



## Detection of Cardiac Arrhythmia from ECG Using CNN and XGBoost

Ravindar Mogili<sup>1\*</sup>      G. Narsimha<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, Jyothishmathi Institute of Technology & Science, India

<sup>2</sup>Department of Computer Science and Engineering, JNTUH University College of Engineering Sultanpur, India

\* Corresponding author's Email: eman.hanfy@azhar.edu.eg

---

**Abstract:** Abnormal behaviour of the heart called arrhythmia can be recorded as an electrocardiogram (ECG) signal. ECG plays a vital role in the diagnosis of heart disease. Advances in machine learning allow the development of computer-aided diagnostic models to identify heart disease type. We have proposed a hybrid model with Convolution Neural Network (CNN) to auto extract features from ECG and use XGBoost to assess the type of arrhythmia. We tested our model to diagnosis eleven kinds of arrhythmia beats from the MIT-BIH arrhythmia database and obtained overall sensitivity of 92.61%, specificity of 99.85%, positive predictive value of 95.99% and accuracy of 99.84%. The robustness of the proposed model is further confirmed by classifying the arrhythmia beats into 5 classes according to the AAMI standard and comparing the results with the state-of-the-art methods. Attained overall Sensitivity of 94.36%, specificity of 99.44%, positive predicative value of 96.40% and Accuracy of 99.69% for 5 AAMI Classes. The results demonstrated that the proposed model could be used in the diagnosis of arrhythmia.

**Keywords:** Arrhythmia, ECG, Convolution neural network, CNN, XGBoost, Heart disease.

---

### 1. Introduction

Cardiovascular disease (CVD) accounts for 32% of all global deaths in 2019 [1]. The premature deaths due to CVD can be prevented if it is detected in the early stage. Early detection of CVD is possible through regular study of heart condition and functionality of an individual. The main function of the heart is to pump purified blood throughout the body with the help of electrical impulses generated in the heart. Heart disease or other heart problems can lead to irregular and poor conduction of electrical impulses in the heart. These irregular impulses cause an abnormal heart rate and rhythm. This abnormality of the heart is called arrhythmia and can be detected using ECG [2]. Based on the abnormality, the arrhythmia is classified into different types. Among them, few arrhythmia types are fatal. Although the other types are not dangerous, the frequent occurrence of such types can lead to life threatening arrhythmia types. Therefore, identifying the correct type of arrhythmia

and providing the proper medications and treatments in early stage can reduce the death rate from CVD.

Nowadays, wearable devices like Fit-bit, smart watches, chest traps, etc are used to regularly monitor a person's heart function. These devices can read a person's ECG signals in real time without the need for medical equipment available in hospitals. The ECG signals collected by these devices must be analyzed to accurately identify the type of arrhythmia. Compared to medical devices, wearable devices have limits in terms of processing power and are dependent on battery power [3]. Therefore, wearable devices cannot more accurately identify the type of arrhythmia. So, wearable devices can only be used to monitor heart function and determine the frequency of abnormalities. They act as warning devices and a person should consult a doctor to know more about the type of heart defect or arrhythmia. Unfortunately, as the number of patients with cardiac arrhythmia increases, the workload of the cardiologist also increases. Therefore, the use of machine learning techniques to detect arrhythmias is crucial. Arrhythmia type can

be detected more precisely when Machine Learning (ML) algorithms are incorporated into medical devices. ML techniques are used in clinical applications for the benefit of physicians, healthcare organizations and individuals.

The objective of this study is to use a hybrid model that combines XGBoost and CNN to improve classification performance of arrhythmia beats when compared to state-of-the-art approaches. XGBoost has excellent generalization performance and supports parallel processing. It can perform classification more accurately as compared to other ML algorithms even datasets have missing values and imbalanced classes. Furthermore, it employs regularization to manage complexity and prevent model overfitting. However, XGBoost is one of the most powerful ML classifier; its performance is influenced by the quality of extracted features supplied to it [4]. In the existing literature, [5, 6] and [7] used XGBoost classifier with hand-crafted features to detect and classify arrhythmias. But, auto extracted feature are extremely effective than hand-crafted features in optimizing classification performance. CNN has the ability to auto extract features from raw ECG signals without human supervision. So, the strategy is to use CNN to extract features and XGBoost to classify and determine the type of arrhythmia.

The organization of the paper is given as follows: Section 2 includes contributions from recent works on the identification and classification of arrhythmia types. Section 3 describes the proposed model, which includes CNN and XGBoost. Section 4 presents experimental results and a comparison of the proposed model to existing methods. Finally, the future work and conclusion is given in Section 5.

## 2. Related work

The general process of classifying arrhythmia from ECG signal involves removing noise and extracting features followed by a classification algorithm. When using the CNN model, it can automatically capture the features from raw ECG without the use of external methods. Recent studies on the classification of arrhythmia are discussed below.

Chiang et al. [8] proposed an Associative Petri net (APN) model for detecting segment-based features and classifying 11 arrhythmia beat categories obtained from MIT-BIH arrhythmia database. APN is a rule-based framework that models and analyzes ECG signals and their dynamics using graphical and mathematical tools.

This model achieved 85.0% sensitivity and 93.5% accuracy. However, APN lack of hierarchy concepts to further enhance classification performance.

Akdeniz et al. [9] extracted features from ECG signals acquired from the MIT-BIH Arrhythmia database using the Zhao Atlas Mark (ZAM) time-frequency distribution. Latter, Cubic SVM is applied to categorize nine types of arrhythmias and achieved accuracy of 96.39%, sensitivity of 94.22%, and positive predictive values of 93.90%. Although ZAM enhances time-frequency resolution and reduces cross-term interferences between two components with the same centre frequency, it fails when the two components have the same time centre. This affects feature quality and reduces classification performance.

Latif et al. [10] classified eleven types of beats collected from MIT-BIH arrhythmia database using five different classifiers such as Multilayer Perceptron, RBF network, decision trees based on linear regression, Naïve Bayes and Random Forest. They achieved best classification performance with Random Forest classifier. The sensitivity of this model is good but the accuracy can be further optimized.

Nurmaini et al. [4] proposed a Deep Learning based model for cardiac arrhythmia identification. They utilized Deep Auto-Encoders (DAEs) to automatically extract features from raw ECG signal, and Deep Neural Networks are employed to classify 10 types of beats collected from the MIT-BIH arrhythmia database. Their model achieved 99.73% accuracy, 91.20% sensitivity, and 93.60% precision. DAEs can filter noise and learn features, but they require more training data and takes time.

