JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 48 – 62
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Extreme Gradient Boosting Model with SMOTE for Heart Disease Classification

Ahmad Ubai Dullah [(1)*], Aditya Yoga Darmawan [(2)], Dwika Ananda Agustina Pertiwi [(3)], Jumanto Unjung [(4)]
[1,2,4] Department of Computer Science, Universitas Negeri Semarang, Semarang, Indonesia
[3] Faculty of Technology Management and Business, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
e-mail : {ubaid,darmoenoyoga}@students.unnes.ac.id, hp220072@student.uthm.edu.my, jumanto@mail.unnes.ac.id.
* Corresponding author.
This article was submitted on 18 October 2024, revised on 9 November 2024, accepted on 10 November 2024, and published on 31 January 2025.

Abstract

Heart disease is one of the leading causes of death worldwide. According to data from the World Health Organisation (WHO), the number of victims who die from heart disease reaches 17.5 million people every year. However, the method of diagnosing heart disease in patients is still not optimal in determining the right treatment. Along with technology development, various models of machine learning algorithms and data processing techniques have been developed to find models that can produce the best precision in classifying heart disease. This research aims to develop a machine learning algorithm model in classifying heart disease to improve the effectiveness of diagnosis and help in determining the right treatment for patients. This research also aims to overcome the limitations of accuracy in existing diagnosis methods by identifying models capable of providing the best results in processing and analysing health data, especially in terms of heart disease classification. In this study, the XGBoost model was identified as the most superior, with an accuracy of 99%. These results show that the XGBoost model has a higher accuracy rate than previous methods, making it a promising solution to improve the accuracy of future heart disease diagnosis and classification.

Keywords: Heart Disease, SMOTE, XGBoost, KNN, SVM

Abstrak

Penyakit jantung adalah salah satu penyebab utama kematian di seluruh dunia. Menurut data dari World Health Organisation (WHO), jumlah korban yang meninggal akibat penyakit jantung mencapai 17,5 juta orang setiap tahunnya. Meski demikian, metode diagnosis penyakit jantung pada pasien masih belum optimal dalam menentukan penanganan yang tepat. Seiring dengan perkembangan teknologi, berbagai model algoritma *machine learning* dan teknik pengolahan data telah dikembangkan untuk menemukan model yang dapat menghasilkan akurasi terbaik dalam mengklasifikasikan penyakit jantung. Penelitian ini bertujuan untuk mengembangkan model algoritma *machine learning* dalam mengklasifikasikan penyakit jantung, sehingga dapat meningkatkan efektifitas diagnosa dan membantu dalam menentukan pengobatan yang tepat bagi pasien. Penelitian ini juga bertujuan untuk mengatasi keterbatasan akurasi pada metode diagnosis yang sudah ada, dengan cara mengidentifikasi model yang mampu memberikan hasil terbaik dalam mengolah dan menganalisa data kesehatan, khususnya dalam hal klasifikasi penyakit jantung. Pada penelitian ini, model XGBoost diidentifikasi sebagai model yang paling unggul, dengan akurasi sebesar 99%. Hasil ini menunjukkan bahwa model XGBoost memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan metode-metode sebelumnya, sehingga dapat menjadi solusi yang menjanjikan dalam meningkatkan akurasi diagnosis dan klasifikasi penyakit jantung di masa depan.

Kata Kunci: Penyakit Jantung, SMOTE, KNN, XGBoost, SVM

1. INTRODUCTION

Heart disease is one of the diseases that is considered to be the main cause of death of a person. Victims of heart disease and stroke are as many as 17.5 million people each year around the world, according to reports from the World Health Organization (WHO) (Baccouche et al., 2020; Xu et al., 2022). Heart disease is a collection of several conditions that affect human heart health (Benhar et al., 2020; Matin Malakouti, 2023). Some of these conditions include diseases of the blood vessels such as heart attack, stroke, heart failure, and arrhythmia. (El-Sofany, 2024; Radhika & Thomas George, 2021; Subathra & Sumathy, 2024). Two terms usually confuse most people, namely, the terms “heart disease” and ‘cardiovascular disease’, which is a situation that can cause heart attack, stroke, and chest pain (Maity et al., 2023; Pan et al., 2020). With the development of science, collecting data on heart disease is easier to obtain and analyse, helping to develop early diagnosis of heart disease (Ammar et al., 2021; Hossain et al., 2023).

In properly diagnosing patients with heart disease, it is necessary to classify heart disease. Research has been conducted on the classification of heart disease by J. P. Li et al. (2020) in 2020 using several machine learning classification models such as Naive Bayes, Support Vector Machine, Logistic Regression, Artificial Neural Network, Decision Tree, And k-Nearest Neighbor which are combined with several feature extractions to assist in data processing. Feature extraction is used to extract features from data that will be used to determine classification parameters. The results obtained have the best accuracy of 92.37% from the SVM model with FCMIM feature extraction. Another research by El-Sofany (2024) aims to employ three different feature selections such as chi-square, analysis of variance (ANOVA), and mutual information. This study also uses various machine learning such as Naive-Bayes, Support Vector Machine (SVM), Voting, XGBoost, AdaBoost, bagging, Decision Tree, K-Nearest Neighbor, Random Forest, Logistic Regression to classify heart disease. Using the SF-2 feature subset that contains 10 of 14 features and XGBoost with SMOTE to oversample the imbalanced data set from the combined Cleaveland Heart Disease Dataset and private dataset, the model reached an accuracy of 97,35%. Manikandan et al. (2024) Using Boruta feature selection and comparing 5 Machine Learning model performance such as Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and XGBoost. The Cleaveland Heart Disease Data Set (Ashtaiwi et al., 2024) was used to train and test the model. The study achieved an accuracy of 88,52% using logistic regression. Another thing from this paper is that Boruta Feature Selection also improve model accuracy for the Support Vector Machine and Decision Tree, but this feature selection method also lower accuracy for the Random Forest (Gárate-Escamila et al., 2020) and XGBoost model, while Logistic Regression receives no improvement on accuracy. Using a newer dataset from Maghdid & Rashid (2022), research from Anshori & Haris (2022) uses logistic regression, support vector machine (SVM) and Linear Discriminant Analysis (LDA) to classify heart disease. The data is considered clean, and the researcher did not specify the training and test split amount. Cross-validation was used to evaluate each models, and Logistic Regression was the best model in their research, reaching 81,35% accuracy.

