Evaluation of Stacked Generalization: Integration of CatBoost with SVM and Random Forest for Arrhythmia Classification in Adolescents

Mei Ditta Kristie Br Bangun[a], Swando Sagala[b], Eddrick Wilbert Solo[c], Delima Sitanggang[d]
[a,b,c,d]Universitas Prima Indonesia
Corresponding Author:
[a]delimasitanggang@unprimdn.ac.id

ABSTRACT
Arrhythmia is a disorder of heart rhythm that can occur in anyone, including adolescents, and may pose serious risks if not detected early. This study aims to evaluate the effectiveness of the CatBoost algorithm, both as a standalone model and in combination with Support Vector Machine (SVM) and Random Forest (RF) using ensemble approaches, namely voting and stacked generalization, for arrhythmia classification based on electrocardiogram (ECG) signals. Data were obtained from 72 students aged 15–19 years using a Smart Holter ECG 5-lead device based on Raspberry Pi 4. The models were developed and tested using the Python programming language. The results show that the standalone CatBoost model achieved strong performance, with an accuracy of 95.45% and a ROC AUC of 99.81%. The CatBoost–RF voting method yielded a higher accuracy (96.97%), while the CatBoost–RF stacking approach demonstrated the best overall performance, achieving an accuracy of 98.48% and a ROC AUC of 99.85%. In contrast, the CatBoost–SVM stacking approach produced the lowest performance, with an accuracy of 89.39%. Based on evaluation using accuracy, F1-score, and ROC AUC metrics, it can be concluded that the ensemble stacking method with Random Forest as the base learner and CatBoost as the meta-learner is the most effective approach for arrhythmia classification in adolescent ECG data. This study is expected to contribute to the development of more accurate machine learning-based automatic arrhythmia detection systems tailored for younger populations.

Keywords : Arrhythmia, ECG, CatBoost, SVM, Random Forest.

INTRODUCTION

Arrhythmia is a disorder of heart rhythm that can have serious consequences, particularly if not detected early (Nagpal et al., 2024). In adolescents, arrhythmia often does not present clear symptoms, making data-driven diagnostic methods necessary to detect patterns in electrocardiogram (ECG) signals. In recent years, machine learning approaches have been widely used to improve the accuracy of arrhythmia classification. Models such as Support Vector Machine (SVM) and Random Forest have demonstrated strong performance in various previous studies (Gajowniczek et al., 2020; Kumari et al., 2021).

However, one of the main challenges in arrhythmia classification is the complexity of data patterns and class imbalance. ECG signals are typically nonlinear and highly variable across

individuals, making simple models insufficient to capture all relevant information in the data (Behera et al., 2025). In addition, the number of arrhythmia samples is usually much smaller than that of normal samples, causing models to be biased toward the majority class (Ma et al., 2022).

To address these challenges, ensemble methods such as stacked generalization (stacking) offer a solution to improve model accuracy and generalization capability. Stacking enables the combination of multiple base learners to produce better predictions (Jahangir et al., 2025). In this context, combining CatBoost with models such as SVM or Random Forest, through either stacking or voting approaches, has the potential to yield more optimal classification performance. SVM excels in maximizing class separation margins, while Random Forest is effective in handling noisy data and complex feature interactions (Anshor & Wiyatno, 2024).

CatBoost, a gradient boosting algorithm based on decision trees, has gained increasing attention due to its ability to handle categorical features and achieve high performance on complex datasets (Boldini et al., 2023). It also incorporates regularization techniques that make it more robust to overfitting compared to other boosting algorithms (Dorogush et al., 2018). Although CatBoost has shown promising results in several classification studies, its integration within a stacking framework alongside SVM and Random Forest for arrhythmia classification, particularly in adolescents, remains underexplored (Prasanna Venkatesh et al., 2023; Wu et al., 2025).

In Indonesia, cardiovascular disease is one of the leading causes of death. According to data from Riset Kesehatan Dasar (Riskesdas), the prevalence of heart disease exceeds 1.5% of the population and continues to increase annually (Wu et al., 2025). Although arrhythmia in adolescents has not been widely reported specifically, risk factors such as stress, physical fatigue, high consumption of sugar- and fat-rich foods, and excessive use of digital devices may indirectly affect adolescent heart health (Schnermann et al., 2021; Presta et al., 2024).

Furthermore, many existing studies rely on clinical data from hospitals, resulting in limited research based on non-clinical populations such as students or adolescents (Santini et al., 2018). Therefore, this study aims to evaluate the effectiveness of a stacked generalization approach by integrating CatBoost, SVM, and Random Forest models for arrhythmia classification in adolescents. This research is exploratory in nature and seeks to assess the extent to which these model combinations can improve classification performance compared to single models. It is expected that the findings will contribute to the development of more accurate machine learning-based decision support systems for arrhythmia diagnosis tailored to younger populations.

METHODS

This study aims to compare the use of ensemble learning-based classification approaches, namely Support Vector Machine and Random Forest, with a gradient boosting model, CatBoost, for predicting arrhythmia classification in adolescent students. CatBoost is a gradient boosting-based algorithm that optimizes the MultiClass loss function with evaluation metrics such as accuracy and is known for its effectiveness in handling categorical features as well as its high efficiency during the training process (Luong et al., 2021). Support Vector Machine (SVM) is a classification algorithm that works by finding an optimal hyperplane

that separates classes in the feature space with the maximum margin; it is effective in high-dimensional spaces and robust to overfitting when an appropriate kernel is selected (Kaushik et al., 2023). Probabilistic SVM is used to generate classification probabilities via Platt scaling (Kull et al., 2017). Random Forest is an ensemble algorithm based on bagging that combines predictions from multiple decision trees to produce a final decision (Salman et al., 2024). It can reduce overfitting from a single decision tree and improve accuracy by aggregating multiple weak learners (Zhu et al., 2023).
This study compares the performance of a standalone CatBoost model, CatBoost combined with SVM, and CatBoost combined with Random Forest. In implementing SVM and Random Forest, two approaches are applied: voting and stacking. Voting combines predictions from multiple models using majority voting or average probabilities, while stacking employs a new model (meta-model) to integrate predictions from base learners (Airlangga, 2025).
The designed models undergo both training and prediction processes using a dataset divided into training and testing sets. The prediction results are then evaluated using several performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. These evaluation metrics are compared to determine the most accurate model. The overall research workflow is illustrated in Figure 1.

