Evaluation of Data Mining in Heart Failure Disease Classification

Nurfadlan Afiatuddin[1], Rahmaddeni*[2], Fitri Pratiwi[3], Rapindra Septia[4], Heri Hendrawan[5]
[1,2,4,5]Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia
[3]Universitas Dumai, Dumai, Indonesia
Email : [1]nurfadlan.afiatuddin096@gmail.com, *[2]**rahmaddeni@sar.ac.id**,
[3]fitrimarten@gmail.com, [4]rapindraseptia@gmail.com, [5]herihendrawan86@gmail.com

Abstract

This study evaluates the effectiveness of data mining algorithms in heart failure disease classification. Various algorithms, including Random Forest, Decision Tree C4.5, Gradient Boosted Machine (GBM), and XGBoost, were applied to a heart failure dataset. The dataset was collected from multiple sources and preprocessed to address imbalances using the SMOTE (Synthetic Minority Over-sampling Technique) technique. The results indicate that employing SMOTE and parameter optimization through grid search significantly enhances the performance of these algorithms. XGBoost and GBM demonstrated superior accuracy, precision, and recall in both balanced and imbalanced data scenarios. In balanced data scenarios, XGBoost achieved an accuracy of 98.75% with an error rate of 1.25%, while GBM achieved an accuracy of 98.60% with an error rate of 1.40%. The study confirms that appropriate data preprocessing and parameter optimization are crucial for improving the accuracy of medical data analysis. These findings suggest that XGBoost and GBM are highly effective for heart disease prediction, supporting early diagnosis and timely medical intervention. Future research should explore alternative preprocessing techniques and additional algorithms to further improve prediction outcomes.

Keywords— *Heart failure, Data mining, SMOTE, XGBoost, Prediction accuracy*

1. INTRODUCTION

Cardiovascular disease (CVD) is the leading cause of death globally, claiming an estimated 17.9 million lives each year. Specifically, CVD encompasses a wide range of heart and blood vessel disorders, including coronary heart disease, cerebrovascular disease, and rheumatic heart disease. Notably, more than four out of five deaths from CVD are caused by heart attacks and strokes, and one-third of those deaths occur prematurely in people under the age of 70[1].

Moreover, heart disease is a common condition that can have a serious impact on health. Various factors, such as age, gender, and blood pressure, have been linked to the risk of this disease [2]. Additionally, unhealthy lifestyles, including poor eating habits and irregular diets, are significant risk factors[3].

In recent years, artificial intelligence (AI) and machine learning (ML) based approaches have proven effective in decision-making and prediction using large medical datasets [4]. Researchers have developed expert systems to improve the diagnostic process of heart disease through early detection[5]. Early detection is critical in facilitating appropriate lifestyle changes and effective medical management[6][7].

Despite advances in heart disease prediction, criticisms regarding the accuracy of existing techniques persist[8]. However, the development of AI and data mining techniques in the healthcare industry has enhanced the evaluation of complex medical data and more accurate identification of heart disease[9]. Data mining is utilized to extract important information from unstructured data, making it effective in medical data analysis for efficient disease

prediction[10]. Therefore, early detection of heart disease is a priority in healthcare, especially in cardiology practice[11]. The World Health Organization has identified CVD as one of the leading causes of global mortality [12].

Prediction and early diagnosis of heart disease present important challenges in clinical data analysis, supporting timely prevention and treatment improvement efforts[13]. Advances in computational intelligence enable the development of pattern recognition systems that can identify hidden health information[14]. Detecting cardiovascular disease symptoms as early as possible is a difficult but essential task, given its global impact on mortality rates[15]. Nonetheless, machine learning and data mining-based approaches promise significant clinical benefits in predicting and detecting heart disease, despite their complex challenges[16].

Based on this observation, a study titled 'A Heart Disease Prediction Model Based on Feature Optimization and Smote-Xgboost Algorithm' by Jian Yang and Jinhan Guan (2022) used the Heart Disease Dataset and XGBoost algorithm. By applying the SMOTE technique to handle data imbalance, the model achieved 85.95% accuracy on training data and 91.80% on test data, showing great potential in effectively improving heart disease prediction[17]. Furthermore, a study entitled 'Development of an Explainable Prediction Model of Heart Failure Survival by Using Ensemble Trees' by Pedro A. Moreno-Sanchez in 2020 utilized a dataset published by Ahmad et al. and available at the UCI Machine Learning repository. In this study, the XGBoost algorithm was employed, and the results showed that XGBoost had the highest accuracy of 83% on new data compared to other ensemble tree methods[18].

Meanwhile, a study titled 'Effective Heart Disease Prediction Using Machine Learning Techniques' by Chintan M. Bhatt, Parth Patel, Tarang Ghetia, and Pier Luigi Mazzeo in 2023 used the Cleveland heart disease dataset. Various algorithms, such as Random Forest, Decision Tree, Multilayer Perceptron, and XGBoost, were used for heart disease prediction. The results indicated that the Multilayer Perceptron model with cross-validation achieved the highest accuracy of 87.28%, outperforming the other algorithms[19]. In addition, a study entitled 'Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization' by Daniyal Asif, Mairaj Bibi, Muhammad Shoaib Arif, and Aiman Mukheimer in 2023 employed the Extra Tree Classifier, XGBoost, and CatBoost algorithms. By applying Grid Search Cross-Validation and Randomized Search Cross-Validation techniques for hyperparameter optimization and data normalization, the proposed model achieved an accuracy of 98.15% [20].