However, the model used is not optimal enough to classify heart disease, so there is a need to increase the resulting accuracy. Machine learning classification methods are increasingly developing, and new models are starting to emerge that can produce more optimal accuracy. Therefore, an analysis is needed to compare the machine learning models that have been developed to obtain more accurate results. Some models that will be used in this study include XGBoost (Chen et al., 2022; Mamun et al., 2022; Muslim et al., 2023), Support Vector Machine (M. Li et al., 2021; Wazrah & Alhumoud, 2021), Decision Tree (Haznedar & Simsek, 2022; Huang & Chen, 2022), Naive Bayes (Gibson et al., 2020), Logistic Regression (Bengesi et al., 2023), and K-nearest Neighbor (Islam et al., 2023).

2. METHODS

The methods used in this study are several machine learning classification models, namely XGBoost, SVM, Decision Tree, Logistic Regression, KNN, and Naive Bayes. Before being processed with the research model, the data will be processed at the preprocessing stage with

several stages such as cleaning and replacing values with numeric. Then sampling is carried out with SMOTE, the data will be trained with the model and produce an evaluation matrix. The methods in this study will be explained in the next section and shown in Figure 1.

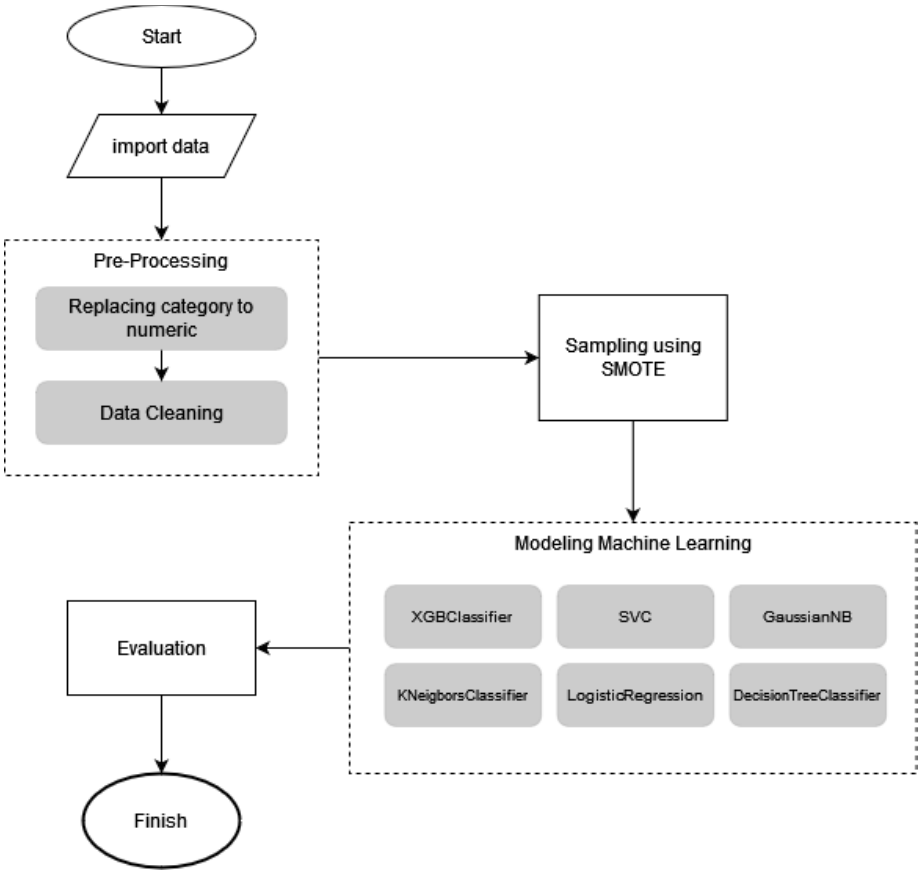

Figure 1 Flowchart for Proposed Method

2.1 Data Collection, Preprocessing and Sampling using SMOTE

The dataset used in this study is "An Extensive Dataset for the Heart Disease Classification System" released on Mendeley Data (Maghdid & Rashid, 2022). This dataset contains 1319 data with nine feature. There are 2 classes, 'positive' for CVD Positive with 810 data and 'negative' for CVD negative with 509 data. In data preprocessing, the data will be changed in value in the class column into binary form, which was originally positive and negative will change to 0 and 1. Negative will become 0, and positive will become 1. Because the data used in this study is text data, it is necessary to check for missing values and duplicates and remove them since missing and duplicate data will decrease model performance. Numerical Feature will be scaled using Formula 1.

$$z = \frac{(x - u)}{s} \quad (1)$$

With z is Scaled data, x is data before scaled, u is the mean of the data, and s is the standard deviation of the data. The Standard Scaler is performed using StandardScaler from the sklearn library.

After cleaning the data and checking the missing values, another Explorative Data Analysis is performed to check the balance of the data. Unbalanced data will affect the results obtained by the classification model. Methods to overcome data imbalance can use sampling. One library that

can be used is SMOTE (Sridhar & Sanagavarapu, 2021). SMOTE is a method for creating data samples to adjust the most data from each category to produce a good data balance. The data used in this study is unbalanced in the category for the class column. Based on the distribution of the class column, there are 61.4% of data with a value of 1 (Positive) and 38.6% of data with a value of 0 (negative). The distribution of data is uneven and needs to be balanced in order to get maximum results.

