ENHANCING STROKE DISEASE PREDICTION PERFORMANCE THROUGH A FUSION OF ADABOOST WITH C4.5 AND K-NEAREST NEIGHBOR ALGORITHMS

[1]Hanny Lutfy Damayanti, [2]Rosita Herawati
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]rosita@unika.ac.id

ABSTRACT

Stroke is one of the most serious medical conditions and has a significant impact on public health. The importance of accurate prediction of stroke risk is to provide appropriate treatment and intervention to individuals at risk of developing the disease. In recent years, the use of machine learning methods has become popular in improving stroke disease prediction. This research implements the Adaboost method to the C4.5 and K-Nearest Neighbor (KNN) algorithms with the aim of improving stroke prediction performance. Using relevant datasets, the C4.5 and KNN algorithms were used separately to perform stroke disease prediction. Furthermore, the Adaboost method is used to combine the prediction results of the two algorithms. The results showed that the implementation of the Adaboost method on the C4.5 and KNN algorithms successfully improved the performance of stroke disease prediction, providing more accurate and reliable predictions to assist in the diagnosis and treatment of stroke disease. With a value of 91% for the combination of KNN with Adaboost and 95% for the combination of C4.5 with Adaboost. Both have a difference in value of 4%. Therefore, C4.5 is more effective in improving the performance of stroke disease prediction.

Keywords: stroke, c4.5, knn, adaboost

INTRODUCTION

Stroke is a significant global health problem, ranking as the second leading cause of death worldwide and contributing significantly to high rates of disability. Indonesia in particular, faces a pressing challenge with increasing stroke cases and high mortality rates.[1] According to data from 208 Riskesdas, North Sulawesi Province has the highest prevalence of stroke (14.2%) while Papua Province (4.1%).[2] Not only that, based on information from the *Centers for Disease Control and Prevention (CDC)*, stroke is also the leading causes of death in the United States. Stroke is a non-communicable disease that accounts for about 11% of all deaths and more than 795,000 individuals in the United States experience the adverse effects of stroke.[3] The C4.5 algorithm can be utilized for predict or classify an event by forming a decision tree.[4] K-Nearest Neighbor performs classification by considering the closest distance between new data and existing data, starting with determining the value of the nearest neighbor.[5] Adaboost is one of the supervised algorithms in the field of data mining that is often used to develop classification models.

With the development of medical technology, it has become possible to utilize machine learning to forecast stroke events. Machine learning algorithms, which are constructive in nature, can produce accurate predictions as well as provide careful analysis. The use of machine learning has proven to be widely applied in classification and optimization topics in creating intelligent systems to improve healthcare providers. The selection of the right method for stroke symptom detection is needed because it affects the results that will be displayed.[6]

The purpose of this research is to apply the Adaboost method to the C4.5 and K-Nearest Neighbor algorithms to improve the performance of stroke disease prediction. In the context of improving stroke prediction performance, the C4.5 algorithm is used to build a decision tree model that can classify stroke symptoms into stroke or non-stroke categories. The K-Nearest Neighbor algorithm measures the distance between new data and old data and performs classification based on predetermined nearest neighbor values. The use of the Adaboost method aims to improve the accuracy of the classification model by combining several weak classification models into one stronger classification model. Accuracy is defined as the degree of agreement between predicted and actual values.[7] In addition, the results of the tests should be analyzed to see how effective the combination algorithm is.

RESEARCH METHODOLOGY

To achieve good results in this research study, a structured research method is essential. Step of problem solving method :

1. Conduct a literature study related to the topic discussed
2. Collecting stroke disease datasets from the kaggle platform, studying the algorithms used
3. Preprocessing the dataset with cleaning data, encoding data, and over sampling using smote
4. Algorithm modeling using C4.5, K-Nearest Neighbors, and Adaboost
5. Analysing implementation results and making conclusions

