Scientific Journal of Informatics
Vol. 11, No. 3, Aug 2024
p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Hyperparameter Tuning Decision Tree and Recursive Feature Elimination Technique for Improved Chronic Kidney Disease Classification

Aries Gilang Saputra[1*], Purwanto[2], Pujiono[3]
[1, 2, 3]Master's Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia

Abstract.
Purpose: This study has the purpose of classifying patients with chronic kidney disease based on specific features and improving the classification models by tuning hyperparameters. This study aims to detect chronic kidney disease at an early stage.
Method: In this study, a machine learning classifier in the form of a decision tree was used to classify chronic kidney disease on the Risk Factor Prediction of Chronic Kidney Disease dataset. After that, the performance of the classifier model was improved by using feature selection, namely Recursive Feature Elimination and Hyperparameter tuning with GridSearchCV.
Result: After tests were conducted 3 times namely testing with Decision Tree, Recursive Feature Elimination, and Hyperparameter tuning GridSearchCV which was the proposed method, then compared to other tests. The results from this study showed that using that method can improve the Decision Tree classifier in classifying chronic kidney disease patients.
Novelty: The dataset used in this study was from the UCI machine learning repository namely Risk Factor Prediction of Chronic Kidney Disease that have 202 instances and 28 features and after being processed and conducting the test, Recursive Feature Elimination and Hyperparameter tuning GridSearchCV can improve the Decision Tree classifier in classifying chronic kidney disease.

Keywords: Chronic kidney disease, Decision tree, Grid search cv, Hyperparameter tuning, Recursive feature elimination
Received August 2024 / **Revised** September 2024 / **Accepted** October 2024

INTRODUCTION

Chronic kidney disease is a disease that should not be taken lightly. According to [1] When a patient has kidney function that has worsened from the condition of normal kidney function in general, this condition is called chronic kidney disease. This is motivated by the fact that according to [2] the kidneys have a very important and main function namely, to filter excess water and waste from the blood and then dispose of them through the process of forming urine if this kidney function does not work well, as in patients with chronic kidney disease, the waste will accumulate in the body. Furthermore, [2] explained that because this disease has been damaged over a long time and gradually, this disease is categorized as a chronic disease.

According to [2], [3] this chronic kidney disease has no symptoms and cannot be diagnosed until the kidneys are severely damaged and cannot be repaired. Several ways can be done to check the condition of chronic kidney disease as mentioned in [2], [4], [5], namely through 3 diagnostic tests in the laboratory carried out by a doctor, including a blood test to determine eGFR (Estimated Glomerular Filtration Rate) in the blood, urine tests to check the Albumin-Creatinine ratio, and blood pressure tests. Apart from that, [3], [5] stated that chronic kidney disease does not require costs small, and these costs apply to patients, health services, and the government. Therefore, [3] said that early detection and prevention are very important to prevent chronic kidney disease from reaching the final stage. To do this, the role of data mining is needed because data mining techniques have been widely used to determine new and understandable patterns that are used to build classification templates [4]. Data mining can be used to mine data on datasets that have very large data, but if the data is small, machine learning is needed to do this [2].

*Corresponding author.
Email addresses: p31202102474@mhs.dinus.ac.id (Saputra)
DOI: 10.15294/sji.v11i3.12990

The classification method that was used in this study is one of the methods found in machine learning models. Decision trees are a very well-known nonparametric type of supervised learning method and can be used to carry out classification [1], [6]. A decision tree is a classification that utilizes a pattern of the tree to decide, it has three main parts, namely root, branches, nodes, and leaves, one type of decision tree is CART, which utilizes the Gini index to create a binary tree and is highly applicable [7]. This method is used because [1] explains that machine learning is often used to translate available data and then process it into useful knowledge, carry out analyses related to various diseases, and diagnose disorders that patients have. Previous Research that used machine learning classifiers related to this disease was carried out by [3] which diagnosed chronic kidney disease in patients using a deep neural network based on a multi-layer perceptron classifier and the performance results were compared with machine learning methods such as Random Forest, Decision Trees, SVM, and Logistic Regression. The result of that research is that Artificial Neural networks have better performance than the machine learning method used as a comparison in that research. In addition, research by [8] also used machine learning algorithms to predict and calculate risk factors for chronic kidney disease. The results of the research found that random forest has the highest accuracy compared to other algorithms and the most significant risk factor for chronic kidney disease is hemoglobin while hypertension has a less significant risk factor for chronic kidney disease.