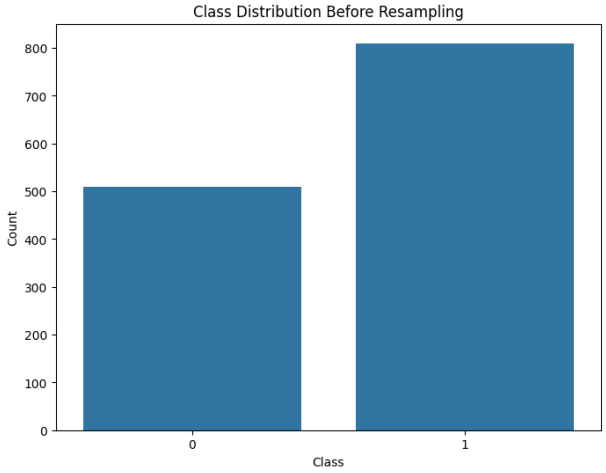

Figure 2 Before Resampling

2.2 Modeling Machine Learning

Modeling process begins with the process of importing machine learning classifier libraries, such as XGBClassifier for XGBoost, SVC for SVM, KNeigborClassifier for KNN, LogisticRegression for Logistic Regression, DecisionTreeClassifier for Decision Tree, and GaussianNB for Naive Bayes. Firstly, one of the machine learning algorithms, Naive Bayes, is based on the Bayes theory and assumes that every feature is independent (naive assumption) (El-Sofany, 2024). Even if the features are frequently not entirely independent, this algorithm is quite robust and effective, especially regarding text classification, such as spam filtering, sentiment analysis, and document generation. Gaussian Naive Bayes (GaussianNB) (Ningsih et al., 2024) is an initialization model. This algorithm summarizes that some parameters agree with a Gaussian (normal) distribution. Gaussian Naive Bayes is typically used when the fit is continuous.

Secondly, Support Vector Machine (SVM) is a machine learning algorithm that is highly effective for classification and regression tasks (Bengesi et al., 2023). SVM operates by searching for a hyperplane that maximizes the margin of error for each data set. This makes SVM extremely effective at solving classification problems, particularly when data cannot be processed linearly (Obiedat et al., 2022). Initialization of the SVM model using a linear kernel (kernel='linear') (Rofik et al., 2024). The type of hyperplane that is used to sift data is determined by the kernel. In this case, the linear kernel that is evaluated means that the model will search for linear terms or linear polynomials.

K-Nearest Neighbors (KNN) is a machine learning algorithm for classification and regression. KNN (El-Sofany, 2024) operates according to the following principle: when new data is provided, KNN determines the class or value of the data based on the k data matched in the training dataset. K-Nearest Neighbors (jabbar et al., 2013) model is analyzed with the parameter n_neighbors=5. Accordingly, the model will employ five lateral tangents to determine the new data set. KNN can't learn like other algorithms; instead, it just provides long-term data that can be used for prediction. The Logistic Regression algorithm is a machine learning technique used for classification (Patidar et al., 2022), primarily for binary classification problems (two classes). Despite being called "regression," logistic regression is a classification model rather than a linear regression. The

model is trained with a 'max_iter=1000' parameter. This parameter sets the maximum number of iterations for the optimization algorithm used in model training.

A decision Tree is a machine learning algorithm for regression and classification. Decision trees break down datasets into smaller subsets based on the current feature until they reach the end (leaf from tree) (Huang & Chen, 2022). This graph's structure is composed of single-simulation (nodes) that monitor a feature or attribute, branch-branch (branches) that monitor a feature's values, and leaf-branch (leaves) that monitor a class or prediction. Decision Tree Classifier from the scikit_learn library is used to create a probabilistic model for classification (Oh, 2021). Lastly, Extreme Gradient Boosting, or XGBoost, is a popular and effective machine learning algorithm for classification and regression tasks (El-Sofany, 2024). XGBoost is a single boosting technique that uses ensemble learning to maximize prediction model performance. XGBoost model initialization for classification using XGBClassifier from the XGBoost source (Zhang & Gong, 2020). This parameter uses the default value from the XGBoost.

2.3 Evaluation Model

The evaluation model uses a confusion matrix with the following composition, F1-score accuracy, recall, and precision. This performance analysis focuses on the accuracy produced by the proposed model compared to previous research. The mathematical formula 2, 3, 4, and 5 is used to analyse the results using confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1\ Score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (5)$$

True Positive (TP) refers to the number of correct instances identified as positive. True Negative (TN) represents the number of incorrect instances identified as negative. False Positive (FP) occurs when correct instances are mistakenly classified as positive, while False Negative (FN) happens when incorrect instances are misclassified as positive.

3. RESULTS AND DISCUSSION

3.1 Result

The results of this study contain the results of processing on the research model. Several stages of the process are passed, such as preprocessing and the results of model testing and evaluation.At the data cleaning stage, checking and cleaning the data is carried out so that the research model can process it. The results of the data-cleaning process can be seen in Table 1. It can be seen that the data is clean from missing values. However, the range age column is not used because it is better to use the age column. Therefore, the column will be dropped, and duplicate data will be cleaned up.

The data that is changed is the data in the class column used as a label. Here the class column contains data in the form of objects with contents, positive and negative. Then, the data must be converted to a numeric form to facilitate data processing in the research model. So, the data will be changed to 1 for positive and 0 for negative. Replacing value results can be seen in Table 2. The algorithm addresses the class imbalance in the target variable using the Synthetic Minority Over-sampling Technique (SMOTE). The training data is first divided into the target variable (y)

and characteristics (X). All columns are included in the features, except the target column "class," and the data from that column is contained in column y. To balance the class distribution, synthetic samples of the minority class are subsequently created using SMOTE. Results are guaranteed to be consistent when random_state=42 is used. The original dataset is smaller than the resampled data, X_resampled and y_resampled. Lastly, the code prints the shapes from the original and resampled datasets to show the modifications. The results of the sampling can be seen in Table 3.