Park et al. [11] proposed a unique heartbeat classification model that incorporates an adaptive feature extraction strategy as well as cascade classifiers to detect arrhythmias accurately. Two distinct random forest classifiers are used to build the cascade classifiers. The initial classifier attempts to classify arrhythmia beats using one of the feature sets. If the first classifier's output has a false positive rate greater than a predefined threshold, then it is used as the input to the next classifier based on other feature set. This model achieved 97.34 percent accuracy when evaluated on ten types of beats collected from the MIT-BIH arrhythmia database. Despite the fact that the k-NN classifier performed better on first feature set, the random forest classifier was chosen to decrease computing cost and memory.

Amorim et al. [12] applied Contourlet and Shearlet transforms to divide ECG signals into different frequency bands and then extracted

features from time-frequency coefficients. They compared the performance of KNN, SVM and Random Forest classifiers on these extracted features to classify seven types of beats collected from MIT-BIH arrhythmia database. They achieved the best performance for random forest using features based on Contourlet Transforms with an accuracy of 91.32% and sensitivity of 90.23%. Even though Contourlet Transform provides multi-resolution and multidirectional features, it has shift variant and poor temporal stability.

The fifteen arrhythmia classes obtained from MIT-BIH arrhythmia database are combined and formed five AAMI classes. Acharya et al. [13] developed a 9-layer 1D-CNN to detect five AAMI beat categories from ECG signals. They achieved 94.03% accuracy with the CNN model trained on augmented data. The artificially augmented data may be slightly different from original ECG. It may influence real-time test data. Romdhane et al. [14] proposed to use Focal Loss function in CNN model to enhance classification accuracy of minority classes and also the overall metrics. This model performance is tested to classify and detect five arrhythmia categories as per AAMI standard. Although the accuracy of minority classes improved, it is less compared to our approach. Qayyum et al. [15] proposed to use 2D CNN model to detect arrhythmia beats according to AAMI standard. They demonstrated that the 2D-CNN model outperformed the 1D-CNN model in arrhythmia classification. But, it requires short time Fourier transform to convert the 1D heartbeat data to a 2D image. Moreover, it needs more memory and execution time as compared to 1D-CNN model.

Sharma et al. [16] proposed weighted KNN classifier with an optimal entropy based features. Their model achieved sensitivity of 85.63% and accuracy of 98.1% for 5 categories of AAMI beats. The accuracy of this model is good but the sensitivity can be further improved by giving importance to minority classes. Li et al. [17] developed a novel combination of AdaBoost with Random Forest model for arrhythmia classification as per AAMI classes. This model achieved average positive predictive value of 99.3% but sensitivity of S and F heartbeat categories needs to be improved.

It has been observed that hand-crafted features limit the power of the classifier in case of arrhythmia classification of AAMI 5 classes or more than five classes. But when auto-extracted features are used, never seen before features and also most relevant features to the machine are extracted. As a result, the classification performance can be enhanced. The second finding is that classification

results vary when the same classifier is employed with different feature sets. This indicates extracted features are more important for classification performance. The power of classification algorithm can be utilized to maximum extent, if best quality features are provided to it.

### 3. Methods and materials

The proposed hybrid model integrates a deep learning model such as CNN to auto extract features and a traditional machine learning classifier such as XGBoost to detect arrhythmia type. In this section, we discussed about CNN, XGBoost and proposed model architecture.

#### 3.1 Convolution neural network

The performance of a classifier is influenced by the set of features selected and also the methods used to extract those features. The traditional feature extraction methods often require complex calculations and ignore weak local information while extracting features from images or signals. To overcome these problems, an automated feature extraction method is required and that is convolution neural network (CNN). CNN conserve the input's spatial locality and neighbourhood relations in their latent higher-level feature representations.

CNN has an input layer that takes ECG segment as input; It is followed by a series of alternative convolution layer and pooling layer; Finally, a fully connected layer is placed. The role of each convolution layer is to successfully extract multi-level features from given input. Each convolution layer can be treated as a fuzzy filter to enhance the original ECG signal segment characteristics and reduced the noise. The weight sharing of convolution kernel can successfully reduce model complexity and the number of parameters required in training. The role of pooling layers is to further reduce the training parameters while maintaining local correlation and useful information effectively. The pooling layer divides the previous layer neurons into a disjoint set of rectangles and performs down sampling operation on every sub-area to acquire one neuron value in the present layer [18]. The commonly used pooling functions are the max-pooling and average-pooling. The max-pooling outputs the maximum value and average-pooling outputs average value from selected sub-region. The dropout layer can also be used in CNN but it is only activated during training. The role of dropout layer is to randomly drop certain number of neurons during forward pass and remember those neurons [19]. In backward pass, it updates only the weights

of non-dropped neurons. The dropout is a feature that brings the regularization. It makes the CNN model to learn vigorous features that are independent to the neurons. After a series of alternative convolution and pooling layer, a fully connected layer is placed to get normalized feature set for given input of ECG segments.

### 3.2 XGBoost classifier

The features extracted from ECG segments by CNN are given as input to the XGBoost classifier to detect the type of beat. XGBoost [20] is a supervised machine learning algorithm that develops an ensemble sequential  $m$  number of trees denoted by  $F = \{f_1, f_2, f_3, \dots, f_m\}$  on dataset  $D$  prepared from extracted features  $(\vec{X}_i)$  and its corresponding ECG beat class labels  $(y_i)$  i.e.,  $D = \{\vec{X}_i, y_i\}_{i=1}^n$  containing  $n$  instance. Let the initial prediction of  $\vec{X}_i$  before developing first tree be zero, i.e.  $p_i^{(0)} = 0$ , then the prediction of a tree  $f_t$  on  $\vec{X}_i$  is given by

$$p_i^{(t)} = p_i^{(t-1)} + f_t(\vec{X}_i) \quad (1)$$

Each new tree  $f_t$  is developed by minimizing the prediction error of its previous tree  $f_{t-1}$  and controlling the over fitting. So, the objective function for  $f_t$  tree is given as

$$\text{obj}^t = \sum_{i=1}^n L(y_i, p_i^{(t)}) + \Omega(f_t) \quad (2)$$

where  $L(y_i, p_i^{(t)})$  be any convex differentiable loss function used to measure the difference between the prediction  $p_i^{(t)}$  and actual outcome  $y_i$  for a given training instance  $\vec{X}_i$ . The regularization term  $\Omega(f_t)$  controls the tree structure to avoid over fitting and it is given by

