Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

Arrhythmia Detection Using XGBoost with Recursive Feature Elimination: A Two-Stage Machine Learning Approach

Suci Mutiarani*, Tikaridha Hardiani

Science and Technology Faculty, Technology Information, Universitas Aisyiyyah Yogyakarta, Yogyakarta, Indonesia
Email: [1,*]mutiaranisuci58@gmail.com, [2,]tikaridha@unisayogya.ac.id
Email Penulis Korespondensi: mutiaranisuci58@gmail.com
Submitted: **04/05/2026**; Accepted: **02/06/2026**; Published**: 05/06/2026**

Abstract−Arrhythmia is a cardiac rhythm disorder that can lead to severe complications, including heart failure and sudden cardiac death. Accurate electrocardiogram (ECG)-based arrhythmia detection remains challenging due to high-dimensional features and class imbalance. Therefore, this study aims to develop a two-stage machine learning approach for arrhythmia detection using Recursive Feature Elimination (RFE) and Extreme Gradient Boosting (XGBoost). The proposed approach performs binary classification to distinguish normal and arrhythmia conditions, followed by multi-class classification to identify arrhythmia subtypes. SMOTE is applied to address class imbalance, while Grid Search with cross-validation is used for hyperparameter optimization. Furthermore, the trained model is implemented in a web-based application for interactive prediction and visualization. Experimental results show that the optimized binary classification model achieves an accuracy of 0.89 and an F1-score of 0.87. Meanwhile, the multi-class classification model achieves an accuracy of 0.69 and a weighted F1-score of 0.66. The results indicate that the proposed approach performs effectively for binary arrhythmia detection. However, performance in multi-class classification remains limited due to imbalance and insufficient samples in several arrhythmia subtypes. This study contributes by proposing an integrated framework that combines Recursive Feature Elimination (RFE) for feature selection, SMOTE for imbalance handling, XGBoost with GridSearchCV-based hyperparameter optimization, and a two-stage classification approach for ECG-based arrhythmia detection and subtype classification. In addition, the proposed model is implemented in a web-based application to support interactive prediction and visualization. Overall, this study demonstrates the potential of integrating RFE, XGBoost, and SMOTE for ECG-based arrhythmia detection and practical web-based implementation.

Keywords: Arrhythmia Detection; XGBoost; Recursive Feature Elimination; SMOTE; Two-stages Classification

1. INTRODUCTION

The heart is a crucial organ responsible for circulating oxygenated blood and nutrients throughout the body. Its condition reflects the overall health of the cardiovascular system. Cardiovascular diseases remain the leading cause of death worldwide, accounting for approximately 17.9 million deaths annually (32% of deaths worldwide) [1]. Heart attacks and strokes account for over 85% of these cases. WHO also emphasizes that most CVDs can be prevented by controlling significant risk factors, including unhealthy eating patterns, sedentary lifestyle, smoking habits, and excessive alcohol use [1]. In Indonesia, heart disease accounts for 26.4% of total deaths, a rate four times higher than cancer-related mortality [2], [3]. Arrhythmia affects 1.5%-5% of the global population. Atrial fibrillation (AF) is the most common type of arrhythmia, affecting more than 46 million people worldwide, with cases projected to increase by 2050. Some types, such as sinus bradycardia, may occur as normal rhythm variations in young adults and athletes, while sick sinus syndrome is more common in older individuals. National data on arrhythmia prevalence in Indonesia remain limited. Several arrhythmias pose high mortality risks, including ventricular tachycardia, which contributes to approximately 25% of sudden cardiac deaths globally (around 4.2 million deaths per year) [4].

Arrhythmia occurs when the heart's electrical activity becomes irregular, resulting in abnormally fast, slow, or erratic heart rhythms. These disturbances can impair blood distribution to vital organs and increase the risk of heart failure and sudden cardiac death [5]. ECG signals are widely used to detect arrhythmias. However, manual interpretation is prone to error due to waveform complexity and inter-individual variation [6]. Therefore, artificial intelligence (AI) and machine learning (ML) techniques are increasingly being utilized as effective solutions for improving arrhythmia detection accuracy [7]. Previous studies have explored various ML techniques for arrhythmia classification. Decision Tree models achieved 67.3% accuracy, indicating limitations in capturing non-linear ECG characteristics [8]. Classical algorithms such as SVM, KNN, and Naïve Bayes achieved up to 71.4% accuracy but still struggled with complex waveform patterns [9]. Deep Neural Networks reached 71.91% accuracy, though performance was limited by the absence of feature selection and imbalanced data handling [10]. Random Forest models achieved 80% accuracy when using interval-based and morphological ECG features [11]. CNN-based approaches using image-transformed ECG data reached 81.21% accuracy but remained affected by feature selection and data imbalance issues [12].

XGBoost has shown strong potential for arrhythmia classification. Using 314 extracted features, it demonstrated improved prediction performance, especially when combined with resampling techniques [13]. It has also outperformed Random Forest, MARS, and Lasso-logistic regression in predicting malignant arrhythmias among heart failure patients [14]. Feature selection methods including Recursive Feature Elimination (RFE) and SHAP-based ranking have further enhanced model performance, achieving accuracies of 98.7%-98.91% through optimal feature reduction [15], [16], [17]. A stacked XGBoost-based model with RFE, PCA, and Chi-Square feature selection reached 99.58% accuracy, demonstrating the effectiveness of multi-stage feature optimization [18]. Class imbalance remains a major challenge in arrhythmia datasets. SMOTE is widely used to generate synthetic samples for minority classes, improving class distribution and reducing model bias [8], [16], [17], [18], [19]. A web-based system was selected for model deployment due to its accessibility across devices and simplified maintenance, as all components run on the server [20].

Although various machine learning methods have been applied to ECG-based arrhythmia classification, several limitations remain in previous studies. Many studies focus solely on classification performance without integrating feature selection, class imbalance handling, and hyperparameter tuning within a unified framework. In addition, most previous studies are limited to binary classification between normal and arrhythmia conditions without identifying specific arrhythmia subtypes. Several studies also evaluate the model only at the experimental stage and do not implement it into an interactive web-based system. Therefore, a comprehensive framework that combines model optimization, arrhythmia subtype classification, and practical implementation remains limited. This study proposes a two-stage machine learning framework using Recursive Feature Elimination (RFE) and XGBoost for ECG-based arrhythmia detection. The first stage performs binary classification to distinguish normal and arrhythmia signals, while the second stage identifies arrhythmia subtypes through multi-class classification. SMOTE is applied to address class imbalance, RFE is used for feature selection, and hyperparameter tuning is performed using GridSearchCV to optimize the XGBoost model. Furthermore, the trained model is implemented in a web-based application to support interactive prediction and result visualization. The main contribution of this study is the development of an integrated framework that combines RFE-based feature selection, SMOTE-based imbalance handling, GridSearchCV-based hyperparameter optimization, and two-stage XGBoost classification for ECG-based arrhythmia detection and subtype classification, along with its implementation in a web-based prediction system.