Table 1 Result from Cleaning Data

Column Name	Value
age	0
gender	0
impluse	0
pressurehight	0
pressurelow	0
glucose	0
kcm	0
troponin	0
class	0
Age_Range	0

Table 2 Result from Replacing Value

Class before replacing	Class after replacing
Negative	0
Positive	1

Table 3 Result from Resampling

Before sampling using SMOTE		After sampling using SMOTE	
Class	Count	Class	Count
1	647	1	647
0	408	0	647

Data sampled with SMOTE will follow the largest amount of data, which is 647 data at value 1. Therefore, the value 0 data will change to 647, originally 408 data, that way the data used will be balanced. As for result without SMOTE, the training data that is not sampled using SMOTE oversampling has been tested using the research model with the results as a classification report as follows.

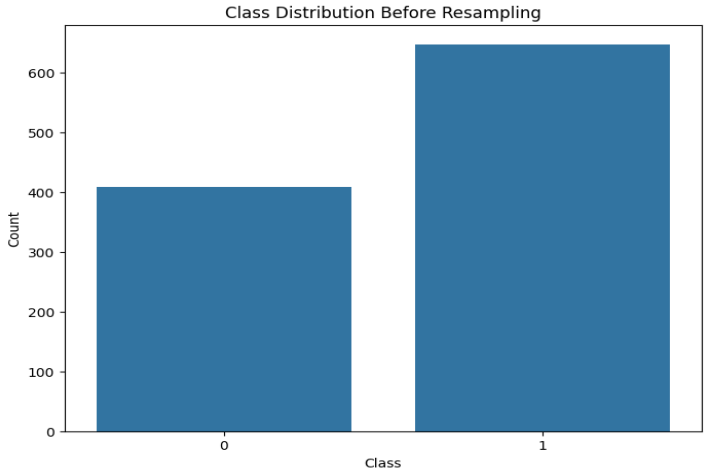

Figure 3 Before Resampling

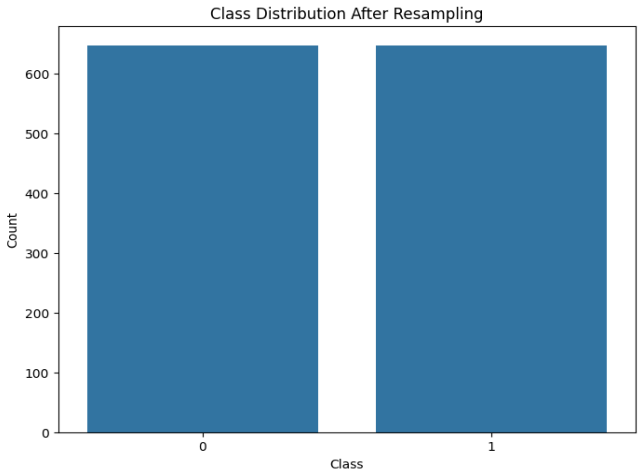

Figure 4 After Resampling

3.1.1 Naïve Bayes without SMOTE

The results obtained by the Naïve Bayes research model using data that is not oversampled can be seen in Table 4 as a classification report. The classification report shows the model's performance in distinguishing between two classes, '0' and '1'. For class 0, the model obtained an average precision (64.52%) but a very high recall rate (99.01%), which means that the model identified the largest number of correct examples in this class, resulting in a decent F1 value of 0.7812. For class 1, the model achieved high precision (99.08%) but lower recall (66.26%), meaning that the model missed a few correct examples from this class, with an F1 value of 0.7941. Overall, the model had a precision of 78.79%. The macro averages (precision 0.8180, recall 0.8263, F1 score 0.7877) show a balanced performance across the two classes, while the weighted average considers the class distribution.

Table 4 Result Naive Bayes without SMOTE

	Precision	Recall	F1-score	Support
0	0.6452	0.9901	0.7812	101
1	0.9908	0.6626	0.7941	163
Accuracy			0.7879	264
Macro avg	0.8180	0.8263	0.7877	264
Weighted avg	0.8586	0.7879	0.7892	264

3.1.2 Support Vector Machine without SMOTE

Table 5 Result from SVM without SMOTE

	Precision	Recall	F1-score	Support
0	0.7282	0.7426	0.7353	101
1	0.8385	0.8282	0.8333	163
Accuracy			0.7955	264
Macro avg	0.7833	0.7854	0.7843	264
Weighted avg	0.7963	0.7955	0.7958	264

The results obtained by the support vector machine research model using data that is not oversampled can be seen in Table 5 as a classification report. This classification report shows the performance of the model for two classes, '0' and '1'. For class 0, the model has an accuracy of 72.82% and a recall of 74.26%, resulting in an F1 score of 0.7353. For class 1, the model achieved a higher accuracy of 83.85% and a higher recall of 82.82% with an F1 value of 0.8333. The overall accuracy of this model was 79.55%. The macro-mean (precision 0.7833, recall

0.7854, F1-score 0.7843) showed a balanced performance for both classes, while the weighted mean (precision 0.7963, recall 0.7955, F1-score 0.7958) reflected the class distribution, suggesting that the model performed quite well for both classes, with a slight advantage for the prediction of class 1.

3.1.3 K-Nearest Neighbor without SMOTE

The results obtained by the k-nearest neighbor research model using data that is not oversampled can be seen in Table 6 as a classification report. This classification report shows the model's performance for classes '0' and '1'. For class 0, the model has an accuracy of 57.45% and a recognition rate of 53.47%, giving an F1 value of 0.5538. Class 1 has a higher accuracy of 72.35% and a recognition rate of 75.46%, giving an F1 value of 0.7387. The overall accuracy was 67.05%. The macro average (precision 0.6490, recall 0.6446, F1 score 0.6463) shows moderate performance for all classes, while the weighted average (precision 0.6665, recall 0.6705, F1 score 0.6680) reflects slightly better performance for class 1 due to greater support, indicating that the model generally prefers class 1 over class 0 in its predictions.