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (3)$$

where  $T$  is the number of leaves of  $f_t$  tree and  $\gamma$  is user parameter to add a constant penalty for each additional leaf to the tree.  $W = \{w_1, w_2, \dots, w_T\}$  is the predicted scores to be assigned at leaf nodes such that, at  $k^{\text{th}}$  node  $w_k$  is assigned and  $\lambda$  is other user parameter that penalizes extreme weights. After substituting Eq. (1) in Eq. (2) we get

$$\text{obj}^{(t)} = \sum_{i=1}^n L(y_i, p_i^{(t-1)} + f_t(\vec{X}_i)) + \Omega(f_t) \quad (4)$$

In order to simplify the calculations of loss function, it is expanded with Taylor series and approximated up to second degree term. This is given by the equation:

$$\text{obj}^{(t)} \cong \sum_{i=1}^n \left[ L(y_i, p_i^{(t-1)}) + g_i f_t(\vec{X}_i) + \frac{1}{2} h_i f_t^2(\vec{X}_i) \right] + \Omega(f_t) \quad (5)$$

where  $g_i = \partial_{p_i^{(t-1)}} L(y_i, p_i^{(t-1)})$  &  $h_i = \partial_{p_i^{(t-1)}}^2 L(y_i, p_i^{(t-1)})$  are first and second order partial derivatives respectively. Each input  $\vec{X}_i$  in a tree leads to one of the leaf nodes, and the weight assigned to that leaf node is an estimate. If  $q$  is function that maps inputs to weights, then we can write

$$f_t(\vec{X}_i) = W_q(\vec{X}_i) = w_j \quad (6)$$

Since the derivative of constant is zero, we can omit constant terms from objective function. From Eqs. (5) and (6) after omitting constant term, we can write

$$\text{obj}^{(t)} \cong \sum_{i=1}^n \left[ g_i w_j + \frac{1}{2} h_i w_j^2 \right] + \Omega(f_t) \quad (7)$$

If  $I_j$  is a set of input instances leads to a particular leaf node  $j$  in  $f_t$ , then the objective function for all the  $T$  leaves in  $f_t$  can be written as

$$\text{obj}^{(t)} \cong \sum_{j=1}^T \left\{ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i \right) w_j^2 \right\} + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (8)$$

Let the sum of first order and second order derivatives at  $j^{\text{th}}$  leaf node represented as  $(\sum_{i \in I_j} g_i) = G_j$ ,  $(\sum_{i \in I_j} h_i) = H_j$  respectively. Then Eq. (8) can be written as

$$\text{obj}^{(t)} \cong \sum_{j=1}^T \left\{ G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right\} + \gamma T \quad (9)$$

The best weight  $w_j$  that optimizes the objective function can be found by differentiating Eq. (9) w.r.t  $w_j$ . It is given as

$$\frac{\partial \text{obj}^{(t)}}{\partial w_j} = G_j + (H_j + \lambda) w_j = 0 \Rightarrow w_j = \frac{-G_j}{H_j + \lambda} \quad (10)$$

Substituting Eq. (10) in Eq. (9) gives the objective function for best tree.

$$\text{obj}^{(t)} \cong -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (11)$$

When a tree structure is given, Eq. (11) can be used to determine how well it predicts. But generating all possible tree structure on D to identify the best predicting tree structure is practically risky process. This job is simplified by XGBoost in a greedy method. The process of designing best tree at each iteration starts with root node by placing all the input instances at it and its objective function is given by Eq. (11). Split the root node into left and right child nodes based on some condition and also split the instances at parent node accordingly. Then the gain due to current split is given by subtracting parent node objective function from the sum of the objective functions of left and right Childs. It is given by

$$\text{Gain} = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{G_L + G_R + \lambda} \right] - \gamma \quad (12)$$

If the gain is positive for a child node, then considering it as parent node, split it into left and right child nodes. Repeat this process to get the best tree structure. After generating m sequential trees (when there is no further improvement in prediction) in a way that the newest tree correct the prediction errors of previously existing trees. The final prediction of  $\vec{X}_1$  is given by

$$p_i = \sum_{i=1}^m f_t(\vec{X}_i) \quad (13)$$

where  $p_i$  is the predicted class label of ECG beat.

### 3.3 Proposed architectural model

The proposed model contains two stages; feature extraction with CNN and classification using XGBoost. This is shown in Fig. 1.

The input layer in CNN model takes 720x1 samples from each ECG segment as an input. The first convolution layer with 6 filters and kernel size of 5 is applied to the input ECG segment to produce an output matrix of 716x6. The max-pooling layer then takes 716x6 matrix as input and produces 238x6 as an output. The second convoluted layer applies 12 filters and kernel size of 5 to input matrix of 238x6 to get matrix of 234x12. The second max pooling layer takes the output of the second convoluted layer as input to generate 77x12 as output. The final convolution layer produces output of 73x24 by applying 36 filters and a kernel size of 5. The third max pooling layer produces matrix of 24x24 as output from 73x24 inputs. Each of the convolution layer and the fully connected layer use "relu" as the activation function to produce output. Each of the max pooling layer uses a pool size of 4