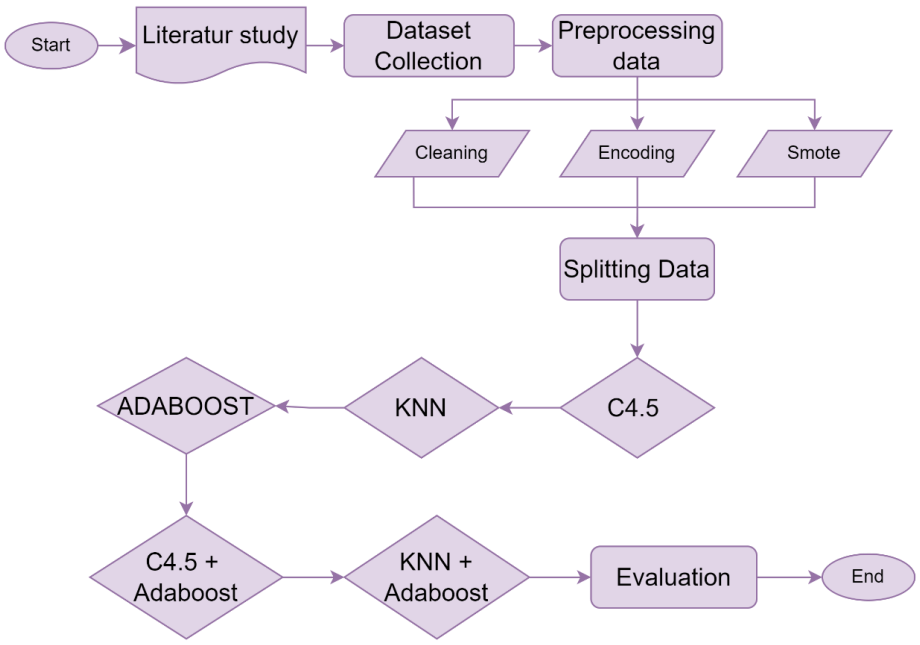

Figure 1. Research Methodology

2.1 Dataset Collection

The dataset used is Stroke Prediction taken from kaggle. The data consists of 43400 observation data with 12 attributes. The data attributes used in this study are presented in the following Table 3.1.

Table 1. Dataset Attribute

No	*Name*	*Information*
1	id	Id pasien
2	gender	Jenis kelamin
3	age	Usia pasien
4	hypertension	Hipertensi/tekanan darah tinggi
5	heart_disease	Penyakit jantung
6	ever_married	Pernah menikah
7	work_type	Jernis pekerjaan
8	residence_type	Tempat tinggal
9	avg_glucose_level	Kadar glukosa
10	bmi	Index massa tubuh
11	smoking_status	Status merokok
12	stroke	Prediksi stroke

2.2 Pre-processing Data

In this research, data preprocessing is done by data cleaning, data coding and smote oversampling. In data cleaning, several attributes and data that are incomplete, inaccurate, and irrelevant are cleaned. Encoding is done because modeling cannot process strings, so it is necessary to change the form of data in some attributes. Smote is done to overcome unbalanced data

2.2.1 Cleaning Data

Data cleaning can be done by various methods such as filling in missing data, normalizing values, or changing variables. However, each case requires a customized assessment to identify the most suitable and efficient data cleansing strategy. The removal of empty data and duplication is a kind of steps that performed during data processing. It contributes to improving data quality. Therefore, for this study, data cleansing was performed by removing the empty data and the duplicate data to ensure efficient processing time.

2.2.2 Encoding Data

Encoding is one of the pre-processing done in this research. In this research, encoding is done because modeling cannot process strings, so it is necessary to change the form of data in some attributes. By doing encoding, it can facilitate data processing.

2.2.3 Smote Oversampling

The last pre-processing is oversampling using smote. This is done to change the amount of data with the label "stroke". The stroke parameter has 2 data contents namely stroke and non-stroke where the number of non-strokes is more than the number of strokes. Therefore it is necessary to do oversampling so that the number becomes the same and produces good accuracy.

2.3 Splitting Data

In splitting the data is divided into 2, namely training and testing. Training is part of the dataset that is trained to predict the function of the machine learning algorithm. Testing is part of the dataset that is tested to see its accuracy. In this research, the module used is sklearn.model_selection.