Table 6 Result from KNN without SMOTE

	Precision	Recall	F1-score	Support
0	0.5745	0.5347	0.5538	101
1	0.7235	0.7546	0.7387	163
Accuracy			0.6705	264
Macro avg	0.6490	0.6446	0.6463	264
Weighted avg	0.6665	0.6705	0.6680	264

3.1.4 Logistic Regression without SMOTE

The results obtained by the logistic regression research model using data that is not oversampled can be seen in Table 7 as a classification report. This classification report with precision, recall, and F1-score metrics for two classes labeled "0" and "1." For class "0," the precision is 0.7609, recall is 0.6931, and F1-score is 0.7254, based on 101 instances. For class "1," the precision is higher at 0.8198, with a recall of 0.8650 and an F1-score of 0.8318, based on 163 instances. The model's overall accuracy is 0.7992, indicating that it correctly classified approximately 79.92% of the samples. Additionally, the macro average (averaging both classes without considering class imbalance) and weighted average (considering class imbalance) for F1-scores are 0.7836 and 0.7973, respectively, reflecting consistent performance across both classes.

Table 7 Result Logistic Regression Without SMOTE

	Precision	Recall	F1-score	Support
0	0.7609	0.6931	0.7254	101
1	0.8198	0.8650	0.8318	163
Accuracy			0.7992	264
Macro avg	0.7903	0.7790	0.7836	264
Weighted avg	0.7972	0.7992	0.7973	264

3.1.5 Decision Tree without SMOTE

The results obtained by the decision tree research model using data that is not oversampled can be seen in Table 8 as a classification report. This classification report shows that the model decision tree has high precision, recall, and F1 scores for both classes. For class "0," precision, recall, and F1-score are all 0.9703, based on 101 instances. For class "1," these metrics are slightly higher, with precision, recall, and F1-score of 0.9816, based on 163 instances. The model achieves an overall accuracy of 0.9773, indicating that it correctly classified approximately 97.73% of the samples. The macro average and weighted average F1-scores are around 0.9759 and 0.9773, respectively, reflecting consistently high performance across both classes.

Table 8 Result Decision Tree without SMOTE

	Precision	Recall	F1-score	Support
0	0.9703	0.9703	0.9703	101
1	0.9816	0.9816	0.9816	163
Accuracy			0.9773	264
Macro avg	0.9759	0.9759	0.9759	264
Weighted avg	0.9773	0.9773	0.9773	264

3.1.6 Extreme Gradient Boosting without SMOTE

The results obtained by the extreme gradient boosting research model using data that is not oversampled can be seen in Table 9 in the form of a classification report. This classification report demonstrates excellent performance across both classes. For class "0," the precision is 0.9800, recall is 0.9703, and the F1-score is 0.9751, based on 101 instances. For class "1," the precision is slightly higher at 0.9817, with a recall of 0.9877 and an F1-score of 0.9847, based on 163 instances. The model achieves an overall accuracy of 0.9811, indicating it correctly classified approximately 98.11% of the samples. The macro average and weighted average F1-scores are 0.9799 and 0.9810, respectively, reflecting strong and consistent performance across both classes.

Table 9 Result XGBoost without SMOTE

	Precision	Recall	F1-score	Support
0	0.9800	0.9703	0.9751	101
1	0.9817	0.9877	0.9847	163
Accuracy			0.9811	264
Macro avg	0.9809	0.9790	0.9799	264
Weighted avg	0.9811	0.9811	0.9810	264

As a result with SMOTE, the model is also trained with training data oversampled using SMOTE, which produces the following classification report. The results of this case are depicted as follows.

3.1.7 Naïve Bayes with SMOTE

The results of testing the naive bayes model with data balanced with SMOTE obtained classification report results, which can be seen in Table 10. The model achieved an accuracy of 80%, indicating that 80% of the predictions were correct. For class 0, the precision is 0.65, meaning 65% of the predicted class 0 instances were correct, with a high recall of 0.99, showing that almost all actual class 0 instances were correctly identified. For class 1, the precision is 0.99, indicating strong performance in predicting class 1, but the recall is lower at 0.68, meaning that only 68% of actual class 1 instances were correctly predicted. The F1 scores, which balance precision and recall, are 0.79 for class 0 and 0.80 for class 1. The macro average, which averages precision, recall, and F1-score across both classes without accounting for class imbalance, shows precision at 0.82, recall at 0.83, and F1-score at 0.80. The weighted average, which considers class imbalance, gives similar values with precision at 0.86, recall at 0.80, and F1-score at 0.80. The model performs well for class 0 but shows weaker recall for class 1, indicating room for improvement in predicting that class.

Table 10 Result from Naive Bayes with SMOTE

	Precision	Recall	F1-score	Support
0	0.6536	0.9901	0.7874	101
1	0.9910	0.6748	0.8029	163
Accuracy			0.7955	264
Macro avg	0.8223	0.8325	0.7952	264
Weighted avg	0.8619	0.7955	0.7970	264

3.1.8 Support Vector Machine with SMOTE

The results of testing the support vector machine model with data balanced with SMOTE obtained classification report results, which can be seen in Table 11. The performance metrics of an SVM (Support Vector Machine) model achieved an accuracy of 78%. This indicates that the model correctly classified 78% of the instances. For class 0, the model has a precision of 0.62, meaning 62% of the predicted class 0 instances were correct, and a high recall of 0.97, indicating that 97% of the actual class 0 instances were accurately identified. For class 1, the precision is higher at 0.97, but the recall is lower at 0.66, meaning only 66% of the actual class 1 instances were correctly predicted. The F1 scores, which balance precision and recall, are 0.77 for class 0 and 0.79 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.81, 0.82, and 0.78, respectively. The weighted average, which accounts for class imbalance, has slightly different results, where the precision, recall, and F1-score are 0.85, 0.78, 0.78.

