BAREKENG: Journal of Mathematics and Its Applications
September 2025 Volume 19 Issue 3 Page 1609-1624
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss3pp1609-1624

GRID SEARCH AND RANDOM SEARCH HYPERPARAMETER TUNING OPTIMIZATION IN XGBOOST ALGORITHM FOR PARKINSON'S DISEASE CLASSIFICATION

Shafa Fitria Aqilah Khansa[1*], Nurissaidah Ulinnuha[2], Wika Dianita Utami[3]

[1,2,3] *Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Sunan Ampel Jln. Dr. Ir. H. Soekarno No.682, Gn. Anyar, Surabaya, 60294, Indonesia*

*Corresponding author's e-mail: * **shafafitria84@gmail.com***

Article History:

Received: 17th December 2024
Revised: 15th January 2025
Accepted: 27th February 2025
Published: 1st July 2025

Keywords:

Classification;
Cross Validation;
Hyperparameter Tuning;
Parkinson;
XGBoost.

ABSTRACT

Parkinson's disease is a neurodegenerative disorder affecting motor abilities, with a prevalence of 329 cases per 100,000 individuals. Early diagnosis is crucial to prevent complications. This study classifies Parkinson's disease using the Extreme Gradient Boosting (XGBoost) algorithm with hyperparameter tuning via Grid Search and Random Search. The dataset from Kaggle consists of 2105 records from 2024 and includes 32 clinical and demographic features such as age, gender, BMI, medical history, and Parkinson's symptoms. The XGBoost method effectively manages large and complex data and reduces. Tuning was performed with 5-fold cross-validation for result validity. After tuning with Grid Search, the model achieved 93.35% accuracy in 44 minutes 51 seconds, with optimal parameters gamma=5, max depth=3, learning rate=0.3, n estimators=100, and subsample=0.7. Meanwhile, Random Search with 50 iterations achieved 93.97% accuracy in 3 minutes 4 seconds with optimal parameters gamma=5, max depth=3, learning rate=0.262, n estimators=58, and subsample=0.631. Random Search also shows better time efficiency thanGrid Search, although with relatively similar accuracy. The results of this study confirm that hyperparameter tuning using Random Search not only produces competitive accuracy performance but also minimizes computation time, making it a more optimal choice for Parkinson's disease classification.

How to cite this article:

S. F. A. Khansa, N. Ulinnuha and W. D. Utami., "GRID SEARCH AND RANDOM SEARCH HYPERPARAMETER TUNING OPTIMIZATION IN XGBOOST ALGORITHM FOR PARKINSON'S DISEASE CLASSIFICATION," *BAREKENG: J. Math. & App.*, vol. 19, no. 3, pp. 1609-1624, September, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Parkinson's disease is a long-term neurodegenerative condition that impacts the motor system by damaging the substantia nigra, leading to a reduction in dopamine levels. This results in symptoms such as tremors, muscle rigidity, slowed movement, and difficulties with balance [1]. In Indonesia, the number of Parkinson's cases is rising alongside the aging population. The 2016 Global Burden of Disease Study estimated 117531 to 178755 cases, marking a 217% increase since 1990. By June 2023, Faculty of Medicine Universitas Airlangga (FK UNAIR) projected 200000 to 400000 cases, with 1100 deaths recorded, placing Indonesia 12th globally and 5th in Asia [2]. Parkinson's commonly affects individuals over 60 years old, with men being more susceptible than women, potentially due to genetic and hormonal factors [3]. Early treatment involving dopaminergic drugs, physical therapy, and occupational therapy can improve quality of life, though long-term costs like brain stimulation pose financial challenges [4].

Diagnosing Parkinson's is complex due to varying symptoms among individuals, requiring physical examinations and medical tests such as blood analyses and brain imaging [5], [6]. Genetic factors, including mutations in genes like LRRK2, GBA, and DJ-1, along with epigenetic influences like lifestyle and medical conditions, contribute to Parkinson's risk [7]. In Indonesia, low public awareness of early symptoms and limited healthcare infrastructure complicate timely diagnosis and treatment [6]. A case of a 53-year-old teacher illustrated these challenges, where overlapping symptoms of hypertension, diabetes, and stroke masked Parkinson's indicators, such as an expressionless face without tremors [8]. This highlights the need for better medical awareness and access to care, particularly in under-resourced regions.

Amidst these challenges, research conducted by [9] shows that the application of machine learning technology, notably the XGBoost algorithm, can provide innovative solutions in analyzing complex data that are difficult to handle by traditional methods. XGBoost works by applying the concept of the boosting method. This is done by building the model sequentially and combining all the models for prediction so that the new model learns from the mistakes of the previous model [10]. XGBoost combines weak classifiers to form stronger models and prevents overfitting [11]. Several methods have been used to classify Parkinson's disease. Research has shown that XGBoost outperforms other algorithms in medical classification tasks. A study [12], on lung disease classification achieved 93.65% accuracy, surpassing Random Forest (90.75%). Similarly, XGBoost demonstrated higher accuracy (95.08%) in heart disease detection [13]. In diabetes classification, hyperparameter tuning increased XGBoost's accuracy to 95% [14]. A recent study on celiac disease, an autoimmune disorder affecting 0.5%-1% of the population, showed that XGBoost, optimized with 5-fold cross-validation, improved accuracy from 98.19% to 98.64%, with 98.43% sensitivity and 99.72% specificity, confirming the effectiveness of parameter tuning [15].

With the increasing number of older adults in Indonesia, Parkinson's cases are predicted to continue to grow, demanding further research into the quality of life of patients as well as more accurate classification methods [16], [17]. This research offers a breakthrough by applying the XGBoost algorithm, which has been proven to excel in handling complex data. Still, it has never been optimized using Grid Search and Random Search hyperparameter tuning for Parkinson's classification in Indonesia. Hyperparameter adjustments, such as learning rate, gamma, max depth, n estimators, and subsample, will be analyzed to maximize model performance. This research aims to measure the impact of hyperparameter tuning on model effectiveness and develop a classification system that is more precise and adaptive to patient characteristics in Indonesia [18]. This approach is expected to create a more accurate and efficient model optimization strategy in detecting Parkinson's disease, making a significant contribution to Indonesia's medical world and research.

2. RESEARCH METHODS

The research process stages undertaken to achieve the objectives of this study are outlined in the research methodology flowchart presented in **Figure 1**.

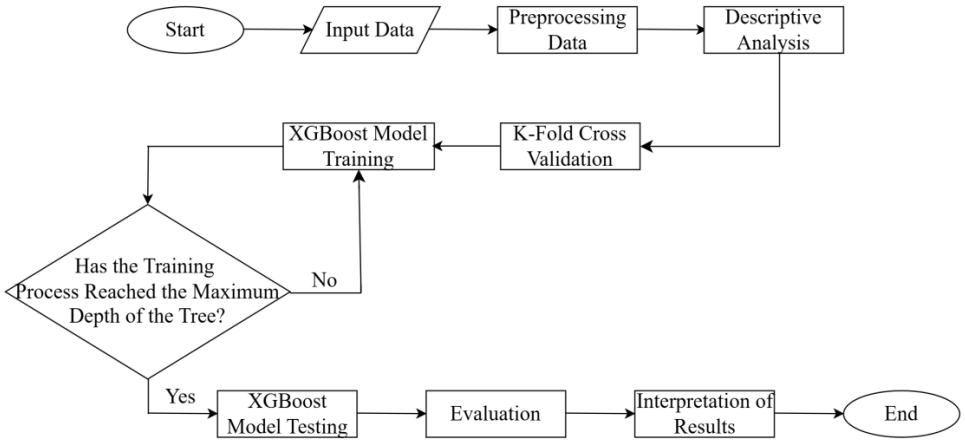

Figure 1. Research Flowchart

Figure 1 shows that the analysis process begins with entering the data to be used, followed by preprocessing to convert categorical data to numerical form to ensure data integrity. Next, descriptive analysis is performed to understand the characteristics of the data before dividing it using K-Fold cross-validation to ensure balanced model validation. The XGBoost model was then trained iteratively until it reached the optimal depth, followed by testing using test data. Parameter optimization was performed through Grid Search and Random Search to find the best configuration. Evaluation of the classification results using a confusion matrix helps assess the model's accuracy in distinguishing between positive and negative classes. Finally, the classification results are interpreted to understand the implications and draw conclusions based on the model predictions.