Lastly, a study titled 'An Optimized XGBoost Based Diagnostic System for Effective Prediction of Heart Disease' by Kartik Budholiya, Shailendra Kumar Shrivastava, and Vivek Sharma in 2022 used the Cleveland heart disease dataset from the University of California, Irvine (UCI) online machine learning and data mining repository. The algorithm employed was XGBoost, with Bayesian Optimization techniques for tuning XGBoost hyperparameters and One-Hot (OH) encoding to process categorical features[21].

In this study, we posit that among the compared algorithms (Random Forest, Decision Tree C4.5, GBM, and XGBoost), one will demonstrate superior performance in predicting heart disease compared to the others. Despite numerous studies utilizing data mining algorithms for heart disease diagnosis, several unresolved issues persist. One major concern is the overall lack of algorithm parameter optimization, which significantly impacts classification accuracy. Previous studies often neglect data preprocessing techniques, such as effectively handling data imbalance using SMOTE. To address these issues, our research will concentrate on employing grid search to optimize algorithm parameters, aiming to enhance accuracy and classification[22] [23].

Moreover, there is a need to compare the performance of various algorithms more comprehensively in the context of heart disease prediction. Many previous studies have only focused on one or two algorithms, thus not providing a complete picture. Therefore, this study will expand its scope by comparing the performance of four different algorithms (Random Forest, Decision Tree C4.5, GBM, and XGBoost) under similar conditions. With this approach, it is expected to find the most suitable algorithm to improve the accuracy and reliability of heart

disease prediction. Furthermore, this research method introduces innovation by employing various data-sharing strategies for model training and testing. By evaluating these strategies with data splits of 60:40, 70:30, 80:20, and 90:10, our study aims to assess the algorithm's reliability in validating the model across diverse dataset scenarios. It is crucial to ensure that the developed model not only performs well on training data but also effectively generalizes the acquired information to new datasets, thereby enhancing the reliability of prediction outcomes in practical applications [24] [25].

2. RESEARCH METHODS

The methodology described using RapidMiner version 10.3 begins with data collection, followed by preprocessing to clean and prepare the data for analysis. The data is then divided into training and testing sets, and an appropriate machine-learning algorithm is selected.

To address data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. This step is followed by hyperparameter optimization to refine the model parameters for optimal performance. The model is then trained with balanced and optimized data. Finally, the model is evaluated using the testing set to assess its performance with appropriate metrics for the specific problem type. RapidMiner's robust suite of tools facilitates each step with an efficient and effective workflow through its intuitive drag-and-drop interface. To understand how to use data for accurate decision-making, follow these steps:

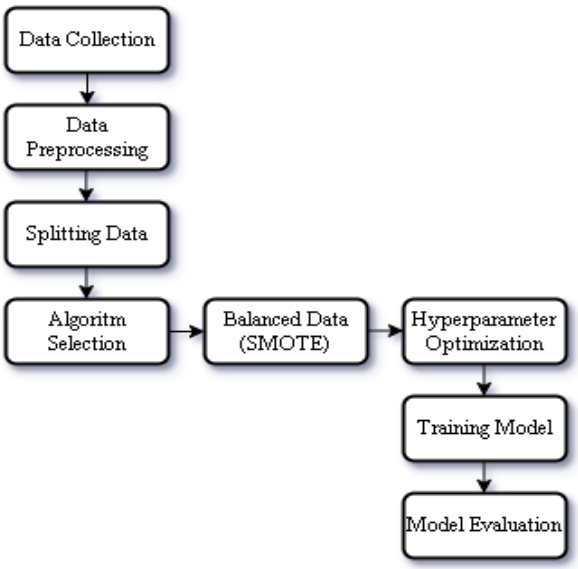

Figure 1. Research Methodology

2.1. Data Collection

The first step in our research is to collect the data. We used the "Heart Failure Prediction Dataset" from Kaggle. This dataset combines data from five sources: Cleveland, Hungarian, Switzerland, Long Beach VA, and Stalog. After removing duplicates, the dataset consists of 918 observations with 11 features used to predict heart disease [26].

2.2. Data Preprocessing

Before modeling, it is essential to ensure the data is properly prepared:

a) Handling Missing Values: Missing values can be addressed by removing affected rows or columns, imputing with the mean, median, or mode, or using predictive models to estimate missing values.

b) Removing Duplicate Data: Ensuring the uniqueness of entries is achieved by identifying and removing duplicate rows from the dataset.
c) Handling Outliers: Outliers, which can distort analysis, are identified using methods such as Z-scores, the Interquartile Range (IQR), and box plots. They can be managed by removal, transformation, imputation, or separate analysis to enhance the accuracy and reliability of the dataset.
d) Filter Examples: After detecting outliers, data examples identified as outliers can be filtered or removed to ensure only relevant data is used for training the model.
e) Select Attributes: Relevant attributes or features are selected from the dataset. All available attributes are used to ensure a comprehensive analysis and maintain data integrity in modeling.
f) Set Role: The role of each attribute in the dataset is defined. For example, the heart disease column is set as the target label, and the other columns are set as features or inputs for the model.

2.3. Algorithm Selection

This research utilizes several data mining algorithms, including Random Forest, Decision Tree C4.5, GBM, and XGBoost, to analyze the data. The analysis is conducted using the RapidMiner tool to facilitate efficient data processing.

2.4. Splitting Data For Evaluation

After cleaning and preparing the data, we split it into training and testing sets to ensure reliable model training and evaluation. Common split ratios include 60:40, 70:30, 80:20, and 90:10. These ratios balance the need for sufficient training data to learn complex patterns and enough testing data to accurately evaluate the model's performance, thereby enhancing its reliability and accuracy.

