Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JACOST**

JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)

Vol. 6 No. 2 (2025) 67 – 73 | ISSN: 2723-1453 (Media Online)

Application of Bagging and Boosting Methods for Heart Disease Classification

Yehezkiel E.A Parapak[1] , Robet[2], Jackri Hendrik[3]
[1,2,3]Informatics Engineering Study Program, STMIK Time
[1]kielparapak@gmail.com, [2]robertdetime@gmail.com, [3]jackri.hendrik@gmail.com

Abstract

Cardiovascular disease remains a primary contributor to global mortality, underscoring the urgent need for accurate and early diagnostic tools. This study aims to develop a robust classification model for heart disease by conducting a comparative analysis of six ensemble machine learning algorithms, comprising three from the Bagging family (Random Forest, Bagged Decision Tree, Extra Trees) and three from the Boosting family (AdaBoost, Gradient Boosting, XGBoost). The research utilizes the publicly available UCI Cleveland Heart Disease dataset, which exhibits a mild class imbalance. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was strategically applied to the training data. The performance of each model was rigorously evaluated using accuracy, precision, recall, and F1-score. Experimental results revealed that the Extra Trees algorithm, when combined with SMOTE, achieved the highest overall performance with 90% accuracy, 96% precision, 82% recall, and an 88% F1-score. The primary contribution of this work lies in its comprehensive analysis demonstrating that the randomization strategy of Extra Trees provides a superior and more reliable framework for this classification task compared to other common ensemble techniques, particularly after data balancing. These findings confirm that an integrated approach of ensemble learning and proper data balancing can significantly enhance the development of fair and effective diagnostic tools to support medical professionals.

Keywords: Heart Disease, SMOTE, Extra Trees, Bagging, Boosting

Abstrak

Penyakit kardiovaskular tetap menjadi kontributor utama kematian global, menggarisbawahi kebutuhan mendesak akan alat diagnostik yang akurat dan dini. Studi ini bertujuan untuk mengembangkan model klasifikasi yang kuat untuk penyakit jantung dengan melakukan analisis komparatif dari enam algoritma pembelajaran mesin ansambel, yang terdiri dari tiga dari keluarga Bagging (Random Forest, Baggged Decision Tree, Extra Trees) dan tiga dari keluarga Boosting (AdaBoost, Gradient Boosting, XGBoost). Penelitian ini menggunakan kumpulan data UCI Cleveland Heart Disease yang tersedia untuk umum, yang menunjukkan ketidakseimbangan kelas ringan. Untuk mengatasi hal ini, Synthetic Minority Over-sampling Technique (SMOTE) diterapkan secara strategis pada data pelatihan. Kinerja setiap model dievaluasi secara ketat menggunakan akurasi, presisi, ingatan, dan skor F1. Hasil eksperimental mengungkapkan bahwa algoritma Extra Trees, ketika dikombinasikan dengan SMOTE, mencapai kinerja keseluruhan tertinggi dengan akurasi 90%, presisi 96%, ingatan 82%, dan skor F1 88%. Kontribusi utama dari pekerjaan ini terletak pada analisis komprehensifnya yang menunjukkan bahwa strategi pengacakan Pohon Ekstra memberikan kerangka kerja yang unggul dan lebih andal untuk tugas klasifikasi ini dibandingkan dengan teknik ansambel umum lainnya, terutama setelah penyeimbangan data. Temuan ini menegaskan bahwa pendekatan terpadu pembelajaran ansambel dan penyeimbangan data yang tepat dapat secara signifikan meningkatkan pengembangan alat diagnostik yang adil dan efektif untuk mendukung profesional medis.

Kata kunci: Penyakit Jantung, SMOTE, Pohon Ekstra, Bagging, Boosting

1. Introduction

In many parts of the world, including Indonesia, heart disease continues to rank as the leading cause of death as shown in Figure 1. Prevention and early detection are crucial steps in reducing the mortality rate associated with this condition. While clinical diagnosis remains the gold standard, it can be time-consuming and reliant on expert availability. In the digital era, data-driven approaches using machine learning offer a promising avenue to supplement traditional methods, helping to automate and improve the efficiency of the diagnostic classification process[1]. Numerous studies have applied machine learning to medical datasets for disease prediction. However, many models face challenges with datasets that are not perfectly balanced between classes, potentially leading to biased predictions that favor the majority class. Furthermore, while many algorithms have been tested individually, a systematic comparison between the two primary families of ensemble learning, Bagging and Boosting, especially when combined with a data balancing technique, remains less explored for the well-established Cleveland Heart Disease dataset. This creates a research gap in identifying the most robust and reliable ensemble framework for this specific, widely used benchmark. This research provides a comprehensive performance evaluation of these six algorithms, comparing their effectiveness before and after data balancing. Performance is measured using accuracy, precision, recall, and F1-score to determine the optimal model for heart disease classification[2], [3].