2.1 Research Data

This research data is health information that includes demographics, lifestyle, medical history, clinical measurements, cognitive assessment, symptoms, and diagnosis indicators. The dataset totaled 2105 with 33 variables, consisting of 32 independent variables and one dependent variable (target). The dataset is from Kaggle, published by Rabie El Kharoua (2024) under a CC BY 4.0 license [19]. From the dataset, 1304 people were diagnosed with Parkinson's, while 801 people were not. Variable details can be seen in Table 1.

Table 1. Sample Data

Age	Gender	Ethnicity	…	SleepDisorders	Constipation	Diagnosis
85	0	3	…	0	0	0
75	0	0	…	1	0	1
70	1	0	…	0	1	1
52	0	0	…	0	1	1
87	0	0	…	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
87	1	0	…	1	0	0
67	0	0	…	1	1	1
65	0	0	…	1	0	1
61	1	0	…	1	1	1
56	0	0	…	0	1	0

***Data source**: Kaggle*

Table 1 shows sample data that includes several variables, such as age, gender, and ethnicity, as well as specific health conditions associated with the diagnosis variables, such as sleep disorders and constipation. The data shows the distribution of patients with varying ages ranging from 50 to 89 years, as well as categorical variables such as gender (0 for male and 1 for female) and ethnicity (expressed in numerical codes). In addition, the data shows the pattern of association between specific health conditions and the final diagnosis outcome, characterized by a binary value (0 or 1). This information forms the basis for research analysis to understand these factors' influence on disease diagnosis.

2.2 Research Attributes

This study utilizes 33 attributes, which include 32 independent attributes and one dependent attribute. Details of these attributes are provided in **Table 2**.

Table 2. Attribute of Data

Attribute	Data Type	Value	Category
Age	Numeric	[50; 90]	-
Gender	Categorical	[0; 1]	0: Male; 1: Female
Ethnicity	Categorical	[0; 1; 2; 3]	0: Caucasian; 1: African American; 2: Asian; 3: Other
Education Level	Categorical	[0; 1; 2; 3]	0: None; 1: High School; 2: Bachelor's; 3: Higher
BMI	Numeric	[15; 40]	-
Smoking	Categorical	[0; 1]	0: No; 1: Yes
Alcohol Consumption	Numeric	[0; 20]	-
Physical Activity	Numeric	[0; 10]	-
Diet Quality	Numeric	[0; 10]	-
Sleep Quality	Numeric	[4; 10]	-
Family History Parkinsons	Categorical	[0; 1]	0: No; 1: Yes
Traumatic Brain Injury	Categorical	[0; 1]	0: No; 1: Yes
Hypertension	Categorical	[0; 1]	0: No; 1: Yes
Diabetes	Categorical	[0; 1]	0: No; 1: Yes
Depression	Categorical	[0; 1]	0: No; 1: Yes
Stroke	Categorical	[0; 1]	0: No; 1: Yes
Systolic BP	Numeric	[90; 180]	-
Diastolic BP	Numeric	[60; 120]	-
Cholesterol Total	Numeric	[150; 300]	-
Cholesterol LDL	Numeric	[50, 200]	-
Cholesterol HDL	Numeric	[20; 100]	-
Cholesterol Triglycerides	Numeric	[50, 400]	-
UPDRS	Numeric	[0; 199]	-
MoCA	Numeric	[0; 30]	-
Functional Assessment	Numeric	[0; 100]	-
Tremor	Categorical	[0; 1]	0: No; 1: Yes
Rigidity	Categorical	[0; 1]	0: No; 1: Yes
Bradykinesia	Categorical	[0; 1]	0: No; 1: Yes
Postural Instability	Categorical	[0; 1]	0: No; 1: Yes
Speech Problems	Categorical	[0; 1]	0: No; 1: Yes
Sleep Disorders	Categorical	[0; 1]	0: No; 1: Yes
Constipation	Categorical	[0; 1]	0: No; 1: Yes
Diagnosis	Categorical	[0; 1]	0: No; 1: Yes

Referring to the independent attributes in **Table 2**, 18 attributes are categorical and 15 are numerical. In this study, the XGBoost algorithm requires transforming categorical data into numerical form, which is designed to process only numerical data.

2.3 Classification

Classification is the process of creating a model to classify objects to predict the class of an unknown object **[20]**. This process consists of two main stages: training and testing. In the training stage, the algorithm analyses the data to generate a classification rule in the form of a function $Y = F(X)$, where Y is the predicted class and X is the features used **[21]**. After the model is formed, the testing stage is carried out using new data to evaluate the accuracy of the model in classifying objects according to the rules made **[22]**.

2.4 Boosting

Boosting was first introduced by Robert Schapire in 1998 [23]. Boosting methods in machine learning aim to improve prediction models' performance by combining several less effective models or classifiers into one more reliable model. This approach is designed to reduce prediction errors and improve overall accuracy by combining models that focus on overcoming data that is difficult to classify [24].

2.5 Extreme Gradient Boosting

XGBoost, developed by Dr Tianqi Chen from the University of Washington in 2014, is an extension of the gradient boosting method that is well-known for its efficiency and effectiveness in solving classification problems [23]. This method is claimed to be 10 times faster than other gradient-boosting techniques [11]. XGBoost is efficient in time and memory usage, and is applied in various fields, such as medicine and credit risk assessment, with advantages in handling data imbalance [25]. The algorithm is designed to handle regression, classification, and ranking problems and prevent overfitting through regularization and parameter adjustment [26]. The following is the flowchart of the XGBoost algorithm, illustrating the step-by-step process of building the model, as shown in Figure 2.