In this study, Machine learning classifiers are not enough. Because of that, a feature selection method is needed to select important features, such as recursive feature elimination. This method can be used to select relevant features by repeatedly eliminating irrelevant features [6]. This method utilizes a trained model and the accuracy of the classification to select relevant features, categorized as a wrapper method because it uses a supervised method and is repetitively wrapped to eliminate the irrelevant features based on the features dataset [9]. Previous Research using this method was conducted by [4], where important features were selected using the recursive feature elimination (RFE) method and Chronic Kidney Disease UCI Machine learning Repository dataset, and then after that, the selected features were tested using classifiers to diagnose chronic kidney diseases such as SVM, KNN, and decision tree. The results of that research show that the classifier used to diagnose Chronic Kidney Disease has good performance as seen from the values produced by Evaluation Metrics when evaluating the performance of each classifier. When compared to the approach methods in previous studies, the method proposed in this study also has convincing performance. In this study, this feature selection method is included in the wrapper method, which uses a classification algorithm to select relevant features and can produce better accuracy than other methods [6], [10]. Besides that, [2] predicted chronic kidney disease using three different feature selection methods, namely the Wrapper Method, Embedded Method (LASSO Regression), and Filter Method (CFS (Correlation-based Feature Selection). This method was also used by [1] combined with a deep neural network, and the results are compared with other machine learning classifiers.

Utilizing hyperparameter tuning can also be used to improve the performance of the classifier model. Hyperparameters are sets of parameters used to describe the design and architecture of the model used, the method used to improve the accuracy of the model is called hyperparameter tuning. Finding the optimal model design requires finding the best hyperparameters [11]. There are several hyperparameter tuning methods such as random search which was used in the research by [12] which shows research related to chronic kidney disease that uses several machine learning classifiers. In addition, other hyperparameter tuning methods such as grid search were conducted by [13] who made predictions for chronic kidney disease using machine learning specifically for small and imbalanced datasets. Grid Search CV can be used to find optimal values of model hyperparameters because choosing the best hyperparameters will have a huge impact on model performance [14]. Grid Search CV combines grid search with cross-validation to reduce overfitting problems in the model to produce unbiased performance estimates [13], [15]. This Grid Search will be able to produce relatively optimal results because it carries out experiments on all combinations of predetermined hyperparameter sets so that it is more detailed and suitable for selecting combinations of small hyperparameter values and sets [16].

The performance of the classifier model can be calculated with the use of performance metrics such as accuracy, precision, recall, and f1-score, which can be used to help evaluate the classifier model [17]. The research mentioned previously commonly used performance metrics to evaluate their respective research. Calculation of correctly classified samples (TP and TN) as well as incorrectly classified samples (FP and FN) will produce this value [4]. True Positive means the sample is correctly predicted as a positive class, true negative means the sample is correctly predicted as a negative class, false positive means the sample is incorrectly predicted as a positive class, and false negative means the sample is incorrectly predicted as

a negative class [18]. Performance metrics are used to evaluate the method used to ensure that the model used matches the dataset and has good performance when tested on unseen data [19], [6].

This research uses a machine learning algorithm with a supervised learning type, namely a decision tree which is used to classify chronic kidney disease. Furthermore, this research also uses a feature selection method to select important and relevant features, namely by using a recursive feature elimination method. In this study, the researcher improved the performance of the classifier model by hyperparameter tuning the decision tree parameters. After testing the performance of the classifier, an evaluation will be conducted with performance metrics such as accuracy, precision, recall, and f1-score.

METHODS

The research procedure for this study consisted of data pre-processing and feature selection using RFE, then hyperparameter tuning was performed, and further classification was performed with a decision tree classifier. After that, performance evaluation was performed using performance evaluation metrics. The results of the proposed method test will be compared with two other tests through the results of the performance evaluation metrics from each test. This test was performed by using Visual Studio Code with the Python programming language. The dataset used for this study was a dataset from the University of California, Irvine Machine Learning Repository, which is available on the UCI website. The dataset employed was Risk Factor Prediction of Chronic Kidney Disease which has 28 features with one target class, namely ckd, and notckd. This dataset was collected from Enam Medical College, Savar, Dhaka, Bangladesh which was patient data in Bangladesh. This dataset has a total of 202 instances which is the number of chronic kidney disease patients.

