Improved Accuracy In Data Mining Decision Tree Classification Using Adaptive Boosting (Adaboost)

Muhammad Riansyah[1)]*, Saib Suwilo[2)], Muhammad Zarlis[3)]
[1)]Master of Informatics Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara
[2)] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sumatera Utara
[3)]Information Systems Management Department, BINUS Graduate Program – Master of Information Systems Management, Bina Nusantara University, Jakarta, 11480, Indonesia
[1)]riansyahmuhammad88@gmail.com, [2)]saib@usu.ac.id, [3)] muhammad.zarlis@binus.edu

Submitted : Dec 22, 2022 | **Accepted** : Feb 20, 2023 | **Published** : Apr 1, 2023

Abstract: The Decision Tree algorithm is a data mining method that is often applied as a solution to a problem in classification. The Decision Tree C5.0 algorithm has several weaknesses, including: The C5.0 algorithm and several other decision tree methods are often biased towards modeling whose features have many levels; some problems for the model can occur, such as over- or under-fit challenges, big changes to decision logic that result in small changes to data training, modeling inconvenience, and data imbalance that causes low accuracy in the C5.0 algorithm. The boosting algorithm is an iterative algorithm that gives different weights to the distribution of training data in each iteration. Each iteration of boosting adds weight to examples of misclassification and decreases weight to examples of correct classification, thereby effectively changing the distribution of the training data. One example of a boosting algorithm is Adaboost. The purpose of this research is to improve the performance of the Decision Tree C5.0 classification method using adaptive boosting (Adaboost) to predict hepatitis disease using the Confusion Matrix. Tests that have been carried out with the Confusion Matrix use the Hepatitis dataset in the Decision Tree C5.0 classification, which has an accuracy rate of 80.58% with a classification error rate of 19.15%. Adaboost has a higher accuracy rate of 82.98% and a classification error rate of 17.02% in the C5.0 classification of the decision tree. This difference is caused by the Adaboost algorithm, because the Adaboost algorithm is able to change a weak classifier into a strong classifier by increasing the weight of the observations, and Adaboost is also able to reduce the classifier error rate.

Keywords: Data Mining; Decision Tree; C5.0 algorithm; Adaptive Boosting; Counfusion Matrix.

INTRODUCTION

Data mining is the process of finding useful knowledge or information from large-scale data. Data mining is also part of the KDD process, which consists of several stages such as data selection, pre-processing, transformation, data mining, and the interpretation of results (Ihsan, 2018).

In solving a computation using a classification technique, of course, there are various algorithms that can be used, including the Nave Bayes algorithm, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), but the algorithm that is quite popular in handling data classification cases is the Decision Tree. The Decision Tree Algorithm is a decision tree classification algorithm that is widely used because it has major advantages over other algorithms.

Decision trees are also able to produce simple decisions from complex ones by turning them into simple ones, and decision trees are very easy to understand when processing small data, which is a method without reducing the quality of the results obtained by using the criteria for each node (Hendra et al., 2020).

Fajri et al. (2022) The C5.0 algorithm is a more advanced version of the ID3 and C4.5 algorithms. When constructing a decision tree, the algorithm will use the highest information gain value as the root for the next node.

*name of corresponding author

so that the performance of this algorithm is superior to the previous algorithm with the boosting phase in the last process of calculating the C5.0 algorithm.

Adaboost is a machine-learning algorithm formulated by Yoav Freund and Robert Schapire. Adaboost could theoretically be used to significantly reduce the errors of some learning algorithms, consistently resulting in better classifier performance. Adaboost (adaptive boosting) is one of the boosting algorithms that has been shown to improve classifier performance (Saifudin & Wahono, 2015). The application of the Adaboost algorithm in feature selection is carried out to give weight to each recommended feature so that features are found that are strong classifiers (Sudarto, 2016).

Pandya and Pandya (2015) conducted previous research in which the ID3, C4.5, and C5.0 algorithms were compared.Among all these classifiers, C5.0 provides more accurate and efficient results.

Rathinasamy & Raj, 2017) The C5.0 algorithm is more accurate, takes less time, and has a lower error rate than the C4.5 algorithm, as demonstrated by this study.

Perveen et al. (2016) stated that the prediction of diabetes in a single classifier or a combination of experimental results showed that overall the performance of the Adaboost method was better than bagging and the J48 decision tree.