The problem of class imbalance in the dataset, where the number of healthy patients is more dominant, was addressed using the Synthetic Minority Over-sampling Technique (SMOTE). This technique creates synthetic data for minority classes so that the class distribution becomes more balanced, allowing the model to learn the patterns of both classes more fairly[4].

This research aims to evaluate and compare the effectiveness of six algorithms in classifying heart conditions using processed and balanced data. Evaluation was carried out using overall accuracy, classification precision, sensitivity (recall), and harmonic mean score (F1-score) metrics to determine how well the model detected the patient's condition as a whole[5].

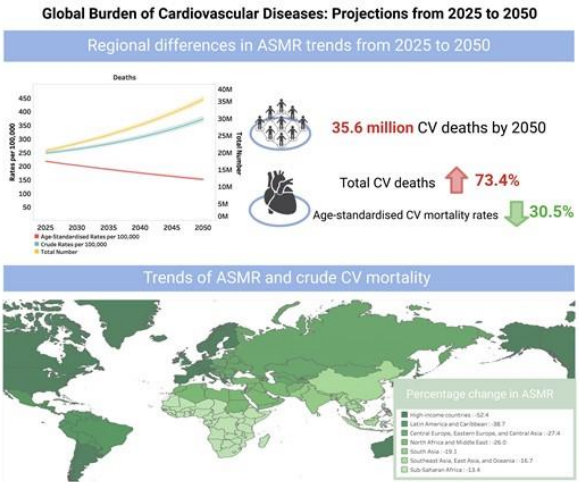


Figure 1.The Number One Cause Of Death In The World

2. Research Methods

2.1. Research Stages

This study was conducted through a systematic sequence of stages, beginning with data acquisition, preprocessing, class balancing, model training, and finally performance evaluation. The overall workflow is illustrated in the research stages (Figure 2), outlining each step from the initial stage to the final review [6], [7].

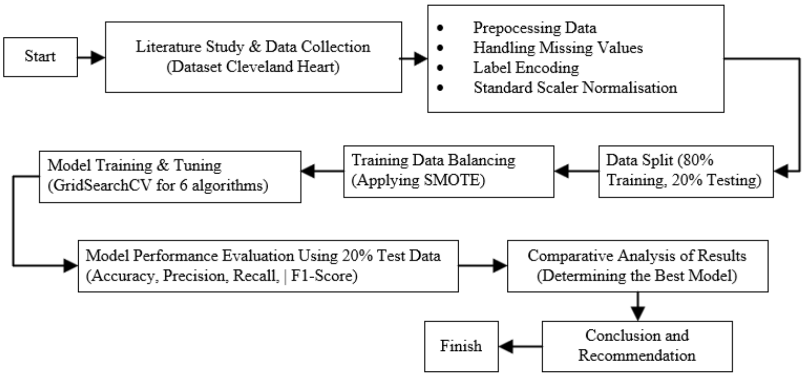
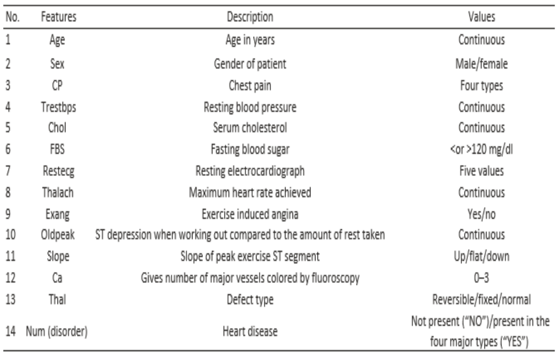

Start
Literature Study & Data Collection
(Dataset Cleveland Heart)
Prepocessing Data
Handling Missing Values
Label Encoding
Standard Scaler Normalisation
Data Split (80% Training, 20% Testing)
Training Data Balancing
(Applying SMOTE)
Model Training & Tuning
(GridSearchCV for 6 algorithms)
Model Performance Evaluation Using 20% Test Data
(Accuracy, Precision, Recall, | F1-Score)
Comparative Analysis of Results
(Determining the Best Model)
Conclusion and Recommendation
Finish

Figure 2. Research Stages

2.2. Load Data

The dataset used is Heart Disease Cleveland, which is publicly available through the Kaggle platform. This dataset contains 303 patient data and has 14 clinical attributes. These attributes consist of 13 predictor variables (features)—which include demographic, symptom, and clinical measurement data—as well as one binary target variable, 'Num (disorder)', which indicates the patient's heart disease status (0 = no disease, 1 = disease). A detailed explanation of the 14 attributes is presented in Figure 3.