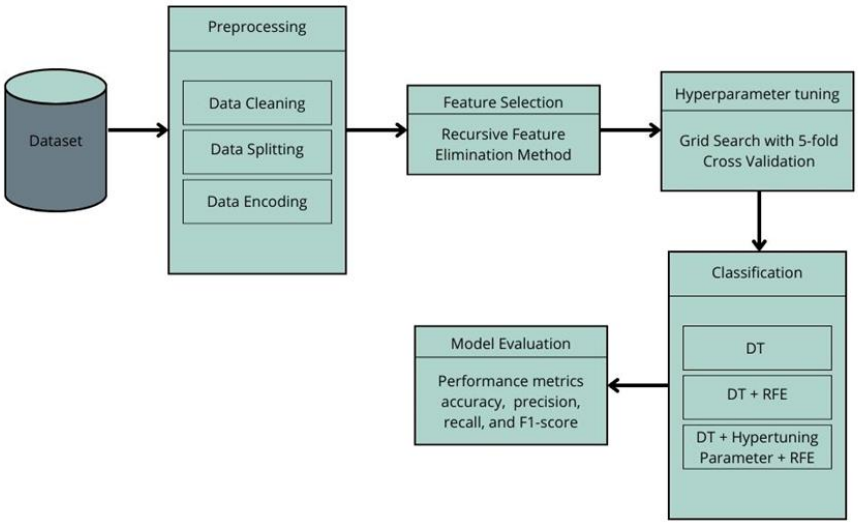

Figure 1. Proposed model

Data preprocessing

The dataset used in this study required data splitting to ensure that the model performs training from a portion of the data that has been set and then evaluation will be carried out using unseen data to assess the generalization performance of the model [18]. This data split works by dividing the dataset used in this study into training data and testing data in the form of percentages for each training data and testing data.

Cleaning the dataset is required, particularly from data that can affect the performance of the classifier. In this study, data cleaning was performed on the dataset by deleting missing values and duplicating data because when the dataset is entered into the software for data processing there are NaN values that are included and counted as missing values or undefined values. Therefore, action needs to be taken. In this study, the action taken to handle the NaN values was to delete them because the values are not relevant to the data in the dataset. In addition, for duplicate data in this dataset, deletion was conducted related to the data for good quality dataset.

Since most machine learning algorithms only accept numeric values as input, values other than numerical must be encoded to become numerical values to be processed further [1]. In the dataset used in this study, there are categorical features. Therefore, categorical features will be encoded into numerical data using the

One-Hot Encoder. For the target variable class, encoding into numerical values will also be performed using the label encoder because the target class is categorical data. The selection feature is needed to select important features that have a strong correlation with the label and remove irrelevant features that can affect the performance of the classifier model [4]. This study conducted the recursive feature elimination selection feature to perform this task. This method is included in the wrapper method, where this wrapper method can remove redundant features and is suitable for non-linear classifiers such as decision trees. The dataset used in this study has 27 features, recursive feature elimination will process the 27 features until the remaining important and relevant features are selected.

This method has two configuration options provided, the first RFE determines the number of features selected, and the second RFE determines the machine learning algorithm that selects the features [20]. The first configuration works by finding a subset of features by considering all features contained in the training dataset and will remove features until the predetermined features are reached, while the second configuration works by using machine learning algorithms and ranking features based on their importance where the way it works is removing features that are not important and then repeating the process of fitting the model, and this entire process will repeat until the remaining number of predetermined features is achieved [20]. In this study, the second configuration is conducted to select relevant features.

Best hyperparameters in the classifier are carried out to improve the performance of the classifier model. In this study, a grid search will be conducted to search for the best hyperparameters, and cross-validation will be used to validate by evaluating the performance of the hyperparameters to validate that the hyperparameters selected are the best. Determining the selected hyperparameters can reduce overfitting by using hyperparameters with stopping criteria. This technique simply works by evaluating using cross-validation on each combination of hyperparameters that have been previously determined by the researcher and then checking each combination until the best combination is found. Before conducting experiments on each hyperparameter combination, a parameter grid containing the range of values of each hyperparameter will be determined first because there are hyperparameters that can have unlimited values [21]. Determination of the range is required because the wider the range and the smaller the interval will make it easier to find the optimal solution, namely the best hyperparameter combination [16]. This method aims to find the best hyperparameters from the parameters included in the decision tree model, namely criterion, max_depth, min_samples_split, and min_samples_leaf. Some of which previously mentioned are stopping criteria that can reduce overfitting on the data. Hyperparameters besides the criterion are in the form of values, but for criterion is the choice between Gini, entropy, and information gain that determined internal node split criteria.

Data classification

The algorithm used for classification in this study was a decision tree. A decision tree is a supervised learning method that is nonparametric and has two types of nodes, namely decision nodes and leaf nodes [1]. The way this decision tree works is by comparing the features in the root node with the feature records in the actual dataset, then based on the results of the comparison, it will determine the decision to move to the next node and then continue by comparing the features in the second node with the features in the sub node and continue until reaching the leaf node [4]. The result of the decision tree is leaf nodes [1]. The decision tree in this study is a CART (Classification and Regression Tree) type, where this type will produce a binary tree where each internal node has only two branches and the internal node split criteria use the Gini criterion [22], [23] . This CART can be used for target variables with discrete and continuous types, can be pruned, handle categorical and numerical values, and can handle outliers in the data. Because of that, CART has many advantages [22] - [24]. [25]. [23]

This study is a study conducted to classify chronic kidney disease patients. The procedure of this study consists of data preprocessing, feature selection using RFE, and then using a classifier, namely a decision tree that has been hyperparameter tuned with grid search and validated using cross-validation to find the best combination of hyperparameters furthermore classification, and prediction are carried out and the performance of the classifier is evaluated using performance metrics such as accuracy, precision, recall, f1-score. The diagram of the proposed research method in this study can be seen in Figure 1.