Taufiqurrahman el argues in 2020 that the DTR model with the Adaboost algorithm outperforms the DTR model without the Adaboost algorithm, with a mean squared error (MSE) value of 0.00454 and an R-Squared value of 0.92847, at the same maximum depth of 8.

Shakeel et al. (2019) reported that feature testing was classified with the help of the Adaboost discrete ensemble. Successful selection of cancer features and reduced feature dimensions helped increase the overall prediction rate and effectively minimize overfitting of cancer features.

In this study, researchers tried to conduct research by increasing the accuracy of the Decision Tree classification method on the C5.0 algorithm by using adaptive boosting (Adaboost) to predict hepatitis, as well as comparing the classification accuracy of the standard C5.0 algorithm with a combination of adaptive boosting.

LITERATURE REVIEW

C5.0 algorithm

The C5.0 algorithm is a data mining algorithm that is specifically applied to decision tree algorithms. This algorithm is a refinement of the previous algorithms created by Ross Quinlan in 1987, namely ID3 and C4.5. In this C5.0 algorithm, attribute selection is processed using the gain ratio.
The formula for finding Entrophy:

$$Entropy(S) = -\sum_{j=1}^{k} P_j * log_2 P_j \qquad (1)$$

With :
S : Case set
k : Number of classes in variable A
Pj : The proportion of Sj and S
Then find the gain value using the equation:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} \text{xEntropy}(S_i) \qquad (2)$$

With :
S : Set of cases
Si: The set of cases in the i-th category
A : Variable
m : Number of categories in variable A
|Si| : Number of cases in category i
|S| : Number of cases in S

After obtaining the entropy and gain values, the next step is to calculate the gain ratio values. The calculation of the gain ratio is as follows:

$$Gain\ Ratio = \frac{Gain(S,A)}{\sum_{i=1}^{m} Entropy(Si)} \qquad (3)$$

With :
Gain(S,A) = The gain value of a variable
$\sum_{i=1}^{m} Entropy(Si)$ = The number of entropy values in a variable

Adaptive Boosting (Adaboost)

Boosting is a machine learning method that combines very weak and poor prediction rules to produce very accurate prediction rules (Schapire, 2013). The Adaboost algorithm creates a classification by combining many classifications. Each classification will have a weight, and when these weights are added together, a new, powerful classification will be created. Although weak classification is poor classification, it outperforms random prediction.

*name of corresponding author

Modifying weak classifications to establish functional classifications that each depend on a single feature is a straightforward way to adapt them. There is no need to have a large database for this method as it uses databases often. (Bahramian & Nikravanshalmani, 2016).
Adaboost algorithm:

```
Input : Dataset $T = \{(x_1, y_1), ((x_2, y_2), . . ., ((x_N, y_N)\}$, $x_i \in R^n$ , $y \in Y = \{-1, +1\}$   (4)
    Output : Classifier kuat G(x)
        (1) Inisialisasi   (5)
            $D_1 = (W_{11}, . . . , W_{1i, . . . ,} W_{1N})$, $W_{1i} = \frac{1}{N}$ , $I = 1,2, . . ., N$
        (2) For m = 1,2, …, M   (6)
            Mendapatkan klasidikasi lemah berdasarkan distribusi bobot $D_m$
                $G_m(x) = \{-1, +1\}$
                a. Menghitung error pada dataset $G_m(x)$   (7)
                   $e_{m =} P(G_m(x) \neq y_i)$
                   $= \sum_{i=1}^{N} w_{mi} 1 (G_m(x) \neq y_i)$
                b. Menghitung bobot $G_m(x)$   (8)
                   $a_m = = \frac{1}{2} \log = \frac{1-em}{em}$
                c. Update $D_m$   (9)
                   $D_{m+1} = (w_{m+1,1}, …, w_{m+1,i}, …, w_{m+1,N)}$
                   $W_{m+1,i} = \frac{Wmi}{Zm} \exp(-\alpha_m y_i G_m(x_i)$
                   $z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i)$
        (3) Mendapatkan klasifikasi kuat   (10)
                   $F(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$
                   $G(x) = \text{sign}(f(x)) = \text{sign}(=\sum_{m=1}^{M} \alpha_m G_m(x))$
```

Confusion matrix

The confusion matrix is a visualization tool commonly used in supervised learning. Each column in the matrix is an example of a predicted class, while each row represents events in the actual class (Gorunescu, 2011). The confusion matrix contains actual and predictable information about the classification system.