No.	Features	Description	Values
1	Age	Age in years	Continuous
2	Sex	Gender of patient	Male/female
3	CP	Chest pain	Four types
4	Trestbps	Resting blood pressure	Continuous
5	Chol	Serum cholesterol	Continuous
6	FBS	Fasting blood sugar	<or >120 mg/dl
7	Restecg	Resting electrocardiograph	Five values
8	Thalach	Maximum heart rate achieved	Continuous
9	Exang	Exercise induced angina	Yes/no
10	Oldpeak	ST depression when working out compared to the amount of rest taken	Continuous
11	Slope	Slope of peak exercise ST segment	Up/flat/down
12	Ca	Gives number of major vessels colored by fluoroscopy	0–3
13	Thal	Defect type	Reversible/fixed/normal
14	Num (disorder)	Heart disease	Not present ("NO")/present in the four major types ("YES")

Figure 3. Attribute

DOI : https://doi.org/10.52158/we9asn06

2.3. Pre-processing and Data Balancing

The preprocessing stage involved cleaning the dataset from missing and duplicate values, transforming categorical variables into numerical form using label encoding, and normalizing all numerical features with *StandardScaler* to ensure uniform scaling across attributes.

Since the dataset exhibited an imbalanced target class distribution (0 = no disease, 1 = heart disease detected), the *Synthetic Minority Over-sampling Technique (SMOTE)* was applied to equalize the minority and majority classes[8] - [10]. Importantly, SMOTE was applied only to the training data to prevent data leakage into the test set. The class distribution before and after applying SMOTE is shown in Figure 4.

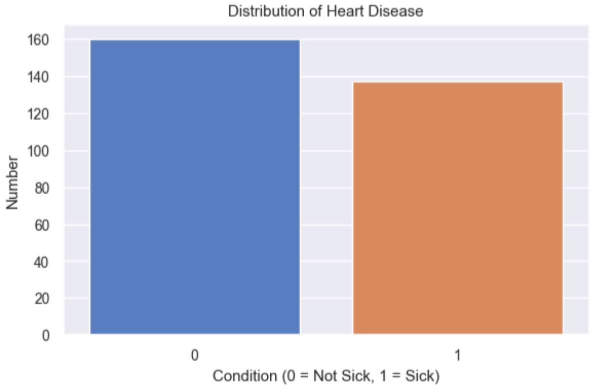


Figure 4. Condition Attributes

2.4 Data Splitting

A stratified random split was employed to preserve class distribution across training and test sets. Specifically, 80% of the data was allocated for model training, while the remaining 20% was reserved for testing.This ensured that the models were trained on sufficient data while retaining an unbiased test set for performance evaluation[11] - [13].

2.5 Ensemble Learning Algorithms

This research explored two main ensemble learning approaches: Bagging and Boosting. The algorithms employed, along with their configurations and mathematical formulations, are summarized below:

Random Forest, to decide the best way to split the data at each node of a tree, it uses a metric called Gini Impurity. The goal is to find splits that make the resulting groups as "pure" or homogenous as possible. A lower Gini value signifies a better, less impure split as shown in Equation 1:

$$Gini(D) \ = 1 - \sum_{i=1}^{c} P^2 i \tag{1}$$

Bagged Decision Tree The final prediction is determined democratically through a process called majority voting. Each of the individual trees gets one "vote" for a classification outcome, and the class that receives the most votes becomes the final prediction for the entire model[14], [15].The model prediction is determined by aggregating the majority vote of all trees. The mathematical formulation for the ensemble output in classification is shown in Equation 2.

$$ý = mode\ (\{ht(x)\}T = 1 \tag{2}$$

Extra Trees,It uses Entropy as its splitting criterion, which is a way to measure the amount of disorder or uncertainty in a data split. The algorithm aims to perform splits that reduce this uncertainty as much as possible (a process known as maximizing "information gain") is shown in Equation 3.