Evaluation

In this study, evaluating the performance of the Machine Learning Classifier can be done by using performance metrics such as accuracy, precision, recall, and F1-score. The calculation of the confusion

matrix values must first be carried out to get performance metrics values. The confusion matrix is True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In addition, because in this study, the test will be carried out three times, a performance evaluation will be conducted for each test to be able to compare the performance of the proposed method with other tests. Thus, the performance results of the proposed method can be known.

RESULTS AND DISCUSSIONS

Data preprocessing

In this study, splitting the data was done by dividing the dataset into two parts, namely training data and testing data. Data splitting of training data and testing data was divided into 80% as training data and 20% as testing data. This was obtained through observations of different ratios, and 80:20 is the correct ratio.

Table 1 shows the results of data splitting, one of which is training data. The training data produced was 161 instances because initially there were 202 instances, then one feature was deleted, namely affected, and because of that, the feature was reduced to 27. Furthermore, the data was split 80% for training data, and the training data obtained was 161 instances. After that, data splitting was not only split into training data but also testing data. Samples of testing data produced through data splitting can be seen in Table 2. The testing data obtained was 41 instances. This data was obtained through the initial data split, the same as in the training data, namely after deleting the affected feature, and because the ratio of this testing data was 20%. In the dataset used in the study, in addition to duplicate data, there were also missing values in the form of NaN values. Many methods can be used to handle these missing values, such as using the mean, median, and mode methods where these methods are used according to the data type of the missing values. In addition, there is also another method that can be used, namely by using the dropna() function method found in one of the Python libraries, namely pandas. This method is specifically designed to delete missing values contained in a dataset. Utilizing this method, the NaN data can be deleted from the dataset. Before the deletion of missing values, there were 202 instances, then after the deletion of missing values became 160 instances. Table 3 shows the data after the deletion of missing values. After handling the missing values, the rows that had NaN data were no longer visible.

Table 1. Data training sample

	Bp (Diastolic)	Bp *Limit*	Sg	Al	…	ane	grf	st*age*	*age*
0	*discrete*	*discrete*	*discrete*	*discrete*	…	*discrete*	*discrete*	*discrete*	*discrete*
1	*NaN*	*NaN*	*NaN*	*NaN*	…	*NaN*	*NaN*	*class*	*NaN*
2	1	1	≥ 1.023	< 0	…	0	51.7832–76.949	s2	≥ 74
3	0	0	≥ 1.023	< 0	…	0	152.446177.612	s1	35 - 43
4	0	0	≥ 1.023	< 0	…	0	102.115127.281	s1	27 - 35
…	…	…	…	…	…	…	…	…	…
160	1	1	≥ 1.023	< 0	…	0	51.7832–76.949	s2	51 - 59

Table 2. Data testing sample

	Bp (Diastolic)	Bp *Limit*	Sg	Al	…	ane	grf	st*age*	*age*
0	1	2	≥ 1.023	01-01	…	1	51.7832 – 76.949	s2	51- 59
1	0	0	1.015-1.017	02-02	…	0	76.949 – 102.115	s2	20-27
2	0	0	≥ 1.023	< 0	…	0	177.612– 202.778	s1	27-35
3	0	0	1.009-1.011	01-01	…	0	<2 6.6175	s4	66-74
4	1	1	1.019-1.021	02-02	…	0	< 26.6175	s5	66-74
…	…	…	…	…	…	…	…	…	…
40	1	1	1.019-1.021	< 0	…	0	51.7832 – 76.949	s2	51-59

Table 2. Data sample after handling missing values

Index	Bp (Diastolic)	Bp Limit	sg	Al	rbc	su	pc	pcc	ba
0	discrete	discrete	discrete	discrete	discrete	discrete	discrete	discrete	discrete
1	1	1	≥ 1.023	<0	0	<0	0	0	0
2	0	0	≥ 1.023	<0	0	<0	0	0	0
3	0	0	≥ 1.023	<0	0	<0	0	0	0
4	0	0	1.009-1.011	<0	0	<0	0	0	0
5	1	2	1.019-1.021	<0	0	<0	0	0	0

In addition to deleting NaN values in the dataset, deletion of unused and recurring data that exist, such as “discrete” instances, was carried out in this study. The method used to delete these instances was by using the drop() function method found in one of the Python libraries, namely pandas, which is almost the same as dropna() but this function can delete the desired rows or columns in the data frame, including deleting one of the features in the dataset that has similarities with the target variable “class” namely the "affected" feature. There were rows with discrete instances that need to be deleted to improve the quality of the dataset. Table 4 shows the data after the deletion of duplicate data. By deleting the duplicate data, the instances in the dataset had now changed to 159 instances from the previous 160 instances. After deleting duplicate data, the current dataset had been free from missing values and duplicate data.