2.4 C4.5 Algorithm

In this research, the classification method uses the C4.5 algorithm to analyze stroke disease. The attribute selection process is done by assigning attributes as nodes, which can be root nodes or internal nodes, based on the highest Gain value possessed by those attributes. The steps of data processing with the C4.5 algorithm involve the calculation of entropy values, the calculation of gain values, and the formation of decision trees and corresponding rules. Equation (1) and (2) are used to calculate entropy and gain values. [8]

$$Entropy(S) = \sum_{i=0}^{n} - pi * log_2(pi) \quad (1)$$

Description :

S : set of cases

n : number of partitions S

pi : proportion of Si to S

$$Gain(S, A) = Entropy(S) - \sum_{I=0}^{n} \frac{|S_i|}{|S|} * Entropy(S_1) \quad (2)$$

Description :

S : set of cases

n : number of partitions of attribute A

|S| : number of cases in S

|Si|: number of cases in the i partition

2.5 K-Nearest Neighbor Algorithm

The K-Nearest Neighbor (KNN) algorithm performs clustering of new data by considering the distance between the data and some nearest neighbors. The number of nearest neighbors is determined by the neighbor parameter, which can be set by the user. KNN operates by finding the minimum distance from the new data to the specified nearest neighbors. The focus of this $A = \pi r2$ algorithm is to classify new objects based on their attributes and training samples. The process of determining neighbor proximity generally uses the Euclidean Distance calculation, which is explained as follows [5] :

$$E(x, y) = \sqrt{\sum_{i}^{n} 0 \ (xi - yi)^2} \quad (3)$$

Description :

xi = sample Data

yi = testing data

n = data dimension

I = variable data

2.6 Adaptive Boosting

Adaboost is used to classify data in their respective classes. Adaboost searches for class categories based on the weight value owned by the class. This process continues to be repeated so that there is a value update on the class. In adaboost, the weight value will continue to increase at each iteration of the wrong weight value at each iteration. Adaboost is a typical ensemble learning algorithm, the results obtained have a strong level of accuracy. To form an adaboost ensemble can use the following formula[1][5] :

$$Ym(x) = sign\left(\sum_{m-1}^{M} a_m y_m(x)\right) \quad (4)$$

2.7 Evaluation

The data that has been processed and tested is then compared. The three main metrics used to evaluate classification models are accuracy, precision, and recall. In this research, the model evaluation uses confusion matrix data. Based on the confusion matrix results, the values of accuracy, recall, precision, and F1 score can be determined.

1. Accuracy

Accuracy is the ratio of true prediction to the overall data.

$$\frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100\% \quad (5)$$

2. Precision

Precision is the ratio of positive true prediction compared to overall positive prediction result.

$$\frac{(TP)}{(TP + FP)} \times 100\% \quad (6)$$

3. Recall

Recall is the ratio of positive true prediction compared to overall positive true data.

$$\frac{(TP)}{(TP + FN)} \times 100\% \quad (7)$$

4. F1 Score

F1 Score is a weighted comparison of average precision and recall.

$$\frac{2 \times (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (8)$$

Based on function (5), (6), (7) TP is True Positive, FP is False Positive, TN is True Negative, FN is False Negative and the result is multiplied by 100% to get the percentage. The calculation result of the Recall (7) and Precision (8) functions will produce an F1 Score (8).

RESULT

3.1 Result C4.5 Algorithm

Result provided start from the beginning of preprocessing, then the data is divided into training and testing then calculated accuracy using C4.5 Algorithm. The Experiment for the high result is using 70%, 60% of training data and 30%, 40% test data. For C4.5 the best max_depth in 5. The following is a table of calculated results.

Table 2. C4.5 Modeling Result

C4.5				
test size	**precision**	**recall**	**f1-score**	**acccuracy**
20%	0.84	0.84	0.84	0.84
30%	0.87	0.87	0.87	0.87
40%	0.87	0.87	0.87	0.87