2. RESEARCH METHODOLOGY

2.1 Research Stages

The stages of this study were structured to provide a clear overview of the implementation of an ECG-based arrhythmia detection system using Recursive Feature Elimination (RFE) and Extreme Gradient Boosting (XGBoost). This research refers to previous studies related to arrhythmia classification frameworks, feature selection techniques, and the application of XGBoost for physiological signal analysis [16], [17], [21]. The dataset used in this study was obtained from the Kaggle platform and consists of 452 samples with 279 features and one diagnostic label. These variables include demographic information such as age, sex, height, and weight, as well as ECG-related parameters including qrs_duration, p-r_interval, q-t_interval, and several waveform indicators such as ΔT, ΔP, and ΔQRST. The diversity of these features provides a comprehensive representation of cardiac electrical activity and supports the identification of arrhythmia patterns [22].

Figure 1 illustrates the overall research workflow used in this study. The process begins with dataset collection, followed by Exploratory Data Analysis (EDA) and preprocessing to improve data quality. The dataset is then divided into training and testing subsets, while normalization is applied to standardize feature values. To overcome class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) is implemented on the training data. Recursive Feature Elimination (RFE) is subsequently utilized to identify the most relevant features for classification. The selected features are then used in the implementation of the XGBoost model with a two-stage classification approach consisting of binary and multi-class classification. Hyperparameter optimization is performed using Grid Search with cross-validation, and model performance is evaluated using accuracy, precision, recall, and F1-score metrics. Finally, the trained model is implemented into a web-based application for arrhythmia prediction and result visualization, as illustrated in Figure 1 following the Web Application Development Life Cycle.

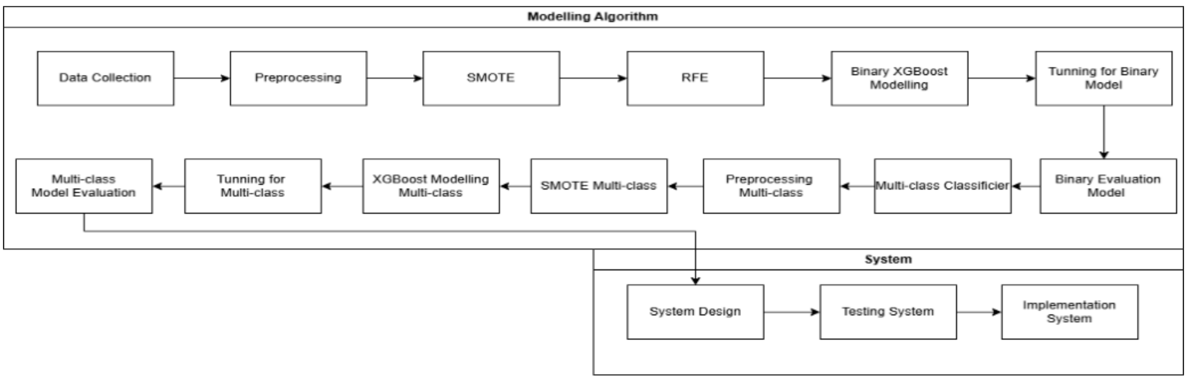


Figure 1. Web Application Development Life Cycle

2.2 Dataset Collection ECG

Data collection is a structured process aimed at obtaining information relevant to the research focus, including the objectives, problem formulation, and established hypotheses. The collected data serve as the foundation for analysis, interpretation, and drawing conclusions. This study employs a quantitative approach using an experimental method to

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

evaluate the effectiveness of the XGBoost classifier utilizing Recursive Feature Elimination (RFE) as a feature selection technique for detecting arrhythmia from ECG signals. The dataset used in this research is the Arrhythmia Dataset from the UCI Machine Learning Repository, which is also available as open-access data on the Kaggle platform. It consists of 452 samples with 279 features and one diagnostic label attribute.

2.3 Preprocessing

To support reliable analysis, the dataset undergoes a preprocessing stage to improve its readiness for subsequent processing electrocardiogram (ECG) signal data for feature extraction and modeling. This stage includes as follows.

2.3.1 Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) phase helps in understanding the dataset before entering the modeling process. At this stage, the structure of the data is examined, the number of available records is reviewed, missing values are checked, and initial patterns that may influence prediction outcomes are identified [2]. EDA also enables the detection of anomalies or data errors that need to be corrected to ensure the model performs more accurately and remains stable.

2.3.2 Data Cleaning

Data cleaning stage, the dataset is examined for duplicate and missing values. Records that are incomplete, invalid, or contain corrupted signals are identified and removed. In addition, non-informative features that do not contribute to the classification process, such as administrative identifiers, are eliminated to reduce noise and improve overall data quality. This step ensures proper preparation of the dataset, while the model demonstrates reliable and relevant performance.

2.3.3 Target Variable Construction

The target variable is constructed based on the diagnosis attribute available in the dataset. In this study, a binary classification approach is applied to simplify the prediction task. Class “1” is categorized as normal (target = 0), while classes “2” to “16” are grouped into the arrhythmia category (target = 1). This transformation is performed to reduce the complexity of multi-class classification and to focus on distinguishing between normal and abnormal heart conditions. After defining the target variable, the dataset is divided into independent variables (X), dependent variable (y) for model training.

2.3.4 Dataset Splitting

The dataset is partitioned into training and testing sets using an 80:20 proportion. This proportion is selected to provide ensuring an adequate amount of data for model training while maintaining a reliable evaluation process. The separation helps reduce the risk of overfitting, as the testing phase is conducted on unseen data. The 80:20 split is commonly used in previous studies and has proven effective for classification tasks, including clinical ECG analysis [5].

2.3.5 Normalization

All features in the dataset will be normalized so that they share a uniform value scale. Normalization is necessary because the amplitude of ECG signals converted into digital form can vary across different ranges, and the model requires consistent values to learn patterns correctly. Normalization is commonly performed by converting values into a range of 0 to 1 [6]. In this study, Min-Max Scaling is planned for use. The final choice of normalization method will be determined based on initial evaluation results, considering model stability and suitability to the dataset’s characteristics.

$$\text{Normalization} = \frac{\text{data} - \min(\text{data})}{\max(\text{data}) - \min(\text{data})} \tag{1}$$

In this normalization process, $Data$ represents the original value to be normalized, while $Min(data)$and $Max(data)$denote the minimum and maximum values within the corresponding feature, respectively. These parameters are used to transform the data into a standardized range, ensuring that each feature contributes proportionally during the model training process.