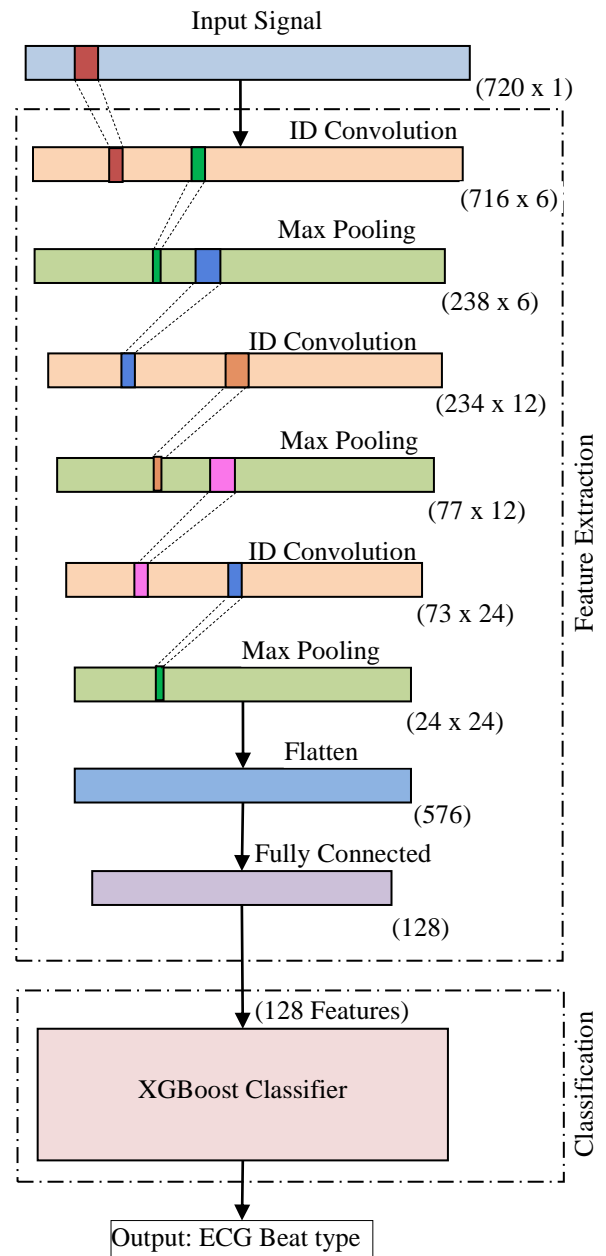


Figure. 1 Proposed model

and a stride of 3. The flatten layer flattens the output of the final max pooling layer and produces 576 outputs. These 576 values are given as input to the fully connected layer to get 128 features. Finally, XGBoost uses 128 features generated by CNN as input to detect the class label of the ECG segment.

### 4. Results and discussion

The performance of proposed model is evaluated using MIT-BIH Arrhythmia Database [21]. ECG recordings in this database are annotated with corresponding beat type at each R peak. The position of R-peaks can also be accurately determined by using algorithms like Pan-Tompkins [22]. Initially, a dataset is created by splitting the

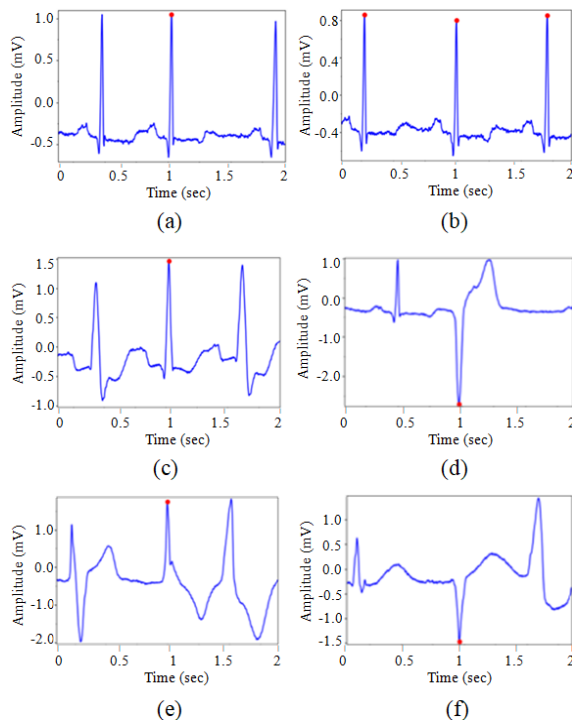


Figure. 2 Sample extracted ECG segments of: (a) Atrial premature beat, (b) Normal beat, (c) Fusion of ventricular and normal beat, (d) Premature ventricular contraction, (e) Fusion of paced and normal beat, and (f) Left bundle branch block beat

raw ECG signals into ECG segments. In the process of preparing dataset, a 2 seconds duration ECG segment was formed from each R peak position such that, a window size of one second to the left and one second to the right from R peak. Each ECG segment assumed to contain one heart beat at the center and the beat type annotated at R peak acts as beat class. Sample ECG segments of window size of 2 seconds are shown in Fig 2.

The initial and final R peaks in each ECG recording may not have either window size of one second to the left or right from R peak. Those R peak segments are not included in the dataset, i.e., only ECG segments of window size of two seconds having left one second and right one second from R peak is chosen to form dataset. In other words an ECG segment contains 720 sampling points with R peak at centre. Finally, dataset is formed with extracted ECG segments as input and their corresponding beat label as output class. The details of these ECG segments are given in Table 1. It contains names of 15 arrhythmia beat types from MIT-BIH arrhythmia database and number of extracted ECG segments for each type. These 15 types of arrhythmia beats are grouped into five super-classes as shown in table 1, according to the Association for the Advancement of Medical Instrumentation (ANSI / AAMI EC57: 1998).

Table 1. Extracted ECG segments

AAMI Beat Class	MIT BIH Beat class	Total No of Beats
Non ectopic beat (N)	Normal beat (NOR)	74926
	Left bundle branch Block Beat (LBB)	8066
	Right bundle branch Block beat (RBB)	7251
	Atrial escape beat (AE)	16
	Nodal Escape (NjE )	229
Supraventricular ectopic beat (S)	Atrial premature beat (AP)	2545
	Aberrated atrial premature beat (aAP)	150
	Nodal (junctional) premature beat (NJP)	83
	Supraventricular premature beat (SVC)	2
Ventricular ectopic beat (V)	Premature ventricular contraction (PVC)	7126
	Ventricular escape beat (VE)	106
Fusion beat (F)	Fusion of ventricular and normal beat (FVN)	802
Unknown beat (Q)	Paced beat ( PB )	7018
	Fusion of paced and normal beat ( fPN )	982
	Unclassified beat (UB)	33