This dataset has features with categorical and numerical data types. Features with categorical types are then encoded so that further processing can be conducted. These features must be encoded into numeric data because there is a string in the data that causes the machine learning algorithm to be unable to process this dataset, so it is necessary to encode it into numeric data. The method used in this study to encode categorical data into numeric data was by using One Hot Encoding. This method can convert data that has a categorical data type into a binary vector where it is given a value of 1 if it is included in the appropriate category, and if it is not included in the category, it will be given a value of 0. Table 5 shows a sample dataset after encoding using one hot encoding.

Table 4. Data sample after deleting duplicate data

Index	Bp (Diastolic)	Bp Limit	sg	Al	rbc	su	pc	pcc	ba
0	1	1	≥ 1.023	<0	0	<0	0	0	0
1	0	0	≥ 1.023	<0	0	<0	0	0	0
2	0	0	≥ 1.023	<0	0	<0	0	0	0
3	0	0	1.009-1.011	<0	0	<0	0	0	0
4	1	2	1.019-1.021	<0	0	<0	0	0	0
5	1	1	≥ 1023	<0	0	<0	0	0	0

Table 5. Data sample result of one hot encoding

Index	sg_<1007	sg_≥1.023	bgr_112-154	bgr_154-196	sod_118-123	sod_123-128	hemo_10-11.3
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	1.0	0.0	0.0	0.0	0.0
2	0.0	1.0	1.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	1.0	0.0	0.0	0.0
5	0.0	1.0	1.0	0.0	0.0	0.0	0.0

Feature selection was conducted using the Recursive Feature Elimination method. In this study, each number of features was observed along with the criteria and accuracy produced to see the best number of features that should be used to improve the classifier model. Table 6 shows the number of features and feature names selected through the recursive feature elimination method. Therefore, the number of features used by the model was five features. The five features were selected after conducting the feature ranking in the Recursive Feature Elimination method. According to this method, hypertension, diabetes mellitus, hemoglobin, albumin, and Red Blood Cell Count were relevant and important features in classifying chronic kidney disease.

Furthermore, in this study, tuning was done on the hyperparameters using the Grid Search Cross Validation method. The hyperparameters that will be tuned are four hyperparameters, namely max_depth, min_samples_split, min_samples_leaf, and criteria. Three of the four hyperparameters are stopping criteria that can be used to prevent overfitting of the data. The max_depth hyperparameter will determine the depth of the tree. The depth of the tree can be seen through the number of branches in the tree, the more branches, the deeper the tree, but it will allow overfitting of the data. The min_samples_split hyperparameter will determine the minimum number of samples needed to be able to split an internal node or can be used as the minimum sample needed to become a leaf node. Furthermore, the min_samples_leaf hyperparameter will be the opposite of min_samples_split because min_samples_leaf will determine the minimum number of samples needed to become a leaf node. The last hyperparameter is the criterion where this hyperparameter will be used to determine the criteria for splitting the tree [25].

The four hyperparameters were entered into the grid parameter according to the range of hyperparameter values to be determined, and the Grid Search method will work by creating a model for each combination of hyperparameter values, then training and evaluation will be carried out and validated using 5-fold cross-validation to get a model containing the best combination of hyperparameter values. Table 7 shows the selected hyperparameter values. According to this method, the criteria using gini impurity, max_depth 5, min_samples_leaf 2, and min_samples_split 2 were the best hyperparameter combinations for the model in this study. These hyperparameters will be used by the classifier model, namely the decision tree, to perform classification and will affect the tree that is built, especially because three of the four hyperparameters are included in the stopping criteria that can prevent overfitting.

Data classification

After getting the best combination of hyperparameters and important and relevant features, the hyperparameters and features will then be used by the classifier model, namely the decision tree, to perform the classification. Because the variant of the decision tree is CART (Classification and Regression Tree), the determination of the root node and split on the internal node uses gini impurity.