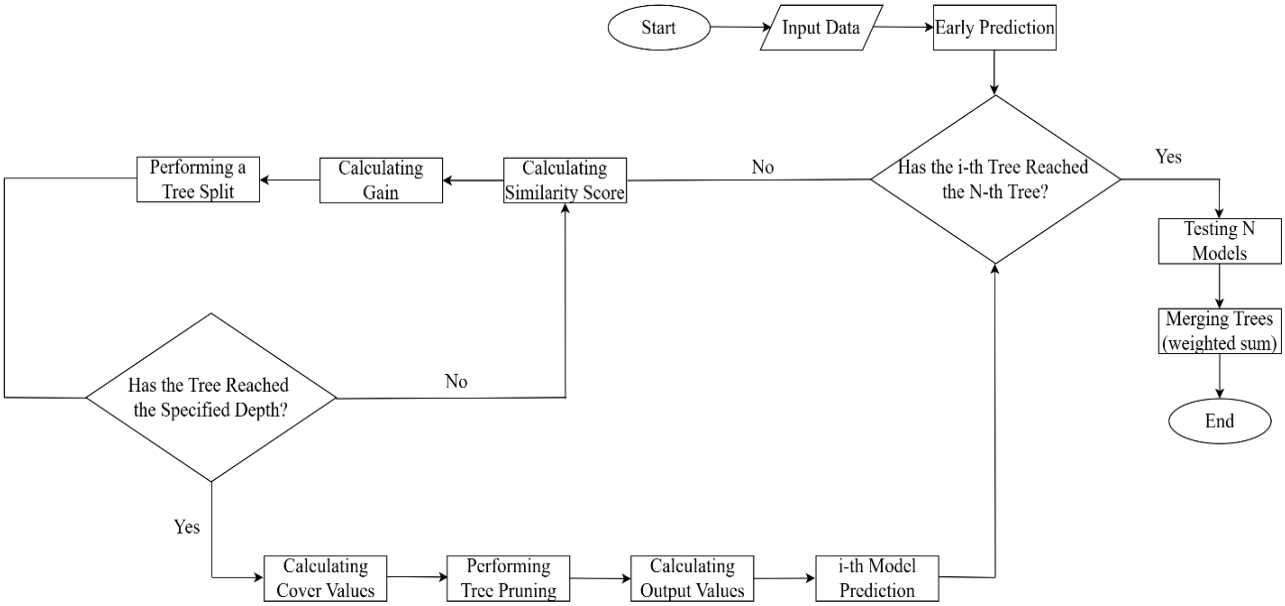

Figure 2. XGBoost Flowchart

Based on Figure 2, the step-by-step process of the XGBoost method can be explained as follows [27]. The first step in building an XGBoost model is to determine the initial prediction probability using Equation (1). Next, the residual is computed as the difference between the actual value and the initial prediction probability, as Equation (2) defines. To assess the quality of branch splitting in the decision tree, the similarity score is calculated using Equation (3). The gain, which evaluates the effectiveness of a split, is then determined by comparing the similarity scores of the root node and its child nodes, as shown in Equation (4). Furthermore, the cover value, which measures the probability distribution at each node, is calculated using Equation (5). Based on Equation (6), tree pruning is applied to prevent excessive complexity. The log-odds transformation is performed using Equation (7), then updating the prediction function described in Equation (8). Finally, the output value, which determines the final prediction of the XGBoost model, is computed using Equation (9). The binary sigmoid function is a mathematical function that produces values between 0 and 1. If the sigmoid value exceeds 0.50, the instance is assigned to class 1, whereas if it is below 0.50, the instance is categorized as class 0.

$$p_{(i)} = \frac{1}{n_c}, \text{with } i = 1, 2, \dots, n \tag{1}$$

$$e_{(i)} = y_{(i)} - p_{(i)} \tag{2}$$

$$SS = \frac{\left(\sum_{i=1}^{n} e_{(i)}\right)^2}{\sum_{i=1}^{n}[p_{(i)} \times (1 - p_{(i)})] + \lambda} \quad (3)$$

$$G = \left(SS_{left} + SS_{right}\right) - SS_{root} \quad (4)$$

$$Cover_{(i)} = \sum_{i=1}^{n}[p_{(i)} \times (1 - p_{(i)})] + \lambda \quad (5)$$

$G - \gamma < 0$, with γ default is 0

$$Output\ Value = \frac{\left(\sum_{i=1}^{n} e_{(i)}\right)}{\sum_{i=1}^{n}[p_{(i)} \times (1 - p_{(i)})] + \lambda} \quad (6)$$

$$\text{Log}(O) = \log\left(\frac{p_{(i)}}{1 - p_{(i)}}\right) \quad (7)$$

$$f(x_i) = \log(O) + \alpha \times Output\ Value \quad (8)$$

$$Sigmoid\left(f(x_i)\right) = \frac{e^{(f(x_i))}}{1 + e^{(f(x_i))}} \quad (9)$$

with

$p_{(i)}$: predicted value; n_c: class size

$e_{(i)}$: residual; $y_{(i)}$: class value

SS: similarity score; SS_{left}: left similarity score; SS_{right}: right similarity score

G: gain

γ: gamma (minimum loss function)

$Cover_{(i)}$: cover value

λ: regularization parameter

n: total data

O: odds

$f(x_i)$: new prediction

α: learning rate

2.6 Hyperparameter Tuning

Hyperparameter tuning is essential for enhancing the performance of machine learning algorithms. It must be determined before the learning process, as it cannot be derived directly from the data. While the interaction between hyperparameters and model performance continues to be researched, tuning is often computationally expensive, especially on large datasets, and customized settings do not always provide significant improvements over default values [28]. The hyperparameters applied in this study include grid search and random search. Grid search is a method used to identify the optimal hyperparameter combinations in a machine-learning model by testing all specified value ranges. It selects the combination that yields the best performance for the optimal model. However, grid search can be time-consuming and resource-intensive, especially with many hyperparameters and value ranges [28]. This algorithm tests all combinations in the sample space to identify the best hyperparameters based on data processing results [29]. Random Search is a hyperparameter method that enables quick adjustments in experimental "resolution," such as adding or skipping experiments. It is simple, easy to implement, and allows for parallel execution, maintaining minimal efficiency in low-dimensional spaces while significantly improving performance in high-dimensional ones. This method samples from the search area using a probability distribution [30]. The combination of parameters used in this study is contained in Table 3.

Table 3. Combination of Parameters

Parameter	Value
Gamma	0; 1; 3; 5
Max depth	3; 4; 5; 6; 7
Learning rate	0.01; 0.1; 0.2; 0.3
N estimator	50; 100
Subsample	0.6; 0.7; 0.8; 0.9

Table 3 shows the parameters used in the hyperparameter tuning process for the XGBoost model. The Gamma parameter was tested with 0, 1, 3, and 5 values to control regularization at split nodes. Max depth was varied between 3 to 7 to determine the maximum depth of the tree. Learning rate was explored with values of 0.01 to 0.3 to control the learning speed of the model. N estimators were tested on 50 and 100 trees, while Subsample had values of 0.6 to 0.9 to determine the proportion of data used at each iteration. This combination of parameters helps optimize the model's classification performance.

2.7 Performance Evaluation

Model performance evaluation assesses how well a model predicts or classifies new data by comparing its predictions to actual values, utilizing various metrics, including the confusion matrix. This table presents the number of correctly and incorrectly classified test data points, enabling an assessment of the classification system's accuracy. Despite its simplicity, the confusion matrix serves as an effective tool for measuring classification performance [31] and evaluating both accuracy and efficiency in grouping test data [32].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (10)$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (11)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (12)$$

With:

$Accuracy$: a measure of how often the model gives correct predictions

$Sencistivity$: a measure of the model's ability to detect all positive cases

$Specificity$: a measure of the model's ability to detect all negative cases while minimizing false positives.