The proposed hybrid model was applied on ECG segment data of eleven beat classes that contain one normal and 10 abnormal classes. The beat classes chosen are Premature Ventricular Contraction (PVC), Fusion of Ventricular and Normal beat (FVN), Atrial Premature beat (AP), Aberrated atrial premature beat (aAP), Nodal (junctional) premature beat (NJP), Right bundle branch block beat (RBB), Left bundle branch block beat (LBB), Paced beat (PB), Ventricular escape beat (VE), Fusion of paced and normal beat ( fPN ) and Normal beat (NOR). As the instances of selected beats are highly imbalanced and number classes to be classified are more, the train-test split strategy with stratified is used. In stratified train-test split, the dataset is split into train and test sets such that each set contains roughly equal proportional instances of target classes. That is, the distribution of each class instances in each set is approximately in same proportions as present in original dataset. This ensures that no set contain either over or under represented instances of a particular class. The selected 11 beat classes are splitted into train data of 70% and test data of 30%. To avoid over-fitting and to guide the training process, the train data set further divided into two



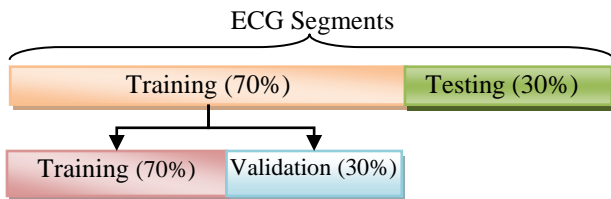
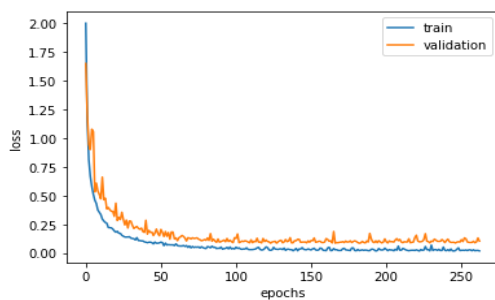


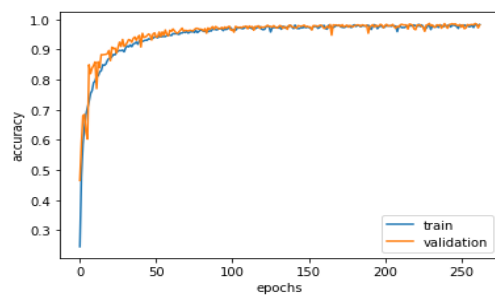
Figure. 3 Division of ECG dataset for training, validating and testing sets

parts. The first part contains 70% of data used to train the model and the remaining 30% data is used to validate the training model. The division of train, validate and test sets is given in Fig. 3.

The 1D-CNN model as described in section 3.3 with a dense layer at the end having “softmax” activation function is fully trained over training and validation data set. The categorical cross entropy loss function and Adam optimizer is used in the process of training the model. Early stopping is employed to avoid over-fitting of CNN model by monitoring validation loss upto 50 epochs. When validation loss does not improve during the last 50 epochs, then the training process is stopped. Latter, the best model is selected based on validation accuracy with the help of a Callbacks API from Keras called as “ModelCheckpoint”. Even though the validation data is not known to the selected training model, it provides best accuracy for the validation data. Therefore, the same training model can also give better accuracy on test data. Based on this assumption, training model is selected. This fully trained model is used to extract features.



(a)



(b)

Figure. 4 Plots for train and validation of CNN: (a) loss curves and (b) accuracy curves

Fig 4(a) and 4(b) shows the loss and accuracy changes of CNN model on training and validation data to better understand the training process. It can be observed from Fig. 4 (a) that the loss of validation data does not decrease after approximately 200 epochs, so the training process continues for another 50 more epochs and then stopped. Figure 4 (b) shows that the accuracy of the validation data and the accuracy of the training data are almost equal approximately from 100 epochs to until the training process stopped. This indicates that the 1D CNN model is well trained. So, the trained model can provide best set of features for classification purpose.

The output from the fully connected layer after flatten layer of the trained CNN model is given as input to the XGBoost model. The flattened layer transforms multidimensional input feature vectors received by it from previous layer into one-dimensional output data. The first fully connected layer reduces the 576 input features received by it into 128 output features. These 128 features are extracted and fed to XGBoost model as input features. While training XGBoost model, to avoid bias towards majority classes, sample weights are assigned to train data using the below formula.

$$\text{class}_i\text{\_weight} = \frac{\text{Total\_no\_of\_samples}}{(\text{no\_of\_classes}) \times (\text{No\_of\_class}_i\text{\_samples})} \quad (14)$$

The number of ECG segments belongs to different arrhythmia types in the data set is highly unbalanced. Therefore the class with the majority samples can influence the overall results of the whole data set. To avoid this, we analyzed class-wise performance besides overall performance. Apart from that we also evaluated using four different metrics such as accuracy, sensitivity, specificity and positive predictive value as per given below equations.

$$\text{Accuracy (Acc)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (15)$$

$$\text{Sensitivity (Sen)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (16)$$

$$\text{Specificity (Spe)} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (17)$$

$$\text{Positive Predictive Value (Ppv)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (18)$$

The analysis of class-wise results from four different evaluation metrics allows us to better understand the performance of the proposed model. In particular, class-wise sensitivity and positive

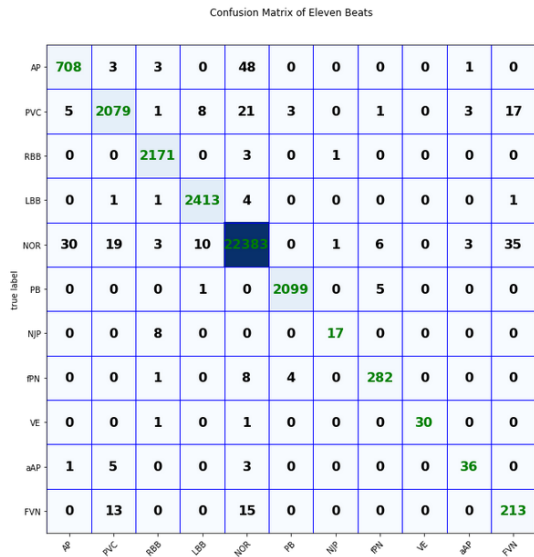


Figure. 5 Confusion matrix for 11-class arrhythmia beats

Table 2. Class-wise evaluation metrics for 11-class arrhythmia beats

Beat Type	No of Test beats	Acc %	Sen %	Spe %	Ppv %
AP	763	99.72	92.79	99.89	95.16
PVC	2138	99.69	97.24	99.87	98.07
RBB	2175	99.93	99.82	99.94	99.18
LBB	2420	99.92	99.71	99.94	99.22
NOR	22490	99.36	99.52	98.99	99.54
PB	2105	99.96	99.71	99.98	99.67
NJP	25	99.97	68.00	99.99	89.47
fPN	295	99.92	95.59	99.96	95.92
VE	32	99.99	93.75	100.00	100.00
aAP	45	99.95	80.00	99.98	83.72
FVN	241	99.75	88.38	99.84	80.08
<b>Average</b>		<b>99.84</b>	<b>92.61</b>	<b>99.85</b>	<b>95.99</b>

Acc=Accuracy, Spe=Specificity, Sen=Sensitivity, Ppv=Positive Predictive Value

predictive value can be used to evaluate the performance of a model when classification data are highly unbalanced.