Table 1. *Confusion Matrix*

Actual Class	**Predicted Class**	
	Predicted. Class 1	Predicted. Class 0
Actual. Class 1	(True Positive)	(False Negative)
Actual. Class 0	(False Positive)	(True Negative)

Where:
True Positive (TP) = Number of positive data correctly classified by the system
True Negative (TN) = Number of negative data correctly classified by the system
False Negative (FN) = Number of negative data but classified as wrong by the system
False Positive (FP) = Number of positive data but classified as wrong by the system
Confusion matrix equation:

$$\textit{Accuracy :} \frac{TP+TN}{TP+TN+FP+FN} \textit{ x } 100\% \qquad (11)$$

$$\textit{Clasification Error :} \frac{FP+TN}{TP+TN+FP+FN} \textit{ x } 100\% \qquad (12)$$

$$\textit{Precision :} \frac{TP}{TP+FP} \textit{ x } 100\% \qquad (13)$$

$$\textit{Recall :} \frac{TP}{TP+FN} \textit{ x } 100\% \qquad (14)$$

*name of corresponding author

METHOD

In data collection, there are data sources collected by researchers. The data is processed according to their respective algorithms. Primary data is data that is first collected by researchers directly in the field, whereas if it is collected secondhand, it is called secondary data. The data used in this study was a dataset obtained from Kaggle.com. The hepatitis dataset consists of 155 data rows and 20 columns.

The research was conducted by improving the performance of the Decision Tree classification method, Algorithm C5.0, using adaptive boosting (Adaboost) to use a confusion matrix. The following describes the general process of research to be carried out to achieve the objectives:

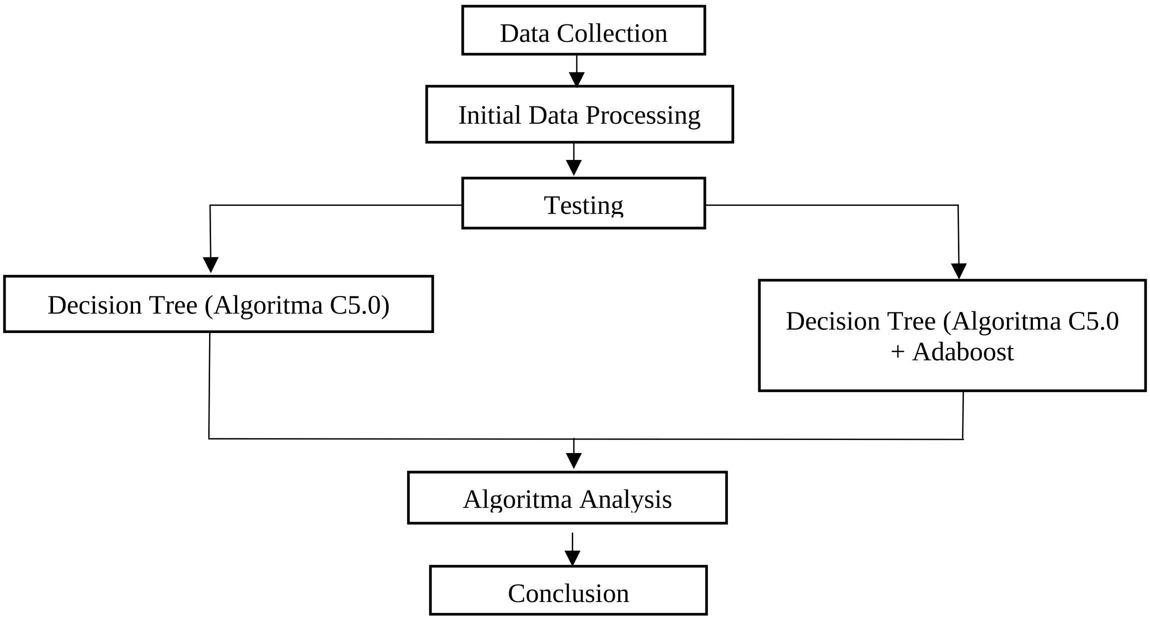

Fig 1. Research Stages

Explanation of the General Description of Research Figure 1 as follows:

1. In the early stages of data collection, the data used in this study was taken from public data taken from https://www.kaggle.com/ totaling 155 rows of data with 20 attributes of supporting data.
2. At the initial data processing stage, the researcher processes the initial data before testing with the model. The data is converted from text value to numeric data, then data that does not match is returned to a value of 0.
3. At the testing stage, the researcher performs prediction calculations using the Decision Tree and Decision Tree models with attribute selection using Adaptive Boosting (Adaboost).
4. From the test results the researcher analyzed the method to draw conclusions, from the Decision Tree method before using Adaboost and after using Adaboost.