2.4 Synthetic Minority Oversampling Technique (SMOTE)

The distribution of arrhythmia data is generally imbalanced, leading the model to be biased toward the majority class. To overcome this issue, this study employs the Synthetic Minority Oversampling Technique (SMOTE), which generates synthetic samples for minority classes by interpolating between existing minority data points and their nearest neighbors, resulting in a more balanced class representation without simply duplicating data [16]. SMOTE is widely used in ECG classification studies because it improves minority class representation and expands the model’s decision boundaries [17]. It is often combined with undersampling to prevent excessive spread of synthetic data in extremely rare classes [18]. In this research, SMOTE is applied exclusively to the training data to equalize the number of samples across classes, ensuring that the model receives more proportional training data and minimizes bias toward the majority class. The formula for SMOTE is as follows:

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

$$\mathrm{x\ sample} = \mathrm{x} + \eta(\mathrm{x\ random} - \mathrm{x}) \tag{2}$$

Here, xsample refers to the generated samples of minor ity classes x. Whereas, xrandom refers to a value chosen randomly from the nearest neighbors of x with 0 ≤ x ≤ 1 [19].

2.5 Recursive Featuring Elimination (RFE)

In feature selection, Recursive Feature Elimination (RFE) is applied through an iterative process of removing features with the lowest contribution to model performance until an optimal subset is obtained [21]. This approach is widely used in medical classification tasks as it reduces data dimensionality, simplifies model complexity, and improves predictive performance [17]. In the context of arrhythmia detection, RFE helps reduce the complexity of electrocardiogram (ECG) data by selecting only the most relevant features, enabling the model to focus on significant patterns [18], [23]. In this study, RFE is integrated with XGBoost to evaluate feature importance. The gain metric is mainly used to compute importance scores, reflecting improvements in model performance when a feature is used for splitting, while weight and cover are used as supporting indicators [15].

2.6 Binary XGBoost Classification

As a gradient boosting-based algorithm, XGBoost is widely recognized for its high efficiency and strong predictive performance, particularly in handling structured and high-dimensional datasets. In this study, XGBoost is applied in the first stage as a binary classification model to distinguish between normal and arrhythmia conditions. This stage functions as an initial filtering process, where only samples classified as arrhythmia are forwarded to the next stage for subtype classification. Through a sequential process, the model builds an ensemble of decision trees, where each tree is trained to reduce the errors of previous trees using gradient-based boosting approach. Additionally, XGBoost incorporates regularization to prevent overfitting, making it suitable for handling complex and imbalanced medical datasets such as ECG data [18].

2.6.1 Formula model additive XGBoost

$$\hat{y} = f(x) = \textstyle\sum_{k=1}^{M} fk(x) \tag{3}$$

In the formula above, $f_k(x)$with $k = 1,2,\dots,M$represents the Mdifferent regression trees. XGBoost trains these trees using a *greedy algorithm*, meaning it adds and trains one tree at each step. At step t, all trees from step 1 to $t - 1$are considered fixed and are no longer modified. If the loss function for a single sample is written as $l(y,\hat{y})$, then the training process at step $t$focuses on correcting the errors made by the previous predictions. The final model is constructed as the sum of multiple regression trees, where each $f_k(x)$represents the k-th regression tree. The model is developed iteratively by adding and training one tree at each step

2.6.2 Objective function

$$Obj^{(t)} = \textstyle\sum_{i=1}^{n} l(y_1, f(x_i)) + \Omega(f\,t))$$
$$= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{\,(t-1)} + f_t(x_i)\right) + \Omega(f\,t) \tag{4}$$

$\hat{y}_i^{\,(k-1)} = \sum_{k=1}^{t-1} fk(x_i)$. It represents the prediction result produced by all the trees that have been built up to iteration $(t-1)$. This means the model already has an initial estimate before a new tree is added. Meanwhile, $\Omega(f_t)$is a penalty term used to measure the complexity of the new tree f_t. This penalty helps limit model complexity and prevents overfitting.
The Regulation

$$\Omega(f) = \gamma T + ½\lambda \textstyle\sum_{j=1}^{T} W_1^2 \tag{5}$$

T represents the number of leaf nodes in the decision tree f. Each leaf node produces a specific output value represented by w_j, where $j = 1,2,\dots,T$ In the XGBoost model, a decision tree is composed of multiple leaf nodes, and each leaf contributes a particular prediction value to the final classification result.These prediction values are denoted as w_j, where $j = 1$to T.

2.7 Performance Metric Evaluation

In medical classification studies, the evaluation of model performance is generally carried out using several quantitative measures, including accuracy, precision, recall (sensitivity), and F1-score. Accuracy indicates the ratio of correctly classified samples to the total number of observations. However, in imbalanced datasets such as ECG data, accuracy alone may not provide a reliable evaluation. Therefore, the F1-score is widely used as it balances between precision and recall, leading to a more reliable performance evaluation [8]. For binary classification, these metrics are derived from the confusion matrix, which consists of true positives, true negatives, false positives, and false negatives. In multi-class scenarios evaluation is extended using averaging methods such as macro and weighted averages to ensure fair performance measurement across all classes.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

2.7.1 Accuracy

Accuracy represents the proportion of correct predictions made by the model (True Positive + True Negative) compared to the total number of predictions in the dataset:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

This metric represents the proportion of correct predictions out of all predictions made

2.7.2 F1-Score

By combining precision and recall as a harmonic mean, the F1-score becomes particularly important in imbalanced classification scenarios and when both types of classification errors need to be carefully balanced.

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{7}$$

With:

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

Description : Recall is also known as sensitivity or the true positive rate (TPR) [7], [16].

2.8 Multi-class Classification

After the binary classification stage, samples identified as arrhythmia (target = 1) are further processed in the second stage using a multi-class classification approach to determine specific arrhythmia subtypes. This approach transforms the problem from binary classification into a multi-class classification task. In this stage, the dataset is filtered to include only arrhythmia cases. The input features consist of 17 selected attributes obtained from the Recursive Feature Elimination (RFE) process, while the diagnosis attribute is used as the target label representing multiple arrhythmia classes. The classification is performed using XGBoost configured to handle multi-class prediction, thereby allowing the model to assign each sample into one of the predefined arrhythmia subtypes.