The confusion matrix of 11 beat classes on test data using the proposed model is given in figure 5 and class-wise accuracy, sensitivity, specificity and positive predictive value for 11 classes are presented in Table 2.

When sensitivities of all the eleven classes from Table 2 are compared, Nodal (junctional) premature beat (NJP) class having low sensitivity of 68%. Even in the existing literature, this class sensitivity is less as compared to other classes [4, 9]. In our case, 8 test beats out of 25 from NJP class are

Table 3. Comparison of 11-class arrhythmia results with other methods

Method used	No of classes	Acc %	Sen %	Spe %	Ppv %
Associative Petri Net Model [8] (2014)	11	93.5	85.0	95.9	85.0
Cascade Random Forrest, Adaptive Feature Extraction [11] (2017)	10	97.34	87.42	99.69	97.98
DNN, Deep Auto-Encoders [4] (2019)	10	99.73	91.2	91.8	93.6
contourlet transforms, Random Forest [12] (2019)	7	91.32	90.23	90.29	-
Random Forest [10] (2020)	11	97.45	97.5	95.9	-
Cubic SVM, Zhao-Atlas mark distribution [9] (2021)	9	96.39	94.22	92.02	93.9
<b>Proposed model</b>	<b>11</b>	<b>99.84</b>	<b>92.61</b>	<b>99.85</b>	<b>95.99</b>

misclassified and treated as PVC class. This can be corrected by increasing numbers samples of NJP class beats both in train and test sets. But, due to non-availability in MITBIH arrhythmia database, it is not possible. Except this class, the sensitivity of other classes is reasonably good and achieved average sensitivity of 92.61%. The class-wise specificity and accuracy is exceptionally good and achieved average specificity and accuracy of 99.85% and 99.84% respectively. In case of positive predictive value, Fusion of paced and normal beat (fPN) class having 80.08% which is lowest when compared with other classes. The average positive predictive value achieved for our model is 95.99%. The comparison of eleven class arrhythmia results of proposed model with other recent existing methods is given in Table 3.

In recent arrhythmia beat classification work, [10-12] used random forest classifier, but the method used to extract features form ECG is different. In case of 11-beat classes, the proposed model outperformed a rule based model such as Associative Petri Net (APN) [8] in all evaluation parameters. Latif et al. [10] achieved high sensitivity using random forest classifier to identify 11 arrhythmia beats, but failed to get better specificity and accuracy compared to our model. Although both Random Forest and XGBoost are ensemble classifiers, the results prove that the latter performs better at detecting arrhythmia category. The positive predictive value of Park et al. [11] is 1.9% higher than proposed model, but its accuracy, specificity and sensitivity are 1.33%, 0.04% and 1.38% lower,



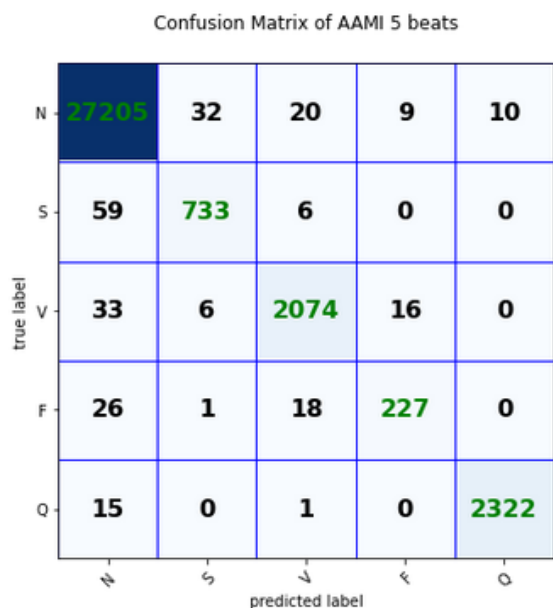


Figure. 6 Confusion matrix for AAMI 5-class beats

respectively. Furthermore, when compared to our approach, the process of applying the features in Park et al. model [11] is more complicated and requires human monitoring. Even though number of beat classes to be classified in our model is greater than [4, 9, 11, 12], our results are better than their results. This indicates the proposed XGBoost classifier with CNN is improving the classification results. But, number of beat classes and their combination in classification task is different from one researcher to other; it may not be a fair comparison. For fair comparison purpose, we also conducted experiment as per standard AAMI classes using proposed model. The details of ECG segments used for classification according to AAMI standard are given in table 1. Here also, we used same strategy of stratified train-test split with 70:30 as described above. The confusion matrix for 5 AAMI classes is given in Fig. 6.

The Confusion Matrix of AAMI categories shows that 252 beats out of 32813 total test beats were misclassified and had correctness of 99.23%. The highest misclassified beats are from Fusion beats (F) of 16.54%. Among them, 6.62% F beats are wrongly predicted as Ventricular ectopic beat (V) and 9.56% F beats are wrongly predicted as Normal beats (N). Table 4 provides a comparison of the proposed model's beat-wise performance with state-of-the-art approaches in classifying ECG beats into five categories. We have highlighted metric-wise highest value for each beat in table 4. All the comparisons performed here used the AAMI standard of 5 types of ECG beats collected from the MIT-BIH arrhythmia database.