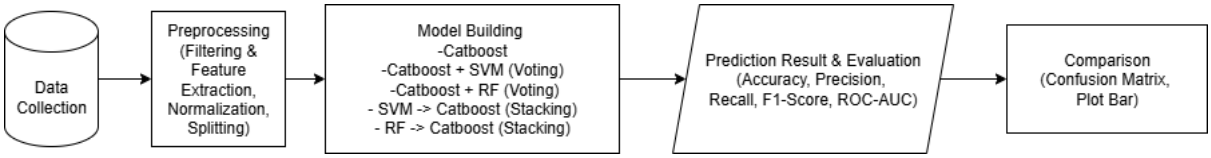


Figure 1. Block Diagram

Data Collection

The study was conducted at SMK Swasta Teladan Sumatera Utara 2, Jl. Pendidikan No.62, Cinta Damai, Kec. Medan Helvetia, Kota Medan, North Sumatra, in collaboration with UNPRI. The partners involved in this study included the students conducting the research and the supervising lecturers. The respondents were 106 students aged 15 to 20 years, consisting of 101 males and 5 females. Data was collected using a Smart Holter ECG 5-lead device based on Raspberry Pi 4, electrode cables, and a laptop used for data recording and operation of the recording software. Six electrodes were attached to the subjects' bodies as shown in Figure 2. Data recording was carried out while the subjects were seated to obtain ECG data under neutral conditions.

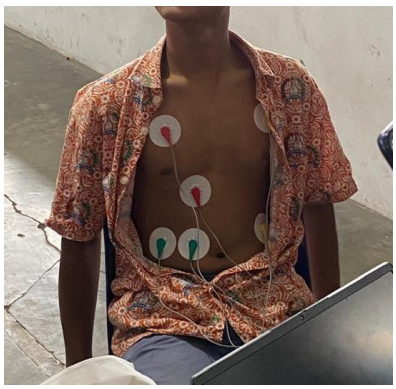


Figure 2. Use of the Smart Holter ECG 5-Lead Device in ECG Data Recording

Data collection was conducted using the Serial Data Acquisition program, which reads and processes raw ECG signals from the Smart Holter ECG 5-lead system with six electrodes. The device records electrical signals from the upper body of the subjects, which are then converted into integer values, assuming the signals have a 2.4-volt range and 24-bit resolution. These signals are transmitted via the ECG device's serial port, as shown in Figure 3. On Jack 1, the red electrode is placed on the upper right chest (RA), the yellow electrode on the upper left chest (LA), and the green electrode on the lower right body (RL). On Jack 2, the red electrode is placed on the sternum area slightly to the right (V1), the yellow electrode on the lower left body (LL), and the green electrode remains on the lower right body (RL).

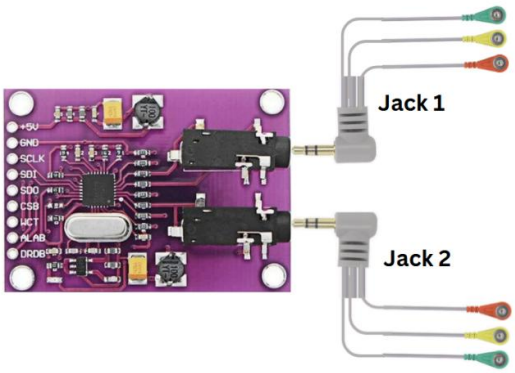


Figure 3. Smart Holter ECG 5-Lead Device

Data Preprocessing

The next stage after data collection is data preprocessing to extract the features required for model training. Preprocessing was carried out using a Python-based pipeline with libraries such as Pandas, NeuroKit2, and SciPy. The signals to be extracted include the P, Q, R, S, and T waves, which are then used to derive features such as heart rate and the durations between signals. A total of seven features were extracted: RR Interval, PR Segment, QRS Duration, QT Interval, ST Segment, R/S Ratio, and heart rate in beats per minute (bpm).

The dataset with extracted features is labeled as part of a supervised learning approach, where the dataset used for training the model is assigned labels representing the ground truth to evaluate prediction accuracy. Labels are determined based on comparisons of the dataset features, categorizing each data point into four classes: Normal, Abnormal, Potential Arrhythmia, and High-Risk Arrhythmia, according to medical standards.

After feature extraction and labeling, the dataset undergoes further processing, including splitting into training and testing sets and normalization. Data is divided into two categories: 90% for training the model and 10% for testing the model's predictive capability. Normalization is performed using StandardScaler from the scikit-learn library, which standardizes the features by removing the mean and scaling to unit variance. This normalization ensures that all features are on a uniform scale, supporting balanced model training.

Model Development

The development of models for arrhythmia prediction was carried out using Visual Studio Code with Python, supported by libraries such as pandas, numpy, scikit-learn, catboost, matplotlib, and seaborn. The model development process includes several main stages: data preprocessing, model selection and training, and performance evaluation.
The initial stage begins with reading the arrhythmia dataset from a .csv file, where numerical features are extracted and class labels (output) are encoded using LabelEncoder for use in classification algorithms. Next, all features are normalized using StandardScaler to ensure each feature is on the same scale, improving machine learning model performance. The data is then split into training and testing sets at a 90:10 ratio using the train_test_split function. Several models were developed and compared in this study, including the base CatBoostClassifier model and ensemble models combining CatBoost with other algorithms such as Support Vector Machine (SVM) and Random Forest (RF). The ensemble approaches used consist of two types: Voting Classifier and Stacking Classifier.
In the Voting Classifier, the final prediction is obtained from the average predicted probabilities of member models such as CatBoost-SVM and CatBoost-RF (soft voting). Meanwhile, the Stacking Classifier uses models such as SVM or RF as base learners, with CatBoost as the meta-learner that learns from the output predictions of the base models.

Model Evaluation

The performance of the models after prediction was evaluated using several common classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The first step in the evaluation process is to create a confusion matrix to compare the model's predictions with the true labels of the test data. From this matrix, fundamental values such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are obtained, which form the basis for calculating the evaluation metrics.
Accuracy measures the proportion of correct predictions out of the total data. However, in cases where class distribution is imbalanced, accuracy alone is often insufficient to reflect the true performance of the model. Therefore, precision and recall are used as complementary metrics. Precision indicates how accurately the model predicts a specific class, while recall shows the model's ability to capture all instances belonging to that class. The F1-score is then used to balance precision and recall, especially when both are important. In addition to these metrics, the ROC-AUC value is calculated to evaluate how well the model distinguishes between classes based on predicted probabilities. ROC-AUC is particularly useful for understanding model prediction quality across different thresholds, especially in multiclass

classification or imbalanced datasets. Using this combination of metrics provides a more comprehensive evaluation and a better understanding of the strengths and limitations of each tested model.

RESULTS

Data Collection Result

In the initial stage of this study, the electrocardiogram (ECG) data obtained from the Smart Holter device were evaluated. The evaluation revealed that some of the subjects' data did not meet completeness criteria. Incomplete data were identified by missing signal segments or certain iterations, indicating technical issues during the recording process, such as excessive body movement, electrode connection problems, or external noise. Therefore, a selection process was conducted to filter out invalid subject data. Out of a total of 106 recorded subjects, only 72 had complete data suitable for further analysis. Although this number was smaller than initially planned, the remaining data were considered to have sufficient signal quality to support ECG parameter extraction and analysis.