Table 3. Selected feature using recursive feature elimination

Feature Name	Feature Name Extension	Data Type
htn	*Hypertension*	*Integer*
dm	*Diabetes mellitus*	*Integer*
hemo_11.3 - 12.6	*Hemoglobin*	*Categorical*
al_< 0	*Albumin*	*Categorical*
Rbcc_6.23 – 6.82	*Red blood cell count*	*Categorical*

Table 4. Selected hyperparameter

No	Hyperparameter Name	Hyperparameter Values
1.	*criterion*	*Gini*
2.	*max_depth*	5
3.	min_*samples_leaf*	2
4.	*min_samples_split*	2

After classification, prediction and evaluation were conducted with performance evaluation metrics. However, before calculating the performance evaluation metrics, it is necessary to obtain the results of the confusion matrix first. Table 8 is the confusion matrix of the proposed method. The confusion matrix value from this test where true positive (TP) had a value of 13, then true negative (TN) had a value of 27, false positive (FP) had a value of 1, and false negative (FN) had a value of 0. After obtaining this confusion matrix value, it was used to calculate the performance evaluation metrics such as accuracy, precision, recall, and f1 score. Table 9 contains the performance evaluation metrics of the proposed method in this study. The table shows that the accuracy produced was 97.56%, with a precision of 92.85%, and recall of 100%, and an f1-score of 96.29%. In addition, a calculation was made for train accuracy to check for overfitting on the data, and the result was a train accuracy of 97.48%. Because it was almost the same as the test accuracy, there was no overfitting on the data.

The purpose of conducting methods explained previously is to solve problems that exist in this study. Risk Factor Prediction of Chronic Kidney Disease dataset has an irrelevant feature that can affect the classification of chronic kidney disease patients. Therefore, performing Recursive Feature Elimination to select relevant features can overcome this problem, and performing hyperparameter tuning to select the best hyperparameter combination can improve the classification performance of the decision tree based on the variables in the dataset. From the result, it is known that conducting previously explained methods can reduce the overfitting problem in the data because train accuracy and test accuracy have similar results of 97.48% and 97.56% and improve the classifier model with an accuracy of 97.56%. Furthermore, to prove the improvements that have occurred, that result will be compared with another test.

Evaluation

In the evaluation section, another test will be conducted which is used as a comparison to the proposed method and proves that there is an improvement in classification. This test will be performed two times. The first one is a decision tree without selection features and hyperparameter tuning and the second one uses a decision tree and selection features Recursive Feature Elimination. After that, a comparison of the

two tests will be carried out with the proposed method and a discussion of the results will be done. The comparison is carried out through the evaluation performance matrix obtained in each experiment.

In this test for comparison with the proposed method, classification is carried out using a decision tree classifier without Recursive Feature Selection and Grid Search CV. After calculating the confusion matrix in Table 8, the evaluation performance matrix of this test is obtained. The accuracy obtained was 95.12%, the precision value obtained was 92.30%, the recall value obtained was 92.30%, and the f1-score value obtained was 92.30%. In this test without RFE and GridSearchCV, the training accuracy obtained was 100%, so it can be concluded that overfitting occurs in the data.

Table 5. Performance evaluation matrix without Recursive Feature Elimination and Grid Search CV

No	Performance Evaluation Matrix	Result
1.	*Accuracy*	0.9512
2.	*Precision*	0.9230
3.	*Recall*	0.9230
4.	*F1-score*	0.9230

The second test was conducted utilizing a decision tree and features selection Recursive Feature Elimination. As with the preceding test, the calculation of the confusion matrix was conducted, and the performance evaluation matrix value from the second test was obtained and presented in Table 9. As can be observed in the table, the resulting accuracy value was 95.12%, the precision value was 87%, the recall value was 100%, and the F1-score value was 92.85%. The accuracy obtained for training was 97.56%. In the initial test of accuracy training, it can be observed that there is overfitting in the data. Conversely, in this second test, the accuracy of the training decreases, indicating reduced overfitting in the data.

Table 6. Performance evaluation matrix second test

No	Performance Evaluation Matrix	Result
1.	*Accuracy*	0.9512
2.	*Precision*	0.87
3.	*Recall*	1.0
4.	*F1-score*	0.9285

Therefore, the addition of a hyperparameter tuning method with grid search CV, such as the proposed method, will likely yield more favorable results. The results of the proposed method can be seen in Table 10 and 11.

Table 7. Proposed method Confusion Matrix

Confusion Matrix	Negative	Positive
Negative	27	1
Positive	0	13

Table 8. Performance evaluation matrix proposed method

No	Performance Evaluation Matrix	Result
1.	*Accuracy*	0.9756
2.	*Precision*	0.9285
3.	*Recall*	1.0
4.	*F1-score*	0.9629

Comparing the proposed method with the test without RFE and GridSearchCV and second tests where in the test without RFE and GridSearchCV, the precision and f1-score had a value of 92.30%, then in the second test, 87% and 92.85%, in the proposed method, the precision and f1-score increased to 92.85%, and 96.29% respectively. The recall value between the second test and the proposed method was the same at 100%. In the test without RFE and GridSearchCV, the resulting train accuracy was 100% with a test accuracy of 95.12%, so there was overfitting in the data. In the second test, the resulting train accuracy was 97.56% with a test accuracy of 95.12% where the training accuracy decreased. Besides that, the proposed method had a train accuracy result of 97.48% with a test accuracy of 97.56%, so there was no overfitting in the data, and accuracy had increased. Figure 2 illustrates the comparative performance evaluation matrix of the proposed method as compared to the other two tests. The results demonstrated that the proposed method produces higher results than test without RFE and GridSearchCV and second test. However, there was no improvement when comparing the test without RFE and GridSearchCV with the second test. There

was an increase in precision and f1-score values when comparing the test without RFE and GridSearchCV with the second test. Thus, using the recursive feature elimination and hyperparameter tuning grid search CV can reduce the overfitting problem in the data and improve the classifier model because there is an increase in the accuracy, precision, recall, and f1-score values between the test without RFE and GridSearchCV, the second test and the proposed method.