Although the sensitivity and positive predictive

Table 4. Comparison of AAMI 5-classes arrhythmia results with other methods

Method	Metric	N	S	V	F	Q	AVG
CNN [13] (2017)	Sen %	91.64	89.04	94.07	<b>95.21</b>	97.39	93.47
	Spe %	96.01	98.77	98.74	98.67	99.61	98.36
	Ppv %	85.17	94.76	95.08	94.69	98.4	93.62
	Acc %	95.14	96.82	97.84	97.97	99.16	97.39
2D CNN [15] (2019)	Sen %	99.63	84.88	94.88	89.75	<b>99.38</b>	93.70
	Spe %	<b>98.22</b>	98.91	98.28	98.59	98.13	98.43
	Ppv %	93.33	95.10	93.24	94.10	92.98	93.75
	Acc %	98.50	96.10	97.60	96.83	98.38	97.48
Entropy features + KNN[16] (2019)	Sen %	99.59	73.64	92.11	64.46	98.32	85.62
	Spe %	91.92	99.84	99.75	99.94	99.93	98.28
	Ppv %	98.34	92.09	96.37	88.38	99.06	94.85
	Acc %	98.27	99.17	99.25	99.68	99.81	99.24
1D CNN with Focal loss[14] (2020)	Sen %	99.49	77.88	94.54	82.10	98.51	90.50
	Spe %	94.99	99.71	99.70	99.88	99.95	98.85
	Ppv %	98.96	87.65	95.73	83.65	99.31	93.06
	Acc %	98.71	99.16	99.36	99.75	99.84	99.36
AdaBoost + Random Forest [17] (2021)	Sen %	<b>99.95</b>	82.61	<b>97.45</b>	70.88	99.24	90.03
	Spe %	95.86	<b>99.99</b>	99.84	<b>100</b>	<b>99.99</b>	99.14
	Ppv %	99.13	<b>99.52</b>	<b>97.95</b>	<b>100</b>	<b>99.87</b>	<b>99.3</b>
	Acc %	99.24	99.58	99.67	<b>99.79</b>	<b>99.94</b>	99.64
1D CNN + XGBoost	Sen %	99.74	<b>91.85</b>	97.42	83.46	99.32	<b>94.36</b>
	Spe %	97.60	99.88	<b>99.85</b>	99.92	99.97	<b>99.44</b>
	Ppv %	<b>99.51</b>	94.95	97.88	90.08	99.57	96.40
	Acc %	<b>99.38</b>	<b>99.68</b>	<b>99.70</b>	<b>99.79</b>	99.92	<b>99.69</b>

value of class F of CNN model [13] are higher than the proposed method, the same metric values of the remaining four classes are lower than the proposed model. Therefore, when the average sensitivity and positive predictive value are taken into account, the proposed model performance is significantly better. Moreover, the specificity and accuracy of all the five beats of the proposed model is also superior to that of the CNN model [13]. This implies that the CNN model results can be further improved when CNN is integrated with XGBoost. The Q beats sensitivity and N beats specificity of the 2D-CNN model [15] are 0.06% and 0.62% respectively, higher than our technique. However, when the overall results considered, our strategy is much better. Moreover 2D-CNN model requires more memory and processing time. Even though the AdaBoost with the Random Forest model [17] has a higher average positive predictive value, its average sensitivity, specificity and accuracy are lower than the proposed model.

Romdhane et al. [14] handled unbalanced dataset classification of AAMI beats using CNN and focal loss function. But our approach outperformed all evaluation metric values across all five beat categories. This shows that our technique to dealing with unbalanced datasets is considerably superior to Romdhane et al. [14] approach. Acharya et al. [13] and Qayyum et al. [15] used balanced dataset for AAMI based beat classification. Despite using an unbalanced data set, we achieved greater or nearly close sensitivity, specificity, accuracy, and positive predictive value for all the beats except category F beats when compared to results from the balanced dataset. However, when the mean metric values are considered to determine overall performance, our model is the winner. This proves that the proposed model performs effectively even when the data set is unbalanced.

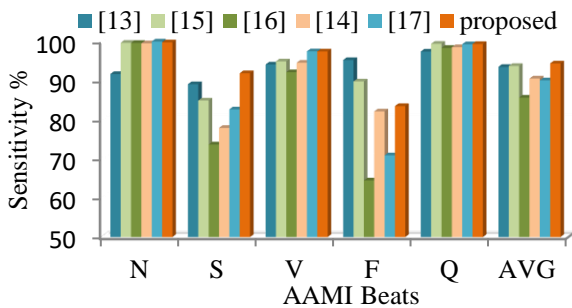


Figure. 7 Comparison of sensitivity of different methods on AAMI 5-classes

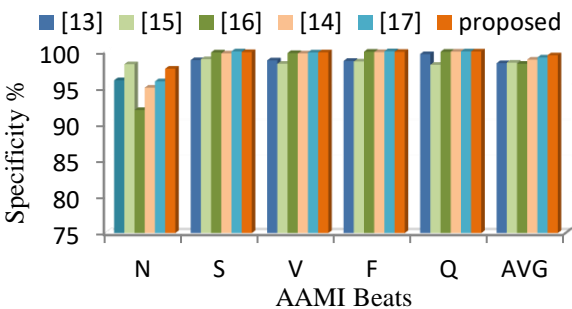


Figure. 8 Comparison of specificity of different methods on AAMI 5-classes

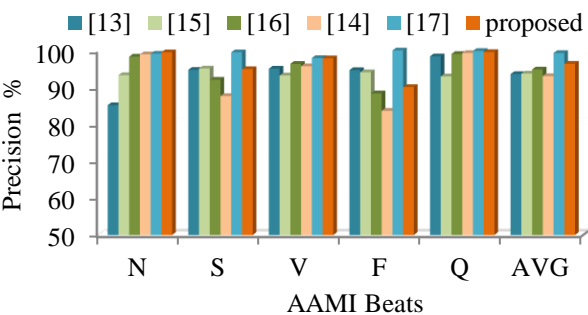


Figure. 9 Comparison of positive predictive value of different methods on AAMI 5-classes

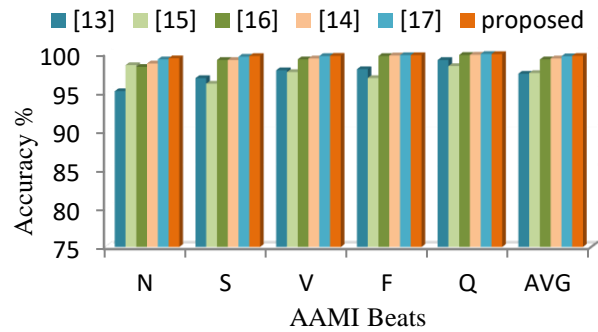


Figure. 10 Comparison of positive predictive value of different methods on AAMI 5-classes

For quick understanding purpose, class-wise sensitivity, specificity, positive predictive value and accuracy of the proposed method along with other recent models on the AAMI 5-classes are graphically presented in Fig. 7, Fig. 8, Fig. 9 and Fig. 10 respectively.