TP: true positive, if the system successfully detects Parkinson's

TN: true negative, if the system correctly doesn't detect Parkinson's in someone who doesn't have it

FP: false positive, if the system incorrectly detects Parkinson's in someone who doesn't have it

FN: false negative, if the system fails to detect Parkinson's

3. RESULTS AND DISCUSSION

3.1 Division of Training Data and Test Data using K-Fold Cross Validation

In this study, the k-fold cross-validation method is used for data partitioning to evaluate model performance, with $k = 5$ and $k = 10$ selected based on [33]. Their study highlights that these values are commonly used, as a higher k reduces bias but may increase variance, while a lower k increases bias. The dataset is divided into multiple folds, where each fold serves as test data once, and the rest are used for training. The dataset division results are shown in **Table 4**.

Table 4. Division Data

Fold	Training Data	Test Data
5	1684	421
10	1895	210

Based on **Table 4**, experimental results show that the model accuracy is higher at $k = 5$ compared to $k = 10$, indicating that splitting the data with $k = 10$ results in better generalization for this dataset.

3.2 Manual Calculation of XGBoost Algorithm

In implementing the XGBoost algorithm, 20 data samples are used as training data and 3 as testing data. In this process, the model is trained using 20 training data samples, where each feature in the data is analyzed to build a sequential decision tree. The model is then optimized to minimize the prediction error. Once the training is complete, the model is tested on three samples of testing data to evaluate its performance, such as accuracy and generalization ability.

1. In this study, the initial prediction probability is set using the following formula:

$$p_{(i)} = \frac{1}{n_c} = \frac{1}{2} = 0.5$$

2. The residuals for the first instance are computed, and the results are obtained in **Table 5**.

$$e_{(i)} = y_{(i)} - p_{(i)}$$

Table 5. Residuals Data

Gender	Tremor	Rigidity	Bradykinesia	Postural Instability	Class	Initial Prediction	Residual
0	1	0	0	0	0	0.5	-0.5
1	1	0	0	0	1	0.5	0.5
0	1	1	1	0	1	0.5	0.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	1	1	0	0	1	0.5	0.5
0	1	0	1	0	0	0.5	-0.5
0	1	0	0	0	1	0.5	0.5

Table 5 presents residual data showing the relationship between input variables, initial predictions, and residuals in the model. Input variables include gender, tremor, rigidity, bradykinesia, and postural instability, which are associated with class as the target label. Initial prediction shows the model's initial predicted value of 0.5 for all data, while residual is calculated as the difference between the actual value (class) and the initial prediction. A positive residual value (0.5) indicates under-prediction, while a negative value (-0.5) indicates over-prediction. This table provides an initial overview of the prediction errors that form the basis of the model update process.

3. Compute the similarity score using this formula:

$$SS = \frac{\left(\sum_{i=1}^{20} e_{(i)}\right)^2}{\sum_{i=1}^{20}\left[p_{(i)} \times \left(1 - p_{(i)}\right)\right] + \lambda} = \frac{\left((-0.5) + 0.5 + 0.5 + \cdots + 0.5 + (-0.5) + 0.5\right)^2}{\sum[0.5 \times (1 - 0.5) + \cdots + 0.5 \times (1 - 0.5)] + 1} = 0.667$$

4. The biggest gain value of the first branch is calculated using the formula:

$$G = (0.692 + 3.267) - 0.667 = 3.292$$

5. Checking the residuals in the leaves to determine branch separation according to the maximum depth of the tree.

6. Calculate the cover value with the following formula by iterating through each observation in the dataset:

$$Cover_{(i)} = \sum_{i=1}^{20} [0.5 \times (1 - 0.5) + 0.5 \times (1 - 0.5) + \cdots + 0.5 \times (1 - 0.5)] = 5$$

7. Performing tree pruning, at this stage, the lowest gain is 0, then:

$$G - \gamma < 0, \text{ with } \gamma \text{ default is } 0$$

$$0 - 0 = 0$$

The pruning result is 0, indicating that no pruning occurs on the branch, and therefore, no additional gains are achieved. The final tree results after the pruning process are shown in **Figure 3**.

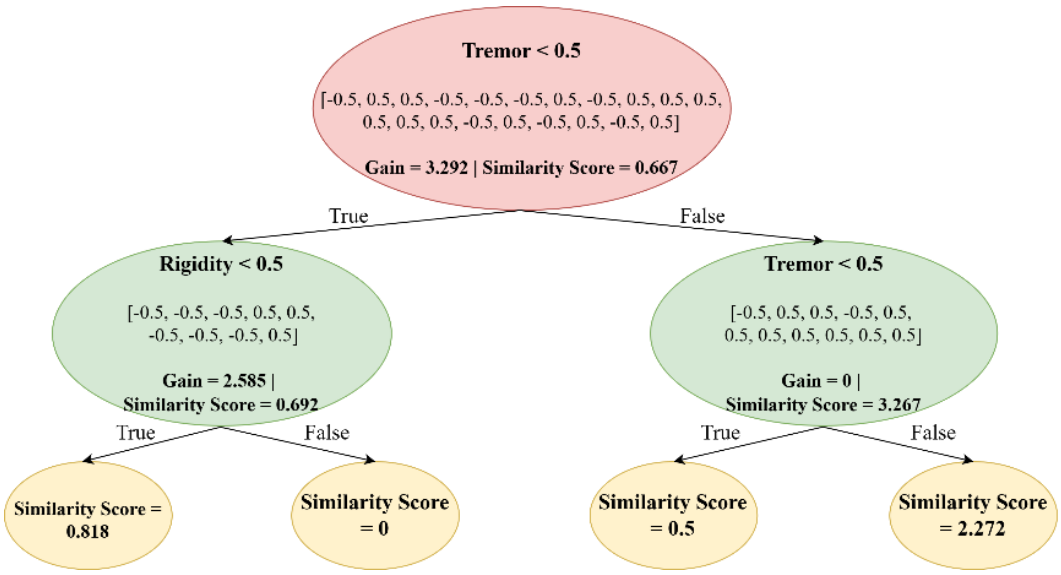

Figure 3. The Final Tree after the Tree Pruning Process

Figure 3 shows the result of the decision tree representation generated after the cover weighting process in the XGBoost algorithm. The decision tree is not pruned because all the gain values in each branch are still greater than zero (the lowest gain is 0). The tree structure shows the separation of data based on features such as Tremor and Rigidity, with a particular threshold value for each feature.

8. Furthermore, the calculation of the output value is obtained using **Equation (6)**, which is written in **Table 6** below.

Table 6. Output Value

Gender	Tremor	Rigidity	Bradykinesia	Postural Instability	Class	Output Value
1	0	0	0	0	0	-2
1	1	1	0	0	1	2
0	0	0	0	0	0	2

Table 6 shows the output values based on various features associated with Parkinson's disease, such as gender, tremor, rigidity, bradykinesia, and postural instability. In the first row, with a feature value of 1 for gender and 0 for other conditions, the output value is -2, indicating a class 0 category. In the second row, with tremor and rigidity conditions valued at 1, the output value is 2, indicating a class 1. In the third row, with feature values all 0, the output value is also 2, indicating a class 0.