2.5. SMOTE (Synthetic Minority Over-sampling Technique)

The SMOTE (Synthetic Minority Over-sampling Technique) technique is used to address class imbalance in a dataset. SMOTE creates new data samples that are similar to the minority class, making the dataset more balanced.

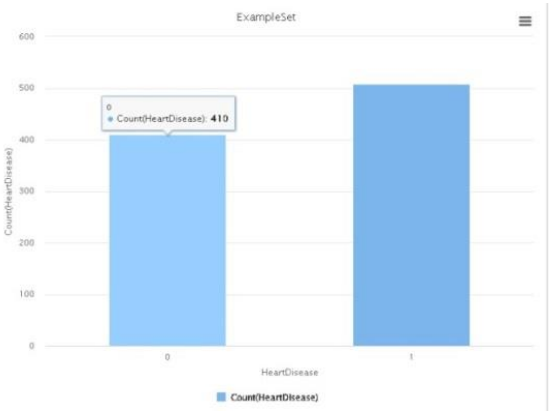

Figure 2. Heart Disease No SMOTE

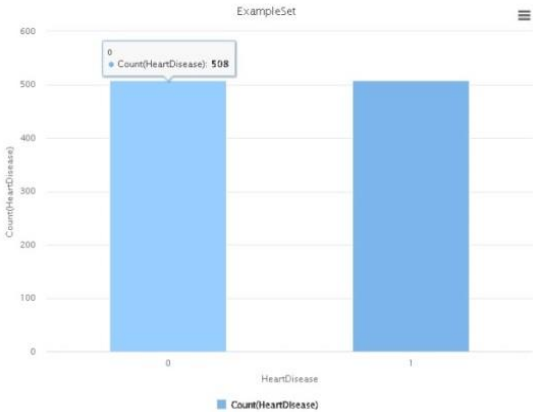

Figure 3. Heart Disease SMOTE

Figures 2 and 3 illustrate the application of the SMOTE technique to the heart disease dataset. Figure 2 shows the dataset before applying SMOTE, with 410 samples for class 0 and 508 samples for class 1. Figure 3 shows the dataset after applying SMOTE, with 508 samples for both classes, achieving balance. Thus, SMOTE effectively addresses class imbalance, ensuring equal representation of classes, which is crucial for improving the performance of machine learning models.

2.6. Hyperparameter Optimization

Hyperparameter optimization involves finding the optimal values for model parameters, such as the learning rate or the number of neurons, before training. Grid Search is a method used to identify the best combination by evaluating all possible options. The following is an explanation of the hyperparameter optimization settings:

***2.6.1** Random Forest*

To optimize the hyperparameters for the Random Forest algorithm in RapidMiner, each set is carefully configured to enhance performance. The number of trees is set to 100, forming a robust ensemble that improves accuracy and reduces overfitting. The maximal depth is limited to 10, preventing overly complex trees and maintaining generalization. Pruning is applied to simplify trees by removing insignificant branches, enhancing interpretability, and reducing overfitting. The minimal leaf size is set to 2, ensuring decisions are based on at least two data points, preventing overfitting. The minimal size for a split is also set to 2, ensuring splits occur only with sufficient data points, balancing bias and variance. These settings are designed to achieve an optimal balance between complexity and generalization, resulting in accurate and reliable predictions.

***2.6.2** Decision Tree C4.5*

To optimize the Decision Tree C4.5 decision tree algorithm, key hyperparameters are carefully configured. The gain ratio is used as the splitting criterion to handle attributes with many values effectively. The maximal depth is set to 10 to prevent overfitting and maintain generalization. Pruning is applied (apply pruning = true) to simplify the tree and enhance interpretability. The minimal leaf size is set to 2, ensuring each leaf node contains at least two data points to avoid overly specific rules. Similarly, the minimal size for a split is set to 2 to balance bias and variance. These settings aim to create an optimal decision tree that balances complexity and generalization, resulting in accurate and reliable predictions.

***2.6.3** Gradient Boosted Machine*

To optimize the Gradient Boosted Machine algorithm, specific hyperparameters are carefully configured. The number of trees is set to 100, ensuring a robust ensemble of decision trees that collectively improve predictive accuracy. The maximal depth is limited to 10, preventing individual trees from becoming too complex and thus maintaining the model's generalization ability. The learning rate is set to 0.01, controlling the contribution of each tree and ensuring the model learns slowly to avoid overfitting. The sample rate is set to 1.0, meaning that each tree is trained using the entire dataset, which helps in achieving accurate predictions. The distribution is set to auto, allowing the algorithm to automatically select the appropriate loss function based on the nature of the prediction task. The min rows parameter is set to 1, ensuring that each node must have at least 1 data point as the minimum sum of instance weights, which helps control overfitting by preventing nodes from being too small and overly specific. These hyperparameter settings are designed to achieve an optimal balance between model complexity and generalization, resulting in accurate and reliable predictions.

***2.6.4** Extreme Gradient Boosting*

To optimize the Extreme Gradient Boosting algorithm, specific hyperparameters are meticulously configured. The booster is set to 'trees', indicating that the model will use tree-based boosting. The number of boosting rounds is set to 100, meaning the model will undergo 100 iterations to build an ensemble of trees, enhancing predictive accuracy. The learning rate is set to 0.01, which controls the contribution of each tree, ensuring gradual learning to prevent overfitting. The tree method is set to 'tree', specifying that the algorithm will use decision trees for boosting. The sub-sample rate is set to 1.0, indicating that each tree is trained using the entire dataset, which helps in achieving accurate predictions. The min child weight is set to 0.5,

which helps control overfitting by requiring a minimum sum of instance weights in each leaf node, ensuring that the model does not learn overly specific patterns. These hyperparameter settings are designed to create an optimal XGBoost model that balances complexity and generalization, resulting in reliable and precise predictions.