Table 11 Result from SVM with SMOTE

	Precision	Recall	F1-score	Support
0	0.6405	0.9703	0.7717	101
1	0.9730	0.6626	0.7883	163
Accuracy			0.7803	264
Macro avg	0.8067	0.8164	0.7800	264
Weighted avg	0.8454	0.7803	0.7819	264

3.1.9 K-Nearest Neighbor with SMOTE

The results of testing the K-nearest Neighbor model with data balanced with SMOTE obtained classification report results, which can be seen in Table 12. The performance metrics of a K-Nearest Neighbors (KNN) classification model achieved an accuracy of approximately 0.65 (65%). This means the model correctly predicted the class for 68% of the instances. For class 0, the precision is 0.54, indicating that 54% of the predicted class 0 instances were correct, while the recall is 0.65, meaning 65% of the actual class 0 instances were identified correctly. For class 1, the precision is 0.75, but the recall is lower at 0.65, meaning only 65% of the actual class 1 instances were predicted correctly. The F1 scores, which balance precision and recall, are 0.59 for class 0 and 0.70 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.64, 0.65, and 0.64, indicating that the model performs similarly for both classes. The weighted averages account for class imbalance and result in 0.67 for precision, 0.65 in recall, and 0.64 in F1-score. Overall, the KNN model has moderate performance, showing some difficulty distinguishing between the two classes, particularly with a lower recall for class 1. This indicates room for improvement in the model's predictive capability.

Table 12 Result from KNN with SMOTE

	Precision	Recall	F1-score	Support
0	0.5366	0.6535	0.5893	101
1	0.7518	0.6503	0.6974	163
Accuracy			0.6515	264
Macro avg	0.6442	0.6519	0.6433	264
Weighted avg	0.6694	0.6515	0.6560	264

3.1.10 Logistic Regression with SMOTE

The results of testing the logistic regression model with data balanced with SMOTE obtained classification report results, which can be seen in Table 13. The performance metrics of the Logistic Regression classification model achieved an accuracy of approximately 0.79 (79%). This means the model correctly predicted the class for 79% of the instances. For class 0, the precision is 0.67, indicating that 67% of the predicted class 0 instances were correct, while the recall is

0.86, meaning 86% of the actual class 0 instances were identified correctly. For class 1, the precision is 0.90, but the recall is lower at 0.74, meaning only 74% of the actual class 1 instances were predicted correctly. The F1 scores, which balance precision and recall, are 0.76 for class 0 and 0.81 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.79, 0.80, and 0.78, indicating that the model performs similarly for both classes. The weighted averages, which account for class imbalance, also result in 0.81 for precision, 0.79 in recall, and 0.79 in F1-score. Overall, the Logistic Regression model has moderate performance, showing difficulty distinguishing between the two classes, particularly with a lower precision for class 0. This indicates room for improvement in the model's predictive capability.

Table 13 Result from Logistic Regression with SMOTE

	Precision	Recall	F1-score	Support
0	0.6744	0.8614	0.7565	101
1	0.8963	0.7423	0.8121	163
Accuracy			0.7879	264
Macro avg	0.7854	0.8019	0.7843	264
Weighted avg	0.8114	0.7879	0.7903	264

3.1.11 Decision Tree with SMOTE

The results of testing the decision tree model with data balanced with SMOTE obtained classification report results, which can be seen in Table 14. Performance summary for a decision tree model. This model achieved a high accuracy of approximately 0.981. The table includes the performance metrics such as 'precision', 'recall', 'f1-score', and 'support' for two classes, labeled '0' and '1'. For Class 0, the precision is 0.98, recall is 0.97, and the f1-score is 0.98. Class 1 shows a precision of 0.98, a recall of 0.99, and an f1-score of 0.99. The model performs exceptionally with high-performance metrics for both macro and weighted average calculation.

Table 14 Result Decision Tree with SMOTE

	Precision	Recall	F1-score	Support
0	0.9800	0.9703	0.9751	101
1	0.9817	0.9877	0.9847	163
Accuracy			0.9811	264
Macro avg	0.9809	0.9790	0.9799	264
Weighted avg	0.9811	0.9811	0.9810	264

3.1.12 Extreme Gradient Boosting with SMOTE

The results of testing the extreme gradient boosting model with data balanced with SMOTE obtained classification report results, which can be seen in Table 15. Performance metrics for an XGBoost model, which has achieved an impressive accuracy of approximately 0.985. The metrics detailed include 'precision', 'recall', and 'f1-score' for two classes, labeled as '0' and '1'. Both classes show outstanding performance with a precision, recall, and f1-score of around 0.98 - 0.99. This summary indicates a highly effective model performance across all evaluated categories.

Table 15 Result XGBoost With SMOTE

	Precision	Recall	F1-score	Support
0	0.9848	0.9802	0.9802	101
1	0.9877	0.9877	0.9877	163
Accuracy			0.9848	264
Macro avg	0.9840	0.9840	0.9840	264
Weighted avg	0.9848	0.9848	0.9848	264

3.2 Discussion

The discussion will be a comparison between research models that have been trained using data that has not been balanced with SMOTE and after being balanced with SMOTE. Then, the best model is used as a proposed model. The proposed model will be compared with the previous research model. A comparison of research models can be seen in Table 16. Based on the comparison table of each research model, the XGBoost with the SMOTE model has very good results. The accuracy obtained reaches 98.48% with Precision, Recall, and F1-Score also around 98%. It can be ascertained that the XGBoost research model is better than other research models. So, it can be said that the XGBoost with SMOTE model is the proposed model. Another thing to point out is SMOTE can increase performance of model, with increased performance in models like Naïve Bayes, Decision Tree, and XGBoost. However, model KNN, Logistic Regression and SVM saw no increase in performance.