9. Log-odds transformation with the following formula:

$$log(O) = log\left(\frac{p_{(i)}}{1 - p_{(i)}}\right) = log\left(\frac{0.5}{1 - 0.5}\right) = log\left(\frac{0.5}{0.5}\right) = log(1) = 0$$

10. A new prediction is generated with the value of the learning rate in this study set as the default of 0.3. The learning rate is a critical parameter in machine learning algorithms, as it determines how much the weights or model parameters are updated during each iteration. The calculation of new predictions is carried out based on **Equation (8)** as follows.

$$f(x_i) = \log(O) + \alpha \times Output\ Value$$

$$f(x_1) = 0 + 0.3 \times -2 = -0.6$$

$$f(x_2) = 0 + 0.3 \times 2 = 0.6$$
$$f(x_3) = 0 + 0.3 \times 2 = 0.6$$

The binary sigmoid function is a mathematical operation that converts input values to an output between 0 and 1. Instances with an output greater than 0.50 are categorized as class 1, while those with an output less than 0.50 are classified as class 0.

$$Sigmoid(f(x_i)) = \frac{\exp(f(x_i))}{1 + \exp\,(f(x_i))}$$
$$Sigmoid(f(x_1)) = \frac{\exp(-0.6)}{1 + \exp(-0.6)} = 0.354 < 0.5$$
$$Sigmoid(f(x_2)) = \frac{\exp(0.6)}{1 + \exp(0.6)} = 0.6457 > 0.5$$
$$Sigmoid(f(x_3)) = \frac{\exp(0.6)}{1 + \exp(0.6)} = 0.6457 > 0.5$$

Table 7. New Prediction for XGBoost Classification

Gender	Tremor	Rigidity	Bradykinesia	Postural Instability	Class	Output Value	New Prediction	Evaluation
1	0	0	0	0	0	-2	0	True
1	1	1	0	0	1	2	1	True
0	0	0	0	0	0	2	1	False

Table 7 shows the new prediction results for XGBoost classification. Each row presents data with gender, tremor, rigidity, bradykinesia, postural instability features, and the model's output value and prediction. In the first row, with gender condition one and other features 0, the model predicts a class of 0 with an output value of -2, corresponding to the actual value, so the evaluation is "True." In the second row, with tremor and rigidity being 1, the output value is 2, and the model prediction also results in a class of 1, which is correct (evaluation "True"). In the third row, although the output value is 2, the model predicts a class 1, but the actual value is 0, so the evaluation is "False".

3.3 Grid Search

The parameter tuning results are summarized in **Table 8**, providing a detailed overview of the performance metrics for each combination of hyperparameters tested. These results highlight the effect of various tuning configurations on the model's accuracy, sensitivity, and specificity, and provide insights into the most effective parameter settings for optimizing the classification task.

Table 8. Grid Search Parameter

Parameter	Best Value	Time	Accuracy	Sensitivity	Specificity
Gamma	5				
Max depth	3				
Learning rate	0.3	44' 51"	93.35%	95.78%	88.14%
N estimator	100				
Subsample	0.8				

Table 8 shows the evaluation results of the XGBoost model, which showed an accuracy of 93.35%, with a sensitivity of 95.78% and a specificity of 88.14%. This suggests the model is highly effective in accurately classifying positive and negative cases. High sensitivity demonstrates the model's capability to identify most positive cases, while strong specificity indicates its precision in recognizing negative cases. The computation time required to achieve this result is 44 minutes and 51 seconds. The optimal tree results of the hyperparameter tuning grid search are in **Figure 4**.