2.7. Training Model

This design outlines the process for training models using RapidMiner, employing four distinct machine-learning algorithms: Random Forest, Decision Tree C4.5, GBM, and XGBoost. The accompanying image illustrates the workflow from data input to model evaluation, with each algorithm represented in a separate module. This process encompasses stages such as data preprocessing, data splitting into training and testing sets, model training with each algorithm, and final model evaluation to determine effectiveness.

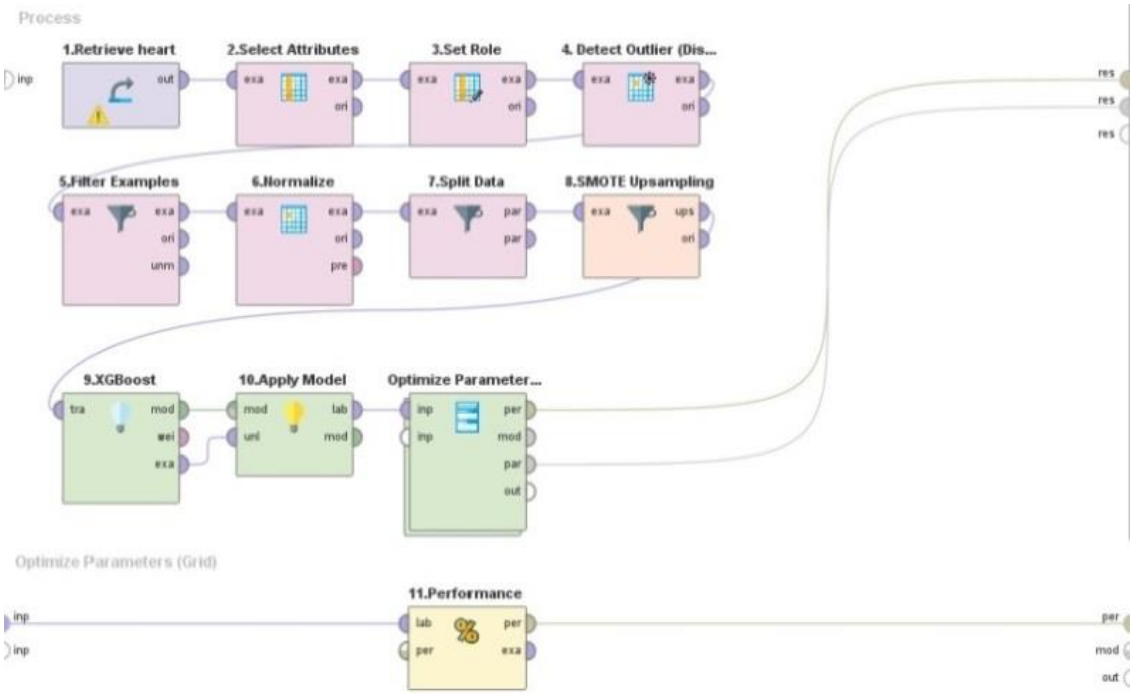

Figure 4. XGBoost with SMOTE and Grid Search

The process for analyzing heart disease data involves several key steps: retrieving the dataset, selecting relevant attributes, defining the role of each attribute, detecting outliers using a distance-based method, filtering outliers, normalizing the data, splitting the data into training and testing sets, applying SMOTE upsampling to address the class imbalance, training models such as Random Forest, Decision Tree C4.5, GBM, and XGBoost for predictions, optimizing model parameters through methods like grid search (e.g., tuning the number of trees, maximum depth, learning rate, and minimum samples for splits), and evaluating the model's performance using metrics such as accuracy, precision, recall, and error rate.

2.8. Model Evaluation (Confusion Matrix)

This design outlines the process for training models using RapidMiner, employing four distinct machine-learning algorithms: Random Forest, Decision Tree C4.5, GBM, and XGBoost. The accompanying image illustrates the workflow from data input to model evaluation, with each algorithm represented in a separate module. This process encompasses stages such as data preprocessing, data splitting into training and testing sets, model training with each algorithm, and final model evaluation to determine effectiveness.

A confusion matrix is a table utilized to evaluate the performance of a classification model. This table allows us to observe how well the model's predictions align with the actual values. The confusion matrix consists of four primary components:

1) True Positive (TP): The number of correct positive predictions, where the model predicts the positive class and the actual value is positive.
2) True Negative (TN): The number of correct negative predictions, where the model predicts the negative class and the actual value is negative.
3) False Positive (FP): The number of incorrect positive predictions, where the model predicts the positive class, but the actual value is negative. This is also referred to as a "Type I error."
4) False Negative (FN): The number of incorrect negative predictions, where the model predicts the negative class, but the actual value is positive. This is also referred to as a "Type II error.

Here's a visual example of a confusion matrix:

Table1. Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FP
Actual Negative	FN	TN

From the confusion matrix, we can calculate various important evaluation metrics such as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Explanation: Accuracy measures the proportion of correct predictions (both true positives and true negatives) made by the model out of all predictions made. It is a general indicator of a model's performance.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Explanation: Precision measures the accuracy of positive predictions. It indicates the proportion of positive identifications that were correct. Precision is particularly useful when the cost of a false positive is high.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Explanation: Recall measures how good the model is at finding all true positive instances. It is the ratio between the number of correct positive predictions (True Positives) to the total number of actual positive data (True Positives + False Negatives).

$$Error\ Rate = \frac{FP+FN}{TP+TN+FP+FN} \quad (4)$$

Explanation: The error rate measures the proportion of incorrect predictions made by the model. It complements accuracy by showing the rate at which the model makes errors, thus providing insight into the instances it fails to predict correctly.