Table 16 Comparison Result

	Without SMOTE				With SMOTE			
Proposed Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	78.79	81.80	82.63	78.77	79.55	82.23	81.64	78.00
KNN	67.05	64.90	64.46	64.63	65.15	64.42	65.19	64.33
Logistic Regression	79.92	79.03	77.90	78.36	78.79	78.54	80.19	78.43
SVM	79.55	78.33	78.54	78.43	78.03	80.67	81.64	78.00
Decision Tree	97.73	97.59	97.59	97.59	98.11	98.09	97.90	97.99
XGBoost	98.11	98.09	97.90	97.99	98.48	98.40	98.40	98.40

Then, the proposed model will be compared with models from previous research. The comparison table of the proposed model with previous research models can be seen in Table 17. The SMOTE technique on XGBoost improves the performance compared to XGBoost itself and surpasses the more complex or simpler methods used by other researchers in the table. This confirms the importance of a good approach in preparing data and choosing the right algorithm for a particular type of data.

Table 17 Comparison with Previous Research

Author	Model Algorithm	Result
El-Sofany (2024)	XGBoost with sampling SMOTE	97.57%
J. P. Li et al. (2020)	FCMIM-SVM	92.37%
Anshori & Haris (2022)	Logistic Regression	81.3%
Proposed Method	XGBoost+SMOTE	98.48%

4. CONCLUSIONS

The results of this study demonstrate that the proposed classification model for heart disease, which integrates the Extreme Gradient Boosting (XGBoost) algorithm with Synthetic Minority Over-sampling Technique (SMOTE), yields superior performance compared to other machine learning models tested. The model achieved a classification accuracy of 98.48%, with precision, recall, and F1-score values consistently above 98%, indicating a high level of reliability and generalizability. These results substantiate the effectiveness of combining advanced ensemble learning with appropriate resampling techniques in addressing class imbalance issues within medical datasets.

Furthermore, the comparative analysis reveals that the XGBoost-SMOTE model outperforms several other baseline classifiers, including Support Vector Machine, Naive Bayes, Logistic Regression, K-Nearest Neighbors, and Decision Tree, both in pre- and post-resampling conditions. The findings also highlight that while SMOTE positively impacts model performance

across most algorithms, its integration with XGBoost delivers the most substantial improvement, thus reinforcing its suitability for the classification of complex, imbalanced clinical data.

When compared to models from prior research, the proposed model exhibits an enhancement in classification performance, surpassing the highest previously reported accuracy of 97.57%. This underscores the significance of meticulous model selection and data preprocessing strategies in developing predictive tools for clinical decision support. Given its empirical robustness and superior accuracy, the XGBoost-SMOTE model proposed in this study holds considerable potential for adoption in real-world diagnostic systems to support early and accurate detection of heart disease.

REFERENCES

Ammar, A., Bouattane, O., & Youssfi, M. (2021). Automatic Cardiac Cine MRI Segmentation and Heart Disease Classification. *Computerized Medical Imaging and Graphics*, *88*(2020), 101864. https://doi.org/10.1016/j.compmedimag.2021.101864

Anshori, M., & Haris, M. S. (2022). Predicting Heart Disease Using Logistic Regression. *Knowledge Engineering and Data Science*, *5*(2), 188. https://doi.org/10.17977/um018v5i22022p188-196

Ashtaiwi, A., Khalifa, T., & Alirr, O. (2024). Enhancing Heart Disease Diagnosis Through ECG Image Vectorization-Based Classification. *Heliyon*, *10*(18), e37574. https://doi.org/10.1016/j.heliyon.2024.e37574

Baccouche, A., Garcia-Zapirain, B., Castillo Olea, C., & Elmaghraby, A. (2020). Ensemble Deep Learning Models for Heart Disease Classification: A Case Study from Mexico. *Information*, *11*(4), 207. https://doi.org/10.3390/info11040207

Bengesi, S., Oladunni, T., Olusegun, R., & Audu, H. (2023). A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion from Twitter Tweets. *IEEE Access*, *11*, 11811–11826. https://doi.org/10.1109/ACCESS.2023.3242290

Benhar, H., Idri, A., & Fernández-Alemán, J. L. (2020). Data Preprocessing for Heart Disease Classification: A Systematic Literature Review. *Computer Methods and Programs in Biomedicine*, *195*, 105635. https://doi.org/10.1016/j.cmpb.2020.105635

Chen, L., Ji, P., & Ma, Y. (2022). Machine Learning Model for Hepatitis C Diagnosis Customized to Each Patient. *IEEE Access*, *10*(10), 106655–106672. https://doi.org/10.1109/ACCESS.2022.3210347

El-Sofany, H. F. (2024). Predicting Heart Diseases Using Machine Learning and Different Data Classification Techniques. *IEEE Access*, *12*(10), 106146–106160. https://doi.org/10.1109/ACCESS.2024.3437181

Gárate-Escamila, A. K., Hassani, A. H. El, & Andrès, E. (2020). Classification Models for Heart Disease Prediction Using Feature Selection and PCA. *Informatics in Medicine Unlocked*, *19*, 100330. https://doi.org/10.1016/j.imu.2020.100330

Gibson, S., Issac, B., Zhang, L., & Jacob, S. M. (2020). Detecting Spam Email with Machine Learning Optimized with Bio-Inspired Metaheuristic Algorithms. *IEEE Access*, *8*, 187914–187932. https://doi.org/10.1109/ACCESS.2020.3030751

Haznedar, B., & Simsek, N. Y. (2022). A Comparative Study on Classification Methods for Renal Cell and Lung Cancers Using RNA-Seq Data. *IEEE Access*, *10*(10), 105412–105420. https://doi.org/10.1109/ACCESS.2022.3211505

Hossain, Md. I., Maruf, M. H., Khan, Md. A. R., Prity, F. S., Fatema, S., Ejaz, Md. S., & Khan, Md. A. S. (2023). Heart Disease Prediction Using Distinct Artificial Intelligence Techniques: Performance Analysis and Comparison. *Iran Journal of Computer Science*, *6*(4), 397–417. https://doi.org/10.1007/s42044-023-00148-7

Huang, Z., & Chen, D. (2022). A Breast Cancer Diagnosis Method Based on VIM Feature Selection and Hierarchical Clustering Random Forest Algorithm. *IEEE Access*, *10*, 3284–3293. https://doi.org/10.1109/ACCESS.2021.3139595