2.9 Preprocessing Multi-class for Arrhythmia Subtypes

In the multi-class classification stage, preprocessing is applied specifically to samples identified as arrhythmia (target = 1) from the previous stage, ensuring that the dataset focuses on arrhythmia subtype classification. Following the previous stage where arrhythmia cases (target = 1) were identified, the dataset is prepared for subtype classification. To support model development, a predefined split is applied to separate the data into training and testing subsets. The training subset is used to build the model, while the testing subset is reserved for performance evaluation on previously unseen data, providing a more reliable indication of the model's generalization ability. Subsequently, feature scaling is performed to standardize the range of input variables, which enhances training stability and accelerates convergence. By doing so, each feature contributes more evenly during the learning process. Through these steps, the dataset is properly prepared for effective multi-class classification.

2.10 SMOTE for Multi-class

In the multi-class classification stage to mitigate class imbalance among arrhythmia subtypes, the Synthetic Minority Oversampling Technique (SMOTE) is employed in a multi-class framework. This method synthesizes new instances for underrepresented classes by utilizing information from their nearest neighbors within the feature space. The value of k_neighbors is dynamically set according to the smallest class size to avoid computational issues in classes with very limited samples. Prior to oversampling, MinMaxScaler is used to normalize the feature values into a uniform range. Following this process, only 17 features selected through the Recursive Feature Elimination (RFE) method are retained for modeling. The oversampling procedure is then carried out solely on the training set, which results in a more balanced class distribution. all arrhythmia subtypes, while the testing data remain unchanged to ensure fair model evaluation.

2.11 XGBoost Classification for Multi-class Subtypes

In the second stage of the proposed system, multi-class classification is performed to identify specific arrhythmia subtypes using the XGBoost algorithm. Prior to model training, the target variable is transformed into a sequential numeric format using a label encoding technique. This step ensures that the configuration remains compatible with the classification model. Subsequently, the XGBoost model is set up for multi-class classification using the multi:softmax objective function with mlogloss employed as the evaluation metric during training. The number of classes is defined based on the encoded labels representing the arrhythmia subtypes. Training is conducted using the resampled dataset obtained from the SMOTE process, which provides a balanced distribution across all classes. This

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

approach enables the model to learn patterns from each subtype more effectively and supports robust multi-class prediction.

2.12 Tuning for Multi-class Subtypes

To improve the performance of the multi-class classification model, the XGBoost algorithm undergoes a hyperparameter optimization process. This process is designed to determine the most suitable parameter configuration that enhances the model's ability to distinguish arrhythmia subtypes effectively. The optimization is carried out using a grid search strategy combined with cross-validation. In this study, GridSearchCV with stratified k-fold cross-validation is utilized to maintain class distribution consistency across each fold during training. The tuning process involves several important XGBoost parameters, such as learning rate, maximum tree depth, number of estimators, and subsampling ratio. Model evaluation is conducted across multiple parameter combinations using a predefined metric to select the best-performing configuration. Finally, the model is trained using the SMOTE-resampled dataset, ensuring a balanced class distribution during optimization. The optimal hyperparameter combination obtained from this process is then used to construct the final multi-class classification model.

2.13 System Design

The arrhythmia prediction system in this study was developed using the Extreme Programming (XP) methodology, which emphasizes iterative development and continuous evaluation [24]. The development process consists of planning, design, coding, and testing phases carried out iteratively to ensure the system meets user requirements. In the planning phase, system requirements, prediction workflows, and data preparation strategies were defined, followed by the design of system architecture with a clear separation between the user interface, processing logic, and data flow. The system was implemented using React.js for the frontend and Flask for the backend, where the backend handles data processing, feature selection, and model inference. The system utilizes a static dataset processed directly without storing user data or prediction history. Therefore, no database is implemented at this stage.

2.14 Testing System

Testing is carried out through an accuracy evaluation by health experts to assess whether the prediction results align with clinical conditions. In addition, black-box testing is performed to ensure that all application features function correctly based on their expected inputs and outputs. The success rate of the system is calculated using the following formula:

$$\text{Success Rate} = \frac{\text{Number of Passed Test Cases}}{\text{Total Number of Test Cases}} \times 100\% \tag{10}$$

This method is commonly used in previous studies, where the outcomes of testing are expressed as percentages of successful and failed test cases to evaluate system performance [25].

2.15 Implementation System

This research implements a web-based arrhythmia prediction system using a client–server architecture. The user interface is developed using React.js to provide an interactive platform for users. The backend system is implemented using Python with the Flask framework as an API to receive and process ECG data and to run the machine learning-based arrhythmia prediction model. The backend handles data preprocessing, feature processing, and prediction inference before returning the results to the frontend in JSON format. The prediction results are then displayed to users through the web interface in a clear and interpretable form.

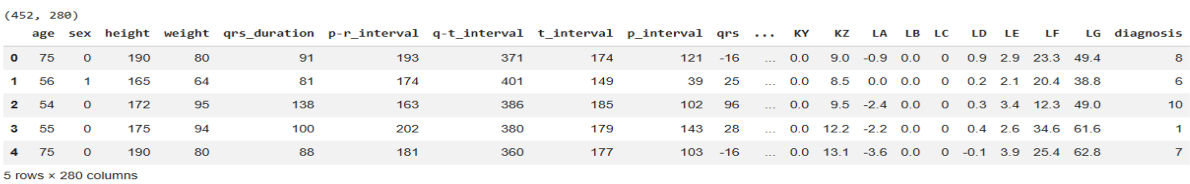
3. RESULT AND DISCUSSION

3.1 Result Preprocessing

The preprocessing stage is conducted to prepare the ECG dataset for feature extraction and machine learning modeling. This stage includes as follows.

3.1.1 Exploratory Data Analysis (EDA)

This stage involves analyzing class distribution, feature distribution, and identifying potential anomalies within the data. The analysis is performed using a pandas and supported by visualization techniques. Figure 2 illustrating the structure and characteristics of the data prior to preprocessing and model development.