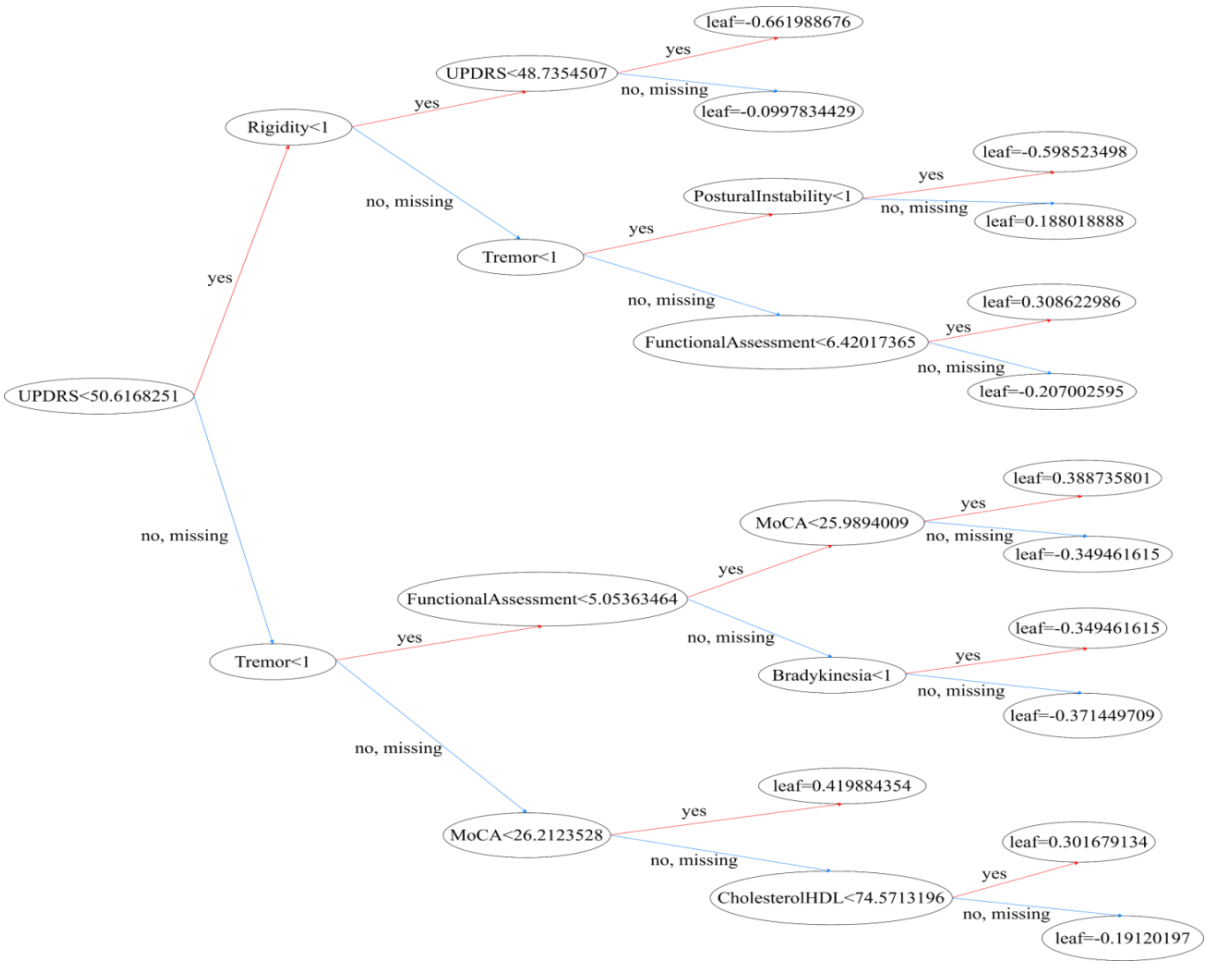

Figure 4. The Optimal Tree Result of Grid Search

Figure 4 shows that the tree starts with the evaluation UPDRS < 50.617. For a correct Rigidity < 1, the tree checks UPDRS < 48.735, resulting in leaf -0.662 (negative class) if correct and leaf -0.100 (negative class) otherwise. If Rigidity ≥ 1 is incorrectly evaluated, the tree evaluates Tremor < 1. When Tremor < 1 is true, the tree checks PosturalInstability < 1, leading to leaf -0.599 (negative class) if true and leaf -0.188 (negative class) otherwise. If Tremor ≥ 1 is incorrect, then the tree evaluates FunctionalAssessment < 6.420, resulting in leaf 0.309 (positive class) if true and leaf -0.207 (negative class) otherwise. On the main right branch (UPDRS ≥ 50.617), when incorrectly evaluated, the tree evaluates Tremor < 1. If correct, the tree checks FunctionalAssessment < 5.054, which, if accurate, leads to an evaluation of MoCA < 25.989. If true, it results in leaf 0.389 (positive class); if false, it results in leaf -0.349 (negative class). If FunctionalAssessment < 5.054 is incorrect, the tree evaluates Bradykinesia < 1, leading to leaf -0.371 (negative class) if true and leaf -0.302 (negative class) otherwise. If MoCA ≥ 26.212 is correctly evaluated, it contributes to a leaf value of 0.420 (positive class). Otherwise, the tree evaluates CholesterolHDL < 74.571, resulting in leaf -0.302 (negative class) if true and leaf -0.191 (negative class). The most influential features in this classification are UPDRS, Rigidity, Tremor, Functional Assessment, and MoCA. UPDRS, which acts as the root node, shows that this feature is the leading indicator in determining the severity of Parkinson's disease. Furthermore, Rigidity, Tremor, and Functional Assessment become distinguishing factors at the internal node level, which helps the model filter patient characteristics based on more specific symptoms. At the final level of classification, MoCA and HDL Cholesterol are used as additional indicators to differentiate patient categories in more detail, ensuring more accurate decisions.

3.4 Random Search

The parameter tuning results using Random Search, which involves sampling hyperparameters from predefined probability distributions, are summarized in Table 9. This approach tests various combinations of hyperparameters in a random search rather than exhaustively and provides a detailed overview of the performance metrics for each configuration.

Table 9. Random Search Parameter

Iteration	Parameter	Best Value	Time	Accuracy	Sensitivity	Specificity
50	Gamma	4.736	3’ 37”	93.97%	96.63%	89.64%
	Max depth	3				
	Learning rate	0.262				
	N estimator	58				
	Subsample	0.631				
100	Gamma	3.898	6’ 46”	93.06%	95.86%	88.51%
	Max depth	6				
	Learning rate	0.160				
	N estimator	53				
	Subsample	0.794				
500	Gamma	4.914	33’ 29”	93.21%	95.78%	89.01%
	Max depth	5				
	Learning rate	0.243				
	N estimator	82				
	Subsample	0.699				
1000	Gamma	4.327	1h 6’ 8”	93.25%	96.40%	89.14%
	Max depth	3				
	Learning rate	0.254				
	N estimator	69				
	Subsample	0.773				

Table 9 shows the XGBoost model evaluation results from the 50th iteration of the Random Search process, which showed the most optimal performance with an accuracy of 93.97%, sensitivity of 96.63%, and specificity of 89.64%. This accuracy reflects the proportion of correct predictions from the overall data, while the high sensitivity indicates the model's ability to detect most positive cases. The high specificity also confirms that the model can classify negative instances accurately. The tuning process took 3 minutes and 37 seconds of computing time, indicating good efficiency in achieving optimal results. The optimal tree generated through hyperparameter tuning using Random Search is shown in **Figure 5**.

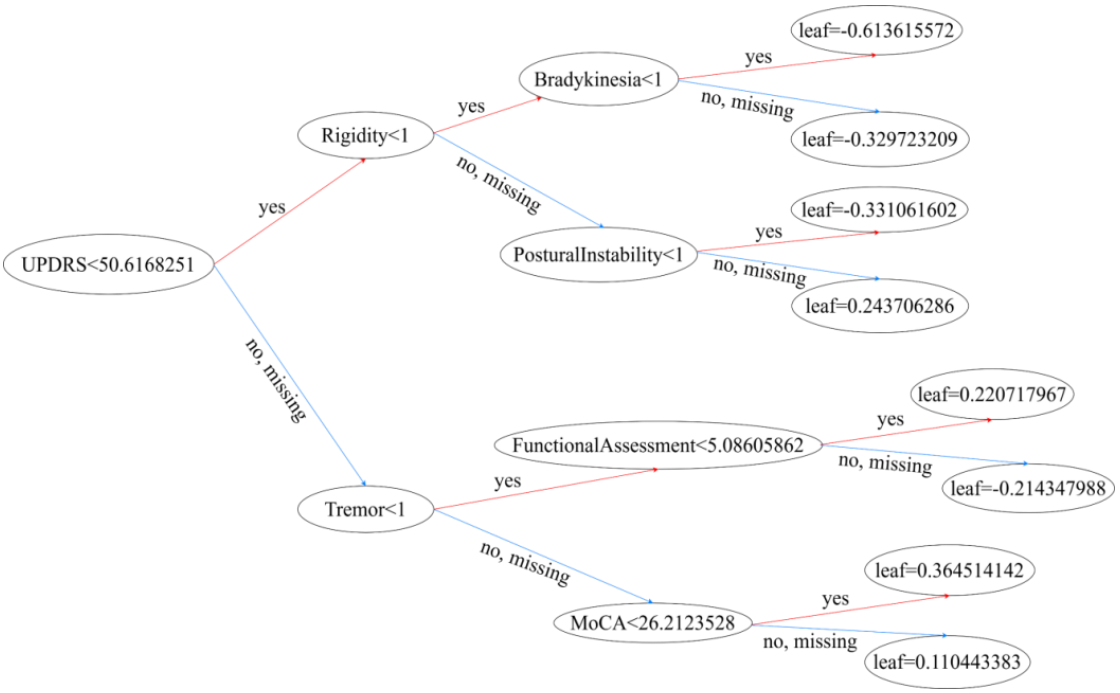

Figure 5. The Optimal Tree Result of Random Search

Figure 5 shows that the tree starts with the evaluation UPDRS < 50.617. For a correct Rigidity < 1, the tree checks Bradykinesia < 1, resulting in leaf -0.614 (negative class) if correct and leaf -0.330 (negative class) if not. If Rigidity ≥ 1 is incorrectly evaluated, the tree evaluates PosturalInstability < 1, leading to leaf -0.331 (negative class) if true and leaf 0.244 (positive class) otherwise. On the main right branch (UPDRS ≥ 50.617), when incorrectly evaluated, the tree evaluates Tremor < 1. If correct, it checks FunctionalAssessment < 5.086, resulting in leaf 0.221 (positive class) if true and leaf -0.214 (negative class) otherwise. If Tremor ≥ 1 is incorrectly evaluated, the tree next evaluates MoCA < 26.212, resulting in leaf 0.365 (positive class) if true and leaf 0.110 (positive class) otherwise. UPDRS acts as the main feature in determining the initial classification. At the same time, Rigidity, Bradykinesia, Postural Instability, Tremor, Functional Assessment, and MoCA are used as internal nodes to differentiate patients based on their symptom characteristics. At each branching, decisions are made based on the values of these features, which ultimately lead to leaf nodes that show the final value to determine the patient's class.