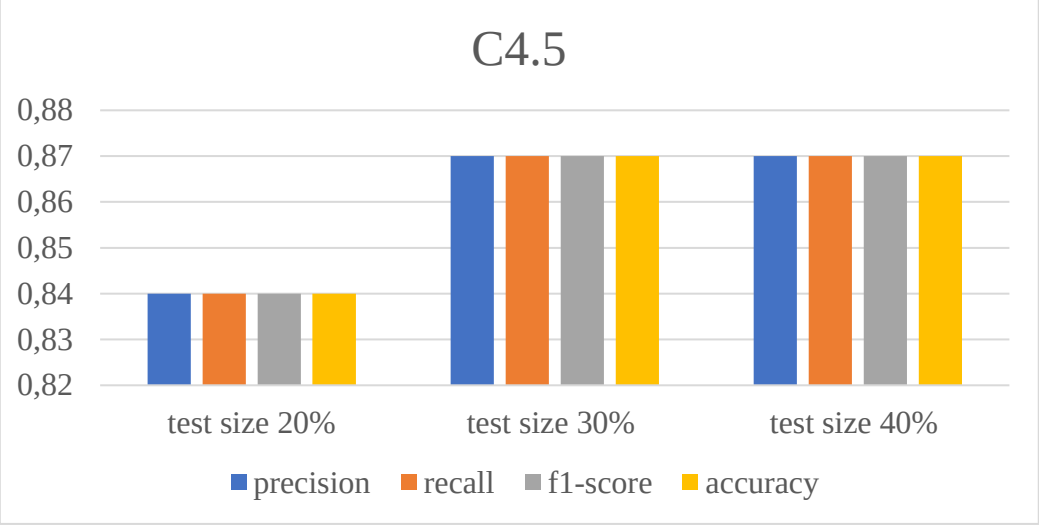

Figure 2. C4.5 Modeling Result

In table 2 and figure 2 researchers used test sizes of 20%, 30% and 40% for the C4.5 algorithm. For the highest value, there are 30% and 40% test sizes with precision, recall, f1-score and accuracy each producing 0.87.

3.2 Result KNN Algorithm

Result provided start from the beginning of preprocessing, then the data is divided into training and testing then calculated accuracy using KNN Algorithm. The Experiment for the high result is using 80%, 70% of training data and 20%, 30% test data. For KNN the best neighbor in 5. The following is a table of calculated results.

Table 3. KNN Modeling Result

KNN				
test size	**precision**	**recall**	**f1-score**	**acccuracy**
20%	0.87	0.86	0.86	0.86
30%	0.87	0.86	0.86	0.86
40%	0.86	0.85	0.85	0.85

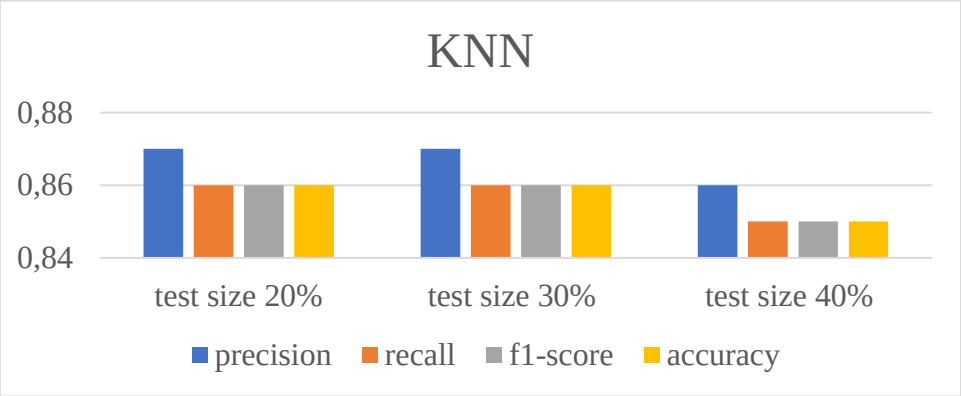

Figure 3. KNN Modeling Result

In table 3 and figure 3 researchers used test sizes of 20%, 30% and 40% for the KNN algorithm. For the highest value, there are 20% and 30% test sizes with precision, recall, f1-score and accuracy each producing 0.86.

3.3 Result Adaptive Boosting

Result provided start from the beginning of preprocessing, then the data is divided into training and testing then calculated accuracy using Adaboost. The Experiment for the high result is using 70% of training data and 30% test data. For Adaboost the best estimator in 20. The following is a table of calculated results.