$$Entropy(S) \ = - \sum_{i=1}^{n} \mathrm{p}i \ \ \log 2\, pi \tag{3}$$

AdaBoost works by learning from its mistakes. It sequentially adds simple models (weak learners), and in each step, it pays more attention to the data points that were misclassified by the previous models. The formula below calculates how much "say" or influence (*αt*) each weak learner has in the final outcome, based on its classification error rate (*εt*). Models with lower error rates are given more influence, calculated using Equation 4.

$$\alpha t = \frac{1}{2} 1n \frac{1-£t}{£t} \tag{4}$$

Gradient Boosting constructs models sequentially by minimizing the residual errors of previous models using a Gradient descent optimization approach. It employs 100 estimators, a learning rate of 0.1, and a maximum tree depth of 3, with a deviance loss function. The update function for adding new learners is defined as Equation 5.

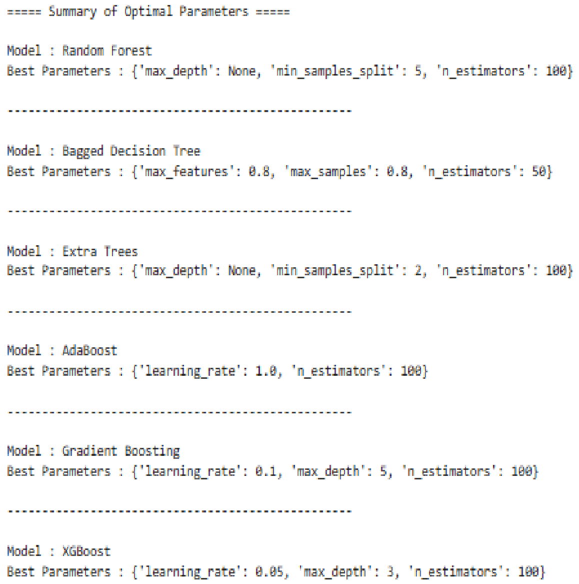
$$Fm(\mathrm{x}) = \ Fm - 1(\mathrm{x}) + \mathrm{v} \cdot hm = (\mathrm{x}) \tag{5}$$

XGBoost is not a fundamentally new algorithm, but a highly efficient and regularized implementation of the Gradient Boosting framework**.** It enhances performance by trying to achieve a balance between two things: being accurate and keeping the model simple to avoid overfitting. This balance is measured by its objective function. The function has two parts: the first part (*l*) measures the prediction error, while the second part (*Ω*) is a penalty for model complexity. The algorithm's goal is to minimize the combined value of this function using Equation 6.

$$Obj(0) = \ \sum_{i=1}^{n} l(\mathrm{y}1\ ) + \sum_{k=1}^{K} \Omega(\mathrm{fk}\ ) \tag{6}$$

2.6 Model Training and Hyperparameter Tuning

In this study, each algorithm was evaluated using the GridSearchCV technique to find the optimal combination of parameters after the data was balanced with the SMOTE method[12], [16] . The selection of optimal parameters is essential because it can affect the performance of the model in performing classification[13]. The results of the parameter search are shown in Figure 5.



```
===== Summary of Optimal Parameters =====

Model : Random Forest
Best Parameters : {'max_depth': None, 'min_samples_split': 5, 'n_estimators': 100}

--------------------------------------------------

Model : Bagged Decision Tree
Best Parameters : {'max_features': 0.8, 'max_samples': 0.8, 'n_estimators': 50}

--------------------------------------------------

Model : Extra Trees
Best Parameters : {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 100}

--------------------------------------------------

Model : AdaBoost
Best Parameters : {'learning_rate': 1.0, 'n_estimators': 100}

--------------------------------------------------

Model : Gradient Boosting
Best Parameters : {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}

--------------------------------------------------

Model : XGBoost
Best Parameters : {'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 100}
```

Figure 5. Customization of the GridSearchCV Model

2.7 Evaluation Metrics

Model performance was evaluated on the unseen test set to ensure unbiased results. The evaluation metrics included *accuracy*, *precision*, *recall*, and *F1-score* are shown in Tables 1 and 2. These metrics were chosen as they collectively provide a comprehensive assessment of classification performance, particularly in imbalanced datasets[17].