Data Preprocessing Result

The preprocessing process began with converting the raw values from the Analog to Digital Converter (ADC) into millivolt (mV) units. However, the initial results showed anomalies with signal values far exceeding the normal ECG range. For illustration, Figure 4 shows raw signals with amplitudes up to 871 mV, whereas physiologically, ECG signals typically range from -5 to +5 mV. To address this issue and obtain more representative signals, a series of filtering steps was applied using three main stages: baseline correction, filtering with a Butterworth filter, and further filtering with a Finite Impulse Response (FIR) filter.

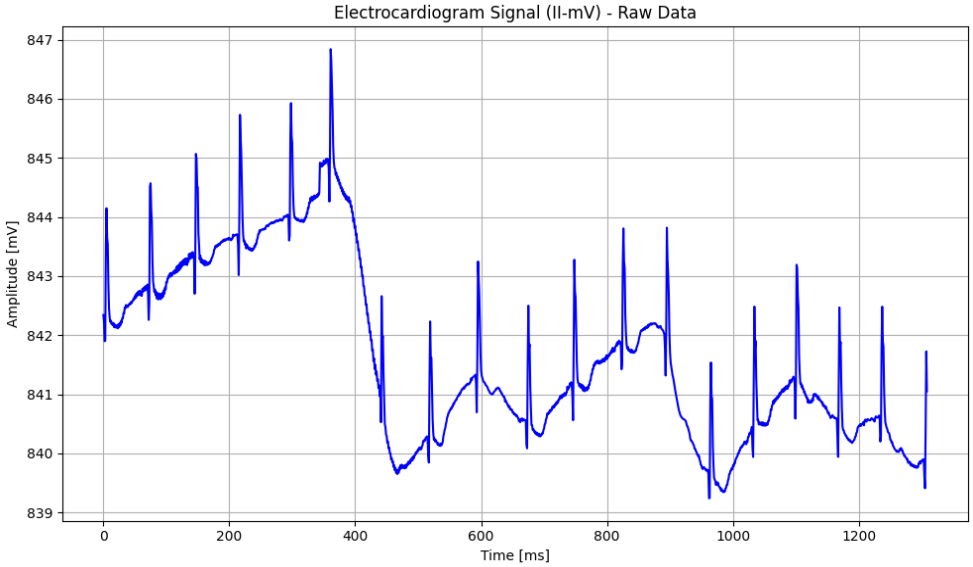


Figure 4. Raw ECG Signal Plot.

The first step, baseline correction, aims to eliminate baseline drift caused by respiration, body movements, or electrode disturbances. This correction is performed using a linear detrending method, which allows the ECG waveform, including the PQRST waves, to appear clearer and more stable, as shown in Figure 5.

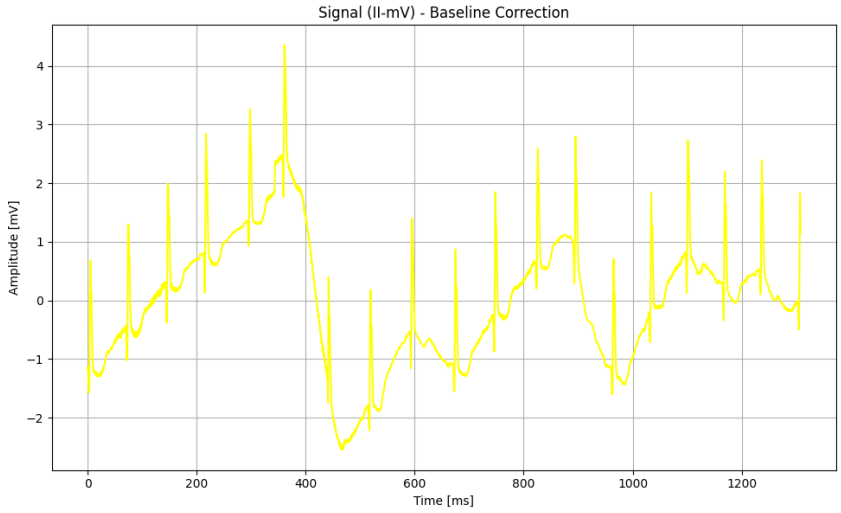


Figure 5. ECG Signal Plot After Baseline Correction.

The next stage involves applying a 4th-order Butterworth filter with a cutoff frequency of 0.6 Hz. This filter is used to eliminate high-frequency noise, such as electrical interference or muscle activity, without distorting the main signal waveform. The filtering process results in a cleaner ECG signal, with PQRST components more clearly recognizable, as shown in Figure 6.

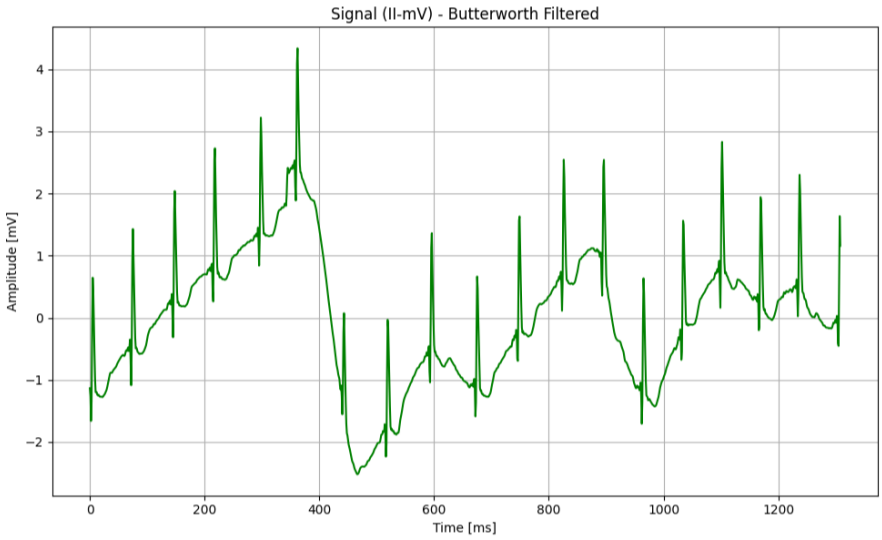


Figure 6. ECG Signal Plot After Butterworth Filtering.

The final step in the filtering process is the use of a Finite Impulse Response (FIR) filter with a Kaiser window and a cutoff frequency of 4 Hz. This filter smooths the signal comprehensively while preserving the integrity of the ECG waveform. The result of this filtering process is shown in Figure 7.