RESULT

In this study, we used hepatitis datasets with the C5.0 algorithm and C5.0 with Adaboost to improve the performance of the C5.0 algorithm. Based on the results of research with the C5.0 algorithm using the hepatitis dataset obtained, The following is the calculation using the confusion matrix:

Table 2. *Confusion Matrix Algoritma C5.0*

Actual Class	**Predicted Class**	
	Predicted. Class 1	Predicted. Class 0
Actual. Class 1	30 (True Positive)	2 (False Negative)
Actual. Class 0	7 (False Positive)	8 (True Negative)

a. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{30+8}{30+8+7+2} = \frac{38}{47} = 0{,}8085 * 100\% = 80{,}85\%$

b. *Classifikasi Error* = $\frac{FP+FN}{TP+TN+FP+FN} = \frac{7+2}{30+8+7+2} = \frac{19}{47}$ = 0.1915 * 100% = 19,15%

c. *Precision* = $\frac{TP}{TP+FP} = \frac{30}{30+7} = \frac{30}{37}$ = 0,8108 * 100% = 81,08%

d. *Recall* = $\frac{TP}{TP+FN} = \frac{30}{30+2} = \frac{30}{32}$ = 0,9375 * 100% = 93,75%

After getting the value with the C5.0 algorithm, it is then calculated with C5.0 with adaptive boosting (Adaboost) to improve the performance of the C5.0 algorithm. The following is the calculation using the confusion matrix:

Table 3. *Confusion Matrix Algoritma C5.0 Adaboost*

Actual Class	**Predicted Class**	
	Predicted. Class 1	Predicted. Class 0
Actual. Class 1	35 (True Positive)	6 (False Negative)
Actual. Class 0	2 (False Positive)	4 (True Negative)

a. *Accuracy* = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{35+4}{35+4+2+6} = \frac{39}{47}$ = 0,8298 * 100% = 82,98%

b. *Classifikasi Error* = $\frac{FP+FN}{TP+TN+FP+FN} = \frac{2+6}{35+4+2+6} = \frac{8}{47}$ = 0,702 * 100% = 17,02%

c. *Positive Predictive Value* = $\frac{TP}{TP+FP} = \frac{35}{35+2} = \frac{35}{37}$ = 0,9459 * 100% = 94,59%

d. *Sensitivity* = $\frac{TP}{TP+FN} = \frac{35}{35+6} = \frac{35}{41}$ = 0.8536 * 100% = 85,36%

DISCUSSIONS

Tests that have been carried out with the confusion matrix use the Hepatitis dataset in the Decision Tree C5.0 classification, which has an accuracy rate of 80.58% with a classification error rate of 19.15%. Whereas in the classification of Decision Tree C5.0, Adaboost has a higher accuracy rate of 82.98% when compared to that of Decision Tree C5.0. The Adaboost Decision Tree C5.0 classification has a misclassification rate of 17.02%. This difference is caused by the Adaboost algorithm, because the Adaboost algorithm is able to change a weak classifier into a strong classifier by increasing the weight of the observations, and Adaboost is also able to reduce the classifier error rate.

.

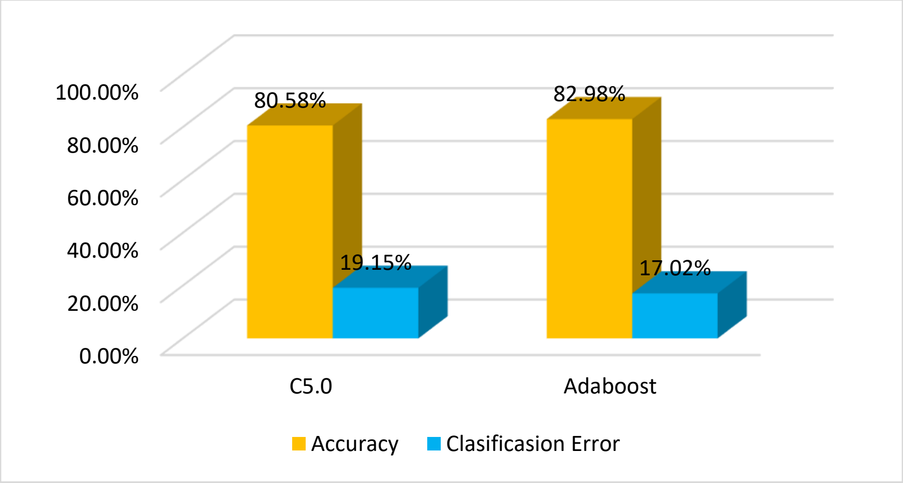

Fig 2. Comparison results of C5.0 algorithm with C5.0 adaptive boosting (Adaboost)