3. RESULT AND DISCUSSION

This study investigates the efficacy of various machine learning techniques, combined with the application of SMOTE (Synthetic Minority Over-sampling Technique) and Grid Search for hyperparameter tuning, in addressing data imbalance and enhancing model performance. The following sections present detailed results for different algorithms and data-splitting ratios, highlighting the significant improvements achieved through these methods.

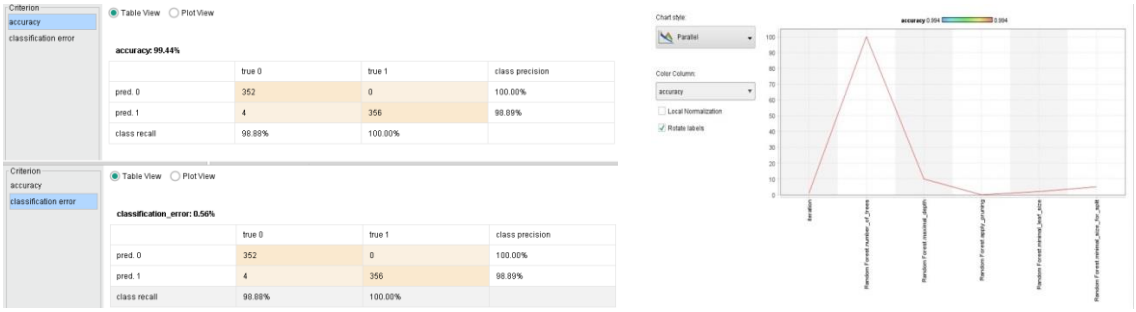

Figure 5 and 6. Random Forest 70:30 with SMOTE and Grid Search

The 70:30 ratio for the Random Forest model yielded the best results, with an accuracy of 99.44%, precision of 98.89%, recall of 100%, and an error rate of 0.56%. The application of SMOTE (Synthetic Minority Over-sampling Technique) effectively addresses data imbalance by generating synthetic data for the minority class, which ultimately enhances the overall performance of the model.

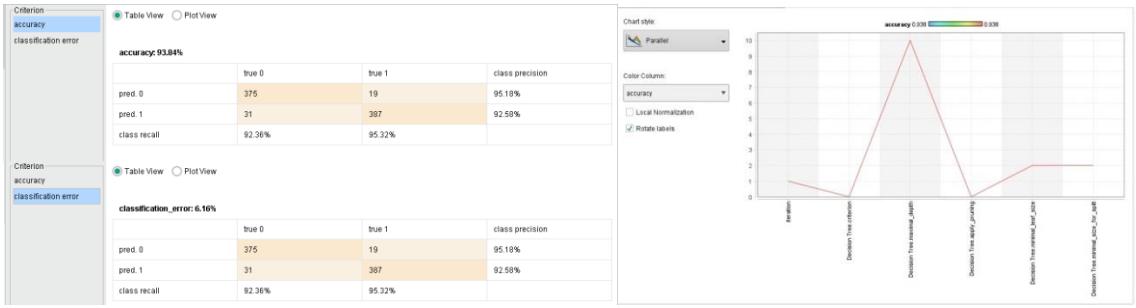

Figure 7 and 8. Decision Tree C4.5 80:20 with SMOTE and Grid Search

The 80:20 ratio for the Decision Tree C4.5 model yielded results, with an accuracy of 93.84%, precision of 92.58%, recall of 95.32%, and an error rate of 6.16%. The application of SMOTE (Synthetic Minority Over-sampling Technique) effectively addresses data imbalance by generating synthetic data for the minority class, which ultimately enhances the overall performance of the model.

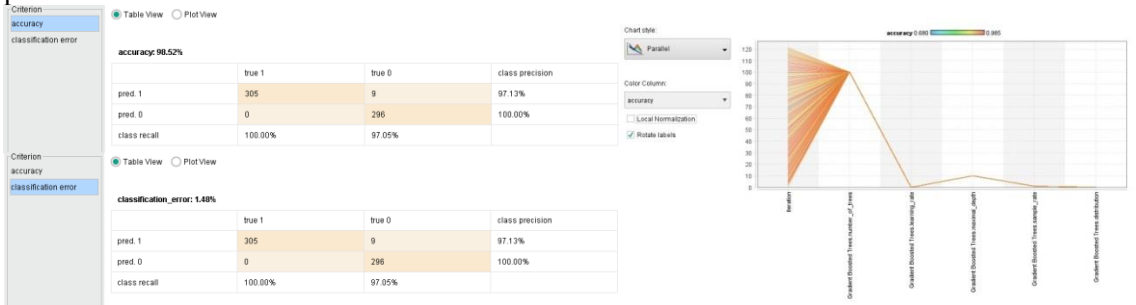

Figure 9 and 10. Gradient Boosted Machine 60:40 with SMOTE and Grid Search

The 60:40 ratio for the Gradient Boosted Machine (GBM) model yielded results, with an accuracy of 98.52%, precision of 97.13%, recall of 100%, and an error rate of 1.48%. The

application of SMOTE (Synthetic Minority Over-sampling Technique) effectively addresses data imbalance by generating synthetic data for the minority class, ultimately enhancing the overall performance of the model.