Table 4. Adaboost Modeling Result

ADABOOST				
test size	**precision**	**recall**	**f1-score**	**acccuracy**
20%	0.91	0.91	0.91	0.91
30%	0.92	0.92	0.92	0.92
40%	0.91	0.91	0.91	0.91

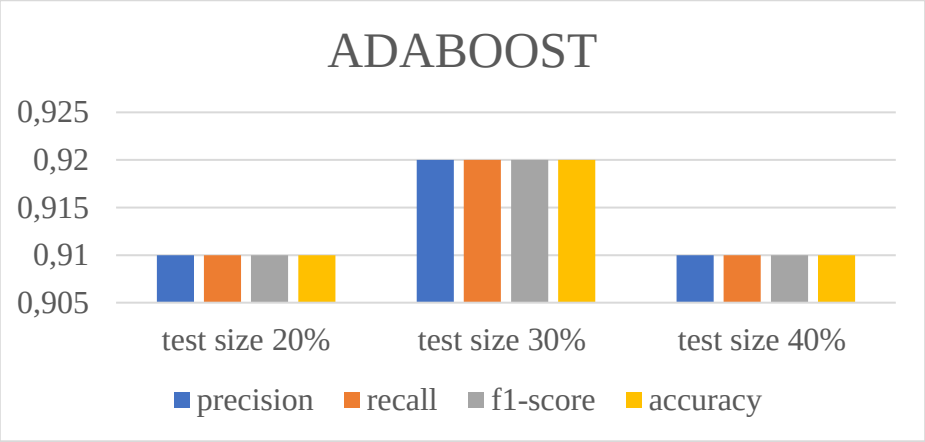

Figure 4. Adaboost Modeling Result

In table 4 and figure 4 researchers used test sizes of 20%, 30% and 40% for the Adaboost. For the highest value, there are 30% test sizes with precision, recall, f1-score and accuracy each producing 0.91.

3. 4 Result C4.5 and Adaboost Combination

The results given start from the beginning of preprocessing, then the data is divided into training and testing and then combined between C4.5 and Adaboost and the accuracy is calculated using the Adaboost Method. The following is a table and figure of calculation results.

Table 5. C4.5 and Adaboost Modeling Result

ADABOOST and C4.5				
test size	**precision**	**recall**	**f1-score**	**acccuracy**
20%	0.95	0.95	0.95	0.95
30%	0.95	0.95	0.95	0.95
40%	0.95	0.95	0.95	0.95

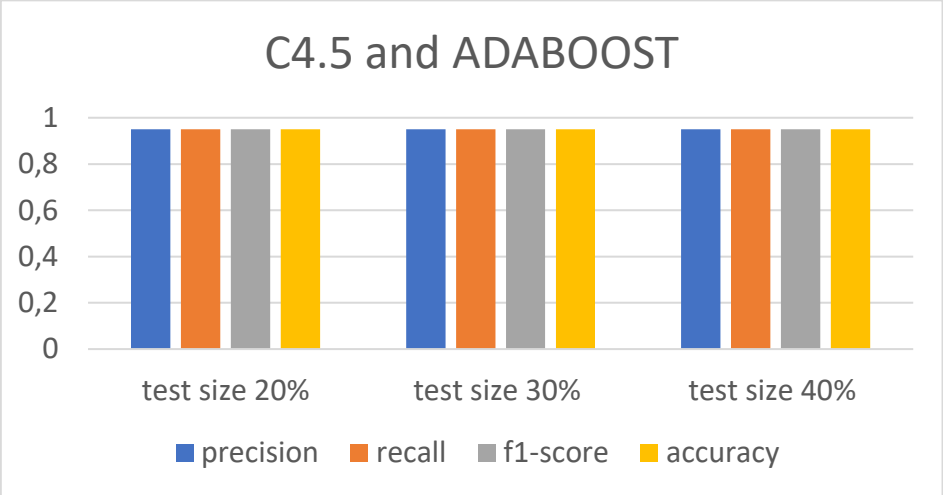

Figure 5. C4.5 and Adaboost Modeling Result

In table 5 and figure 5 researchers used test sizes of 20%, 30% and 40% for combonation C4.5 and Adaboost. For each test size has the same value in each precision, recall, f1-score and accuracy with a result of 0.95.

3.6 Result KNN and Adaboost Combination

The results given start from the beginning of preprocessing, then the data is divided into training and testing and then combined between KNN and Adaboost and the accuracy is calculated using the Adaboost Method. The following is a table of calculation results.