(452, 280)

	age	sex	height	weight	qrs_duration	p-r_interval	q-t_interval	t_interval	p_interval	qrs	...	KY	KZ	LA	LB	LC	LD	LE	LF	LG	diagnosis
0	75	0	190	80	91	193	371	174	121	-16	...	0.0	9.0	-0.9	0.0	0	0.9	2.9	23.3	49.4	8
1	56	1	165	64	81	174	401	149	39	25	...	0.0	8.5	0.0	0.0	0	0.2	2.1	20.4	38.8	6
2	54	0	172	95	138	163	386	185	102	96	...	0.0	9.5	-2.4	0.0	0	0.3	3.4	12.3	49.0	10
3	55	0	175	94	100	202	380	179	143	28	...	0.0	12.2	-2.2	0.0	0	0.4	2.6	34.6	61.6	1
4	75	0	190	80	88	181	360	177	103	-16	...	0.0	13.1	-3.6	0.0	0	-0.1	3.9	25.4	62.8	7

5 rows × 280 columns

Figure 2. Dataset ECG

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

Figure 3 and 4 the analysis of the target variable shows that the dataset contains a slight class imbalance, where arrhythmia cases are fewer than normal cases. Furthermore, the distribution of arrhythmia subtypes reveals a more significant imbalance, with several classes being underrepresented while others dominate. This condition indicates the complexity of the classification problem and justifies the need for imbalance handling techniques. Additionally, it was found that subtype classes 11, 12, and 13 contain no samples in the dataset. Therefore, these classes were removed from the classification process to avoid bias and potential errors during model training and evaluation. The exclusion of these empty classes ensures that the model focuses only on valid and representative categories, thereby improving the reliability of the classification results.

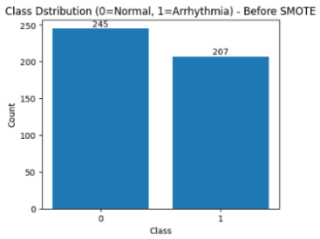


Figure 3. Binary Class Distribution

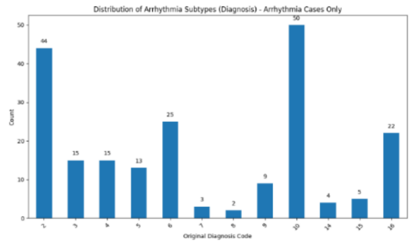


Figure 4. Multi-class Distribution of Arrhytmia Subtipes

3.1.2 Data Cleaning

Data cleaning was performed to ensure data quality for machine learning modeling. Invalid values such as "?", blank spaces, and empty strings were standardized into NaN, and all feature columns (except the diagnosis label) were converted into numerical format using coercion. Features containing entirely missing values were removed, reducing the total number of features from 280 to 279. The remaining missing values were handled using median imputation due to its robustness against outliers. Table 1 presents the dataset after the cleaning process, illustrating the refined data structure used for subsequent preprocessing and model development.

Table 1. Datasets After Cleaning

Target	Count
0 (Normal)	245
1 (Arrhythmia)	207

3.1.3 Target Variable Construction

The target variable is defined based on the diagnosis attribute. A binary classification scheme is applied, where class "1" is categorized as normal (target = 0), and classes "2" to "16" are categorized as arrhythmia (target = 1). The dataset is then split across independent variables (X) dependent variable (y) for model training.

3.1.4 Dataset Splitting

After preprocessing, the dataset was partitioned into training and testing sets with an 80:20 ratio. The training data (X_train) consists of 361 samples with 279 features, along with 361 corresponding labels (y_train). Meanwhile, the testing data (X_test) contains 91 samples with 279 features and 91 labels (y_test). This distribution ensures sufficient data allocated for model training while a portion was kept for evaluation purposes. Class imbalance treatment was performed only to the training data, while the testing data retained its original distribution to ensure an objective evaluation and better reflect the model's generalization ability.

3.1.5 Normalization

Feature scaling is performed using the MinMaxScaler to normalize all numerical features into a range between 0 and 1. The scaler fit_transform() is applied to the training data, whereas transform() is used for the testing data in order to avoid data leakage. After normalization, the data is converted back into DataFrame format to preserve the original structure. The results show that the shape of X_train_scaled is (361, 279) and X_test_scaled is (91, 279), indicating that the normalization process does not change the data dimensions.

3.2 Synthetic Minority Oversampling Technique (SMOTE) for Binary Class

The SMOTE technique was employed to mitigate class imbalance in the training data. After resampling, the number of training samples increased to 392, while the number of features remained 279. Table 2 presents the class distribution.

Table 2. Class Distribution

Target	y_train_resampled:
0 (Normal)	196
1 (Arrhythmia)	196

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

The class distribution became balanced, with 196 samples for each class after applying SMOTE. This indicates that synthetic samples were successfully generated for the minority class using the k-nearest neighbors approach. As a result, the model is expected to learn minority-class patterns more effectively and reduce prediction bias toward the majority class, thereby improving classification performance on imbalanced ECG data.

3.3 Recursive Feature Elimination (RFE)

Once the class imbalance problem is mitigated through SMOTE, Recursive Feature Elimination (RFE) is applied for feature selection by employing XGBoost as the base model to extract the most relevant features. A total of 15 features were selected by RFE, with two additional features (age and sex) included based on their clinical relevance, resulting in 17 features. The selected features include C3, DN, DZ, GJ, HR, IH, IV, JB, JD, JZ, KD, KV, LE, age, heart_rate, qrs_duration, and sex. Following feature selection, the dataset was reduced to 392 training samples and 91 testing samples, each with 17 features. This reduction improves model efficiency while preserving important information for classification. Table 3 presents the top selected features identified by the RFE method, highlighting the most influential features used in the arrhythmia classification process.

Table 3. Top 5 Selected Features by RFE

1. qrs_duration
2. heart_rate
3. CJ
4. DN
5. DZ

3.4 Binary XGBoost Classifier

The classification model was constructed using the XGBoost algorithm. The model was configured with the binary:logistic objective function and evaluated using the logloss metric. A random_state of 42 was applied in order to maintain result reproducibility. The model was trained using the selected features obtained from the RFE process and the resampled training data generated by SMOTE. This approach enables the model to learn from relevant features while addressing class imbalance.

3.5 Binary Model Evaluation Before and After Hyperparameter Tuning

Before hyperparameter tuning, the XGBoost model achieved an accuracy of 85.71%, with a precision of 0.8537, recall of 0.8333, and F1-score of 0.8434. Figure 5 shows the confusion matrix before tuning, where several false positives and false negatives were still observed, indicating that the default parameter configuration had not fully captured the ECG characteristics. After hyperparameter tuning using Grid Search with 5-fold cross-validation, the best configuration was obtained with learning_rate = 0.1, max_depth = 3, n_estimators = 100, and subsample = 0.8. As shown in Figure 6, the optimized model achieved improved performance with an accuracy of 89.01%, precision of 0.9000, recall of 0.8571, and F1-score of 0.8780. This improvement indicates that hyperparameter optimization enhanced the ability of XGBoost to learn more representative ECG patterns while reducing misclassification errors. In addition, the relatively shallow tree depth and controlled learning rate likely helped prevent overfitting, resulting in better generalization performance. The obtained results are also competitive with several previous machine learning-based arrhythmia classification studies.