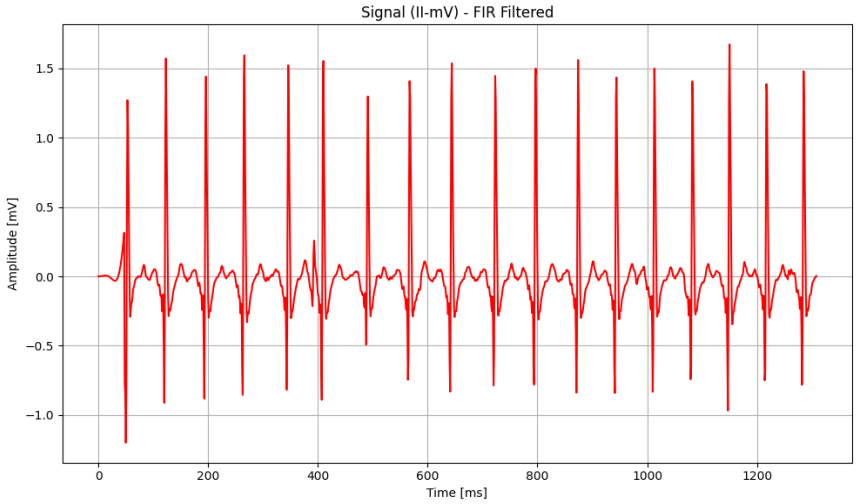


Figure 7. ECG Signal Plot After FIR Filtering.

After the filtering stage, the ECG signal was analyzed to detect the P, Q, R, S, and T wave peaks along with their onset and offset. This detection was performed using the Discrete Wavelet Transform (DWT) method, implemented via the NeuroKit2 library. The processed signal had an amplitude range more consistent with physiological standards, between -1.0 mV and 2.0 mV, with clearly marked PQRST peaks, as shown in Figure 8.

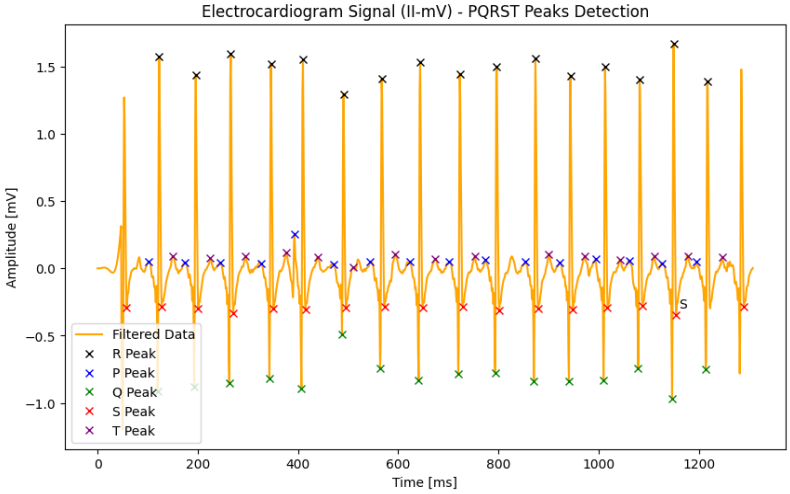


Figure 8. Final ECG Signal Plot with Detected Peaks.

Based on the peak detection results, feature extraction was performed, including time and interval parameters from the ECG signals. From each of the 665 ECG recordings in the dataset, seven numerical parameters reflecting cardiac signal characteristics were obtained: RR interval, PR interval, QT interval, QRS complex duration, ST segment, R-to-S ratio, and heart rate. The average ST interval was derived from Lead I, while RR, PR, QT intervals, QRS complex duration, and heart rate (bpm) were obtained from Lead II. Additionally, the R/S ratio was calculated from Lead V1. Each parameter was evaluated against medically recognized physiological ranges, as shown in Table 1, to determine the type of signal classification. The data were then categorized into four classes: Normal, Abnormal, Potential Arrhythmia, and High-Risk Arrhythmia.

Table 1. ECG Value Range.

Output Label	RR Interval (ms)	PR Interval (ms)	QRS Complex (ms)	QT Interval (ms)	ST Segment (ms)	R/S Ratio	Heart Beat (bpm)
Normal	600-1000	120-200	60-100	350-440	80-120	<1	60-100
Abnormal	<600	<120	<60	<350	<80	>1	<60
Potential Arrhythmia	600-1000	190-200	101-120	441-460	120-130	>1	101-110
High-Risk Arrhythmia	600-1000	>200	>120	>460	>130	>1	>111

After preprocessing, feature extraction, and labeling were completed, the final dataset showed an uneven class distribution, with 236 entries for class 0 (Normal), 312 entries for class 1 (Abnormal), and 107 entries for class 3 (High-Risk Arrhythmia). No data were found for class 2 (Potential Arrhythmia). All data were consolidated into a single CSV file, ready for use in the training and testing process, as illustrated in Table 2.

Table 2. Example of data with extracted features and labels.

Subjek	rr	pr	qs	qt	st	r/s ratio	heartrate	output
1	629.7376	115.3846	87.91209	295.1895	217.93	0.50375	95.27778	0
1	494.6667	103.5294	80.47059	221.7778	160.8889	0.482003	121.2938	1
1	511.1972	114.4551	73.41541	238.1568	173.5573	0.900144	117.3715	1
	...							
106	506.6138	87.76844	71.42857	213.4039	181.2169	7.587139	118.4334	1
106	472.0648	76.11336	78.13765	241.2955	180.7692	11.03366	127.1012	1
106	451.9531	65.37829	81.82566	243.75	177.3438	9.05171	132.7571	1

Model Design and Prediction Results

The models used in this study were developed using the Python programming language. Initially, the dataset was read using the pandas library and stored as a dataframe. The variable x was taken from all columns except the subject identification and output label columns, as only feature columns were needed as input for the models. Meanwhile, the variable y was taken from the output column representing the actual class values. The x dataset was then normalized using the StandardScaler method from the scikit-learn library, which adjusts feature values to have a mean of 0 and a standard deviation of 1. This normalization is important to enhance the learning process's effectiveness and maintain model performance stability.

Data splitting was performed using Stratified K-Fold Cross Validation with the StratifiedKFold function. This method divides the dataset into several folds of nearly equal

size while maintaining class proportions within each fold. In this study, the data were split into five folds. Choosing five folds aims to balance model evaluation reliability and computational efficiency. Each fold was alternately used as test data, while the remaining folds were used for training the model. The random state parameter was set to ensure consistent and reproducible data splitting.

To evaluate and compare the performance of CatBoost, Support Vector Machine (SVM), and Random Forest models, the Confusion Matrix was used as the primary method. The Confusion Matrix illustrates model prediction performance by comparing predicted labels with actual labels. This method is effective for assessing classification model performance, especially in multiclass classification cases, as in this study.

Additionally, several quantitative evaluation metrics were used to measure the accuracy and consistency of predictions, including Accuracy, Precision, Recall, and F1 Score. All metrics were calculated based on True Positive, False Positive, True Negative, and False Negative values, providing a comprehensive view of model classification performance.

Furthermore, the ROC-AUC (Receiver Operating Characteristic – Area Under the Curve) metric was used. ROC-AUC evaluates how well a classification model can distinguish between classes, especially in cases of imbalanced or multiclass data. ROC (Receiver Operating Characteristic) plots the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold values. TPR indicates the proportion of positive data correctly classified, while FPR indicates the proportion of negative data incorrectly classified as positive. AUC (Area Under the Curve) measures the area under the ROC curve, ranging from 0 to 1. An AUC close to 1 indicates excellent classification performance, while a value near 0.5 indicates performance equivalent to random guessing. In multiclass classification, ROC-AUC is usually calculated using the one-vs-rest approach to assess the model's ability to distinguish each class individually based on predicted probabilities.