Islam, N., Fatema-Tuj-Jahra, M., Hasan, Md. T., & Farid, D. Md. (2023). KNNTree: A New Method to Ameliorate K-Nearest Neighbour Classification Using Decision Tree. *2023 International*

Conference on Electrical, Computer and Communication Engineering (ECCE), 1–6. https://doi.org/10.1109/ECCE57851.2023.10101569

jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2013). Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. *Procedia Technology*, *10*, 85–94. https://doi.org/10.1016/j.protcy.2013.12.340

Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare. *IEEE Access*, *8*(2), 107562–107582. https://doi.org/10.1109/ACCESS.2020.3001149

Li, M., Ma, X., Chen, C., Yuan, Y., Zhang, S., Yan, Z., Chen, C., Chen, F., Bai, Y., Zhou, P., Lv, X., & Ma, M. (2021). Research on the Auxiliary Classification and Diagnosis of Lung Cancer Subtypes Based on Histopathological Images. *IEEE Access*, *9*, 53687–53707. https://doi.org/10.1109/ACCESS.2021.3071057

Maity, A., Pathak, A., & Saha, G. (2023). Transfer Learning Based Heart Valve Disease Classification from Phonocardiogram Signal. *Biomedical Signal Processing and Control*, *85*(2022), 104805. https://doi.org/10.1016/j.bspc.2023.104805

Mamun, M., Farjana, A., Al Mamun, M., & Ahammed, M. S. (2022). Lung Cancer Prediction Model Using Ensemble Learning Techniques and a Systematic Review Analysis. *2022 IEEE World AI IoT Congress (AIIoT)*, *2022*, 187–193. https://doi.org/10.1109/AIIoT54504.2022.9817326

Manikandan, G., Pragadeesh, B., Manojkumar, V., Karthikeyan, A. L., Manikandan, R., & Gandomi, A. H. (2024). Classification Models Combined with Boruta Feature Selection for Heart Disease Prediction. *Informatics in Medicine Unlocked*, *44*(2023), 101442. https://doi.org/10.1016/j.imu.2023.101442

Matin Malakouti, S. (2023). Heart Disease Classification Based on ECG Using Machine Learning Models. *Biomedical Signal Processing and Control*, *84*(2022), 104796. https://doi.org/10.1016/j.bspc.2023.104796

Muslim, M. A., Nikmah, T. L., Pertiwi, D. A. A., Subhan, Jumanto, Dasril, Y., & Iswanto. (2023). New Model Combination Meta-Learner to Improve Accuracy Prediction P2P Lending with Stacking Ensemble Learning. *Intelligent Systems with Applications*, *18*(2022), 200204. https://doi.org/10.1016/j.iswa.2023.200204

Ningsih, M. R., Unjung, J., Farih, H. al, & Muslim, M. A. (2024). Classification Email Spam Using Naive Bayes Algorithm and Chi-Squared Feature Selection. *Journal of Applied Intelligent System*, *9*(1), 74–87. https://doi.org/10.33633/JAIS.V9I1.9695

Obiedat, R., Qaddoura, R., Al-Zoubi, A. M., Al-Qaisi, L., Harfoushi, O., Alrefai, M., & Faris, H. (2022). Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution. *IEEE Access*, *10*, 22260–22273. https://doi.org/10.1109/ACCESS.2022.3149482

Oh, H. (2021). A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model. *IEEE Access*, *9*, 144121–144128. https://doi.org/10.1109/ACCESS.2021.3121508

Pan, Y., Fu, M., Cheng, B., Tao, X., & Guo, J. (2020). Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform. *IEEE Access*, *8*, 189503–189512. https://doi.org/10.1109/ACCESS.2020.3026214

Patidar, S., Kumar, D., & Rukwal, D. (2022). Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction. In *Advanced Production and Industrial Engineering* (pp. 64–69). https://doi.org/10.3233/ATDE220723

Radhika, R., & Thomas George, S. (2021). Heart Disease Classification Using Machine Learning Techniques. *Journal of Physics: Conference Series*, *1937*(1), 012047. https://doi.org/10.1088/1742-6596/1937/1/012047

Rofik, R., Hakim, R. A., Unjung, J., Prasetiyo, B., & Muslim, M. A. (2024). Optimization of SVM and Gradient Boosting Models Using GridSearchCV in Detecting Fake Job Postings. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *23*(2), 419–430. https://doi.org/10.30812/matrik.v23i2.3566

Maghdid, S. S., & Rashid, T. A. (2022). *An Extensive Dataset for the Heart Disease Classification System*. *2*. https://doi.org/10.17632/65GXGY2NMG.2

Sridhar, S., & Sanagavarapu, S. (2021). Handling Data Imbalance in Predictive Maintenance for Machines Using SMOTE-Based Oversampling. *2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*, 44–49. https://doi.org/10.1109/CICN51697.2021.9574668

Subathra, R., & Sumathy, V. (2024). An Offbeat Bolstered Swarm Integrated Ensemble Learning (BSEL) Model for Heart Disease Diagnosis and Classification. *Applied Soft Computing*, *154*(2023), 111273. https://doi.org/10.1016/j.asoc.2024.111273

Wazrah, A. Al, & Alhumoud, S. (2021). Sentiment Analysis Using Stacked Gated Recurrent Unit for Arabic Tweets. *IEEE Access*, *9*, 137176–137187. https://doi.org/10.1109/ACCESS.2021.3114313

Xu, W., Yu, K., Ye, J., Li, H., Chen, J., Yin, F., Xu, J., Zhu, J., Li, D., & Shu, Q. (2022). Automatic Pediatric Congenital Heart Disease Classification Based on Heart Sound Signal. *Artificial Intelligence in Medicine*, *126*(2021), 102257. https://doi.org/10.1016/j.artmed.2022.102257

Zhang, D., & Gong, Y. (2020). The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure. *IEEE Access*, *8*, 220990–221003. https://doi.org/10.1109/ACCESS.2020.3042848