Table 10. Result Comparison of Hyperparameter Tuning, Grid Search, and Random Search

	Grid Search	Random Search			
		50	100	500	1000
Accuracy	93.35%	93.97%	93.06%	93.21%	93.25%
Sensitivity	95.78%	96.63%	95.86%	95.78%	96.40%
Spesificity	88.14%	89.64%	88.51%	89.01%	89.14%
Time	44’ 51”	3’ 37”	6’ 44”	33’ 29”	1h 6’ 8”

Interestingly, from **Table 10**, Random Search proved to be more optimal than Grid Search in this study. Despite Grid Search achieving a comparable accuracy of 93.35% with carefully evaluated parameters, its 44-minute and 51-second computation time was significantly longer than Random Search, which achieved higher accuracy in less time. For instance, with 50 iterations, Random Search attained an accuracy of 93.97%, sensitivity of 96.63%, and specificity of 89.64% within just 3 minutes and 37 seconds. This efficiency stems from Random Search's ability to explore a broader range of hyperparameter combinations in a shorter time frame, as it does not exhaustively test every possible combination like Grid Search.

These findings suggest important implications for hyperparameter tuning strategies in machine learning applications. Given its efficiency and superior performance, Random Search is a more practical choice for optimizing complex models, particularly in resource-constrained environments. This is especially relevant for medical diagnostics and real-time applications, where minimizing computational overhead is crucial for timely decision-making. Moreover, the results highlight the importance of balancing accuracy with computational efficiency, reinforcing the need for adaptive search techniques that maximize performance while reducing processing time. Future research could explore hybrid approaches that combine the strengths of both methods, potentially enhancing hyperparameter tuning strategies for various machine learning models.

4. CONCLUSIONS

Testing with Random Search for 50 iterations resulted in optimal performance for Parkinson's disease classification using the XGBoost method, achieving an accuracy of 93.97%, sensitivity of 96.63%, and specificity of 89.64%. Compared to Grid Search, Random Search improved accuracy by 0.62%, sensitivity by 0.85%, and specificity by 0.5% while reducing computation time by 41 minutes and 14 seconds. These findings indicate that Random Search enhances model-tuning efficiency by exploring a broader range of parameter combinations, leading to improved classification accuracy and computational efficiency.

However, despite these advantages, Random Search has potential limitations. Since it selects hyperparameters randomly, there remains a risk of suboptimal parameter tuning, which could lead to inconsistent performance across different datasets. Additionally, while XGBoost inherently includes regularization techniques to mitigate overfitting, improper hyperparameter selection, especially with excessive boosting rounds or an overly complex tree structure, can still lead to overfitting, particularly on smaller datasets.

To address these limitations, future research could explore alternative hyperparameter optimization techniques, such as Bayesian Optimization or Tree-structured Parzen Estimators (TPE), which utilize

probabilistic models to refine the search process and potentially yield more optimal parameter configurations. Additionally, applying this model to more extensive and diverse datasets or extending its use to classify other neurological diseases could provide further insights into its generalizability and robustness in medical diagnostics.

In conclusion, while Random Search offers advantages in efficiency and performance, exploring alternative optimization strategies and testing the model on broader datasets can provide deeper insights into improving machine learning-based disease classification.

REFERENCES

[1] S. A. Tanazza and L. M. Erawati, "ANALISIS INTERVENSI FISIOTERAPI PADA PENYAKIT PARKINSON MENGGUNAKAN VOSVIEWER," *Physio J.*, vol. 2, no. 2, pp. 49–60, 2022, https://doi.org/10.30787/phyjou.v2i2.877.

[2] P. Tristiawati, *KENALI PENYEBAB, GEJALA, DAN PENGOBATAN PENYAKIT PARKINSON YANG PENDERITANYA CAPAI 400 RIBU DI INDONESIA*, vol. Liputan6. 2024, p. [Online]. Available: https://www.liputan6.com/health/read/5580264/kenali-penyebab-gejala-dan-pengobatan-penyakit-parkinson-yang-penderitanya-capai-400-ribu-di-indonesia

[3] I. S. Zein and Khairunnisa., "PARKINSON DISEASE," *American family physician*, vol. 102, no. 11, pp. 679–691, 2023. https://doi.org/10.55606/jurrike.v2i2.1701.

[4] A. Lees *et al.*, "OPTIMIZING LEVODOPA THERAPY, WHEN AND HOW? PERSPECTIVES ON THE IMPORTANCE OF DELIVERY AND THE POTENTIAL FOR AN EARLY COMBINATION APPROACH," *Expert Rev. Neurother.*, vol. 23, no. 1, pp. 15–24, 2023, https://doi.org/10.1080/14737175.2023.2176220.

[5] M. J. Armstrong and M. S. Okun, "DIAGNOSIS AND TREATMENT OF PARKINSON DISEASE: A REVIEW," *JAMA - J. Am. Med. Assoc.*, vol. 323, no. 6, pp. 548–560, 2020, https://doi.org/10.1001/jama.2019.22360.

[6] F. N. A. N. Hati, *MENELUSURI LABIRIN PARKINSON: DEFINISI, GEJALA, DAN TANTANGAN MASA KINI*, vol. RSA UGM. 2024, p. [Online]. Available: https://rsa.ugm.ac.id/2024/06/menelusuri-labirin-parkinson-definisi-gejala-dan-tantangan-masa-kini/

[7] J. Jankovic and E. K. Tan, "PARKINSON'S DISEASE: ETIOPATHOGENESIS AND TREATMENT," *J. Neurol. Neurosurg. Psychiatry*, vol. 91, no. 8, pp. 795–808, 2020, https://doi.org/10.1136/jnnp-2019-322338.

[8] F. F. Sutantoyo, P. Nugraha, and M. Hamdan, "GAMBARAN PASIEN PARKINSONISME VASKULAR: SEBUAH LAPORAN KASUS," *J. Aksona*, vol. 1, no. 5, pp. 140–145, 2020.

[9] M. I. Sodikin, "PENERAPAN DAN MANFAAT MACHINE LEARNING DI RUMAH SAKIT," *Multiverse Open Multidiscip. J.*, vol. 2, no. 2, pp. 262–265, 2023, https://doi.org/10.57251/multiverse.v2i2.1207.

[10] F. Nurrahman, H. Wijayanto, A. H. Wigena, and N. Nurjanah, "PRE-PROCESSING DATA ON MULTICLASS CLASSIFICATION OF ANEMIA AND IRON DEFICIENCY WITH THE XGBOOST METHOD," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 2, pp. 0767–0774, 2023, https://doi.org/10.30598/barekengvol17iss2pp0767-0774.

[11] Muhamad Fikri, "KLASIFIKASI STATUS STUNTING PADA ANAK BAWAH LIMA TAHUN MENGGUNAKAN EXTREME GRADIENT BOOSTING," *Merkurius J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 4, pp. 173–184, 2024, https://doi.org/10.61132/merkurius.v2i4.159.

[12] R. Harahap *et al.*, "PERBANDINGAN KINERJA ALGORITMA RANDOM FOREST DAN XGBOOST DALAM KLASIFIKASI PENYAKIT PARU-PARU BERDASARKAN DATA DEMOGRAFI PASIEN," no. 02, pp. 130–141.

[13] Y. Amelia, "PERBANDINGAN METODE MACHINE LEARNING UNTUK MENDETEKSI PENYAKIT JANTUNG," *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 220–225, 2023, https://doi.org/10.36080/idealis.v6i2.3043.

[14] G. Abdurrahman, H. Oktavianto, and M. Sintawati, "OPTIMASI ALGORITMA XGBOOST CLASSIFIER MENGGUNAKAN HYPERPARAMETER GRIDESEARCH DAN RANDOM SEARCH PADA KLASIFIKASI PENYAKIT DIABETES," *INFORMAL Informatics J.*, vol. 7, no. 3, p. 193, 2022, https://doi.org/10.19184/isj.v7i3.35441.