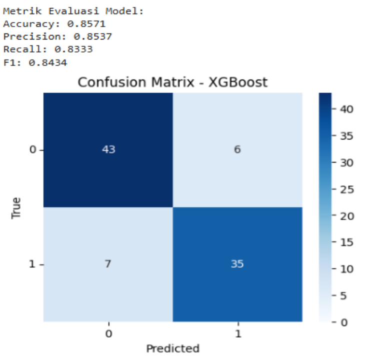


Figure 5. Confusion Matrix Before Tuning

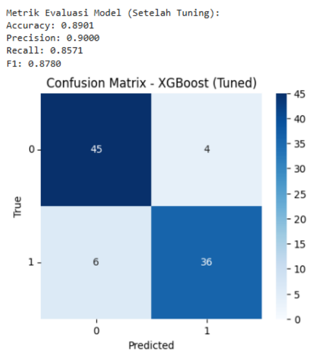


Figure 6. Confusion Matrix After Tuning

3.6 Multi-class Classification for Predicting Arrhythmia Subtypes

After detecting arrhythmia, a multi-class classification stage was conducted to identify specific arrhythmia subtypes. The dataset was filtered to include only arrhythmia cases (target = 1), using the 17 selected features obtained from the RFE process as input variables, with the diagnosis attribute as the target. The resulting dataset consists exclusively of arrhythmia samples with 17 features, and the class distribution indicates the presence of imbalance among the subtypes.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

3.7 Data Splitting for Multi-class Model

The multi-class dataset was split into training and testing sets using an 80:20 ratio, resulting in 165 and 42 samples, respectively, each with 17 selected features. Stratified sampling was implemented to preserve class distribution across both sets. However, the data remains imbalanced, with some classes having significantly fewer samples, which may negatively affect model performance, particularly in predicting minority classes.

3.8 Scaling Data for Multi-class Model

Feature scaling was carried out using the same MinMaxScaler fitted on the training data to maintain consistency across models. Because the scaler was originally trained on the full feature set, the multi-class dataset was temporarily reconstructed before normalization. After scaling, only the 17 features selected through the RFE process were retained. The final dataset consists of 165 training samples and 42 testing samples, each with 17 normalized features, ensuring consistent feature representation for model training.

3.9 Handling Imbalance in Multi-class Data with SMOTE

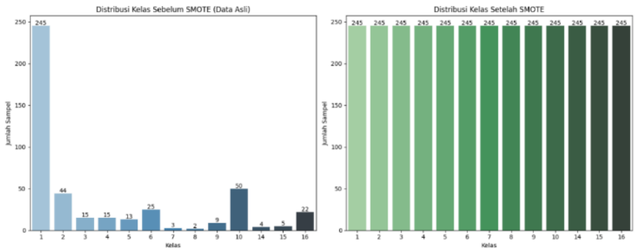


Figure 7. Dataset Multi-class After SMOTE

The dataset is characterized by class imbalance, with several classes having significantly fewer samples. Accordingly, SMOTE was employed on the training data using the imblearn library with k_neighbors = 5. As a result, the number of training samples increased from 165 to 480 while retaining the 17 selected features, leading to a more balanced class distribution and improved learning across classes. Figure 7 illustrates the multi-class dataset distribution after the application of SMOTE, showing that the minority classes were successfully augmented to achieve a more balanced distribution across all classes. This balancing process helped reduce the dominance of majority classes and provided the model with more representative training data for minority class learning.

3.10 Handling the Multi-class XGBoost Model

A multi-class classification model was implemented using XGBoost to predict arrhythmia subtypes. The target variable was converted into 12 numerical classes using LabelEncoder to meet the model requirements. The model was configured with the multi:softmax objective and mlogloss as the evaluation metric. Training was conducted on the dataset resampled using SMOTE to handle class imbalance, enabling the model to learn all classes more effectively and proceed to the evaluation stage.

3.11 Evaluation of Multi-class Model Before and After Hyperparameter Tuning Using Grid Search

Before tuning, the XGBoost model achieved an accuracy of 0.64 with a weighted F1-score of 0.62, indicating moderate performance. However, the macro F1-score was lower at 0.46, suggesting uneven classification performance across different classes. Figure 8 presents the confusion matrix before tuning, showing that the model performed well on several majority classes but struggled to generalize across minority classes. Meanwhile, Figure 9 illustrates the evaluation performance of the model before hyperparameter tuning, where prediction errors can be observed in several minority classes. Although classes 5, 6, and 9 were classified relatively well, classes 7, 14, and 15 showed poor prediction performance, indicating that the model tended to favor majority classes during classification.

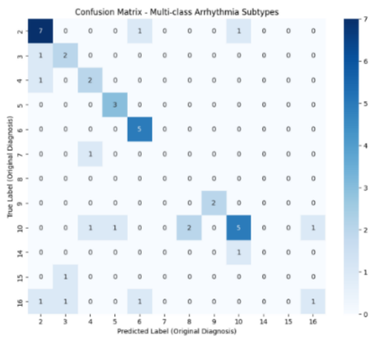


Figure 8. Confusion Matrix Before Tuning

```
Multi-class Classification Report:
              precision    recall  f1-score   support

           2       0.70      0.78      0.74         9
           3       0.50      0.67      0.57         3
           4       0.50      0.67      0.57         3
           5       0.75      1.00      0.86         3
           6       0.71      1.00      0.83         5
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         0
           9       1.00      1.00      1.00         2
          10       0.71      0.50      0.59        10
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.50      0.25      0.33         4

    accuracy                           0.64        42
   macro avg       0.45      0.49      0.46        42
weighted avg       0.63      0.64      0.62        42

Multi-class Accuracy: 0.6429
```

Figure 9. Evaluation Performance Before Tuning

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

Using GridSearchCV, hyperparameter tuning was performed with 5-fold stratified cross-validation by evaluating 81 parameter combinations (405 model fits). The best configuration achieved a weighted F1-score of 0.9254 during cross-validation. After tuning, the model achieved an accuracy of 0.69 and a weighted F1-score of 0.66. However, the macro F1-score remained relatively low at 0.48, indicating inconsistent performance across classes. Figure 10 illustrates the evaluation performance after tuning, where improvements in accuracy and weighted F1-score can be observed. Figure 11 presents the confusion matrix after hyperparameter tuning, demonstrating improved classification performance for several classes compared to the untuned model. Nevertheless, some minority classes still experienced misclassification, indicating that the model tended to favor majority classes.

```
Multi-class Classification Report (Setelah Tuning):
              precision    recall  f1-score   support

           2       0.70      0.78      0.74         9
           3       0.50      0.67      0.57         3
           4       0.67      0.67      0.67         3
           5       0.75      1.00      0.86         3
           6       0.71      1.00      0.83         5
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         0
           9       1.00      1.00      1.00         2
          10       0.70      0.70      0.70        10
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       1.00      0.25      0.40         4

    accuracy                           0.69        42
   macro avg       0.50      0.51      0.48        42
weighted avg       0.68      0.69      0.66        42

Multi-class Accuracy (Setelah Tuning): 0.6905
```