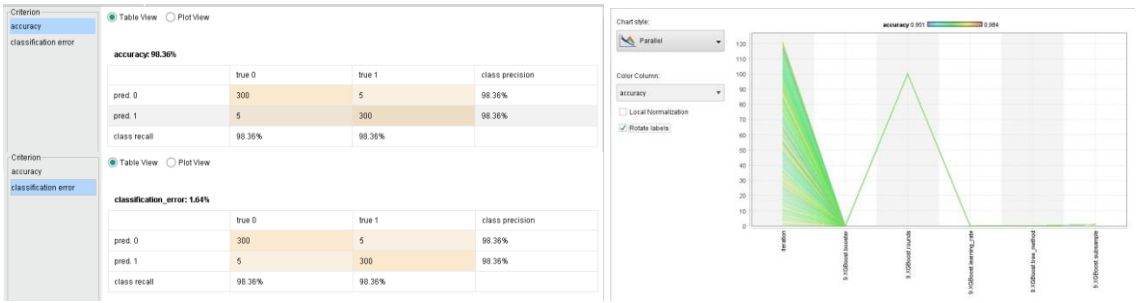

Figure 11 and 12. Extreme Gradient Boosting 60:40 with SMOTE and Grid Search

The 60:40 ratio for the XGBoost model yielded results, with an accuracy of 98.36%, precision of 98.36%, recall of 98.36%, and an error rate of 1.64%. The application of SMOTE (Synthetic Minority Over-sampling Technique) effectively addresses data imbalance by generating synthetic data for the minority class, thereby enhancing the overall performance of the model.

Table 2. Imbalance Class

Algoritma	Splitting Data						
	Training	*Testing*	*Class Label*	*Accuracy*	*Precision*	*Recall*	*Error Rate*
Random Forest	60%	40%	Imbalance	90,56%	89,66%	93,77%	9,94%
	70%	30%	Imbalance	90,82%	89,91%	94,10%	9,18%
	80%	20%	Imbalance	90,60%	89,46%	94,09%	9,40%
	90%	10%	Imbalance	90,19%	89,00%	93,87%	9,81%
Decision Tree C4.5	60%	40%	Imbalance	92,01%	93,65%	91,80%	7,99%
	70%	30%	Imbalance	92,07%	92,72%	92,98%	7,93%
	80%	20%	Imbalance	92,37%	91,47%	95,07%	7,63%
	90%	10%	Imbalance	94,07%	92,86%	96,72%	5,93%
Gradient Boosted Machine	60%	40%	Imbalance	95,83%	92,99%	100%	4,17%
	70%	30%	Imbalance	95,33%	92,67%	99,44%	4,67%
	80%	20%	Imbalance	92,37%	88,89%	98,52%	7,63%
	90%	10%	Imbalance	95,16%	91,95%	100%	4,84%
XGBoost	60%	40%	Imbalance	96,37%	95,53%	98,03%	3,63%
	70%	30%	Imbalance	95,65%	94,81%	97,47%	4,35%
	80%	20%	Imbalance	96,73%	95,69%	98,52%	3,27%
	90%	10%	Imbalance	96,97%	95,96%	98,69%	3,03%

The results of this study demonstrate the performance of data mining algorithms based on different training and testing data splits, particularly in the context of imbalanced data. Specifically, for the Random Forest algorithm, accuracy varied from 90.19% to 90.82%, with error rates ranging from 9.18% to 9.94%. Precision and recall were also consistent, indicating stable predictive capabilities.

In contrast, the Decision Tree C4.5 algorithm showed an increase in accuracy with a higher proportion of training data, reaching 94.07% with a 90:10 split, and the lowest error rate of 5.93%. Moreover, precision and recall varied significantly, with the highest precision at 93.65% and recall at 96.72%.

Furthermore, the Gradient Boosted Machine (GBM) algorithm exhibited excellent performance, achieving an accuracy of 95.83% with a 60:40 split and 95.16% with a 90:10 split.

Additionally, precision and recall were exceptionally high, with recall reaching 100% in some splits, and the lowest error rate recorded at 4.17%.

Moreover, XGBoost demonstrated the best overall performance among all the algorithms, with the highest accuracy of 96.97% at a 90:10 split and the lowest error rate of 3.03%. Precision and recall were also very high, with precision reaching 95.96% and recall at 98.69%, making it a very robust algorithm for handling imbalanced data.

In summary, XGBoost showed superior performance compared to the other algorithms, followed closely by the Gradient Boosted Machine, which also performed exceptionally well, particularly in recall. Meanwhile, Decision Tree C4.5 and Random Forest also performed well, but not as well as XGBoost and GBM. Ultimately, the choice of the best algorithm depends on specific needs and use case contexts, but these results indicate that XGBoost and GBM are very strong choices for data with class imbalance.

Table 3. SMOTE Class & Grid Search

Algoritma	**Splitting Data**						
	Training	***Testing***	***Class Label***	***Accuracy***	***Precision***	***Recall***	***Error Rate***
Random Forest	60%	40%	SMOTE	98,20%	96,82%	99,67%	1,80%
	70%	30%	SMOTE	99,44%	98,89%	100%	0,56%
	80%	20%	SMOTE	98,77%	97,83%	99,75%	1,23%
	90%	10%	SMOTE	98,69%	97,85%	99,56%	1,31%
Decision Tree C4.5	60%	40%	SMOTE	92,46%	94,81%	95,08%	7,54%
	70%	30%	SMOTE	92,42%	92,66%	92,13%	7,58%
	80%	20%	SMOTE	93,84%	92,58%	95,32%	6,16%
	90%	10%	SMOTE	91,47%	89,23%	94,31%	8,53%
Gradient Boosted Machine	60%	40%	SMOTE	98,52%	97,13%	100%	1,48%
	70%	30%	SMOTE	98,46%	98,32%	98,60%	1,54%
	80%	20%	SMOTE	97,54%	96,62%	98,52%	2,46%
	90%	10%	SMOTE	97,37%	96,76%	98,03%	2,63%
XGBoost	60%	40%	SMOTE	98,36%	98,36%	98,36%	1,64%
	70%	30%	SMOTE	98,03%	97,50%	98,60%	1,97%
	80%	20%	SMOTE	97,66%	96,40%	99,01%	2,34%
	90%	10%	SMOTE	98,03%	97,41%	98,69%	1,97%