Table 6. KNN and Adaboost Modeling Result

ADABOOST and KNN				
test size	**precision**	**recall**	**f1-score**	**acccuracy**

20%	0.91	0.91	0.91	0.91
30%	0.91	0.91	0.91	0.91
40%	0.91	0.91	0.91	0.91

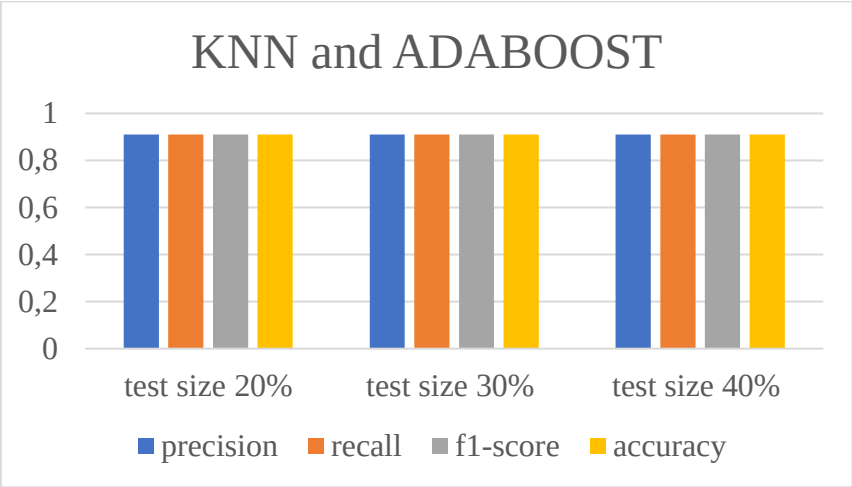

Figure 6. KNN and Adaboost Modeling Result

In table 6 and figure 6 researchers used test sizes of 20%, 30% and 40% for combination KNN with Adaboost. For each test size has the same value in each precision, recall, f1-score and accuracy with a result of 0.91.

3.7 Conclusion of the Two Combination Result

Based on the algorithm testing above which uses a max depth value of 5, neighbors 5, estimator 20 and the research was processed with all test size has good results. Although each algorithm has very small difference in precision, recall, and f1-score values. For more details, can see the chart below.

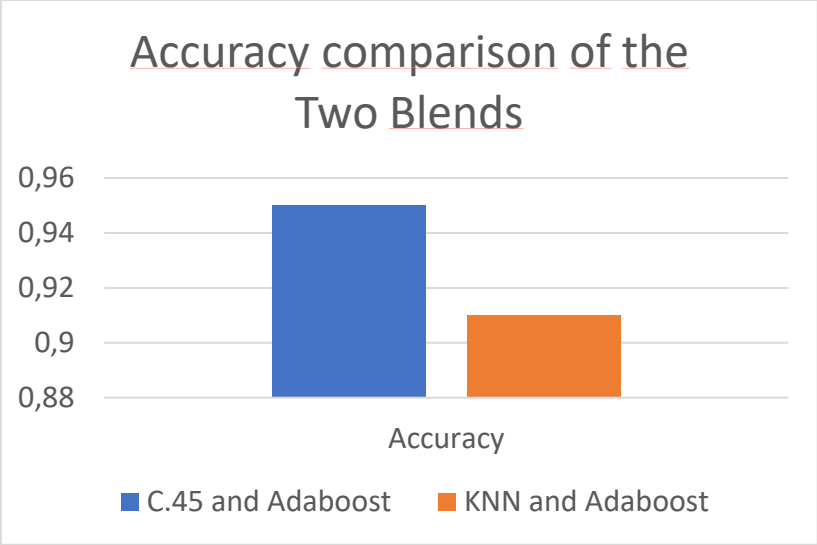

Figure 7. Two Combination Result

Based on figure 7 the combination of the C4.5 algorithm has higher results than the KNN algorithm combination. Each has a value of 91% and 95%. Where the two combinations have a difference of 4%.

CONCLUSION

Based on the test results of combining the two algorithms, it can be concluded that both combinations can help improve performance in predicting stroke disease. The regular C4.5 algorithm has an optimal accuracy value at 87%, but after C4.5 is combined with Adaboost, the accuracy value increases into 95%. Combining C4.5 with Adaboost can increase the accuracy value at 8%. While the KNN algorithm has an optimal accuracy value at 86%, after combining with Adaboost the accuracy value increases into 91%. Therefore, the combination of the C4.5 algorithm with Adaboost is the best combination for improving accuracy performance in the stroke disease prediction.