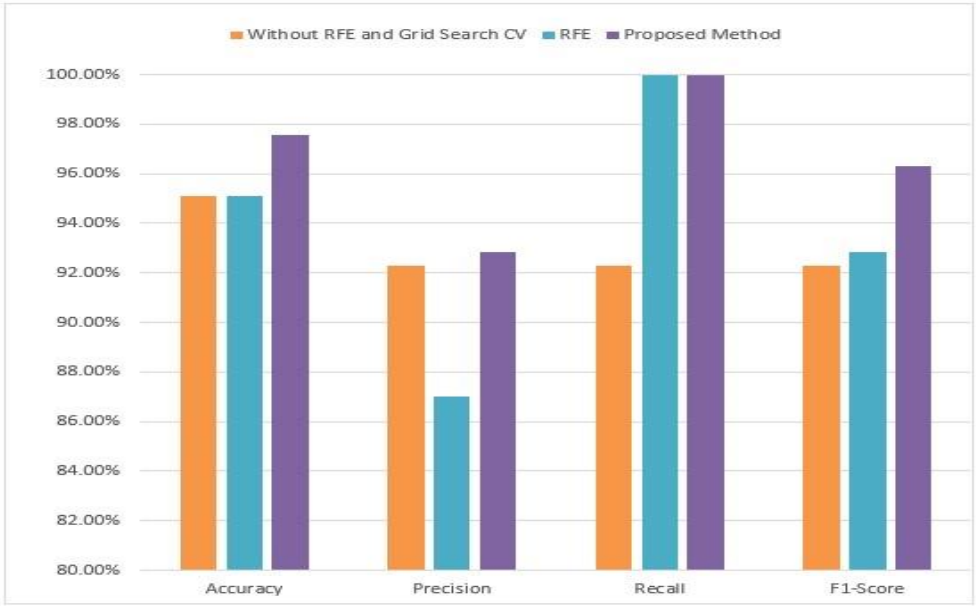

Figure 1. Comparison of Confusion Matrix between two test and proposed method

CONCLUSION

The results of this study showed that it is possible to improve the decision tree classifier model by using the recursive feature elimination selection method and hyperparameter tuning grid search cross-validation. Recursive feature elimination was conducted to select relevant features, and hyperparameter tuning was conducted to select the best combination of hyperparameters to improve classification performance based on variables in the dataset. It can be seen through the increase in accuracy of the proposed method when compared to the test without RFE and GridSearchCV and the second test. The accuracy in the first and second tests was 95.12%, but in the proposed method reached 97.56%.

REFERENCES

[1] V. Singh, V. K. Asari, and R. Rajasekaran, "A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease," *Diagnostics*, vol. 12, no. 1, Jan. 2022, doi: 10.3390/diagnostics12010116.

[2] P. Chittora *et al.*, "Prediction of Chronic Kidney Disease - A Machine Learning Perspective," 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2021.3053763.

[3] R. Sawhney, A. Malik, S. Sharma, and V. Narayan, "A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease," *Decision Analytics Journal*, vol. 6, Mar. 2023, doi: 10.1016/j.dajour.2023.100169.

[4] E. M. Senan *et al.*, "Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques," *J Healthc Eng*, vol. 2021, 2021, doi: 10.1155/2021/1004767.

[5] S. Krishnamurthy *et al.*, "Machine learning prediction models for chronic kidney disease using national health insurance claim data in Taiwan," *Healthcare (Switzerland)*, vol. 9, no. 5, May 2021, doi: 10.3390/healthcare9050546.

[6] D. A. Debal and T. M. Sitote, "Chronic kidney disease prediction using machine learning techniques," *J Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00657-5.

[7] U. Salamah, A. Ayu Ningrum, and M. Yunianto, "Classification of Early-Stage Lung Cancer Based on First and Second Order Statistical Variations Using the Gaussian Smoothing Filter and Decision Tree Method," *Scientific Journal of Informatics*, vol. 10, no. 4, 2023, doi: 10.15294/sji.v10i4.48026.

[8] M. A. Islam, S. Akter, M. S. Hossen, S. A. Keya, S. A. Tisha, and S. Hossain, "Risk factor prediction of chronic kidney disease based on machine learning algorithms," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 952–957. doi: 10.1109/ICISS49785.2020.9315878.