*name of corresponding author

CONCLUSION

In this study, classification was carried out using Decision Tree C5.0 and Decision Tree C5.0 Adaboost using the Hepatitis dataset as training and testing data. The dataset was divided into 70% of the data as training data and 30% of the data as test data. Tests that have been carried out with the confusion matrix using the Hepatitis dataset in the Decision Tree C5.0 classification, which has an accuracy rate of 80.58% with a classification error rate of 19.15%, Adaboost has a higher accuracy rate of 82.98% and a classification error rate of 17.02% in the C5.0 classification of the decision tree. This difference is caused by the Adaboost algorithm, because the Adaboost algorithm is able to change a weak classifier into a strong classifier by increasing the weight of the observations, and Adaboost is also able to reduce the classifier error rate.

REFERENCES

Bahramian, S. and Nikravanshalmani, A. 2016. Hybrid Algorithm based on K-Nearest Neighbor Algorithm and Adaboost with Selection Of Feature By Genetic Algorithms for the Diagnosis of Diabetes. International Journal Of Mechatronics, Electrical and Computer Technology (IJMEC) 6(21):2977-2986.

Fajri, M., Utami,l.T and Maruf, M. 2022. Comparison of C4.5 and C5.0 Algorithm Classification Tree Models for Analysis of Factors Affecting Auction. Indonesian Journal of Statistics and Its Applications (IJSA), Vol. 6, No. 1, 13-22.

Gorunescu, F. Data Mining: Concepts Models, and Techniques.: Springer, 2011

Hendra., Azis, M.A. and Suhardjono. 2020. Analysis of Student Graduation Predictions Using a Decission Tree Based on Particle Swarm Optimization. Journal of SISFOKOM (Computer and Information Systems), pp. 102-107.

Hepatitis B virus DNA is formed through distinct repair processes of each strand

Ihsan. 2018. Attribute Reduction in the K-Nearest Neighbor (Knn) Algorithm Using a Genetic Algorithm. [Thesis]. Medan: University of North Sumatra, Postgraduate.

Pandya, R., and Pandya, J. (2015). C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning. International Journal of Computer Applications (IJCA). Volume 117 – No. 16, May 2015, pp. 18-21.

Perveen, S., Shahbaz, M., Guergachi.A. and Keshavjee. K. 2016. Performance Analysis of Data Mining Classification Techniques to Predict Diabetes. *Procedia Computer Science* 82 (2016), pp. 115 – 121.

Rathinasamy, R. and Raj, L. 2017. Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data: International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE). Vol. 5, Special Issue 1, March 2017, pp. 50-58.

Saifudin, A., and Wahono, R. S. (2015). Application of Ensemble Techniques to Handle Class Imbalances in Software Flaw Prediction. Journal of Software Engineering, 1(1), 28–37. https://doi.org/10.1016/S1896-1126(14)00030-3n

Schapire, R.E. 2013. Explaining Adaboost. Dept of Computer Science. Princeton University: USA.

Shakeel. P.M., Tolba, A., Al-Makhadmeh, Z. and Jaber, M.M. 2019. Automatic Detection Of Lung Cancer From Biomedical Data Sets Using Discrete Adaboost Optimized Ensemble Learning Generalized Neural Networks. Neutral Computing and Applications. Springer.

Sudarto. 2016. Analysis of Handling Class Imbalance Using Density Based Feature Selection (DBFS) and Adaptive Boosting (Adaboost). [Thesis]. Medan: University of North Sumatra, Postgraduate Program

Taufiqurrahman, A., Putrada, A.G and Dawani, F. 2020. Decison Tree Regression with Adaboost Ensemble Learning For Water Temperature Forecasting in Aquaponic Ecosystem. 2020 6th International Conference on Interactive Digital Media (ICIDM), Vol 12 No 1, February 2020, pp. 1-10.

www://Kaggle.com.