The results presented in Table 3 demonstrate the performance of various data mining algorithms after applying SMOTE (Synthetic Minority Over-sampling Technique) and grid search optimization, evaluated based on different training and testing data splits. The Random Forest algorithm shows consistently high accuracy, ranging from 98.20% to 99.44%, with both precision and recall being high. The recall reaches 100% with a 70:30 split, and the error rate is low, with the lowest at 0.56%. The Decision Tree C4.5 algorithm displays accuracy ranging from 91.47% to 93.84%, with relatively high but more variable precision and recall. The highest precision is 94.81% with a 60:40 split, and the error rate is higher compared to Random Forest, with the highest at 8.53% for the 90:10 split. The Gradient Boosted Machine (GBM) algorithm exhibits excellent performance, with accuracy between 97.37% and 98.52%. Both precision and recall are high, with recall reaching 100% for the 60:40 split, and the error rate consistently low, the lowest being 1.48%. XGBoost demonstrates the highest overall accuracy, ranging from 97.66% to 98.36%. Precision and recall are also very high, indicating robust performance in identifying positive instances, with a low error rate, the lowest at 1.64% for the 60:40 split.

The application of SMOTE and grid search optimization significantly improves the performance of all algorithms. XGBoost and GBM consistently show superior performance with high accuracy, precision, and recall, and low error rates. Random Forest also performs well but slightly lags behind XGBoost and GBM. While Decision Tree C4.5 performs adequately, it

shows more variability in its metrics and generally lower performance compared to the other algorithms. Overall, XGBoost and GBM are the top performers in handling imbalanced data with SMOTE, making them suitable choices for tasks requiring high accuracy and reliability in prediction.

The comparison between the two sets of results clearly shows that applying SMOTE and grid search optimization significantly enhances the performance of data mining algorithms in handling imbalanced data. The accuracy, precision, recall, and error rates of all algorithms improved substantially. Notably, XGBoost and Gradient Boosted Machine (GBM) algorithms exhibited superior performance in both scenarios, with XGBoost emerging as the most robust algorithm overall. These findings underscore the effectiveness of SMOTE and grid search optimization in improving model performance for imbalanced datasets.

4. CONCLUSION

This study acknowledges several limitations. Firstly, the dataset used may not fully represent the broader population, which could limit the generalizability of the results. Secondly, the SMOTE technique employed to address data imbalance has inherent limitations, as it may not fully replicate natural variations in minority data. Additionally, the algorithms and parameter optimization techniques applied in this study may not encompass all potential approaches that could yield better results.

Therefore, future research should explore alternative data preprocessing techniques, such as adaptive resampling or advanced generative models, and test a broader range of machine learning algorithms. Specifically, deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), could be investigated, as they have demonstrated significant potential in handling complex patterns in large medical datasets. Moreover, ensemble approaches that combine multiple models may further enhance prediction accuracy.

This research outlines the process of data generation and processing. Initially, data was collected from reputable sources, including the heart failure prediction dataset from Kaggle. The dataset was cleansed of duplicates and processed to address imbalance using the SMOTE technique, ensuring balanced class representation, which is essential for improving the performance of heart disease prediction models.

The findings of this study show that employing SMOTE and parameter optimization through grid search significantly improves the performance of data mining algorithms in handling imbalanced data. These results highlight the critical role of appropriate data preprocessing and parameter optimization in enhancing prediction accuracy for medical data analysis. Moreover, the findings align with other studies suggesting that effective optimization and preprocessing techniques can mitigate challenges in heart disease prediction.

The implications of this study are significant in both theory and application. Theoretically, the results confirm the importance of data preprocessing and parameter optimization in improving the accuracy of machine learning models. Practically, algorithms such as XGBoost and Gradient Boosted Machines with SMOTE can support the early detection of heart disease, aiding clinical decision-making and enabling timely medical intervention. Furthermore, this research paves the way for exploring and testing alternative algorithms, preprocessing techniques, and parameter tuning methods, including deep learning architectures like CNNs and RNNs, across various medical datasets. These future efforts could further enhance predictive performance, providing new insights for medical research and healthcare applications.

REFERENCES

[1] World Health Organization, "Cardiovascular diseases," 2021, [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

[2] S. Suangli, F. Fahmi, and E. M. Zamzami, "Performance Analysis of Support Vector Machine and Xgboost Classifier Algorithms in Predicting Data Heart Disease," in *2023 29th International Conference on Telecommunications (ICT)*, IEEE, Nov. 2023, pp. 1–6. doi: 10.1109/ICT60153.2023.10374048.

[3] V. Jain and M. Agrawal, "Heart Failure Prediction Using XGB Classifier, Logistic Regression and Support Vector Classifier," in *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, IEEE, May 2023, pp. 1–5. doi: 10.1109/InCACCT57535.2023.10141752.

[4] S. Parthasarathy, V. Jayaraman, and J. P. Princy R, "Predicting Heart Failure using SMOTE-ENN-XGBoost," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, IEEE, Jan. 2023, pp. 661–666. doi: 10.1109/IDCIoT56793.2023.10053458.