[9] F. E. P. Nadya, M. F. I. Ferdiansyah, V. R. S. Nastiti, and C. S. K. Aditya, "Implementation of Feature Selection Strategies to Enhance Classification Using XGBoost and Decision Tree," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 18–194, Feb. 2024, doi: 10.15294/sji.v11i1.48145.

[10] A. Al-Nafjan, "Feature selection of EEG signals in neuromarketing," *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/peerj-cs.944.

[11] D. Swain *et al.*, "A Robust Chronic Kidney Disease Classifier Using Machine Learning," *Electronics (Switzerland)*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010212.

[12] M. M. Nishat *et al.*, "A Comprehensive Analysis on Detecting Chronic Kidney Disease by Employing Machine Learning Algorithms," *EAI Endorsed Trans Pervasive Health Technol*, vol. 7, no. 29, Nov. 2021, doi: 10.4108/eai.13-8-2021.170671.

[13] A. C. M. da Silveira, Á. Sobrinho, L. D. da Silva, E. de B. Costa, M. E. Pinheiro, and A. Perkusich, "Exploring Early Prediction of Chronic Kidney Disease Using Machine Learning Algorithms for Small and Imbalanced Datasets," *Applied Sciences (Switzerland)*, vol. 12, no. 7, Apr. 2022, doi: 10.3390/app12073673.

[14] G. N. Ahmad, H. Fatima, Shafiullah, A. Salah Saidi, and Imdadullah, "Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques with and Without GridSearchCV," *IEEE Access*, vol. 10, pp. 80151–80173, 2022, doi: 10.1109/ACCESS.2022.3165792.

[15] G. Saranya and A. Pravin, "Grid Search based Optimum Feature Selection by Tuning hyperparameters for Heart Disease Diagnosis in Machine learning", doi: 10.2174/18741207-v17-e230510-2022-HT28-4371-8.

[16] H. S. Kang, Y. S. Choi, J. S. Yu, S. W. Jin, J. M. Lee, and Y. J. Kim, "Hyperparameter Tuning of OC-SVM for Industrial Gas Turbine Anomaly Detection," *Energies (Basel)*, vol. 15, no. 22, Nov. 2022, doi: 10.3390/en15228757.

[17] E. Dritsas and M. Trigka, "Machine Learning Techniques for Chronic Kidney Disease Risk Prediction," *Big Data and Cognitive Computing*, vol. 6, no. 3, Sep. 2022, doi: 10.3390/bdcc6030098.

[18] M. S. Arif, A. Mukheimer, and D. Asif, "Enhancing the Early Detection of Chronic Kidney Disease: A Robust Machine Learning Model," *Big Data and Cognitive Computing*, vol. 7, no. 3, Sep. 2023, doi: 10.3390/bdcc7030144.

[19] L. Antony *et al.*, "A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction," *IEEE Access*, vol. 9, pp. 126481–126501, 2021, doi: 10.1109/ACCESS.2021.3109168.

[20] R. C. Poonia *et al.*, "Intelligent Diagnostic Prediction and Classification Models for Detection of Kidney Disease," *Healthcare (Switzerland)*, vol. 10, no. 2, Feb. 2022, doi: 10.3390/healthcare10020371.

[21] L. Japa, M. Serqueira, I. Mendonca, M. Aritsugi, E. Bezerra, and P. H. Gonzalez, "A Population-Based Hybrid Approach for Hyperparameter Optimization of Neural Networks," *IEEE Access*, vol. 11, pp. 50752–50768, 2023, doi: 10.1109/ACCESS.2023.3277310.

[22] S. Moslehi, N. Rabiei, A. R. Soltanian, and M. Mamani, "Application of machine learning models based on decision trees in classifying the factors affecting mortality of COVID-19 patients in Hamadan, Iran," *BMC Med Inform Decis Mak*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01939-x.

[23] M. Hamdi, I. Hilali-Jaghdam, B. E. Elnaim, and A. A. Elhag, "Forecasting and classification of new cases of COVID 19 before vaccination using decision trees and Gaussian mixture model," *Alexandria Engineering Journal*, vol. 62, pp. 327–333, Jan. 2023, doi: 10.1016/j.aej.2022.07.011.

[24] Y. Yang, F. Yi, C. Deng, and G. Sun, "Performance Analysis of the CHAID Algorithm for Accuracy," *Mathematics*, vol. 11, no. 11, Jun. 2023, doi: 10.3390/math11112558.

[25] F. Aaboub, H. Chamlal, and T. Ouaderhman, "Statistical analysis of various splitting criteria for decision trees *," *J Algorithm Comput Technol*, vol. 17, Jan. 2023, doi: 10.1177/17483026231198181.