Figure 10. Evaluation Performance After Tuning

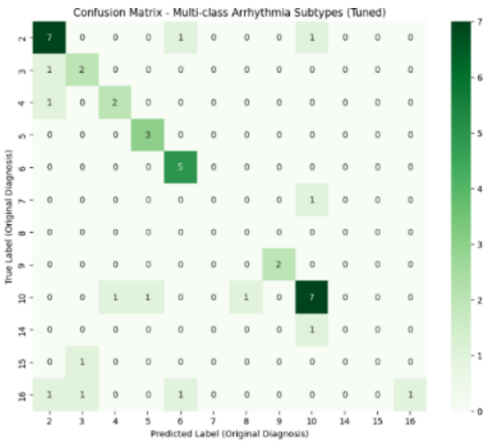
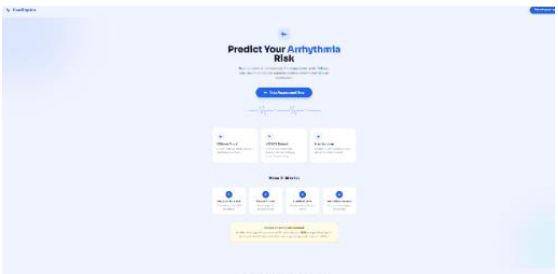


Figure 11. Confusion Matrix After Tuning

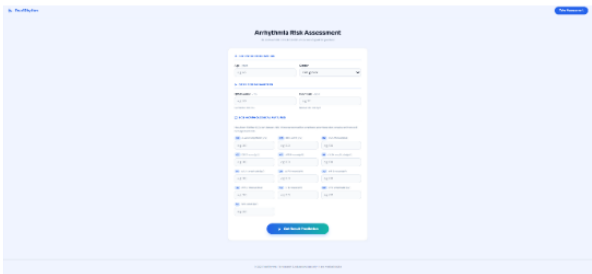
These findings indicate that multi-class arrhythmia classification is more challenging than binary classification due to the complex characteristics of ECG signals across different arrhythmia subtypes. Several minority classes had limited samples, making it difficult for the model to learn representative subtype patterns effectively. In addition, similarities in ECG characteristics among certain subtypes may increase inter-class confusion. Although SMOTE improved the class distribution, the limited number of samples in several minority classes still affected the overall classification performance. The obtained performance is comparable to several previous machine learning-based arrhythmia classification studies. However, the relatively low macro F1-score indicates that handling minority classes in multi-class ECG classification remains a significant challenge, particularly when the dataset distribution is highly imbalanced.

3.12 System Design

To support arrhythmia prediction, a web-based application is implemented, allowing users to initiate the prediction process via the "Take Assessment" feature on the interface. Users are required to input 17 selected parameters obtained from the Recursive Feature Elimination (RFE) method. After all required data is entered, the input is processed by the trained model within the system, and the prediction results are subsequently displayed. Figure 12 illustrates the dashboard interface of the arrhythmia detection website, while Figure 13 presents the prediction interface for arrhythmia classification.



Picture 1. Dashboard Interface Website Arrhythmia Detection



Picture 2. Prediction Interface for Arrhythmia Classification

3.13 Testing System

System testing shows that the prediction results are consistent with expert evaluations, indicating that the system produces outputs aligned with clinical assessments and can support arrhythmia detection. In addition, black-box testing confirms that all system functionalities, including data input, processing, and prediction output, operate as expected without errors. However, the 17 input parameters required by the system must be entered manually by healthcare professionals based on manual ECG interpretation results prior to the prediction process.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

4. CONCLUSION

This research successfully developed an ECG-based arrhythmia detection system using Recursive Feature Elimination (RFE) and Extreme Gradient Boosting (XGBoost) within a two-stage machine learning framework. The proposed approach demonstrated strong performance in binary classification, achieving an accuracy of 0.89 and an F1-score of 0.87 after hyperparameter optimization using GridSearchCV. Meanwhile, the multi-class classification model achieved an accuracy of 0.69 and a weighted F1-score of 0.66, although the macro F1-score indicates inconsistent performance across several arrhythmia subtypes due to class imbalance and limited sample availability. This study contributes by proposing an integrated framework that combines RFE-based feature selection, SMOTE for imbalance handling, XGBoost with GridSearchCV-based hyperparameter optimization, and a two-stage classification architecture for ECG-based arrhythmia detection and subtype classification. Furthermore, the use of RFE successfully reduced the feature space into 17 important features, improving model efficiency while maintaining predictive performance. From a practical perspective, the trained model was implemented into a web-based system for interactive arrhythmia prediction and result visualization. However, the performance of the multi-class classification model remains limited due to severe class imbalance and insufficient samples in several arrhythmia subtypes. In addition, subtypes 11, 12, and 13 were excluded from the analysis because no samples were available in the dataset in order to avoid bias and training instability. These limitations reduce the model's ability to generalize across all arrhythmia categories. Therefore, future studies are recommended to expand the dataset, particularly by increasing the number of samples for underrepresented arrhythmia subtypes, as well as exploring more advanced feature selection and imbalance handling techniques to improve classification robustness and overall predictive performance.