CatBoost Prediction Results

The CatBoost model testing on the dataset demonstrated excellent predictive performance. The model achieved an accuracy of 95.45%, indicating its strong ability to classify test data with a low error rate. Precision and recall were 95.57% and 95.45%, respectively, showing that the model produced accurate predictions while recognizing samples from each class in a balanced manner. The F1 Score of 95.47% reflects a good balance between precision and recall. Additionally, the overall ROC AUC was 99.81%. This overall ROC AUC of 99.81% was further supported by class-specific ROC AUC values of 1.0 for classes 0, 1, and 3, demonstrating the model’s excellent capability in distinguishing each class.

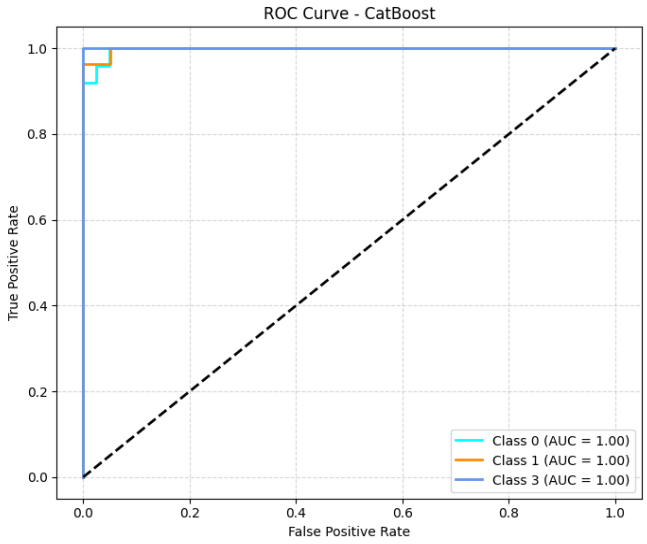


Figure 9. ROC curve of the CatBoost model.

Based on the Confusion Matrix, for class 0, 24 samples were correctly predicted, while 1 sample was misclassified as class 1. For class 1, 26 samples were correctly predicted, and 1 sample was misclassified as class 0. For class 3, 13 samples were correctly predicted, and 1 sample was misclassified as class 0.

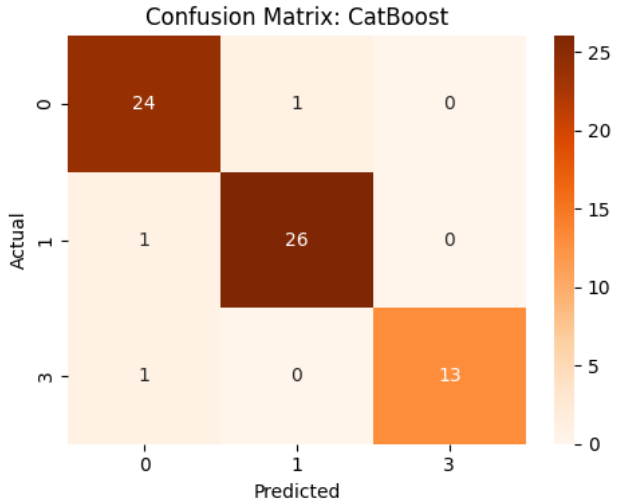


Figure 10. Confusion matrix of the CatBoost model.

Voting Prediction Results for CatBoost with SVM

The Voting Ensemble model combining CatBoost and SVM also demonstrated strong performance, achieving an accuracy of 93.94%. Precision and recall were 94.78% and 93.94%, respectively, while the F1 Score reached 93.88%, indicating balanced predictive performance. The overall ROC AUC was 99.21%, supported by class-specific ROC AUC values of 0.99 for class 0, 1.0 for class 1, and 0.99 for class 3. Although slightly lower than the standalone CatBoost model, this ensemble model still effectively distinguished between classes.

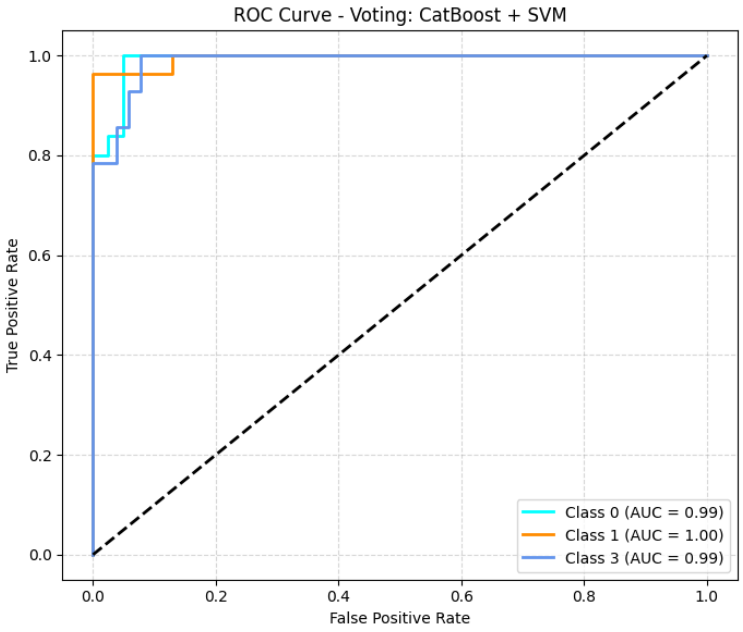


Figure 11. ROC curve of the Voting model combining CatBoost and Random Forest.

Based on the Confusion Matrix, class 0 had 25 correctly classified samples with no misclassifications. Class 1 had 26 correctly predicted samples, with 1 sample misclassified as class 0. For class 3, 11 samples were correctly predicted, while 3 samples were misclassified as class 0.

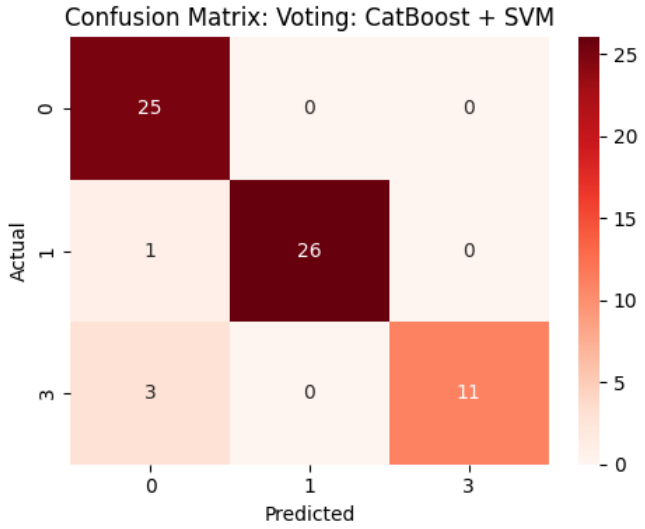


Figure 12. Confusion Matrix of the Voting model combining CatBoost and SVM.