[15] R. J. Alfirdausy, N. Ulinnuha, and W. D. Utami, "IMPLEMENTATION OF THE EXTREME GRADIENT BOOSTING ALGORITHM WITH HYPERPARAMETER TUNING IN CELIAC DISEASE CLASSIFICATION," vol. 24, no. 1, pp. 117–128, 2024, https://doi.org/10.30812/matrik.v24i1.4031.

[16] D. Kurnia, M. Itqan Mazdadi, D. Kartini, R. Adi Nugroho, and F. Abadi, "SELEKSI FITUR DENGAN PARTICLE SWARM OPTIMIZATION PADA KLASIFIKASI PENYAKIT PARKINSON MENGGUNAKAN XGBOOST," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 5, pp. 1083–1094, 2023, https://doi.org/10.25126/jtiik.20231057252.

[17] M. D. Mubarak, A. Afdal, D. Pertiwi, N. R. Masnadi, Z. D. Rofinda, and T. Handayani, "GAMBARAN GEJALA DEPRESI PADA PENDERITA PARKINSON DISEASE DI RSI IBNU SINA PADANG," *J. Ilmu Kesehat. Indones.*, vol. 5, no. 2, pp. 170–177, 2024, https://doi.org/10.25077/jikesi.v5i2.1153.

[18] A. Desiani, I. Ramayanti, M. Arhami, P. Studi Matematika, and F. Matematika dan Ilmu Pengetahuan Alam, "DIAGNOSIS OF PARKINSON'S DISEASE USING K-NEAREST NEIGHBOR AND DECISION TREE C4.5 ALGORITHMS," vol. 12, no. 1, pp. 47–58, 2023,https://doi.org/10.21107/simantec.v12i1.21167.

[19] R. el Kharoua, *PARKINSON'S DISEASE DATASET ANALYSIS*. 2024. [Online]. Available: https://www.kaggle.com/datasets/rabieelkharoua/parkinsons-disease-dataset-analysis?select=parkinsons_disease_data.csv

[20] S. Amelia, "PENERAPAN METODE MODIFIED K-NEAREST NEIGHBOR PADA PENGKLASIFIKASIAN STATUS PEMBAYARAN KREDIT BARANG ELEKTRONIK DAN FURNITURE," *Stat. J. Theor. Stat. Its Appl.*, vol. 22, no. 1, pp. 95–104, 2022, https://doi.org/10.29313/statistika.v22i1.345.

[21] Heliyanti Susana, "PENERAPAN MODEL KLASIFIKASI METODE NAIVE BAYES TERHADAP PENGGUNAAN AKSES INTERNET," *J. Ris. Sist. Inf. dan Teknol. Inf.*, vol. 4, no. 1, pp. 1–8, 2022,

https://doi.org/10.52005/jursistekni.v4i1.96.
[22] H. F. Putro, R. T. Vulandari, and W. L. Y. Saptomo, "PENERAPAN METODE NAIVE BAYES UNTUK KLASIFIKASI PELANGGAN," *J. Teknol. Inf. dan Komun.*, vol. 8, no. 2, 2020, https://doi.org/10.30646/tikomsin.v8i2.500.
[23] S. E. Herni Yulianti, Oni Soesanto, and Yuana Sukmawaty, "PENERAPAN METODE EXTREME GRADIENT BOOSTING (XGBOOST) PADA KLASIFIKASI NASABAH KARTU KREDIT," *J. Math. Theory Appl.*, vol. 4, no. 1, pp. 21–26, 2022, https://doi.org/10.31605/jomta.v4i1.1792.
[24] A. Shahraki, M. Abbasi, and Ø. Haugen, "BOOSTING ALGORITHMS FOR NETWORK INTRUSION DETECTION: A COMPARATIVE EVALUATION OF REAL ADABOOST, GENTLE ADABOOST AND MODEST ADABOOST," *Eng. Appl. Artif. Intell.*, vol. 94, no. February, p. 103770, 2020, https://doi.org/10.1016/j.engappai.2020.103770.
[25] C. Wang, C. Deng, and S. Wang, "IMBALANCE-XGBOOST: LEVERAGING WEIGHTED AND FOCAL LOSSES FOR BINARY LABEL-IMBALANCED CLASSIFICATION WITH XGBOOST," *Pattern Recognit. Lett.*, vol. 136, pp. 190–197, 2020, https://doi.org/10.1016/j.patrec.2020.05.035.
[26] A. F. A. Naibaho and A. Zahra, "PREDIKSI KELULUSAN SISWA SEKOLAH MENENGAH PERTAMA MENGGUNAKAN MACHINE LEARNING," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 173–183, 2023, https://doi.org/10.23960/jitet.v11i3.3056.
[27] M. W. Dwinanda, N. Satyahadewi, and W. Andani, "CLASSIFICATION OF STUDENT GRADUATION STATUS USING XGBOOST ALGORITHM," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 3, pp. 1785–1794, 2023, https://doi.org/10.30598/barekengvol17iss3pp1785-1794.
[28] W. Nugraha and A. Sasongko, "HYPERPARAMETER TUNING PADA ALGORITMA KLASIFIKASI DENGAN GRID SEARCH HYPERPARAMETER TUNING ON CLASSIFICATION ALGORITHM WITH GRID SEARCH," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, 2022, [Online]. Available: http://sistemasi.ftik.unisi.ac.id
[29] U. Sunarya and T. Haryanti, "PERBANDINGAN KINERJA ALGORITMA OPTIMASI PADA METODE RANDOM FOREST UNTUK DETEKSI KEGAGALAN JANTUNG," *J. Rekayasa Elektr.*, vol. 18, no. 4, pp. 241–247, 2022, https://doi.org/10.17529/jre.v18i4.26981.
[30] M. A. Abubakar, M. Muliadi, A. Farmadi, R. Herteno, and R. Ramadhani, "RANDOM FOREST DENGAN RANDOM SEARCH TERHADAP KETIDAKSEIMBANGAN KELAS PADA PREDIKSI GAGAL JANTUNG," *J. Inform.*, vol. 10, no. 1, pp. 13–18, 2023, https://doi.org/10.31294/inf.v10i1.14531.
[31] D. Normawati and S. A. Prayogi, "IMPLEMENTASI NAÏVE BAYES CLASSIFIER DAN CONFUSION MATRIX PADA ANALISIS SENTIMEN BERBASIS TEKS PADA TWITTER," *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 697–711, 2021, https://doi.org/http://dx.doi.org/10.30645/j-sakti.v5i2.369.
[32] R. S. Amardita, A. Adiwijaya, and M. D. Purbolaksono, "ANALISIS SENTIMEN TERHADAP ULASAN PARIS VAN JAVA RESORT LIFESTYLE PLACE DI KOTA BANDUNG MENGGUNAKAN ALGORITMA KNN," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 1, p. 62, 2022, https://doi.org/10.30865/jurikom.v9i1.3793.
[33] Yurivo Rianda Saputra, Syafriandi Syafriandi, Dony Permana, and Zilrahmi, "CLASSIFICATION OF PROGRAM KELUARGA HARAPAN RECIPIENT HOUSEHOLDS IN PADANG USING K-NEAREST NEIGHBORS," *UNP J. Stat. Data Sci.*, vol. 2, no. 2, pp. 187–195, 2024, https://doi.org/10.24036/ujsds/vol2-iss2/167.