### 5. Conclusion

The novelty of this work is the integration of a deep learning model with traditional machine learning model. The CNN model is used to auto extract features and XGBoost is used to classify the arrhythmia beats. The proposed model obtained overall sensitivity of 92.61%, specificity of 99.85%, precision of 95.99% and accuracy of 99.84% to detect eleven arrhythmia beats of MIT-BIH arrhythmia database. We also tested to classify the five arrhythmia beats according to the AAMI standard and attained overall Sensitivity of 94.36%, specificity of 99.44%, precision of 96.40% and Accuracy of 99.69%. This study demonstrated that combining XGBoost with CNN can improve the classification performance compared to state-of-the-art methods, particularly when the dataset is highly imbalanced.

The main issue with the proposed model is that it takes long time to train. However, the testing time is suitable for use in real-time applications. In future work, we would like to investigate the integration of CNN with other deep learning techniques such as LSTM and BiLSTM in the arrhythmia detection process.

### Conflicts of Interest

The authors declare no conflict of interest.

### Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data collection, writing original draft

preparation, writing review and editing, visualization, have been done by 1<sup>st</sup> author. The supervision and project administration have been done by 2<sup>nd</sup> author.

## References

- [1] [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [2] S. Vallem, "Multi-Class Arrhythmia Detection using a Hybrid Spatial-Temporal Feature Extraction Method and Stacked Auto Encoder", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 2, pp. 82-94, 2021, doi: 10.22266/ijies2021.0430.08.
- [3] O. Cheikhrouhou, R. Mahmud, R. Zouari, M. Ibrahim, A. Zaguia, and T. N. Gia, "One-dimensional CNN approach for ECG arrhythmia analysis in fog-cloud environments", *IEEE Access*, Vol. 9, pp. 103513-103523, 2021.
- [4] S. Nurmaini, R. U. Partan, W. Caesarendra, T. Dewi, M. N. Rahmatullah, A. Darmawahyuni, V. Bhayyu, and F. Firdaus, "An automated ECG beat classification system using deep neural networks with an unsupervised feature extraction technique", *Applied Sciences* Vol. 9, No. 14, p. 2921, 2019.
- [5] Z. Yue and Z. Jinjing, "Atrial Fibrillation Detection Based on EEMD and XGBoost", *Journal of Physics: Conference Series*, Vol. 1229, No. 1, p. 012074, 2019.
- [6] H. Shi, H. Wang, Y. Huang, L. Zhao, C. Qin, and C. Liu, "A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification", *Computer Methods and Programs in Biomedicine*, Vol. 17, pp. 1-0, 2019.
- [7] H. Yang and Z. Wei, "A Novel Approach for Heart Ventricular and Atrial Abnormalities Detection via an Ensemble Classification Algorithm Based on ECG Morphological Features", *IEEE Access*, Vol. 9, pp. 54757-54774, 2021.
- [8] H. S. Chiang, D. H. Shih, B. Lin, and M. H. Shih, "An APN model for arrhythmic beat classification", *Bioinformatics*, Vol. 30, pp. 1739-1746, 2014.
- [9] F. Akdeniz, I. Kayikcioglu, and T. Kayikcioglu, "Classification of cardiac arrhythmias using Zhao-Atlas-Marks time-frequency distribution", *Multimedia Tools and Applications*, Vol. 80, pp. 1-15, 2021
- [10] G. Latif, F. Y. A. Anezi, M. Zikria, and J. Alghazo, "EEG-ECG Signals Classification for Arrhythmia Detection using Decision Trees", In: *Proc. of 2020 Fourth International Conference on Inventive Systems and Control*, pp. 192-196, 2020.
- [11] J. Park, M. Kang, J. Gao, Y. Kim, and K. Kang, "Cascade classification with adaptive feature extraction for arrhythmia detection", *Journal of Medical Systems*, Vol. 41, No. 1, pp. 1-12, 2017.
- [12] P. Amorim, T. Moraes, D. Fazanaro, J. Silva, and H. Pedrini, "Shearlet and contourlet transforms for analysis of electrocardiogram signals", *Computer Methods and Programs in Biomedicine*, Vol. 161, pp. 125-132, 2018.
- [13] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and T. R. San, "A deep convolutional neural network model to classify heartbeats", *Computers in Biology and Medicine*, Vol. 89, pp. 389-396, 2017.
- [14] T. F. Romdhane, H. Alhichri, R. Ouni, and M. Atri, "Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss", *Computers in Biology and Medicine*, Vol. 123, 2020.
- [15] A. B. A. Qayyum, T. Islam, and M. A. Haque, "ECG Heartbeat Classification: A Comparative Performance Analysis between One and Two Dimensional Convolutional Neural Network", In: *Proc. of 2019 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health*, pp. 93-96, 2019.
- [16] M. Sharma, R. S. Tan, and U. R. Acharya, "Automated heartbeat classification and detection of arrhythmia using optimal orthogonal wavelet filters", *Informatics in Medicine Unlocked*, Vol. 16, p. 100221, 2019.
- [17] R. Li, W. Zhang, S. Shen, J. Yao, B. Li, B. Zhou, G. Chen, and Z. Wang, "An Intelligent Heartbeat Classification System Based on Attributable Features with AdaBoost+ Random Forest Algorithm", *Journal of Healthcare Engineering*, Vol. 2021, 2021.
- [18] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing", *IEEE Transactions on Dependable and Secure Computing*, Vol. 18, No. 3, pp. 1441-1455, 2019.
- [19] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning", In: *Proc. of 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, pp. 2319-2323, 2018.

- [20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", In: *Proc. of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining 2016*, pp. 785-794, 2016.
- [21] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database", *IEEE Engineering in Medicine and Biology Magazine*, Vol. 20, No. 3, pp. 45-50, 2001.
- [22] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm", *IEEE Transactions on Biomedical Engineering*, Vol. BME-32, No. 3, pp. 230-236, 1985.