Voting Prediction Results for CatBoost with Random Forest

The Voting Ensemble model combining CatBoost and Random Forest achieved higher performance, with an accuracy of 96.97%. Precision, recall, and F1 Score were all 96.97%, demonstrating consistent performance in classifying the data. The overall ROC AUC was 99.78%, and the per-class ROC AUC was 1.0 for all classes, indicating the model's ability to perfectly distinguish each class.

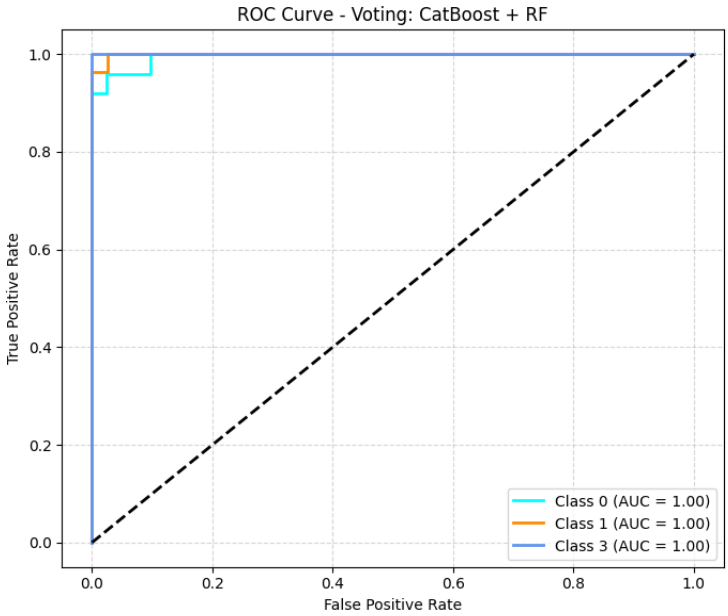


Figure 13. ROC Curve of the Voting model combining CatBoost and Random Forest.

Based on the Confusion Matrix, for class 0, 24 data points were correctly classified, and 1 data point was misclassified as class 1. For class 1, 26 data points were correctly classified, and 1 was misclassified as class 0. All data points in class 3 (14 data points) were correctly classified.

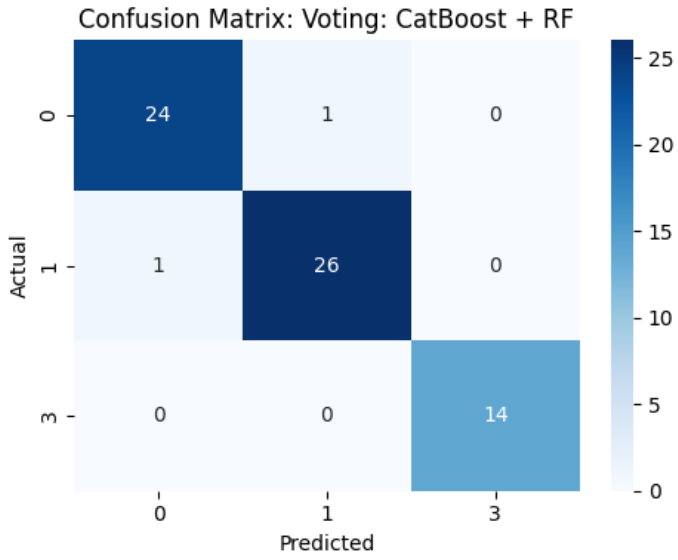


Figure 14. Confusion Matrix of the Voting model combining CatBoost and Random Forest.

Stacking Prediction Results for CatBoost with SVM

The Stacking Ensemble model with SVM as the base learner and CatBoost as the meta-learner showed slightly lower performance compared to the other ensemble methods. This model achieved an accuracy of 89.39%, with a precision of 89.45% and a recall of 89.39%. The F1 Score was 89.40%, and the ROC AUC reached 98.98%, with a per-class ROC AUC

of 0.99 for all classes, indicating that although the model still separates classes relatively well, its performance is not as accurate as the other models.

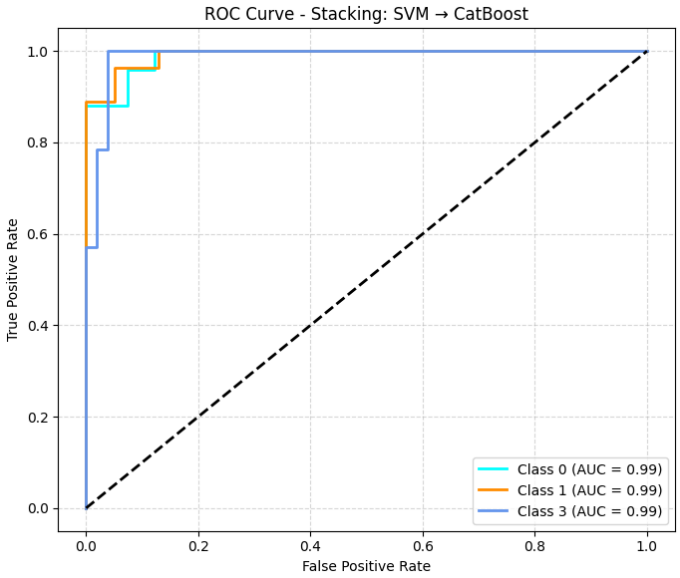


Figure 15. ROC Curve of the Stacking model with CatBoost and SVM.

Based on the Confusion Matrix, class 0 had 23 correctly predicted samples, with 1 sample misclassified as class 1 and 1 as class 3. Class 1 had 24 correctly predicted samples, while 2 samples were misclassified as class 0 and 1 as class 3. Class 3 had 12 correctly predicted samples, with 1 sample misclassified as class 0 and 1 as class 1.

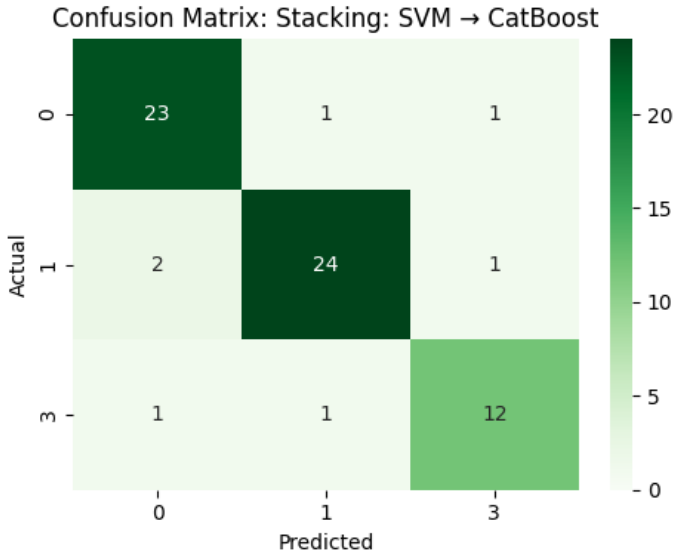


Figure 16. Confusion Matrix of the Stacking model with CatBoost and SVM.

Stacking Prediction Results for CatBoost with Random Forest

Meanwhile, the Stacking Ensemble model with Random Forest as the base learner and CatBoost as the meta-learner achieved the highest performance among all tested models. This model recorded an accuracy of 98.48%, with precision of 98.54%, recall of 98.48%, and an F1 Score of 98.48%, indicating consistent and accurate results. The ROC AUC reached

99.85%, with per-class ROC AUC values of 1.0 for all classes, demonstrating that the model was able to perfectly distinguish each class.

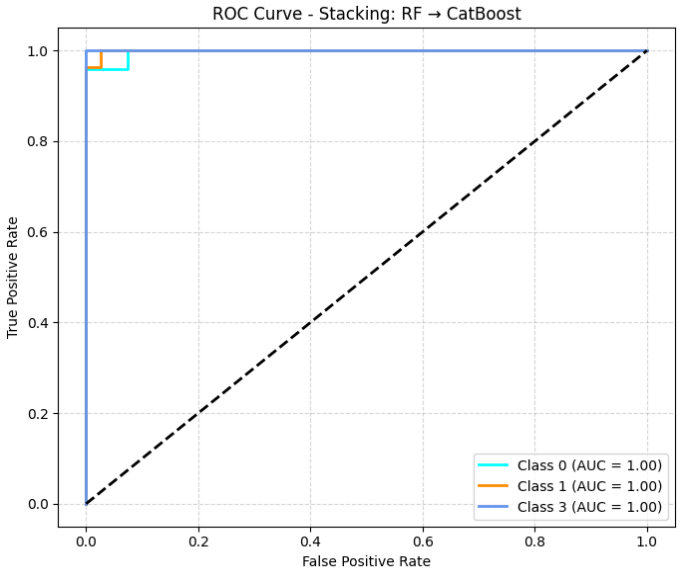


Figure 17. ROC Curve of the Stacking model with CatBoost and Random Forest.

Based on the Confusion Matrix, class 0 had 24 correctly classified samples and 1 misclassified into class 1. Class 1 was classified perfectly, with all 27 samples correctly predicted. Class 3 was also perfectly classified, with all 14 samples accurately identified.

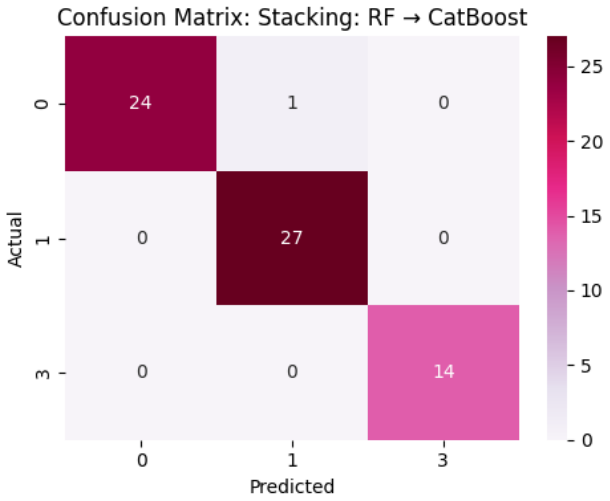


Figure 18. Confusion Matrix of the Stacking model with CatBoost and Random Forest.

DISCUSSION

Evaluation of various models for arrhythmia classification using the ECG dataset shows that the CatBoost algorithm provides strong predictive performance. The model achieved an accuracy of 95.45%, with precision, recall, and F1 score values of 95.57%, 95.45%, and 95.47%, respectively. The consistency of these metrics indicates that CatBoost as a standalone model is reliable for classification on this dataset.
The application of ensemble techniques produced varying performance. The voting method

combining CatBoost with Random Forest achieved the most optimal results, with accuracy, precision, recall, and F1 score all reaching 96.97%. This model also attained a high ROC AUC of 99.78%, demonstrating excellent overall classification capability. In comparison, the voting approach combining CatBoost with SVM showed lower performance, with an accuracy of 93.94% and an F1 score of 93.88%. Meanwhile, the stacking approach using Random Forest as the base learner and CatBoost as the meta-learner provided the best results, achieving 98.48% accuracy, 98.54% precision, and an almost perfect ROC AUC of 99.85%. Conversely, the stacking configuration with SVM as the base learner and CatBoost as the meta-learner exhibited the lowest performance among all tested models, with an accuracy of only 89.39%. Based on these results, the ensemble stacking approach (Random Forest → CatBoost) can be concluded as the most effective strategy for the classification task in this study.

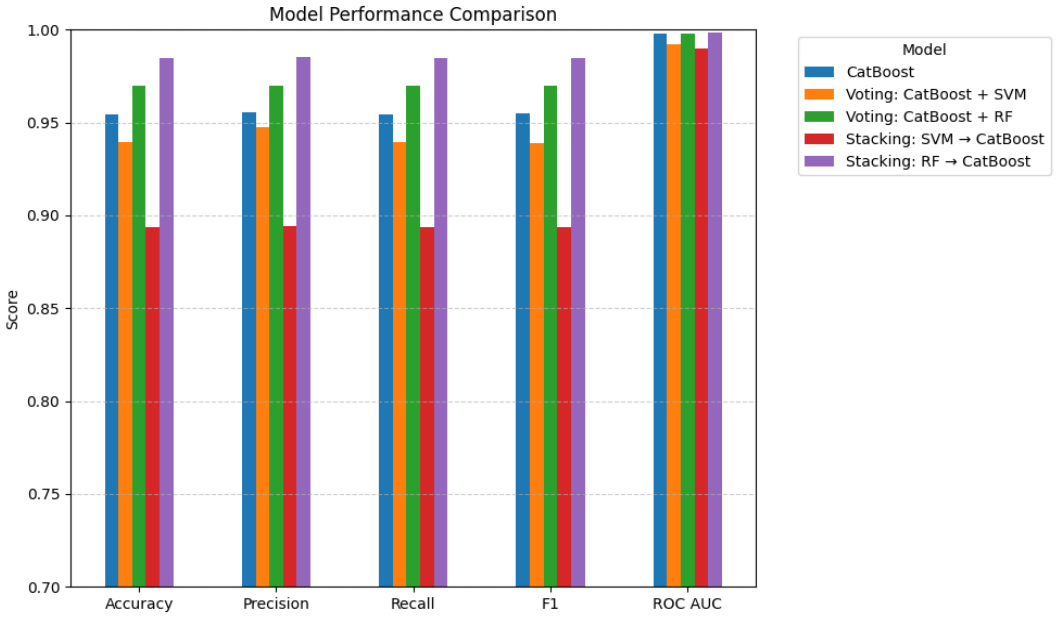


Figure 19. Comparison plot of evaluation metrics from the prediction results.

CONCLUSION

This study evaluated the performance of several machine learning models based on boosting and ensemble methods for multi-class ECG signal classification, specifically targeting four arrhythmia categories: normal, abnormal, potential arrhythmia, and high-risk arrhythmia. The models tested included CatBoost as a standalone model, as well as several ensemble approaches such as voting (CatBoost with SVM, CatBoost with Random Forest) and stacking (SVM with CatBoost, Random Forest with CatBoost). Overall, CatBoost demonstrated strong performance as a single model, with consistent evaluation metrics across accuracy, precision, recall, F1 score, and ROC AUC. This is supported by CatBoost's advantages in handling categorical features and its ability to prevent overfitting through built-in regularization. The best results were achieved using the stacking method with Random Forest as the base learner and CatBoost as the meta-learner, yielding high accuracy and near-perfect ROC AUC. This combination effectively captured the complexity of the data while enhancing prediction stability. In contrast, stacking with SVM produced the lowest performance, likely due to

SVM's limitations in handling large, non-linear datasets. Based on these findings, it can be concluded that a machine learning approach using ensemble stacking of Random Forest with CatBoost is the most effective method for ECG signal classification for detecting arrhythmia risk in adolescent students.

LIMITATION

This study has several limitations that should be considered for further development of machine learning–based ECG arrhythmia analysis. One major constraint encountered was incomplete subject data due to disturbances during field ECG recordings. This highlights the need for improved recording conditions through better coordination between researchers, participants, and school authorities. In addition, one class, "potential arrhythmia," was entirely absent in the dataset, preventing the model from being trained or evaluated on this category. Furthermore, testing the models on entirely new (unseen) data is recommended to more accurately assess their generalization capability in real-world scenarios.

REFERENCES

Nagpal, A. K., Pundkar, A., Singh, A., & Gadkari, C. (2024). Cardiac arrhythmias and their management: An in-depth review of current practices and emerging therapies. Cureus, 16(8), e66549. https://doi.org/10.7759/cureus.66549

Gajowniczek, K., Grzegorczyk, I., Ząbkowski, T., & Bajaj, C. (2020). Weighted random forests to improve arrhythmia classification. Electronics, 9(1). https://doi.org/10.3390/electronics9010099

Kumari, C. U., Murthy, A. S. D., Prasanna, B. L., Reddy, M. P., & Panigrahy, A. K. (2021). An automated detection of heart arrhythmias using machine learning technique: SVM. Materials Today: Proceedings, 45(Part 2), 1393–1398. https://doi.org/10.1016/j.matpr.2020.07.088

Behera, S. K., et al. (2025). Non-linear dynamics in ECG: a novel approach for robust classification of cardiovascular disorders. *npj Cardiovascular Health*. https://doi.org/10.1038/s44325-024-00038-2

Ma, S., Cui, J., Chen, C.-L., Chen, X., & Ma, Y. (2022). An effective data enhancement method for classification of ECG arrhythmia. Measurement, 203, 111978. https://doi.org/10.1016/j.measurement.2022.111978

Jahangir, R., Islam, M. N., Islam, M. S., et al. (2025). ECG-based heart arrhythmia classification using feature engineering and a hybrid stacked machine learning. BMC Cardiovascular Disorders, 25, 260. https://doi.org/10.1186/s12872-025-04678-9

Anshor, A. H., & Wiyatno, T. N. (2024). Comparative analysis of random forest and support vector machine for Sundanese dialect classification using speech recognition features. Sisfokom (Jurnal Sistem Informasi dan Komputer), 14(2). https://doi.org/10.32736/sisfokom.v14i2.2347

Boldini, D., Grisoni, F., Kuhn, D., et al. (2023). Practical guidelines for the use of gradient boosting for molecular property prediction. Journal of Cheminformatics, 15, 73. https://doi.org/10.1186/s13321-023-00743-7

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support (Version 1). arXiv. https://doi.org/10.48550/arXiv.1810.11363

Prasanna Venkatesh, P. V. N., Pradeep Kumar, R., Neelapu, B. C., Pal, K., & Sivaraman, J. (2023). CatBoost-based improved detection of P-wave changes in sinus rhythm and tachycardia conditions: A lead selection study. Physical and Engineering Sciences in Medicine, 46(2), 925–944. https://doi.org/10.1007/s13246-023-01274-z

Wu, J., Tao, G., Xie, S., Yang, H., Qi, F., Bao, N., Li, Z., Chang, G., & Xiao, H. (2025). Prediction of three-year all-cause mortality in patients with heart failure and atrial fibrillation using the CatBoost model. BMC Cardiovascular Disorders, 25(1), 466. https://doi.org/10.1186/s12872-025-04928-w

Schnermann, M. E., Schulz, C. A., Herder, C., Alexy, U., & Nöthlings, U. (2021). A lifestyle pattern during adolescence is associated with cardiovascular risk markers in young adults: Results from the DONALD cohort study. Journal of Nutritional Science, 10, e92. https://doi.org/10.1017/jns.2021.84

Presta, V., Guarnieri, A., Laurenti, F., Mazzei, S., Arcari, M. L., Mirandola, P., Vitale, M., Chia, M. Y. H., Condello, G., & Gobbi, G. (2024). The impact of digital devices on children's health: A systematic literature review. Journal of Functional Morphology and Kinesiology, 9(4), 236. https://doi.org/10.3390/jfmk9040236

Santini, M., Di Fusco, S. A., Colivicchi, F., & Gargaro, A. (2018). Electrocardiographic characteristics, anthropometric features, and cardiovascular risk factors in a large cohort of adolescents. Europace, 20(11), 1833–1840. https://doi.org/10.1093/europace/euy073

Luong, T. T., et al. (2021). Analysis of noisy and complex student ECG signals using ML methods. Biomedical Signal Processing and Control, 70. https://doi.org/10.1016/j.bspc.2021.103061

Kaushik, C., McRae, A. D., Davenport, M. A., & Muthukumar, V. (2023). New equivalences between interpolation and SVMs: Kernels and structured features. arXiv. https://doi.org/10.48550/arXiv.2305.02304

Kull, M., Silva Filho, T. M., & Flach, P. (2017). Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics, 11*(2), 5052–5080. https://doi.org/10.1214/17-EJS1338SI

Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random Forest algorithm overview. *Babylonian Journal of Machine Learning, 2024*, 69–79. https://doi.org/10.58496/BJML/2024/007

Zhu, J., Zhang, A., & Zheng, H. (2023). Research on predictive model based on ensemble learning. Highlights in Science, Engineering and Technology, 57, 311–319. https://doi.org/10.54097/hset.v57i.10023

Airlangga, G. (2025). Comparative analysis of voting and stacking ensemble learning for heart disease prediction: A machine learning approach. Jurnal Teknologi Informatika dan Komputer, 11(1), 1–10. https://doi.org/10.37012/jtik.v11i1.2584