sistem Informasi,* vol. 11, no. 2, pp. 391-401, 2022.

[9] D. Ismafillah, T. Rohana and Y. Cahyana, "Analisis algoritma pohon keputusan untuk memprediksi penyakit diabetes menggunakan oversampling smote," *Infotech: Jurnal Informatika & Teknologi,* vol. 4, no. 1, pp. 27-36, 2023.

[10] A. T. Akbar, H. Prapcoyo and R. Husaini, "SMOTE and K-Means Preprocessing for Classification by Logistic Regression on Pima Indian Diabetes Dataset," *Telematika: Jurnal Informatika dan Teknologi Informasi,* vol. 20, no. 2, pp. 238-249, 2023.

[11] D. C. P. Buani, "Deteksi Dini Penyakit Diabetes dengan Menggunakan Algoritma Random Forest," *Evolusi: Jurnal Sains dan Manajemen ,* vol. 12, no. 1, pp. 1-8, 2024.

[12] S. Marimuthu, T. Mani, T. D. Sudarsanam, S. George and L. Jeyaseelan, "Preferring Box-Cox transformation, instead of log transformation to convert skewed distribution of outcomes to normal in medical research," *Clinical Epidemiology and Global Health,* vol. 15, pp. 1-5, 2022.

[13] I. Tasin, T. U. Nabil, S. Islam and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthcare Technology Letters,* vol. 10, pp. 1-10, 2022.

[14] H. Bichri, A. Chergui and M. Hain, "Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," *International Journal of Advanced Computer Science and Applications,* vol. 15, no. 2, pp. 331-339, 2024.

[15] "Comparison of Feature Selection with Information Gain Method in Decision Tree, Regression Logistic and Random Forest Algorithms," *Journal of Applied Business and Technology,* vol. 5, no. 3, pp. 146-153, 2023.

[16] E. Afrianto, J. E. Suseno and B. Warsito, "Decision Tree Method with C4.5 Algorithm for Students Classification Who is Entitled to Receive Indonesian Smart Card (KIP)," in *INCITEST 2020*, 2020.

[17] P. Gulati , A. Sharma and M. Gupta, "Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review," *International Journal of Computer Applications,* vol. 141, no. 14, pp. 19-25.

[18] G. S. Reddy and S. Chittineni, "Entropy based C4.5-SHO algorithm with information gain optimization in data mining," *PeerJ Computer Science,* pp. 1-22, 2021.

[19] H.-B. Wang and Y.-J. Gao, "Research on C4.5 algorithm improvement strategy based on MapReduce," *Procedia Computer Science,* vol. 183, pp. 160-165, 2021.

[20] X. Meng, P. Zhang and D. Zhang, "Decision Tree for Online Voltage Stability Margin Assessment Using C4.5 and Relief-F Algorithms," *Energies,* pp. 1-13, 2020.

[21] A. Mohanty and G. Gao, "A survey of machine learning techniques for improving Global Navigation Satellite Systems," *EURASIP Journal on Advances in Signal Processing,* vol. 2024, no. 73, pp. 1-40, 2024.

[22] I. K. Pious, A. Rajalakshmi, P. K. R, V. C. M, M. Nalini and S. S. R, "Enhancing Prediction Accuracy Through Random Forest in Classification and Regression," in *Smart Technologies for Sustainable Development Goals (ICSTSDG)*, 2024.

[23] A. Efendi, I. Fitri and G. W. Nurcahyo, "Improvement of Machine Learning Algorithms with Hyperparameter Tuning on Various Datasets," in *Future Technologies for Smart Society (ICFTSS)*, 2024.