[5] S. Doki, S. Devella, S. Tallam, S. S. Reddy Gangannagari, P. Sampathkrishna Reddy, and G. P. Reddy, "Heart Disease Prediction Using XGBoost," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, IEEE, Aug. 2022, pp. 1317–1320. doi: 10.1109/ICICICT54557.2022.9917678.

[6] K. Karthick, S. K. Aruna, R. Samikannu, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, "Implementation of a Heart Disease Risk Prediction Model Using Machine Learning," *Comput Math Methods Med*, vol. 2022, pp. 1–14, May 2022, doi: 10.1155/2022/6517716.

[7] A. Tiwari, A. Chugh, and A. Sharma, "Ensemble framework for cardiovascular disease prediction," *Comput Biol Med*, vol. 146, p. 105624, Jul. 2022, doi: 10.1016/j.compbiomed.2022.105624.

[8] M. O. Butt, A. Ur Rehman, S. Javaid, T. M. Ali, and A. Nawaz, "An Application of Artificial Intelligence for an Early and Effective Prediction of Heart Failure," in *2022 Third International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/INTELLECT55495.2022.9969182.

[9] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, and M. A. Moni, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," *Comput Biol Med*, vol. 136, p. 104672, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104672.

[10] S. S. Panigrahi and N. Kaur, "Hybrid Classification Method for the Heart Disease Prediction," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, Dec. 2022, pp. 494–499. doi: 10.1109/ICAC3N56670.2022.10074324.

[11] U. Nagavelli, D. Samanta, and P. Chakraborty, "Machine Learning Technology-Based Heart Disease Detection Models," *J Healthc Eng*, vol. 2022, pp. 1–9, Feb. 2022, doi: 10.1155/2022/7351061.

[12] F. Tasnim and S. U. Habiba, “A Comparative Study on Heart Disease Prediction Using Data Mining Techniques and Feature Selection,” in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, IEEE, Jan. 2021, pp. 338–341. doi: 10.1109/ICREST51555.2021.9331158.

[13] H. H. Alalawi and M. S. Alsuwat, "Detection of cardiovascular disease using machine learning classification models," *Int. J. Eng. Res. Technol.*, vol. 10, no. 7, pp. 151-157, 2021.

[14] B. Martins, D. Ferreira, C. Neto, A. Abelha, and J. Machado, “Data Mining for Cardiovascular Disease Prediction,” *J Med Syst*, vol. 45, no. 1, p. 6, Jan. 2021, doi: 10.1007/s10916-020-01682-8.

[15] A. Alqahtani, S. Alsubai, M. Sha, L. Vilcekova, and T. Javed, “Cardiovascular Disease Detection using Ensemble Learning,” *Comput Intell Neurosci*, vol. 2022, pp. 1–9, Aug. 2022, doi: 10.1155/2022/5267498.

[16] K. M. Mohi Uddin, R. Ripa, N. Yeasmin, N. Biswas, and S. K. Dey, “Machine learning-based approach to the diagnosis of cardiovascular vascular disease using a combined dataset,” *Intell Based Med*, vol. 7, p. 100100, 2023, doi: 10.1016/j.ibmed.2023.100100.

[17] J. Yang and J. Guan, "A heart disease prediction model based on feature optimization and SMOTE-XGBoost algorithm," *Information*, vol. 13, no. 10, p. 475, 2022. [Online]. Available: https://doi.org/10.3390/info13100475.

[18] P. A. Moreno-Sanchez, “Development of an Explainable Prediction Model of Heart Failure Survival by Using Ensemble Trees,” *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, pp. 4902–4910, 2020, doi: 10.1109/BigData50022.2020.9378460.

[19] K. Shiwangi, J. K. Sandhu, and R. Sahu, “Effective Heart-Disease Prediction by Using Hybrid Machine Learning Technique,” in *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, IEEE, Aug. 2023, pp. 1670–1675. doi: 10.1109/ICCPCT58313.2023.10245785.

[20] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, “Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization,” *Algorithms*, vol. 16, no. 6, p. 308, Jun. 2023, doi: 10.3390/a16060308.

[21] K. Budholiya, S. K. Shrivastava, and V. Sharma, “An optimized XGBoost based diagnostic system for effective prediction of heart disease,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4514–4523, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.013.

[22] Narayanan and Jayashree, “Implementation of Efficient Machine Learning Techniques for Prediction of Cardiac Disease using SMOTE,” *Procedia Comput Sci*, vol. 233, pp. 558–569, 2024, doi: 10.1016/j.procs.2024.03.245.

[23] R. Valarmathi and T. Sheela, "Heart disease prediction using hyperparameter optimization (HPO) tuning," *Biomed. Signal Process. Control*, vol. 70, p. 103033, 2021. [Online]. Available: https://doi.org/10.1016/j.bspc.2021.103033.

[24] N. Afiatuddin, M. T. Wicaksono, V. R. Akbar, R. Rahmaddeni, and D. Wulandari, "Komparasi Algoritma Machine Learning dalam Klasifikasi Kanker Payudara," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 2, p. 889, Apr. 2024, doi: 10.30865/mib.v8i2.7457.

[25] A. Nugroho, "Analisa Splitting Criteria Pada Decision Tree dan Random Forest untuk Klasifikasi Evaluasi Kendaraan," *JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer*, vol. 1, no. 1, pp. 41–49, Dec. 2022, doi: 10.53624/jsitik.v1i1.154.

[26] Fedesoriano, "Heart Failure Prediction Dataset," 2021, [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction