Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.045975

ARTICLE

AutoRhythmAI: A Hybrid Machine and Deep Learning Approach for Automated Diagnosis of Arrhythmias

S. Jayanthi* **and S. Prasanna Devi**

Department of Computer Science and Engineering, SRM Institute of Science and Technology—Vadapalani Campus, Chennai, Tamil Nadu, India

*Corresponding Author: S. Jayanthi. Email: js8683@srmist.edu.in

Received: 13 September 2023 Accepted: 27 October 2023 Published: 27 February 2024

ABSTRACT

In healthcare, the persistent challenge of arrhythmias, a leading cause of global mortality, has sparked extensive research into the automation of detection using machine learning (ML) algorithms. However, traditional ML and AutoML approaches have revealed their limitations, notably regarding feature generalization and automation efficiency. This glaring research gap has motivated the development of AutoRhythmAI, an innovative solution that integrates both machine and deep learning to revolutionize the diagnosis of arrhythmias. Our approach encompasses two distinct pipelines tailored for binary-class and multi-class arrhythmia detection, effectively bridging the gap between data preprocessing and model selection. To validate our system, we have rigorously tested AutoRhythmAI using a multimodal dataset, surpassing the accuracy achieved using a single dataset and underscoring the robustness of our methodology. In the first pipeline, we employ signal filtering and ML algorithms for preprocessing, followed by data balancing and split for training. The second pipeline is dedicated to feature extraction and classification, utilizing deep learning models. Notably, we introduce the 'RRI-convoluted transformer model' as a novel addition for binary-class arrhythmias. An ensemble-based approach then amalgamates all models, considering their respective weights, resulting in an optimal model pipeline. In our study, the VGGRes Model achieved impressive results in multi-class arrhythmia detection, with an accuracy of 97.39% and firm performance in precision (82.13%), recall (31.91%), and F1-score (82.61%). In the binary-class task, the proposed model achieved an outstanding accuracy of 96.60%. These results highlight the effectiveness of our approach in improving arrhythmia detection, with notably high accuracy and well-balanced performance metrics.

KEYWORDS

Automated machine learning; neural networks; deep learning; arrhythmias

1 Introduction

Arrhythmias, abnormal heart rhythms ranging from benign to life-threatening, can be triggered by factors like undiagnosed cardiac disease, adverse drug reactions, and metabolic abnormalities [1]. Analyzing the ECG data for heart electrical activity anomalies is crucial for arrhythmia detection. Traditional methods involve continuous manual monitoring of ECG signals, but researchers firmly believe that leveraging AI-guided ECGs to diagnose conditions like atrial fibrillation has several benefits. According to the World Health Organization (WHO) [2], cardiovascular diseases, including arrhythmias, account for approximately 17.9 million deaths annually, representing 31% of all global deaths as shown in Fig. 1. These alarming statistics underscore the urgency of developing efficient and accurate arrhythmia diagnosis methods. To determine the applicability of AI-ECG procedures in various clinical contexts and patient demographics, researchers are planning multicenter hybrid experiments that incorporate AI algorithms, such as machine learning and deep learning models [3,4]. Swift and accurate arrhythmia diagnosis requires powerful automatic algorithms, and AI promises to fulfill this critical need.

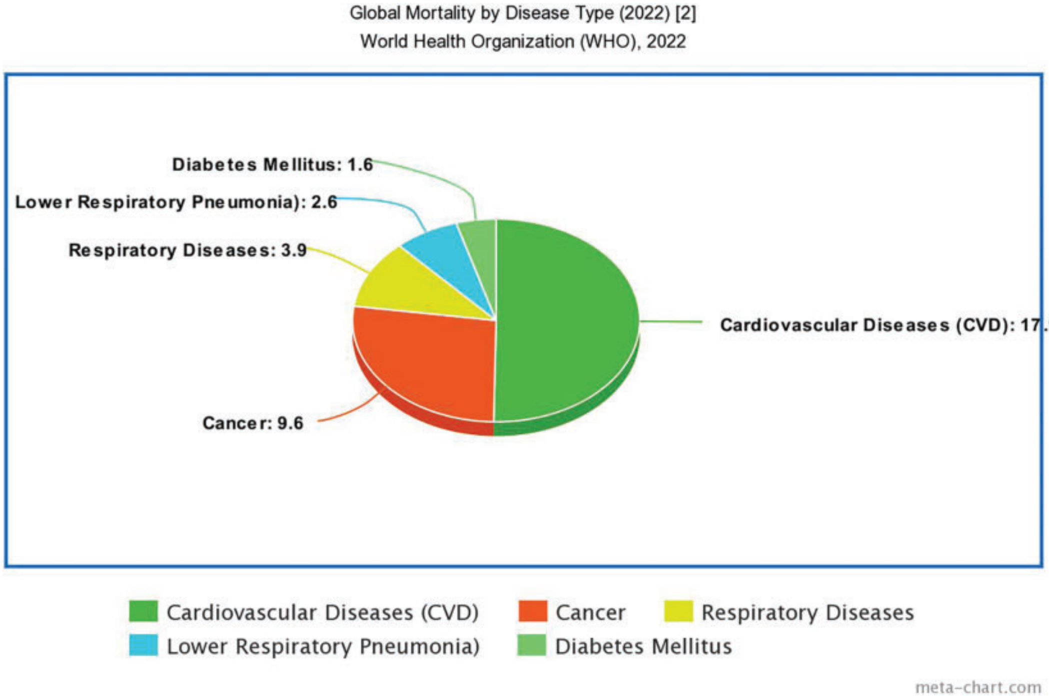


Figure 1: Statistics report by WHO

AutoML is becoming increasingly popular because it can drastically save the time and expense needed to create machine-learning models while increasing their efficacy and accuracy [5]. Electrocardiogram (ECG) signal analysis using AutoML has attracted a lot of interest because it can potentially increase the precision and effectiveness of this type of analysis [6]. Research strongly suggests the Auto-ML techniques for automation and ease of quick detection of disease diagnosis [7]. However, those techniques have the following issues. The issues include:

1. Low availability of ECG datasets: Strict regulations govern the access and distribution of the data due to its sensitive nature. Because of this, creating and testing auto-ML models with large, varied ECG datasets may be challenging [8].

2. Managing noisy ECG signals: Movement artifacts, electrode impedance, and electromagnetic interference are some conditions that might cause noisy ECG signals [9]. As a result, utilizing auto-ML models to recognize and categorize ECG characteristics may be challenging. Research is required to create reliable autoML models that handle noisy ECG readings.

3. Powerful automation algorithms: Several ML and deep learning algorithms are available. AutoML supports several automation tools, such as Auto-Sklearn [10], TPOTS [11], and Auto-Keras [12]. From an ECG research point of view, these tools must provide robustness and be more accurate. The AutoML framework initiates with the first step of pre-processing, wherein user input facilitates the conversion of categorical data into integers (e.g., via a label encoder) [13]. Nonetheless, the autoML pipeline necessitates the incorporation of multiple pre-processing tasks. Auto-Keras frameworks do not support generalized feature extraction. So, a separate ECG auto-ML tool is needed to design powerful automation algorithms for processing and classification. Our study aims to collect generic ECG features using extraction and pre-processing techniques to eliminate noisy signals.

This work aims at detecting automatic heart disease by both binary and multi class classification of arrhythmias on ECG using proposed AutoRhythmAI techniques.

1. Focus on multimodal ECG datasets, enhancing AutoML algorithms' robustness to noise in ECG Signals by incorporating various data types, improving diagnostic accuracy.

2. Introduction of a custom AutoRhythmAI model pipeline, featuring advanced deep learning models for both binary and multi-class arrhythmia detection.

3. Introduction of a novel ensemble method combining multiple deep learning models, such as ResIncept, InceptVGG, VGGRes, and LenetAlexLSTM, to achieve accurate multi-class arrhythmia diagnosis and a novelty hybrid RRI Peak detection + CNN + Transformer model for binary arrhythmia detection.

The paper provides an overview as follows: Section 2 presents the pertinent literature, while Section 3 entails a comprehensive exploration of detailed methodology, discussion of multi-model databases, comparisons, analyses of study groups, elucidation of critical attributes, and examination of research challenges. Section 4 outlines the outcomes and their synthesis, while Section 5 encapsulates the study's discussion and summarizes its prospects.

2 Related Work

Signal Processing Techniques: ECG data generally contains noise that requires elimination. Various signal processing techniques form the foundation for reducing noise, extending beyond elimination to serve as feature extraction methods. Filtering, the first and most prevalent technique [14], removes noise and artifacts from ECG signals to enhance signal quality. Standard filters encompass high-pass, low-pass, and notch filters. Feature extraction involves deriving pertinent attributes from ECG signals for arrhythmia detection [15]. Standard features contain RR intervals, QRS complex duration, and ST segment deviation. Wavelet analysis decomposes ECG signals into wavelet coefficients at different scales [16] for feature extraction, denoising, and compression. Time-frequency analysis scrutinizes ECG signals across both time and frequency domains, utilizing methods like the short-time Fourier transform and the continuous wavelet transform. Time-frequency analysis assists in feature extraction and denoising of ECG signals. Primary AI techniques using ML algorithms [17] independently acquire the ability to learn and classify ECG signals into various arrhythmia classes. Standard ML algorithms encompass support vector machines, decision trees, and neural networks. Hence, filtering and feature extraction techniques enhance the diagnostic accuracy of ECG arrhythmia detection. The resulting signal becomes more straightforward for analysis, extracting relevant features. These techniques offer continuous surveillance of patients with arrhythmias for real-time ECG signal monitoring, aiding in real-time arrhythmia detection and timely intervention.

Deep Learning Techniques: Deep learning Models are part of the machine learning models mainly used to automate and detect the features [18,19] and classify them [20] accordingly. Deep learning techniques have shown great potential in ECG signal classification. Several algorithms support the ECG classification. Some of them are CNN, RNN, autoencoders, and GAN [21]. Hybrid deep learning algorithms offer several advantages over single deep learning techniques, such as improved accuracy, reduced overfitting, transfer learning, and scalability. Those algorithms include the Convolutional Neural Network—Recurrent Neural Network (CNN—RNN), the Convolutional Neural Network—Long Short-Term Memory (CNN—LSTM), the Residual Neural Network—Long Short-Term Memory (ResNet—LSTM) [22], and so on. The choice of a particular hybrid algorithm depends on the specific requirements of the ECG signal classification task at hand. Furthermore, there is a lack of time series-based deep learning algorithms for continuous monitoring of ECG signals. Hybrid models typically have many parameters, which can lead to overfitting and reduced generalization performance, especially when dealing with small datasets.

AutoML Techniques: AutoML for ECG classification aims to automate the machine learning pipeline, including feature engineering, algorithm selection, hyperparameter tuning, and model ensembling. In reference [23], Auto-Keras, TPOT, Auto-Sklearn, and H2O commonly use predefined auto-ML tools. They automate feature engineering, model selection, and hyperparameter tuning to enhance accuracy in arrhythmia detection from ECG data. Researchers also employ AutoML for signal processing, ensemble building, transfer learning, hyperparameter optimization, and addressing class imbalance. These techniques streamline arrhythmia diagnosis and contribute to cardiac health monitoring.

The Table 1 states the so far techniques used for ECG classification for arrhythmias and their gaps are tabulated.

Table 1: Techniques used in ECG classification and its limitations

Techniques	Ref. No.	Methods	Advantages	Limitations	Research gaps
Signal processing techniques	[24]	ML algorithm: PCA for R peak detection	PCA provides more accurate and robust results than traditional peak detection algorithms.	One limitation of the paper is that it only evaluates the proposed PCA-based R-peak detection method on a single dataset.	Studies on a broader range of datasets may be needed to fully evaluate the effectiveness of the proposed method.

(Continued)

Table 1 (continued)

Techniques	Ref. No.	Methods	Advantages	Limitations	Research gaps
	[25]	Filter technique: Matched filter Feature Extraction: CNN	The proposed approach uses a matched filter to pre-process ECG signals and extract informative features related to arrhythmia patterns.	The paper does not provide a detailed analysis of the features learned by the CNN or how they relate to arrhythmia patterns.	Enhanced algorithms could help improve the interpretability of the proposed approach.
Deep learning techniques	[26]	CNN RNN FCNN	CNN,RNN and FCNN are trained separately and then combined for final classification.	Do not provide a detailed analysis of the features learned by the CNN or how they relate to arrhythmia.	Detailed patterns that need to be analyzed could help improve the interpretability of the proposed approach.
	[27]	ResNet-50 model	The proposed approach on a large dataset of 12-lead ECGs and report high accuracy in ECG diagnosis, including classification of arrhythmias and detection of ischemic heart disease.	ResNet-50 model parameters are missing for ECG patterns analysis.	ResNet-50 may increase the computational complexity for real-time applications.

(Continued)

Table 1 (continued)

Techniques	Ref. No.	Methods	Advantages	Limitations	Research gaps
Auto-ML techniques	[28]	Auto-ML tools	Stated the usage of auto-ML for large medical dataset and its risk of overfitting.	Potential risks and ethical considerations associated with the use of machine learning in healthcare and laboratory medicine.	For the multi-modal dataset auto-Ml supports the generalization of the proposed model.
	[29]	Auto-keras, TPOT, and H2O.ai	Used butterworth filter for pre-processing and then classified different ECG leads.	The need for high-quality labeled data.	Does not ensure interpretability of the resulting models.

Researchers have given limited focus to auto-ML techniques for ECG classification. Since auto-ML is a hot topic in research and arrhythmia detection plays a vital role in the survival of human health. The combination of these will give a platform for the development of AI techniques.

3 Materials and Methods

3.1 Multi Modal ECG Dataset

A multimodal ECG dataset collects electrocardiogram (ECG) data that includes additional modalities or signals, such as respiratory signals or blood pressure. These different signals can provide complementary information that can improve the accuracy of ECG analysis, such as detecting arrhythmias, identifying cardiac abnormalities, or monitoring vital signs during medical procedures. Multimodal ECG datasets are often publicly available, which allows researchers and developers to benchmark their algorithms against existing methods and collaborate with other experts in the field. In this research, we utilized datasets such as the China Physiological Signal Challenge 2018 (CPSC2018), the St. Petersburg INCART 12-lead Arrhythmia Database, the Georgia 12-lead ECG Challenge Database (CinC2020), the PhysioNet Computing in Cardiology Challenge 2017 (CinC2017), and the Contextual Arrhythmia Database (CACHET-CADB) [30] as shown in Fig. 2. The above databases have two data formats: header files (.hea) and mat files (.mat). The mat file consists of patient ECG signals. The header file consists of 12 lead signal values and their diagnostic values. Table 2 shows the summarization of the multimodal ECG database used.

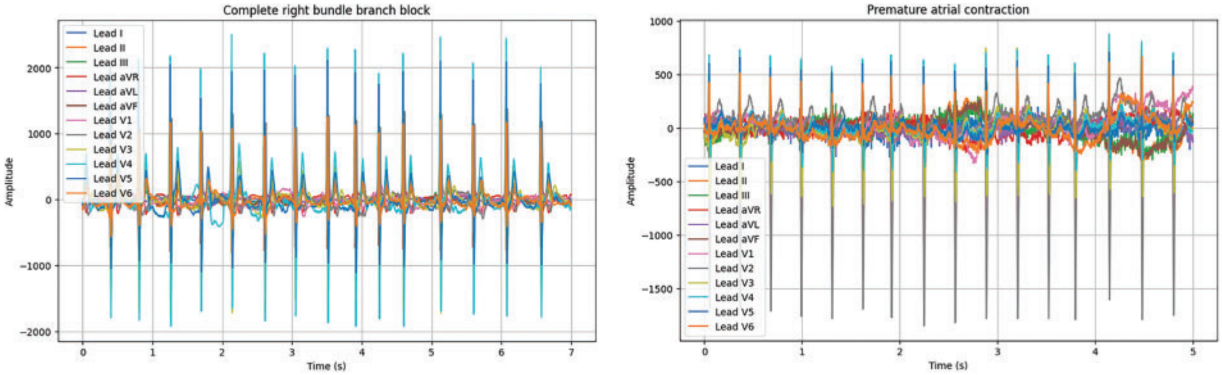


Figure 2: Different types of arrhythmias signals used (X-axis = time Y-axis = amplitude)

Table 2: Summarization of multimodal ECG database

Dataset name	Number of subjects	Number of female (F) and male (M) subjects	Length of ECG signals recorded (in seconds (s))
CPSC2018	6877	M: 3699 F: 3178	6–60 s
St. Petersburg INCART	32	M: 17 F: 15	1800 s
Georgia	10,344	M: 5551 F: 4793	10 s
CACHET-CADB	24	M:15 F:9	10 s (till 24h)

3.2 Multimodal Dataset Collaboration and Data Balancing

This section focuses on merging multiple datasets while retaining all data attributes. We initialize empty lists for data, labels, filenames, genders, and ages. For each dataset, we load it separately and append its data and associated information to their respective lists. We concatenate these lists into single NumPy arrays, ensuring the correct order. The resulting merged dataset retains all the original data and labels. Regarding data balancing, we extracted approximately 3,252 abnormal and 2,568 standard signals from each dataset, rendering additional data-balancing techniques unnecessary as the dataset already maintains a balanced class distribution.

3.3 Custom AutoRhythmAI Model for ECG Arrhythmias

AutoML capabilities are poised to take the first step towards AI standards by delivering ML with the push of a button or, at the very least, ensuring that they conceal algorithm execution, data pipelines, and code from view [31]. As a result, it provides a topic for research proposal methodology. As AutoML is a framework still under development for automating tasks, we cannot build a hybrid model for automation based on our own ECG customization. The proposed Custom AutoRhythmAI steps, which suit ECG arrhythmias detection for multi-model databases and hybrid model automation, are discussed below. Fig. 3 illustrates the working pipeline of the designed AutoML.

Steps: AutoRhythmAI model for ECG arrhythmias

Step 1: Import Python packages such as pandas, NumPy, and so on for computation and processing of the dataset.

Step 2: Set the parameters required to understand the problem and computations.

Step 3: Design the preprocessed pipelines for processing the ECG signals. In this model, there are two preprocessed pipelines based on filtering techniques.

Step 4: Construct the class for feature extraction and classification—one class for multi-class data models and another type for binary class data models.

Level 1: Multi-Class Model: Create a multi-class class on one pipeline. We have used algorithms such as ANN, ResNet-50, Lenet-5, AlexNet, VGG-16, Inception, LSTM, ResIncept, InceptVGG, VGGRes, and LenetAlexLSTM model on that pipeline. This class processes all the test data, yielding the output of the best model for detection.

Level 2: Binary class Model: Create a Binary class on another pipeline. The novelty algorithm called "RRI-Convoluted Transformer Model" is on that pipeline, which takes the RRI features along with the highlighted CNN features for classification. Since the ECG signals are time series data, a transformer model is used.

Step 5: Train the custom model on the training set using the defined loss function and optimization algorithm.

Step 6: Evaluate the performance of the hybrid model on the validation set.

Step 7: Calculate the test set's performance metrics, such as accuracy, sensitivity, and specificity.

So, even for non-data science experts, we may get good accuracy at the end of these procedural steps with minimal computing time.

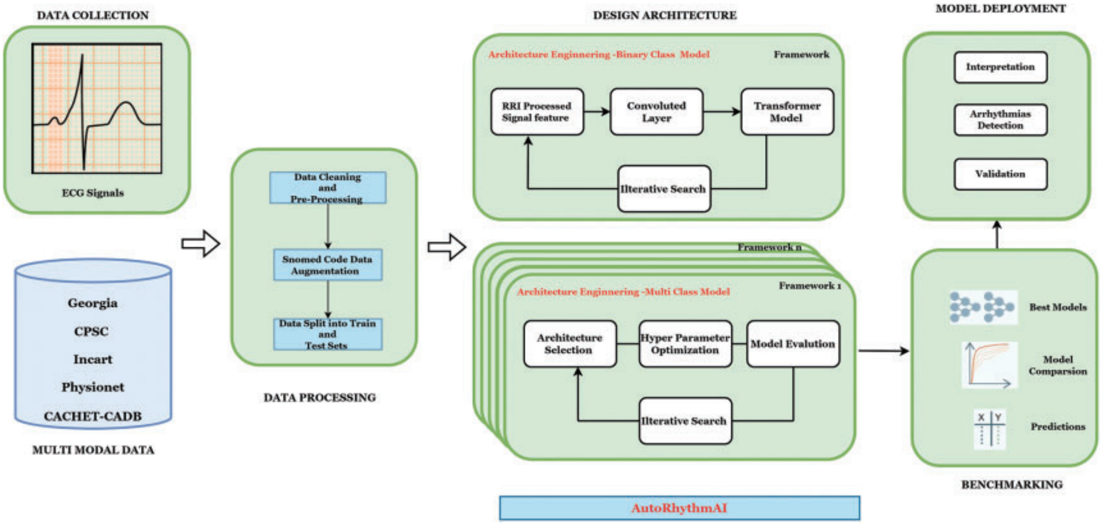


Figure 3: Proposed model architecture

3.4 Pre-Processing Levels

In the existing research, auto-ML pre-processing [32] includes label encoders, one-hot encoding, target encoders, compression technique [33], etc. Some techniques will only be suitable for some ECG processing, so the traditional AutoML will not work for our applications. Thus, the research aims to develop filtering technique pipelines to eliminate signal noise. In this context, we categorize the pre-processing stages of channels into binary class and multi-class classifications. We apply the bandpass filtering technique for binary-type pre-processing to eliminate signal noise and artifacts. The primary purpose of using bandpass is to consolidate the lowpass and high-pass filter values as described in Pseudocode 1 and this act as base combining signals with different sample sizes.

On the other hand, the multi-class pre-processing technique has 27 distribution classes. So, we ensured the signal length and performed labeling of the data. After that, we used machine learning techniques named one hot encoding with a function called Multilabel binarize to process the data as described in Pseudocode 2.

Pseudocode 1: Binary Class Pre-Processing Technique.

```
    1. Set the signal length range as between 6 and 60 s.
    2. Define a function to normalise the signal samples to help with deep learning.
    3. Truncate the length of the signal to 10 s.
    4. Apply a bandpass filter (FIR filter) to retain frequencies between 0.5 and 40 Hz.
    5. Resample individual truncated segments to 300 Hz.
    6. Convert the resampled signal data points into a one-dimensional array.
    7. Normalise the signals to a range of 0 and 1.
    8. Plot the preprocessed signals using a graph, where the x-axis represents the sampling rate of
300 Hz, and the y-axis shows the voltage difference.
    9. Distinguish normal and abnormal arrhythmia signals based on the signal pattern on the x-axis,
rather than the amplitude on the y-axis.
    10. Train a deep learning model using the preprocessed signals as input data.
```

Pseudocode 2: Multi-Class Pre-Processing Technique—signal length and one hot encoding.

```
Label Transformation:
1. Define a function named make_undefined_class that takes labels and df_unscored as parameters.
2. Create a Pandas DataFrame called df_labels from the labels. Iterate through it, replacing all
occurrences of an element with “undefined class” using replace (). Return df_labels.
One-Hot Encoding:
1. Define a function named onehot_encode that takes df_labels as a parameter.
2. Create an instance of MultiLabelBinarizer () called one_hot.
3. Use fit_transform () on one_hot to one-hot encode df_labels by splitting them using commas.
4. Assign the result to a variable called y.
5. Print “The classes we will look at are encoded as SNOMED CT codes:” followed by
one_hot.classes_.
6. Return y and an array of one_hot.classes_ except the last element.
```

3.5 Validation Schema

In health analysis data and AutoML, the traditional IID assumption may not hold due to rapidly changing ecosystem activities. This requires alternative validation methods like K-fold cross-validation or custom splits. In this case, we employed a 10-fold stratified cross-validation on a dataset of 52,369 ECGs. Custom splits can also be used based on the order of data and signal sequences for training and validation. These methods are crucial for accurate model performance estimation and hyperparameter tuning in AutoML.

3.6 Feature Selection and Classification

Feature selection or Feature extraction [34] is an essential step in any machine learning model, including deep learning models. In deep learning, feature selection selects the most relevant features from a dataset to improve the model's performance and efficiency. Deep learning models can easily overfit the training data if trained on too many irrelevant or redundant features. Feature selection helps to reduce overfitting by selecting only the most informative features and removing unrelated data, which improves the model's ability to generalize to new data. In this work, we have two types of feature processing levels.

Level 1 Feature selection for Multi-class ECG data: In our comprehensive pipeline, we have harnessed a diverse array of algorithms, ranging from traditional machine learning approaches such as the Artificial Neural Network (ANN) to cutting-edge deep learning architectures like ResNet-50, Lenet-5, AlexNet, VGG-16, Inception, and LSTM. Moreover, we've ingeniously combined these models into hybrid configurations, such as the ResIncept Model (ResNet-50 + Inception), InceptVGG Model (Inception + VGG-16), VGGRes Model (VGG-16 + ResNet-50), and the LenetAlexLSTM Model (Lenet-5 + AlexNet + LSTM) as described on algorithm [1]. This multi-faceted approach enables us to harness the strengths of diverse algorithms simultaneously, resulting in enhanced performance, adaptability, and resilience to tackle complex tasks effectively. We will simultaneously pass each pre-processed input through these algorithms, merge the outcomes of those data, and evaluate the best-sorted features.

Level 2 Feature selection for Binary-class ECG data: For Binary class ECG data classification, we introduced the novel hybrid combination approach named "Novel RRI Convoluted Transformer model" for time series ECG signal classification. Fig. 4 shows the proposed methodology, which includes the main stages of detection of RR features, feature extraction, and feature classification using the newly proposed deep learning techniques.

Algorithm 1: Proposed Ensemble Model

```
Input: models (list of pre-trained models), input_shape, learning_rate
Output: ensemble_model
Start Algorithm
    model = Sequential ()
    model_outputs = []
    model_input = tf.keras.Input(shape=input_shape)
    For (each model in models) do
    model_output = model(model_input)
    model_outputs.Append(model_output)
    ensemble_output = tf.keras.layers.Average()(model_outputs)
    ensemble_model = tf.keras.Model(inputs=model_input, outputs=ensemble_output)
Phase 1, Compile the Model
    opt = keras.optimizers.Adam(learning_rate=learning_rate)
    Parameters:
      - Loss: 'sparse_categorical_crossentropy'
      - Metrics: ['accuracy']
      - Optimizer: opt
Phase 2, Model Fit
```

(Continued)

Algorithm 1 (continued)

```
Fit the ensemble model to the training data
   ensemble_model.fit(
      training_data [0],   # Training input data
      training_data [1],   # Training labels
      batch_size= 64,      # Batch size
      epochs=100           # Number of training epochs
   )
ensemble_model.summary()
```

Phase 3, Test Evalution

```
evaluation_results = ensemble_model.evaluate(
      test_data [0],       # Test input data
      test_data [1]        # Test labels
   )
Return the evaluation_results as the   predicted output
End ; /   Algorithm
```

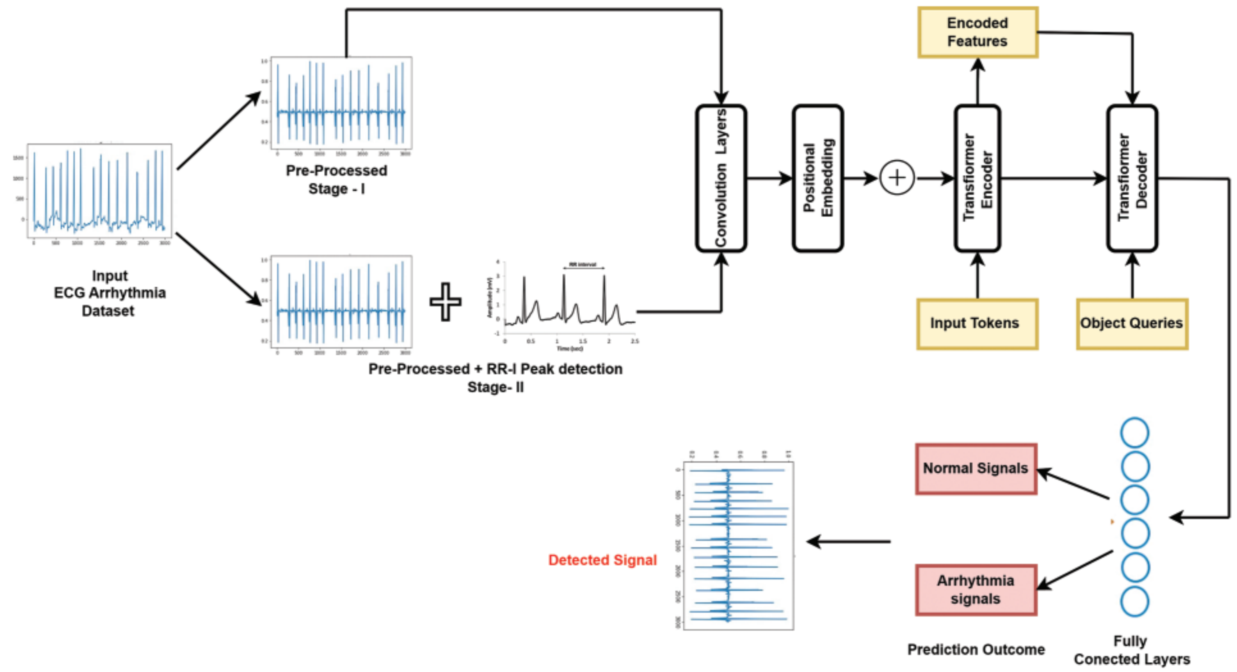


Figure 4: Working model of simulated deep convolutional transformer model

We have shown the variation in detecting the RRI peak values and using raw signals on this model. The purpose of this proposed model is to capture the local and global temporal features of ECG signals. The model receives a 1D ECG signal as input and processes it through convolutional layers. These convolutional layers employ filters of different sizes to capture various time-scale features. Following the convolutional layers, the model guides the output through a sequence of transformer blocks. These transformer blocks consist of self-attention and feed-forward layers, allowing the model to grasp dependencies between signal parts and acquire higher-level features. Overall, the convolutional transformer model is a robust architecture for ECG signal classification, as it can effectively capture both local and global temporal features and has shown promising results in various studies. The generalized features that are extracted are described in Table 3. The layers and their corresponding equations are described in Table 4.

Table 3: Feature extracted components

Component	Description
Input	RRI signals (R-R interval signals)
Feature extraction	Convolutional neural network (CNN) layers for local pattern extraction and transformer architecture for capturing long-range dependencies
Features extracted	Heart rate variability metrics, temporal patterns, and statistical properties of RRI intervals
Model layers	Multiple layers combining CNN and transformers layers
Output	Predictions or classifications related to arrhythmia detection

Table 4: RRI Convoluted transformer model layers and its descriptions

Layers and components	Equations	Description
Embedding	$E_i = \text{Embedding}(X_i)$ X_i = Input ECG sequences	$E \in \mathbb{R}^{(T \times d)}$ where d is the embedding dimension
Positional encoding	$PE_{(pos,\, i)} = \sin(pos/10000^\wedge(2 * i/d))$ for even $PE_{(pos,\, i)} = \cos(pos/10000^\wedge(2 * (i-1)/d))$ for odd	$PE_{(pos,\, i)}$ represents the (i,j)-th element of the positional encoding matrix
Convolutional layer	$H = \text{Convolution}\ (E_i + PE_{(pos,\, i)})$	The convolutional layers, typically used for feature extraction, process the ECG sequence to capture relevant patterns
Multihead attention layer	$Q = HW\_Q$, $K = HW\_K$, $V = HW\_V$,	Q‘, ‘K‘, and ‘V‘ are the query, key, and value matrices, respectively W_Q‘, ‘W_K‘, and ‘W_V‘ are weight matrices for projecting the input into query, key, and value spaces

(Continued)

Table 4 (continued)

Layers and components	Equations	Description
Attention score	$A_h = \text{softmax}((Q_h * K_h$^T)/sqrt(d))	The A_h is calculated by taking the softmax of the scaled dot product between the query and key vectors, divided by the square root of the dimensionality (d)
Weighted sum	$V_h = A_h * V_h$	The V_h is computed by element-wise multiplying the attention scores A_h with the value vectors V_h
MultiHeadOutput	MultiHeadOutput = Concatenate(V_1, V_2,..., V_H) * W_O	Concatenating the output value vectors from multiple attention heads (V1, V2, ..., VH)
Position-wise feedforward networks	FFN_output = ReLU(FFN(W_1 * MultiHeadOutput + b_1) * W_2 + b_2)	The feedforward network typically consists of two linear layers with a ReLU activation
Residual connections and layer normalization	Output1 = LayerNorm(MultiHeadOutput + Sublayer(MultiHeadOutput))	Residual connections are used to add the original input to the output of each sub-layer before applying layer normalization
Layer stacking	Output = self.self_att_pool(Output1)‘	Multiple layers of these components are stacked sequentially to form the complete TransformerEncoder

3.7 Hyper-Parameter Used

Hyperparameters directly impact the performance of a model [35]. The choice of hyperparameters for the ECG signal processing model was made based on a combination of domain knowledge and experimentation for both classifications. For binary class system a window size of 10 s was selected to capture sufficient ECG data. A batch size of 64 balances computational efficiency and convergence. A learning rate of 0.0001 ensures gradual convergence during training. Two convolutional layers capture essential features, and an embedding dimension of 64 provides sufficient representation capacity. Four encoder and decoder layers handle sequence dependencies, with four attention heads capturing different aspects of the input. A feed-forward dimension of 256 balances model complexity, and a dropout rate of 0.25 regularizes the model. The log_interval of 1 indicates logging after each training step. These values aim to optimize model performance for ECG analysis.

3.8 Ensembling

Ensemble methods for ECG signals merge all used deep learning models to predict [36]. This work employs blending, which utilizes weighted averages of the best features from proposed algorithms to create a blender model. The innovative training model, incorporating algorithmic elements, is seamlessly integrated into the proposed framework.

3.9 Experimental Setup

The suggested model in this research uses the deep learning libraries NumPy Community, Mathematics, and Seaborn for implementation on the Python backend. With the assistance of Kaggle Respiratory, the model was trained. The model is trained on the Tesla P100 GPU in a Kaggle cloud computing environment because the data set is more than 1.5 GB.The raw ECG data were scaled in the 0–1 range before normalization. Using the WandB cloud service, the test data was analyzed. WandB is a centralized interface for tracking and exchanging measurements, forecasts, and hyperparameters from our models in a competitive real-time setting. Test findings like sensitivity, specificity, epochs, and accuracy are available through the WandB service.

3.10 Evaluation

1. The Adam optimization technique [37] will choose the optimum parameters during model search and training and evaluate the best dropout strategy and model performance. The AdamW Optimizer improves on approaches like momentum and root-mean-square propagation by using their strengths to provide a more efficient gradient descent. This implies that we set AdamW with a weight decay of 1e-6 for this proposed model, and (0.9, 0.98) is configured to optimize the simulated model. The optimized model is initialized with a StepLR scheduler at 0.95.

2. The AutoRhythmAI model framework will execute a few operations during the experiment. The optimization strategy is based on hyperparameters, dropout, learning rate, and performance metrics.

3.The performance metrics of this model are evaluated using AUC, accuracy, recall, and precision metrics.

4 Empirical Results and Discussion

We are posing the following empirical queries after describing the experiment in full:

Question 1: What is the impact of separate preprocessing over arrhythmia detection in ECG signals?

Question 2: Which Deep Learning method outperforms the benchmark compared to the fundamental approach, enabling the selection of the best base model for further testing?

Question 3: Which category of features, like raw signals convoluted features or RRI convoluted features in binary class, have the best impact?

Thus, address questions 1 through 3 in this section by presenting and analyzing your experimental results.

4.1 Results

Question 1: What is the impact of separate preprocessing over arrhythmia detection in ECG signals?

Table 5 presents the performance metrics such as accuracy, precision, recall, AUC score, and Loss [38] based on different groups of preprocessing, which we evaluated after training the models. It demonstrates that the AutoRhythmAI framework automates the evaluation of the model using Adam optimization. Notably, the VGGRes Model emerges as the top performer, boasting the highest accuracy of 97.39% and vital precision, recall, and AUC score, making it a robust choice for accurate diagnostics. Additionally, the Inception model demonstrates remarkable results with an accuracy of 96.39% and high precision and recall values. These findings underscore the effectiveness of deep learning architectures, especially the VGGRes Model, in achieving superior performance in ECG signal classification tasks. We plotted a boxplot chart to compare the model's overall performance in accuracy, precision, and AUC values for binary class distribution. The RRI preprocessed signals work well on the proposed model, with a model accuracy of 96.8%, which can be inferred from the Fig. 5, and concluded that preprocessing techniques used for different classifications in the framework achieved higher performance in detecting the arrhythmias.

Table 5: Multi-class arrhythmias detection model performance metrics

Models	Accuracy (%)	Precision (%)	Recall (%)	AUC score (%)	Loss (%)
ANN	95.50	81.27	17.44	73.60	14.20
Lenet-5	95.48	75.12	20.61	76.85	13.70
AlexNet	95.73	79.11	24.15	78.67	12.66
VGG-16	96.07	83.57	30.90	81.61	11.731
ResNet-50	95.96	78.66	28.04	81.07	11.97
Inception	96.39	88.18	35.54	86.54	10.48
LSTM	95.85	84.46	24.90	77.44	13.09
ResIncept model	96.40	78.61	28.04	81.07	11.97
VGGRes model	97.39	82.13	31.91	82.61	10.12
InceptVGG	96.86	89.18	38.45	88.12	10.88
LenetAlexLSTM	95.89	80.12	21.61	79.85	12.70

Question 2: Which Deep Learning method outperforms benchmarking when compared to the fundamental approach, allowing the best base model to be chosen for additional testing?

For multi-class:

Table 6 shows that among the various machine learning approaches and ensemble methods evaluated, the proposed hybrid models, specifically the InceptVGG and VGGRes Models, consistently exhibit superior AUC scores on both the training and validation datasets. Notably, the VGGRes Model emerges as the standout performer, achieving the highest AUC score of 0.89 on the testing dataset. This outstanding result underscores the effectiveness of the VGGRes Model in the context of ECG signal classification, highlighting its potential for accurate and reliable diagnostics. Importantly, VGGRes models exhibit adaptability to various types of lead ECG signals, as visualized in Figs. 6–8, which presents the performance metrics graph, confusion matrix, and visualization of ECG plots generated by the VGGRes with 27 class labels for the data.

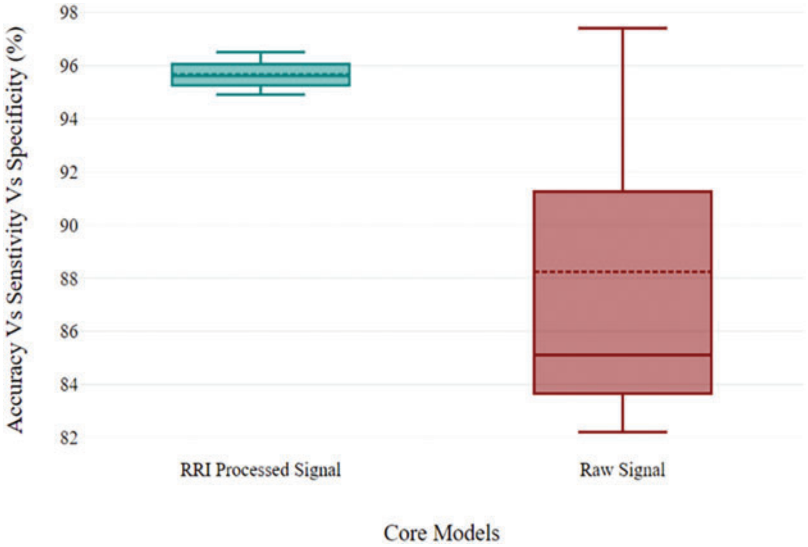


Figure 5: Performance metrics of proposed novelty models for binary class distribution

Table 6: Comparison of proposed ensemble model with previous models using arrhythmia dataset

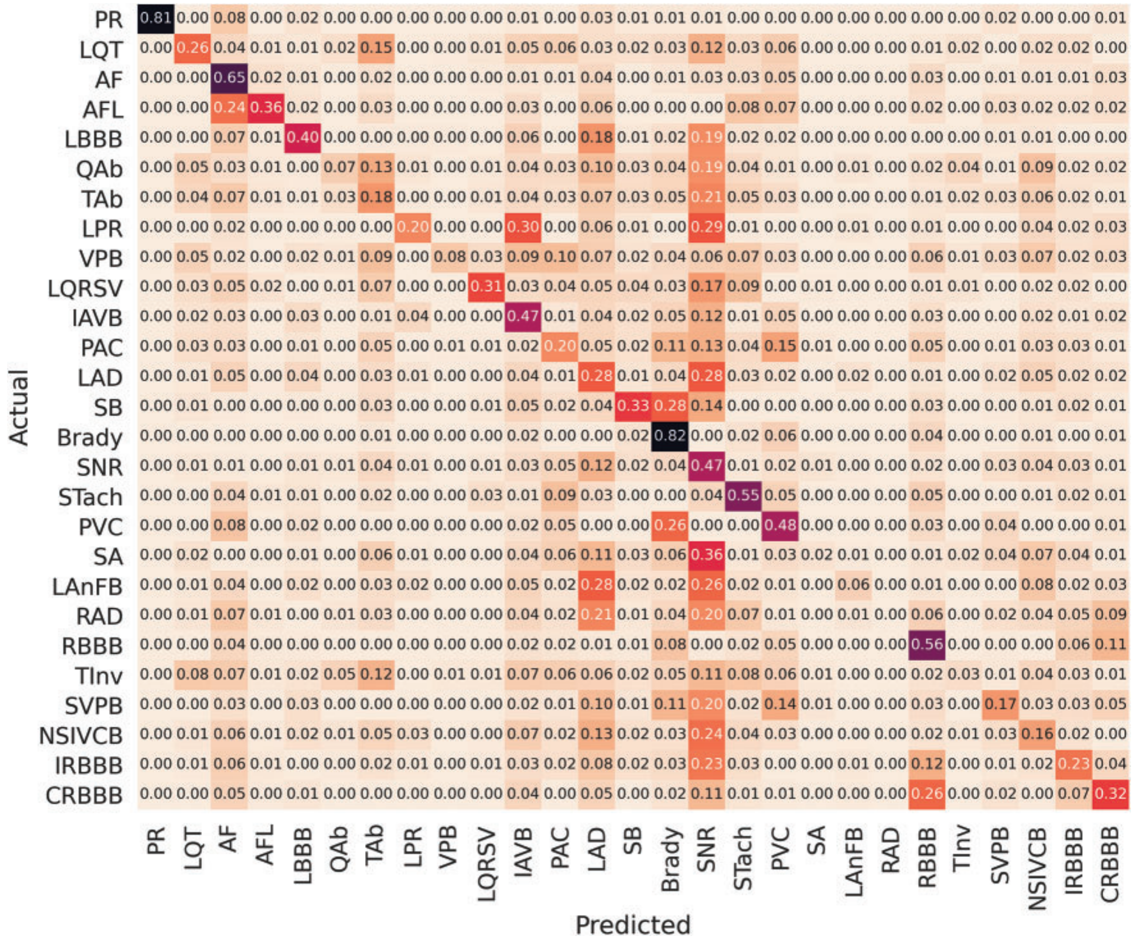
Method	Training AUC	Validation AUC	Testing AUC
Different DL approaches [39]	0.76	0.68	0.72
ATI-CNN [40]	0.67	0.70	0.72
MBSF-Net [41]	0.79	0.71	0.69
Proposed ResIncept model	0.81	0.74	0.87
Proposed VGGRes model	0.81	0.82	0.89
Proposed InceptVGG model	0.80	0.80	0.86
Proposed LenetAlexLSTM model	0.77	0.78	0.79

Binary Class: novelty algorithm:

The ability of transformer models to attend to various sequence elements with varied degrees of priority allows them to handle variable-ECG length sequences. The transformer-based hybrid combination allowed this framework to achieve its higher performance. Table 7 shows that the fold values of the convoluted transformer model have earned an excellent sensitivity score compared to traditional methods. This underscores the model's robustness, primarily due to innovative algorithmic features and carefully designed pre-processing stages.

Question 3: Which category of features like raw signals convoluted features or RRI convoluted features in binary class over single and multi-modal dataset have the best impact?

The proposed novelty algorithm was observed to perform well on RRI-featured signals displayed in Table 8. To prove the best impacts of this working model, we have tested with various datasets for raw and RRI-detected signals.



	PR	LQT	AF	AFL	LBBB	QAb	TAb	LPR	VPB	LQRSV	IAVB	PAC	LAD	SB	Brady	SNR	STach	PVC	SA	LAnFB	RAD	RBBB	TInv	SVPB	NSIVCB	IRBBB	CRBBB
PR	0.81	0.00	0.08	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.03	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01
LQT	0.00	0.26	0.04	0.01	0.01	0.02	0.15	0.00	0.00	0.01	0.05	0.06	0.03	0.02	0.03	0.12	0.03	0.06	0.00	0.00	0.00	0.01	0.02	0.00	0.02	0.02	0.00
AF	0.00	0.00	0.65	0.02	0.01	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.04	0.00	0.01	0.03	0.03	0.05	0.00	0.00	0.00	0.03	0.00	0.01	0.01	0.01	0.03
AFL	0.00	0.00	0.24	0.36	0.02	0.00	0.03	0.00	0.00	0.00	0.03	0.00	0.06	0.00	0.00	0.00	0.08	0.07	0.00	0.00	0.00	0.02	0.00	0.03	0.02	0.02	0.02
LBBB	0.00	0.00	0.07	0.01	0.40	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.18	0.01	0.02	0.19	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00
QAb	0.00	0.05	0.03	0.01	0.00	0.07	0.13	0.01	0.00	0.01	0.04	0.03	0.10	0.03	0.04	0.19	0.04	0.01	0.00	0.01	0.00	0.02	0.04	0.01	0.09	0.02	0.02
TAb	0.00	0.04	0.07	0.01	0.01	0.03	0.18	0.00	0.00	0.01	0.04	0.03	0.07	0.03	0.05	0.21	0.05	0.03	0.00	0.00	0.00	0.01	0.02	0.03	0.06	0.02	0.01
LPR	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.30	0.00	0.06	0.01	0.00	0.29	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.04	0.02	0.03
VPB	0.00	0.05	0.02	0.00	0.02	0.01	0.09	0.00	0.08	0.03	0.09	0.10	0.07	0.02	0.04	0.06	0.07	0.03	0.00	0.00	0.00	0.06	0.01	0.03	0.07	0.02	0.03
LQRSV	0.00	0.03	0.05	0.02	0.00	0.01	0.07	0.00	0.00	0.31	0.03	0.04	0.05	0.04	0.03	0.17	0.09	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.02	0.02	0.00
IAVB	0.00	0.02	0.03	0.00	0.03	0.00	0.01	0.04	0.00	0.00	0.47	0.01	0.04	0.02	0.05	0.12	0.01	0.05	0.00	0.00	0.00	0.03	0.00	0.00	0.02	0.01	0.02
PAC	0.00	0.03	0.03	0.00	0.01	0.00	0.05	0.00	0.01	0.01	0.02	0.20	0.05	0.02	0.11	0.13	0.04	0.15	0.01	0.00	0.00	0.05	0.00	0.01	0.03	0.03	0.01
LAD	0.00	0.01	0.05	0.00	0.04	0.00	0.03	0.01	0.00	0.00	0.04	0.01	0.28	0.01	0.04	0.28	0.03	0.02	0.00	0.02	0.00	0.01	0.00	0.02	0.05	0.02	0.02
SB	0.00	0.01	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.05	0.02	0.04	0.33	0.28	0.14	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.02	0.01
Brady	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.82	0.00	0.02	0.06	0.00	0.00	0.00	0.04	0.00	0.00	0.01	0.00	0.01
SNR	0.00	0.01	0.01	0.00	0.01	0.01	0.04	0.01	0.00	0.01	0.03	0.05	0.12	0.02	0.04	0.47	0.01	0.02	0.01	0.00	0.00	0.02	0.00	0.03	0.04	0.03	0.01
STach	0.00	0.00	0.04	0.01	0.01	0.00	0.02	0.00	0.00	0.03	0.01	0.09	0.03	0.00	0.00	0.04	0.55	0.05	0.00	0.00	0.00	0.05	0.00	0.00	0.01	0.02	0.01
PVC	0.00	0.00	0.08	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.05	0.00	0.00	0.26	0.00	0.00	0.48	0.00	0.00	0.00	0.03	0.00	0.04	0.00	0.00	0.01
SA	0.00	0.02	0.00	0.00	0.01	0.00	0.06	0.01	0.00	0.00	0.04	0.06	0.11	0.03	0.06	0.36	0.01	0.03	0.02	0.01	0.00	0.01	0.02	0.04	0.07	0.04	0.01
LAnFB	0.00	0.01	0.04	0.00	0.02	0.00	0.03	0.02	0.00	0.00	0.05	0.02	0.28	0.02	0.02	0.26	0.02	0.01	0.00	0.06	0.00	0.01	0.00	0.00	0.08	0.02	0.03
RAD	0.00	0.01	0.07	0.01	0.00	0.01	0.03	0.00	0.00	0.00	0.04	0.02	0.21	0.01	0.04	0.20	0.07	0.01	0.00	0.01	0.00	0.06	0.00	0.02	0.04	0.05	0.09
RBBB	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.01	0.01	0.08	0.00	0.02	0.05	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.06	0.11
TInv	0.00	0.08	0.07	0.01	0.02	0.05	0.12	0.00	0.01	0.01	0.07	0.06	0.06	0.02	0.05	0.11	0.08	0.06	0.01	0.00	0.00	0.02	0.03	0.01	0.04	0.03	0.01
SVPB	0.00	0.00	0.03	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.10	0.01	0.11	0.20	0.02	0.14	0.01	0.00	0.00	0.03	0.00	0.17	0.03	0.03	0.05
NSIVCB	0.00	0.01	0.06	0.01	0.02	0.01	0.05	0.03	0.00	0.00	0.07	0.02	0.13	0.02	0.03	0.24	0.04	0.03	0.00	0.00	0.00	0.02	0.01	0.03	0.16	0.02	0.00
IRBBB	0.00	0.01	0.06	0.01	0.00	0.00	0.02	0.01	0.00	0.01	0.03	0.02	0.08	0.02	0.03	0.23	0.03	0.00	0.00	0.01	0.00	0.12	0.00	0.01	0.02	0.23	0.04
CRBBB	0.00	0.00	0.05	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.05	0.00	0.02	0.11	0.01	0.01	0.00	0.00	0.00	0.26	0.00	0.02	0.00	0.07	0.32

Figure 6: Confusion matrix of VGGRes models

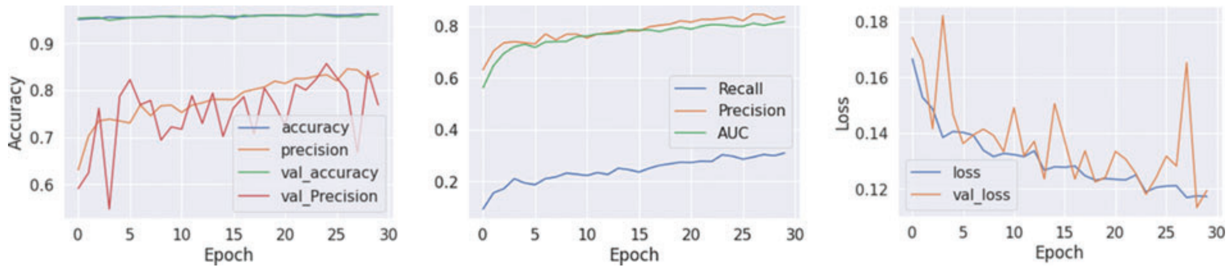


Figure 7: Training, testing and loss graph of VGGRes models

Table 8 briefly illustrates the hybrid combination "CNN + Transformer model" performance with the single dataset value and multi-modal ECG signal dataset. All the datasets performed well on this working model. The best method of this model is given by a multi-model dataset with the pipeline setup of "Filtering technique + RRI detected signals + CNN features + transformer time series model" with the accuracy of 96.8 seen by Auto ML & DL framework. Compared to raw signals, which show a significant variation of difference, 11.8 states that peak detection has the best impact in detection for binary class arrhythmia. The training loss of the model is also 20%, and, following that, set the dropout rate at 0.25, which performed efficiently over other models.

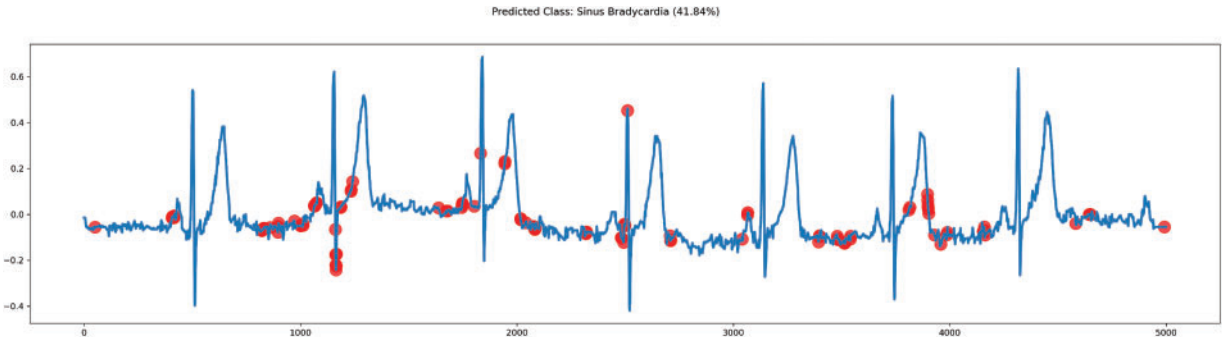


Figure 8: Plot of ECG signals predicted by VGGRes models

Table 7: K-fold values of proposed binary class model with traditional methods

Folds	CNN	ELM-RNN	LSTM-Autoencoders	Proposed model
Fold 2	81.22	82.11	80.44	86.286
Fold 4	83.59	85.03	84.12	90.286
Fold 6	85.78	86.89	85.32	90.721
Fold 8	87.91	88.67	87.29	93.276
Fold 10	89.45	90.12	89.93	96.60

Table 8: Comparison of “Convoluted Transformer model” with raw and RRI signals

Dataset	Epochs	Test accuracy		Test sensitivity		Test specificity		Test loss	
		Raw signal	RRI signal	Raw signals	RRI signal	Raw signal	RRI signals	Raw signal	RRI signal
China physiological signal challenge 2018	100	84.57	88.5	87.8	88.9	71.9	71.9	0.151	0.189
PTB-XL electrocardiography database	100	82.3	83.2	85.5	86.0	77.1	80.5	0.154	0.164
Multi-model database	100	85	96.60	89.1	97	91.0	95.94	0.143	0.155

To prove the best impacts of this working model we have further tested with various dataset for both raw and RRI detected signals. Table 8 illustrated the performance of hybrid combination “CNN + Transformer model” with the single dataset value and multi modal ECG signal dataset.

4.2 Implications of Findings

Our findings offer crucial implications for the field of arrhythmia detection in ECG signals:

Enhanced Diagnosis Precision: Our meticulous approach to preprocessing, model selection, and feature engineering improves diagnostic accuracy, benefiting patient care through quicker, tailored treatment plans.

Streamlined Healthcare Delivery: Our guidance on selecting deep learning models streamlines the development of automated arrhythmia detection systems, reducing the workload on healthcare professionals, especially in resource-constrained settings.

Feature Engineering Insight: Emphasizing the importance of features, especially RRI convoluted features, enhances our understanding of critical elements in arrhythmia detection applicable to this field and other ECG-based diagnostic tasks.

4.3 Discussion

Our study has unveiled valuable insights into arrhythmia detection in ECG signals, shedding light on the advantages and considerations associated with the model and its findings. These insights could revolutionize the accuracy and efficiency of arrhythmia diagnosis in healthcare.

Enhanced Diagnostic Accuracy: One of the most significant advantages of our model is the substantial improvement in diagnostic accuracy achieved through meticulous preprocessing, model selection, and feature engineering. Notably, the VGGRes Model demonstrated a remarkable precision of 97.39% in specific configurations.

Model Selection Guidance: Our study provides invaluable guidance for selecting appropriate deep-learning models for ECG signal classification. The VGGRes Model, Inception model, and RRI transformer-based hybrid models emerged as top performers. This guidance simplifies the decision-making process for researchers and healthcare professionals, enabling them to choose models tailored to their specific diagnostic needs.

Despite its numerous advantages, our study does present certain disadvantages and considerations:

1. The potential variability in ECG signal data across different patient populations and healthcare settings could affect the generalizability of our findings.
2. Some of the deep learning models utilized, particularly those with complex architectures, may demand significant computational resources, which could be limiting in resource-constrained healthcare environments.
3. The ethical and regulatory aspects surrounding data privacy and security, especially when handling sensitive medical data, must be diligently addressed.

5 Conclusion and Future Work

The AutoRhythmAI model framework conducted a hyperparameter optimization test to validate its novel approach. The framework demonstrated superior performance, showcasing its capability to automate ECG signal classification by combining binary and multi-class arrhythmia detection. Initially, we amalgamated diverse databases from various geographical locations, ensuring data balance through machine learning balancing algorithms. Subsequently, we subjected the data to two levels of preprocessing within the custom-designed auto-ML and deep learning techniques. We considered 27 groups for binary-class classification: 26 related to heart problems (including those merged with aberrant signals) and one associated with a healthy condition. All 27 groups underwent multi-label binarization. In feature extraction, we introduced the "RRI-Convoluted Transformer model" for binary ECG signal classification, demonstrating significantly higher accuracy. The ensemble model named the VGGRes Model for multi-class type also yielded the best accuracy. The obtained results validate the effectiveness and applicability of the proposed model, with comparisons and validation against recent literature. Despite its high computational complexity and limited interpretability, the model represents a pioneering step towards improving utility in the healthcare industry. Future studies will focus on refining this method. The research introduces the AutoRhythmAI framework, marking a groundbreaking advancement in arrhythmia detection. Its fusion of traditional and deep learning techniques, automated feature engineering, flexible pipeline, innovative algorithms, ensemble-based approach, real-time potential, and data integration highlights its potential impact on accurate and efficient diagnostics.

Designing this AutoRhythmAI model requires significant computational resources and can be computationally expensive. Small businesses or organizations with limited computational resources may find this challenging. This model is hard to interpret due to the complexity of the optimization process [42]. It is also clear that including RRI features improves model performance significantly but does so at the cost of more than a nine-fold increase in training and inference time. One of this model's main limitations is the delayed Pan-Tompkins approach, used to conduct R-peak identification and is the reason for the time increase. According to the suggested system, the classification also depends on two pipelines, which allows for a slow detection rate. In the future, the hybrid combination of different deep learning models and various preprocessing can be constructed as a single pipeline to achieve higher computation and accuracy for detection.

Acknowledgement: The authors would like to appreciate all anonymous reviewers for their insightful comments and constructive suggestions to polish this paper in high quality.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: S. Jayanthi, S. Prasanna Devi; data collection: S. Jayanthi; analysis and interpretation of results: S. Jayanthi; draft manuscript preparation: S. Jayanthi, S. Prasanna Devi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Leclercq *et al.,* "Wearables, telemedicine, and artificial intelligence in arrhythmias and heart failure," *Proc. Eur. Soc. Cardiol. Cardiovasc. Round Table, Europace*, vol. 24, no. 9, pp. 1372–1383, 2022. doi: 10.1093/europace/euac052.

[2] R. Lalo, I. Zekja, and F. Kamberi, "Association of cardiovascular disease risk and health-related behaviors in stroke patients," *Int. J. Environ. Res. Public Health*, vol. 20, no. 4, pp. 3693, 2023. doi: 10.1016/j.est.2021.103009.

[3] S. K. Saini and R. Gupta, "Artificial intelligence methods for analysis of electrocardiogram signals for cardiac abnormalities: State-of-the-art and future challenges," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 1519–1565, 2022. doi: 10.1016/j.est.2021.103009.

[4] Y. L. Liu, C. S. Lin, C. C. Cheng, and C. Lin, “A deep learning algorithm for detecting acute pericarditis by electrocardiogram,” *J. Pers. Med.*, vol. 12, no. 7, pp. 1150, 2022. doi: 10.3390/jpm12071150.

[5] S. Jayanthi and S. P. Devi, “Automated machine learning on high dimensional big data for prediction tasks,” in *Proc. ICCES*, Coimbatore, India, 2022, pp. 995–1001.

[6] K. M. Aamir *et al.,* “Automatic heart disease detection by classification of ventricular arrhythmias on ECG using machine learning,” *Comput. Mater. Continua*, vol. 71, no. 1, 2022. doi: 10.32604/cmc.2022.018613.

[7] J. P. Consuegra-Ayala, Y. Gutiérrez, Y. Almeida-Cruz, and M. Palomar, “Intelligent ensembling of auto-ML system outputs for solving classification problems,” *Inform. Sci.*, vol. 609, pp. 766–780, 2022. doi: 10.1016/j.ins.2022.07.061.

[8] T. Siriborvornratanakul, “Human behavior in image-based road health inspection systems despite the emerging AutoML,” *J. Big Data*, vol. 9, no. 1, pp. 96, 2022. doi: 10.1186/s40537-022-00646-8.

[9] X. Wang *et al.,* “An ECG signal denoising method using conditional generative adversarial net,” *IEEE J. Biomed. Health Inform.*, vol. 26, no. 7, pp. 2929–2940, 2022. doi: 10.1109/JBHI.2022.3169325.

[10] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, “Auto-sklearn 2.0: Hands-free automl via meta-learning,” *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 11936–11996, 2022.

[11] H. Jin, Q. Song, and X. Hu, “Auto-keras: An efficient neural architecture search system,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, New York, NY, USA, 2019, pp. 1946–1956.

[12] J. D. Romano, T. T. Le, W. Fu, and J. H. Moore, “TPOT-NN: Augmenting tree-based automated machine learning with neural network estimators,” *Genet. Program. Evolvable Mach.*, vol. 22, pp. 207–227, 2021. doi: 10.1007/s10710-021-09401-z.

[13] S. Naseem, K. Javed, M. J. Khan, S. Rubab, M. A. Khan and Y. Nam, “Integrated CWT-CNN for epilepsy detection using multiclass EEG dataset,” *Comput. Mater. Continua*, vol. 69, no. 1, pp. 471–486, 2021. doi: 10.32604/cmc.2021.018239.

[14] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, “Review of noise removal techniques in ECG signals,” *IET Signal Process*, vol. 14, no. 9, pp. 569–590, 2020.

[15] M. U. Ali *et al.,* “Correlation-filter-based channel and feature selection framework for hybrid EEG-fNIRS BCI applications,” *IEEE J. Biomed. Health*, 2023. doi: 10.1109/JBHI.2023.3294586.

[16] M. Loria-Romero, H. Peregrina-Barreto, J. Rangel-Magdaleno, and H. D. Rico-Aniles, “Ventricular fibrillation characterization for sudden cardiac death risk prediction based on wavelet analysis,” in *2022 IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Messina, Italy, IEEE, 2022, pp. 1–6. doi: 10.1109/MeMeA54994.2022.9856497.

[17] N. Salari *et al.,* “Detection of sleep apnea using machine learning algorithms based on ECG signals: A comprehensive systematic review,” *Expert Syst. Appl.*, vol. 187, pp. 115950, 2022. doi: 10.1016/j.eswa.2021.115950.

[18] S. Bibi *et al.,* “MSRNet: Multiclass skin lesion recognition using additional residual block based fine-tuned deep models information fusion and best feature selection,” *Diagnostics*, vol. 13, no. 19, pp. 3063, 2023.

[19] M. Hussain *et al.,* “SkinNet-INIO: Multiclass skin lesion localization and classification using fusion-assisted deep neural networks and improved nature-inspired optimization algorithm,” *Diagnostics*, vol. 13, no. 18, pp. 2869, 2023.

[20] V. Dillshad, M. A. Khan, M. Nazir, O. Saidani, N. Alturki and S. Kadry, “D2LFS2Net: Multi-class skin lesion diagnosis using deep learning and variance-controlled marine predator optimisation: An application for precision medicine,” *CAAI Trans. Intell. Technol.*, 2023. doi: 10.1049/cit2.12267.

[21] S. Dey, R. Pal, and S. Biswas, “Deep learning algorithms for efficient analysis of ECG signals to detect heart disorders,” in *Biomedical Engineering*, London, UK: IntechOpen, 2022. doi: 10.5772/intechopen.103075.

[22] O. M. A. Ali, S. W. Kareem, and A. S. Mohammed, “Evaluation of electrocardiogram signals classification using CNN, SVM, and LSTM algorithm: A review,” in *2022 8th Int. Eng. Conf. Sustain. Technol. Development (IEC)*, Erbil, Iraq, 2022, pp. 185–191.

[23] A. Alqahtani, S. Alsubai, M. Sha, M. A. Khan, M. Alhaisoni and S. R. Naqvi, “Automated white blood cell disease recognition using lightweight deep learning,” *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 107–123, 2023.

[24] V. Gupta, N. K. Saxena, A. Kanungo, P. Kumar, and S. Diwania, "PCA as an effective tool for the detection of R-peaks in an ECG signal processing," *Int. J. Syst. Assur. Eng. Manage.*, vol. 13, no. 5, pp. 2391–2403, 2022. doi: 10.1007/s13198-022-01650-0.

[25] M. M. Farag, "A matched filter-based convolutional neural network (CNN) for inter-patient ECG classification and arrhythmia detection at the edge," *SSRN Electron. J.*, 2022. doi: 10.2139/ssrn.4070665s.

[26] S. Irfan, N. Anjum, T. Althobaiti, A. A. Alotaibi, A. B. Siddiqui and N. Ramzan, "Heartbeat classification and arrhythmia detection using a multi-model deep-learning technique," *Sens.*, vol. 22, no. 15, pp. 5606, 2022. doi: 10.3390/s22155606.

[27] M. S. Islam, M. N. Islam, N. Hashim, M. Rashid, B. S. Bari *et al.,* "New hybrid deep learning approach using BiGRU-BiLSTM and multilayered dilated CNN to detect arrhythmia," *IEEE Access*, vol. 10, pp. 58081–58096, 2022. doi: 10.1109/ACCESS.2022.3178710.

[28] H. H. Rashidi, N. Tran, S. Albahra, and L. T. Dang, "Machine learning in health care and laboratory medicine: General overview of supervised learning and auto-ML," *Int. J. Lab. Hematol.*, vol. 43, pp. 15–22, 2021. doi: 10.1111/ijlh.13537.

[29] M. Bodini, M. W. Rivolta, and R. Sassi, "Classification of ECG signals with different lead systems using autoML," in *Computing in Cardiology (CinC)*, Brno, Czech Republic, 2021, pp. 1–4.

[30] D. Kumar, S. Puthusserypady, H. Dominguez, K. Sharma, and J. E. Bardram, "CACHET-CADB: A contextualized ambulatory electrocardiography arrhythmia dataset," *Front. Cardiovasc. Med.*, vol. 9, pp. 893090, 2022. doi: 10.3389/fcvm.2022.893090.

[31] Y. Xia *et al.,* "An automatic cardiac arrhythmia classification system with wearable electrocardiogram," *IEEE Access*, vol. 6, pp. 16529–16538, 2018. doi: 10.1109/ACCESS.2018.2807700.

[32] D. K. Atal and M. Singh, "Arrhythmia classification with ECG signals based on the optimization-enabled deep convolutional neural network," *Comput. Meth. Prog. Bio.*, vol. 196, pp. 105607, 2020. doi: 10.1016/j.est.2021.103009.

[33] J. Hua, B. Chu, J. Zou, and J. Jia, "ECG signal classification in wearable devices based on compressed domain," *PLoS One*, vol. 18, no. 4, pp. e0284008, 2023. doi: 10.1371/journal.pone.0284008.

[34] M. R. Rajeshwari and K. S. Kavitha, "Arrhythmia ventricular fibrillation classification on ECG signal using ensemble feature selection and deep neural network," *Cluster Comput.*, vol. 25, no. 5, pp. 3085–3102, 2022. doi: 10.1007/s10586-022-03547-w.

[35] M. Hammad, S. Meshoul, P. Dziwiński, P. Pławiak, and I. A. Elgendy, "Efficient lightweight multimodel deep fusion based on ECG for arrhythmia classification," *Sens.*, vol. 22, no. 23, pp. 9347, 2022. doi: 10.3390/s22239347.

[36] S. Jayanthi and S. P. Devi, "Ensemble of deep learning models for classification of heart beats arrhythmias detection," *J. Theor. Appl. Inform. Technol.*, vol. 101, no. 8, 2023.

[37] I. H. Tsai and B. I. Morshed, "Scalable and upgradable AI for detected beat-by-beat ECG signals in smart health," in *Proc. AIIoT Seattle*, WA, USA, 2023, pp. 0409–0414. doi: 10.1109/AIIoT58121.2023.10174482.

[38] N. Salankar, D. Koundal, C. Chakraborty, and L. Garg, "Automated attention deficit classification system from multimodal physiological signals," *Multimed. Tools Appl.*, vol. 82, no. 4, pp. 4897–4912, 2023. doi: 10.1007/s11042-022-12170-1.

[39] A. K. Singh and S. Krishnan, "ECG signal feature extraction trends in methods and applications," *Biomed. Eng. Online*, vol. 22, no. 1, pp. 1–36, 2023. doi: 10.1186/s12938-023-01075-1.

[40] Y. Dong *et al.,* "Detection of arrhythmia in 12-lead varied-length ECG using multi-branch signal fusion network," *Physiol. Meas.*, vol. 43, no. 10, pp. 105009, 2022. doi: 10.1088/1361-6579/ac7938.

[41] Q. Yao, R. Wang, X. Fan, J. Liu, and Y. Li, "Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network," *Inform. Fusion*, vol. 53, pp. 174–182, 2020. doi: 10.1016/j.inffus.2019.06.024.

[42] S. Jayanthi and S. P. Devi, "Automated ECG arrhythmia classification using resnet and autoML learning model," in *Proc. ICSTCEE*, Bengaluru, India, 2022, pp. 1–6.