Suggestions for future research are to try using other approaches, such as filling missing data. Exploring other improvement methods. Trying to predict in other cases. And can try some other algorithms to find out better prediction performance.

REFERENCES

[1] A. Byna and M. Basit, "PENERAPAN METODE ADABOOST UNTUK MENGOPTIMASI PREDIKSI PENYAKIT STROKE DENGAN ALGORITMA NAÏVE BAYES," *SISFOKOM*, vol. 9, no. 3, pp. 407–411, Nov. 2020, doi: 10.32736/sisfokom.v9i3.1023.

[2] Y. Oktarina and S. Mulyani, “EDUKASI KESEHATAN PENYAKIT STROKE PADA LANSIA,” vol. 3, 2020, doi: 10.22437/medicaldedication.v3i2.11220.
[3] K. L. Kohsasih and Z. Situmorang, “Analisis Perbandingan Algoritma C4.5 dan Naïve Bayes Dalam Memprediksi Penyakit Cerebrovascular,” *Jurnal Penelitian Teknik Informatika, Manajemen Informatika dan Sistem Informasi*, vol. 9, no. 1, pp. 13–17, Apr. 2022, doi: 10.31294/inf.v9i1.11931.
[4] R. Novita, “Teknik Data Mining : Algoritma C 4.5”.
[5] N. Novianti, M. Zarlis, and P. Sihombing, “Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes,” *mib*, vol. 6, no. 2, p. 1200, Apr. 2022, doi: 10.30865/mib.v6i2.4017.
[6] A. Puspitawuri, E. Santoso, and C. Dewi, “Diagnosis Tingkat Risiko Penyakit Stroke Menggunakan Metode K-Nearest Neighbor dan Naïve Bayes”.
[7] L. Pebrianti, F. Aulia, and H. Nisa, “Implementasi Metode Adaboost untuk Mengoptimasi Klasifikasi Penyakit Diabetes dengan Algoritma Naïve Bayes”, vol. 7, no. 2, 2022, doi: 10.32528/justindo.v7i2.8627.
[8] R. E. Pambudi, “Klasifikasi Penyakit Stroke Menggunakan Algoritma Decision Tree C.45,” vol. 16, no. 02, doi: 10.5281/zenodo.7535865.
[9] A. Rohman, V. Suhartono, and C. Supriyanto, “PENERAPAN ALGORITMA C4.5 BERBASIS ADABOOST UNTUK PREDIKSI PENYAKIT JANTUNG,” vol. 13, 2017, doi: 10.25126/jtiik.2020752379.
[10] A. F. Hermawan, F. R. Umbara, and F. Kasyidi, “Prediksi Awal Penyakit Stroke Berdasarkan Rekam Medis menggunakan Metode Algoritma CART(Classification and Regression Tree)”, vol. 7, no. 2, 2022, doi: 10.26760/mindjournal.v7i2.151-164.
[11] D. C. P. B. - STMIK Nusa Mandiri Jakarta, “Prediksi Penyakit Hepatitis Menggunakan Algoritma Naïve Bayes Dengan Seleksi Fitur Algoritma Genetika,” *EVOLUSI*, vol. 6, no. 2, Sep. 2018, doi: 10.31294/evolusi.v6i2.4381.
[12] P. Handayani, E. Nurlelah, M. Raharjo, and P. M. Ramdani, “Prediksi Penyakit Liver Dengan Menggunakan Metode Decision Tree dan Neural Network,” *Com, Engine, Sys, Sci*, vol. 4, no. 1, p. 55, Feb. 2019, doi: 10.24114/cess.v4i1.11528.
[13] D. Larassati, A. Zaidiah, and S. Afrizal, “Sistem Prediksi Penyakit Jantung Koroner Menggunakan Metode Naive Bayes,” *jipi. jurnal. ilmiah. penelitian. dan. pembelajaran. informatika.*, vol. 7, no. 2, pp. 533–546, May 2022, doi: 10.29100/jipi.v7i2.2842.