REFERENCES

[1] World Health Organization, "Cardiovascular Diseases (CVDs)." Accessed: May 08, 2026. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
[2] E. P. Silmina and L. Perkasa, "EDA and Tableau Analysis for Identification of Heart Disease Risk Factors," *J. Artif. Intell. Softw. Eng. J-AISE*, vol. 5, no. 1, p. 79, Mar. 2025, doi: 10.30811/jaise.v5i1.6389.
[3] S. Liza Munira, D. Puspasari, and Trihono, "Survei Kesehatan Indonesia (SKI) Tahun 2023," Slideshare. Accessed: May 21, 2026. [Online]. Available: https://www.slideshare.net/slideshow/survei-kesehatan-indonesia-ski-tahun-2023/269424965
[4] Michael Sintong Halomoan and D. Rahmah, "Epidemiologi Aritmia." Accessed: May 08, 2026. [Online]. Available: https://www.alomedika.com/penyakit/kardiologi/aritmia/epidemiologi
[5] M. Fira, L. Goraș, L. Fira, R. F. Popa, and H.-N. Costin, "Arrhythmia Classification With Single-Channel Features Extracted from 'A Large-Scale 12-Lead ECG Database for Arrhythmia Study,'" *Sensors*, vol. 25, no. 18, p. 5621, Sep. 2025, doi: 10.3390/s25185621.
[6] A. Darmawahyuni *et al.*, "An Improved Electrocardiogram Arrhythmia Classification Performance With Feature Optimization," *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, p. 412, Dec. 2024, doi: 10.1186/s12911-024-02822-7.
[7] M. Martínez-Sellés and M. Marina-Breysse, "Current and Future Use of Artificial Intelligence in Electrocardiography," *J. Cardiovasc. Dev. Dis.*, vol. 10, no. 4, p. 175, Apr. 2023, doi: 10.3390/jcdd10040175.
[8] S. Sattar *et al.*, "Cardiac Arrhythmia Classification Using Advanced Deep Learning Techniques on Digitized ECG Datasets," *Sensors*, vol. 24, no. 8, p. 2484, Apr. 2024, doi: 10.3390/s24082484.
[9] R. M. Devadas, "Cardiac Arrhythmia Classification Using SVM, KNN and Naive Bayes Algorithms," *Int. Res. J. Eng. Technol. IRJET*, vol. 8, no. 5, 2021.
[10] M. F. Z. Nugraha, H. F. Tsp, and R. Magdalena, "Deteksi Aritmia Menggunakan Algoritma Deep Neural Network (DNN) Pada Sinyal Elektrokardiogram," *E-Proceeding Eng.*, vol. 10, no. 5, p. 4155, Oct. 2023.
[11] A. Sitorus, P. A. R. Simanjuntak, M. C. Angelina. H., K. A. Situmorang, and A. Z. Putra, "ECG-Based Arrhythmia Classification in Students Using Random Forest: A Case Study with Class Imbalance Analysis," *Sinkron*, vol. 9, no. 2, pp. 905–913, Jun. 2025, doi: 10.33395/sinkron.v9i2.14793.
[12] A. A. Pratama and A. M. Abadi, "Classification of Arrhythmia Diseases by the Convolutional Neural Network Method Based on ECG Images," *BAREKENG J. Ilmu Mat. Dan Terap.*, vol. 17, no. 2, pp. 0625–0634, Jun. 2023, doi: 10.30598/barekengvol17iss2pp0625-0634.
[13] D. Asmawati, L. Arif Sanjani, C. Dimas Renggana, C. Fatichah, and T. Mustaqim, "Arrhythmia Classification With ECG Signal Using Extreme Gradient Boosting (XGBoost) Algorithm," *J. Technol. Inform. JoTI*, vol. 6, no. 1, pp. 36–42, Oct. 2024, doi: 10.37802/joti.v6i1.792.
[14] Q. Wang *et al.*, "Machine Learning-Based Risk Prediction of Malignant Arrhythmia in Hospitalized Patients With Heart Failure," *ESC Heart Fail.*, vol. 8, no. 6, pp. 5363–5371, Dec. 2021, doi: 10.1002/ehf2.13627.
[15] X. Yu, "ECG Signal Classification Based on DWT Denoising and XGBoost," *Appl. Comput. Eng.*, vol. 95, no. 1, pp. 57–67, Oct. 2024, doi: 10.54254/2755-2721/95/2024BJ0057.
[16] R. Xiao, M. Yang, C. Ma, L. Zhao, J. Li, and C. Liu, "Interpretable XGBoost-SHAP Model for Arrhythmic Heartbeat Classification," presented at the 2024 Computing in Cardiology Conference, Dec. 2024. doi: 10.22489/CinC.2024.186.
[17] M. Saleem, W. Aslam, M. I. U. Lali, H. T. Rauf, and E. A. Nasr, "Predicting Thalassemia Using Feature Selection Techniques: A Comparative Analysis," *Diagnostics*, vol. 13, no. 22, p. 3441, Nov. 2023, doi: 10.3390/diagnostics13223441.
[18] R. Jahangir, M. N. Islam, Md. S. Islam, and Md. M. Islam, "ECG-Based Heart Arrhythmia Classification Using Feature Engineering and a Hybrid Stacked Machine Learning," *BMC Cardiovasc. Disord.*, vol. 25, no. 1, p. 260, Apr. 2025, doi: 10.1186/s12872-025-04678-9.
[19] S. A. Hadi and T. Hardiani, "Arrhythmia Disease Detection Using SVM With Recursive Feature Elimination," *Int. J. Inform. Comput. IJICOM*, vol. 7, no. 2, 2025, doi: 10.35842/ijicom.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 246−257
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9840

[20] A. Mahmoudian Azar Sharabiani and S. M. Mousavi, “A Web-Based Decision Support System for Project Evaluation With Sustainable Development Considerations Based on Two Developed Pythagorean Fuzzy Decision Methods,” *Sustainability*, vol. 15, no. 23, p. 16477, Dec. 2023, doi: 10.3390/su152316477.

[21] A. M. Priyatno and T. Widiyaningtyas, “A Systematic Literature Review: Recursive Feature Elimination Algorithms,” *JITK J. Ilmu Pengetah. Dan Teknol. Komput.*, vol. 9, no. 2, pp. 196–207, Feb. 2024, doi: 10.33480/jitk.v9i2.5015.

[22] B. Eralp, “Arrhythmia Detection From ECG Data.” Accessed: May 08, 2026. [Online]. Available: https://kaggle.com/code/batuhaneralpofficial/arrythmia-detection-from-ecg-data

[23] J. Reddekopp, “Machine Learning: Feature Selection — Recursive Feature Elimination,” Simplifying Complexity: Feature Selection with Recursive Feature Elimination. Accessed: May 09, 2026. [Online]. Available: https://readmedium.com/machine-learning-feature-selection-recursive-feature-elimination-88750767c05f

[24] M. L. Abdillah and D. Wijayanto, “Perancangan Sistem Informasi Cuti Berbasis Web pada Rumah Sakit Asri Medical Center,” *J. Sist. Inf. Dan Bisnis Cerdas*, vol. 18, no. 1, Feb. 2025, doi: 10.33005/sibc.v18i1.419.

[25] P. A. D. A. Santi, R. Afwani, Moh. A. Albar, S. E. Anjarwani, and A. Z. Mardiansyah, “Black Box Testing with Equivalence Partitioning and Boundary Value Analysis Methods (Study Case: Academic Information System of Mataram University),” in *Proceedings of MIMSE 2022 (Informatics and Computer Science)*, Dordrecht, 2022, pp. 207–219. doi: 10.2991/978-94-6463-084-8_19.