Table 1. Before SMOTE

Algorithms	Accuracy	Precision	Recall	F1-Score
Random Forest	88%	96%	79%	86%
Bagged Decision Tree	80%	86%	68%	76%
Extra Trees	88%	96%	79%	86%
AdaBoost	85%	95%	71%	82%
Gradient Boosting	87%	92%	79%	85%
XGBoost	85%	88%	79%	83%

Table 2.After SMOTE

Algorithms	Accuracy	Precision	Recall	F1-Score
Random Forest	88%	96%	79%	86%
Bagged Decision Tree	80%	86%	68%	76%
Extra Trees	90%	96%	82%	88%
AdaBoost	83%	88%	75%	81%
Gradient Boosting	85%	91%	75%	82%
XGBoost	85%	88%	79%	83%

An illustration of the results after the SMOTE above can be seen from Figures 6 to 11.

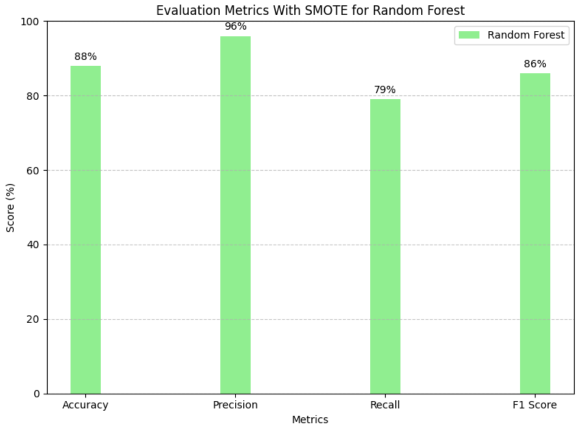


Figure 6. Model Evaluation Graphics Random Forest

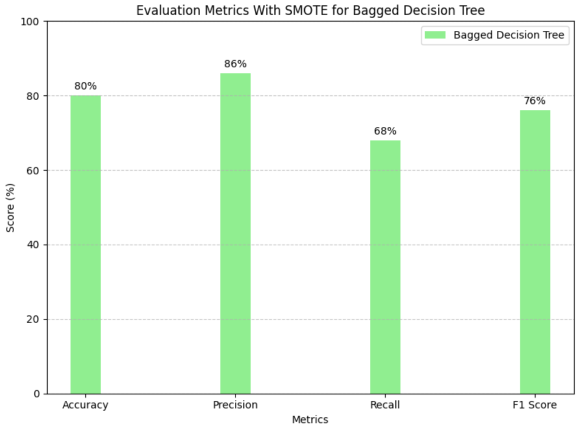


Figure 7. Model Evaluation Graphics Bagged Decision Tree

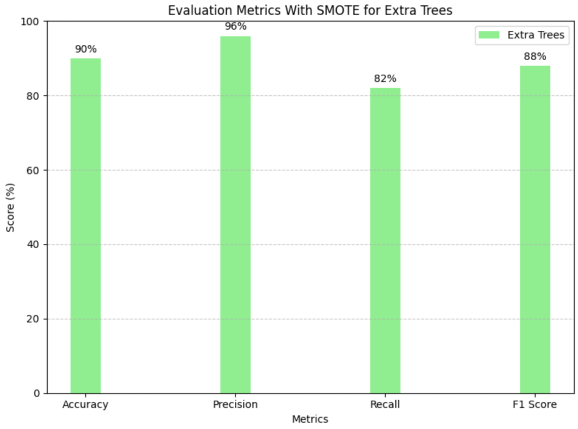


Figure 8. Model Evaluation Graphics Extra Trees

The technique is applied to equalize the distribution of data between minority and majority classes[18]. Categorization of heart condition status (0 = no disease, 1 = heart disease detected). Some of the equations of the metrics that will be used are Equations 7 to 10.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Presicion = \frac{TP}{TP+FP} \quad (8)$$

DOI : https://doi.org/10.52158/we9asn06

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$\text{F1} - \text{Score} = 2 \text{ x } \frac{Presicion \text{ } x \text{ } Recal}{Presicion+Recal} \quad (10)$$

TP(True Positive) refers to the number of correct predictions of positive cases; TN(True Negative) refers to the number of correct predictions of negative cases; FP(False Positive) indicates the count of incorrectly predicted positive cases among actual negative instances; and FN(False Negative) represents the amount of positive data that was erroneously identified as a negative category

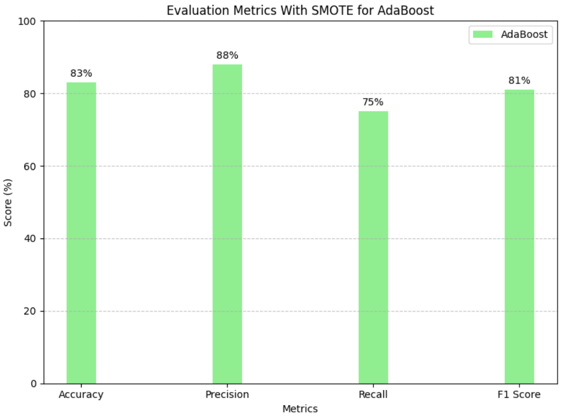


Figure 9. Model Evaluation Graphics AdaBoost

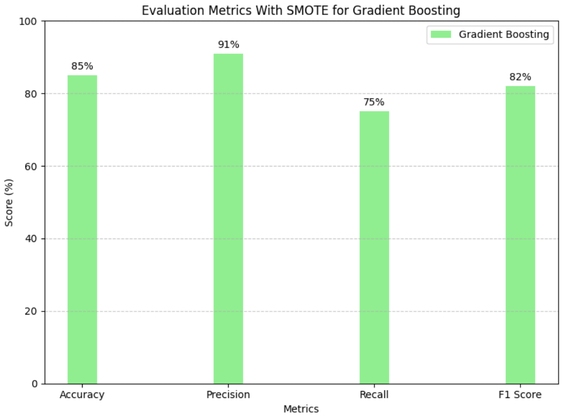


Figure 10. Model Evaluation Graphics Gradient Boosting

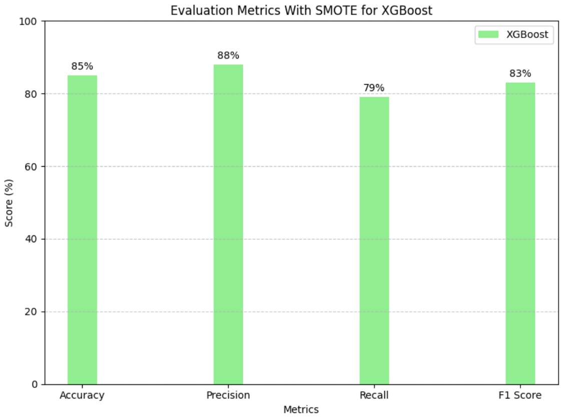


Figure 11. Model Evaluation Graphics XGBoost

3. Results and Discussions

The performance of the six ensemble models was evaluated both before and after applying SMOTE to the training data. The results, summarized in Table 1 and Table 2, indicate that the data balancing step had a notable and selective impact on model performance. Extra Trees model emerged as the clear top performer, achieving the highest accuracy (90%) and F1-Score (88%). Its superior performance can be attributed to its dual-randomization mechanism. By randomizing both the feature subset and the split threshold at each node, Extra Trees builds more diverse trees, which effectively reduces model variance and makes it less susceptible to overfitting on this dataset. This finding is consistent with literature suggesting that increased randomness can lead to more robust models in complex classification tasks. The application of SMOTE provided a meaningful improvement for Extra Trees, increasing its recall from 79% to 82% without compromising its high precision of 96%.

This shows SMOTE successfully helped the model identify more true positive cases,Random Forest model also demonstrated strong and stable performance, maintaining its accuracy at 88% and F1-Score at 86% both before and after SMOTE.This stability suggests that Random Forest is inherently robust to the mild class imbalance present in the original dataset. Similarly, XGBoost The boosting algorithms showed varied responses to SMOTE,AdaBoost saw a trade-off: its accuracy slightly decreased from 85% to 83%, but its recall improved from 71% to 75%.

This indicates that while SMOTE made the model slightly less accurate overall, it became more balanced in its ability to detect positive cases,Gradient Boosting performance remained largely unchanged, suggesting that its internal mechanism of correcting residual errors was not significantly enhanced by the oversampling technique, Bagged Decision Tree consistently remained the weakest algorithm, with an accuracy of 80% and the lowest recall of 68%, indicating a persistent difficulty in avoiding false negatives [19], [20].

Table 4 is the result of the confusion matrix after the implementation of SMOTE and the selection of the best parameters via GridSearchCV.

Table 4. Confusion Matrix Model After SMOTE

Algorithms	Confusion Matrix (TN, FP, FN, TP)
Random Forest	[[96, 8], [7, 110]]
Bagged Decision Tree	[[94, 10], [9, 108]]
Extra Trees	[[97, 7], [8, 109]]
AdaBoost	[[92, 12], [11, 106]]
Gradient Boosting	[[95, 9], [8, 109]]
XGBoost	[[96, 8], [7, 110]]

DOI : https://doi.org/10.52158/we9asn06

Figure 12 illustrates the age distribution of patients in the dataset. The histogram shows that the majority of patients fall within the age range of 50–60 years, with the highest frequency observed around 58 years old. This indicates that middle-aged and older individuals constitute the largest proportion of cases, which is consistent with clinical evidence suggesting that the risk of heart disease increases with age. In contrast, the number of patients under the age of 40 is relatively small, highlighting that heart disease tends to be less common in younger populations. This age distribution provides important context for the study, as it reflects the demographic characteristics of patients most affected by the condition and may influence the performance of predictive models.

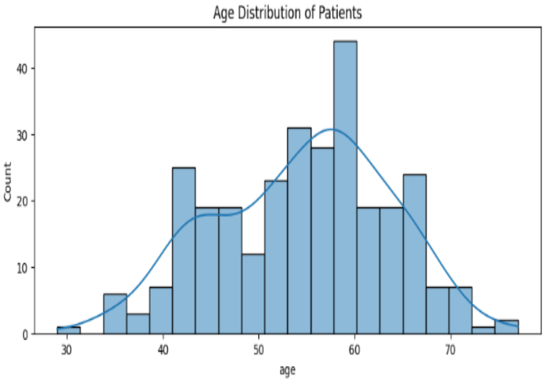


Figure 12. Age Distribution of Patients

4. Conclusions

This study conducted a comprehensive comparative analysis of six ensemble learning algorithms for heart disease classification, with a focus on the impact of the SMOTE data balancing technique. The key finding is that the Extra Trees algorithm, after the application of SMOTE on the training data, delivered the most optimal and reliable performance, achieving 90% accuracy and an 88% F1-Score. This demonstrates that a model leveraging high randomization is particularly effective for this type of clinical dataset.

The application of SMOTE had a selective impact. It provided a clear benefit to Extra Trees by improving its recall by 3% without sacrificing precision. However, for inherently robust algorithms like Random Forest and XGBoost, their performance remained stable, indicating that data balancing is not universally required for all models on this mildly imbalanced dataset. The results confirm that the combination of a suitable ensemble method and a careful data balancing strategy is crucial for developing effective clinical decision support systems. Future work could involve applying these models to larger, more diverse datasets and exploring advanced feature selection techniques to further enhance predictive accuracy.

References

[1] N. Nasution, F. Nasution, and M. A. Hasan, “Heart Disease Risk Prediction: Evaluating Machine Learning Algorithms With Feature Reduction Using Lda,” *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 11, no. 1, pp. 9–16, 2024, doi: 10.33330/jurteksi.v11i1.3498.

[2] M. K. Dwipa Jaya, “Perbandingan Random Forest, Decision Tree, Gradient Boosting, Logistic Regression untuk Klasifikasi Penyakit Jantung,” *Jnatia*, vol. 2, no. November, pp. 1–5, 2023.

[3] Irpanudin, Reka, R. Nur Anggraeni, P. Pratama, A. Sujjada, and A. Fergina, “Prediksi Penyakit Jantung Menggunakan Metode Deep Neural Network dengan Memanfaatkan Internet of Things,” *J. Inf. dan Teknol.*, vol. 5, pp. 45–55, 2023, doi: 10.37034/jidt.v5i2.330.

[4] N. H. Alfajr and S. Defiyanti, “Prediksi Penyakit Jantung Menggunakan Metode Random Forest Dan Penerapan Principal Component Analysis (Pca),” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3S1, 2024, doi: 10.23960/jitet.v12i3s1.5055.

[5] A. Rahmada and E. R. Susanto, “Peningkatan Akurasi Prediksi Penyakit Jantung dengan Teknik SMOTEENN pada Algoritma Random Forest,” *J. Pendidik. dan Teknol. Indones.*, vol. 4, no. 12, pp. 795–803, 2025, doi: 10.52436/1.jpti.524.

[6] G. Fernando, H. M. Zaydan, I. Akbar, and R. Dwi Irawan, “Biner : Jurnal Ilmu Komputer, Teknik dan Multimedia Klasifikasi Penyakit Jantung Menggunakan Extreme Gradient Boosting,” vol. 2, no. 5, pp. 667–670, 2024, [Online]. Available: https://journal.mediapublikasi.id/index.php/Biner

[7] D. Sitanggang, N. Nicholas, V. Wilson, A. R. A. Sinaga, and A. D. Simanjuntak, “Implementasi Data Mining Untuk Memprediksi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor Dan Logistic Regression,” *J. Tek. Inf. dan Komput.*, vol. 5, no. 2, p. 493, 2022, doi: 10.37600/tekinkom.v5i2.698.

[8] C. A. Ramadan, F. E. Fahriza, F. Hidayatullah, and M. T. Amru, “Literatur Review : Pendekatan Ensemble Learning untuk Klasifikasi Penyakit Jantung Koroner,” vol. 2, no. 3, pp. 482–486, 2024.

[9] S. A. T. Al Azhima, D. Darmawan, N. F. Arief Hakim, I. Kustiawan, M. Al Qibtiya, and N. S. Syafei, “Hybrid Machine Learning Model untuk memprediksi Penyakit Jantung dengan Metode Logistic Regression dan Random Forest,” *J. Teknol. Terpadu*, vol. 8, no. 1, pp. 40–46, 2022, doi: 10.54914/jtt.v8i1.539.

[10] A. A. Surya and Y. Yamasari, “Penerapan Algoritma Naïve Bayes (NB) untuk Klasifikasi Penyakit Jantung,” *J. Informatics Comput. Sci.*, vol. 5, no. 03, pp. 447–455, 2024, doi: 10.26740/jinacs.v5n03.p447-455.

[11] S. A. Kamila, R. S. Sulistijowati, and I. Susanto, “Klasifikasi Penyakit Jantung Menggunakan Decision Tree dan Random Forest,” *Semin. Nas. Teknol. &Amp; Sains*, vol. 2, no. 1, pp. 7–12, 2023.

[12] M. I. Aziz, A. Z. Fanani, and A. Affandy, “Analisis Metode Ensemble Pada Klasifikasi Penyakit Jantung Berbasis Decision Tree,” *J. Media Inform. Budidarma*, vol. 7, no. 1, p. 1, 2023, doi: 10.30865/mib.v7i1.5169.

[13] A. F. Anjani, D. Anggraeni, and I. M. Tirta, “Implementasi Random Forest Menggunakan SMOTE untuk Analisis Sentimen Ulasan Aplikasi Sister for Students UNEJ,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 2, pp. 163–172, 2023, doi: 10.25077/teknosi.v9i2.2023.163-172.

[14] F. M. Natsir, R. Y. Bakti, and T. Wahyuni, “Analisis Deteksi Dini Penyakit Jantung dengan Pendekatan Support Vector Machine pada Data Pasien,” *Arus J. Sains dan Teknol.*, vol. 2, no. 2, pp. 437–446, 2024, doi: 10.57250/ajst.v2i2.669.

[15] S. G. Barus, “Klasifikasi Sentimen Data Tidak Seimbang Menggunakan Algoritma Smote Dan K-Nearest Neighbor Pada Ulasan Pengguna Aplikasi Pedulilindungi,” *Senamika*, pp. 162–173, 2022.

[16] D. Wijayanto, R. Marco, A. Sidauruk, and M. Sulistiyono, “The Effect of SMOTE and Optuna Hyperparameter Optimization on TabNet Performance for Heart Disease Classification,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 14, no. 2, pp. 156–

164, 2025, doi: 10.32736/sisfokom.v14i2.2348.

[17] G. F. Fahrudin, S. Suroso, and S. Soim, “Pengembangan Model Support Vector Machine untuk Meningkatkan Akurasi Klasifikasi Diagnosis Penyakit Jantung,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 7, no. 3, pp. 1418–1428, 2024, doi: 10.32493/jtsi.v7i3.42254.

[18] Y. Amelia, “Perbandingan Metode Machine Learning Untuk Mendeteksi Penyakit Jantung,” *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 220–225, 2023, doi: 10.36080/idealis.v6i2.3043.

[19] R. Waluyo and A. S. Munir, “Optimasi Prediksi Kematian pada Gagal Jantung Analisis Perbandingan Algoritma Pembelajaran Ensemble dan Teknik Penyeimbangan Data pada Dataset,” *J. Sist. dan Teknol. Inf.*, vol. 12, no. 2, p. 365, 2024, doi: 10.26418/justin.v12i2.75158.

[20] A. S. Prabowo and F. I. Kurniadi, “Analisis Perbandingan Kinerja Algoritma Klasifikasi dalam Mendeteksi Penyakit Jantung,” *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 7, no. 1, pp. 56–61, 2023, doi: 10.47970/siskom-kb.v7i1.468.