Detection and classification of Cardiac Arrhythmia using electrocardiogram (ECG)

Paras Nagpal[a], Pradumn Gupta[b], Paawan Singh Sohal[c]

[a,b,c]Delhi Technological University

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract: Cardiac Arrhythmia refers to a condition in which the heart beats irregularly, fast or slow. There are several types of arrhythmia (bradyarrhythmia, premature heartbeat, tachycardia, ventricular fibrillation, etc.). Out of these some can be very dangerous and even life threatening if not treated quickly. In this paper we try to propose a solution for detecting and then classifying various types of arrhythmia. We have used UCI Arrhythmia dataset to train and test our machine learning models.

Keywords: Electrocardiogram (ECG), Support Vector Machine (SVM), K-nearest neighbors (KNN), Convolutional neural network (CNN), classification.

1. Introduction

People are suffering from various chronic disease, nowadays, and heart diseases are one of such diseases which affect a large part of the population. Early diagnosis and detection can be lifesaving in these cases. Most commonly used method of diagnosing the heart function is an ECG which is made up of P waves, T waves and QRS complex. Irregularities in this can be a clear symptom of Cardiac Arrhythmia.

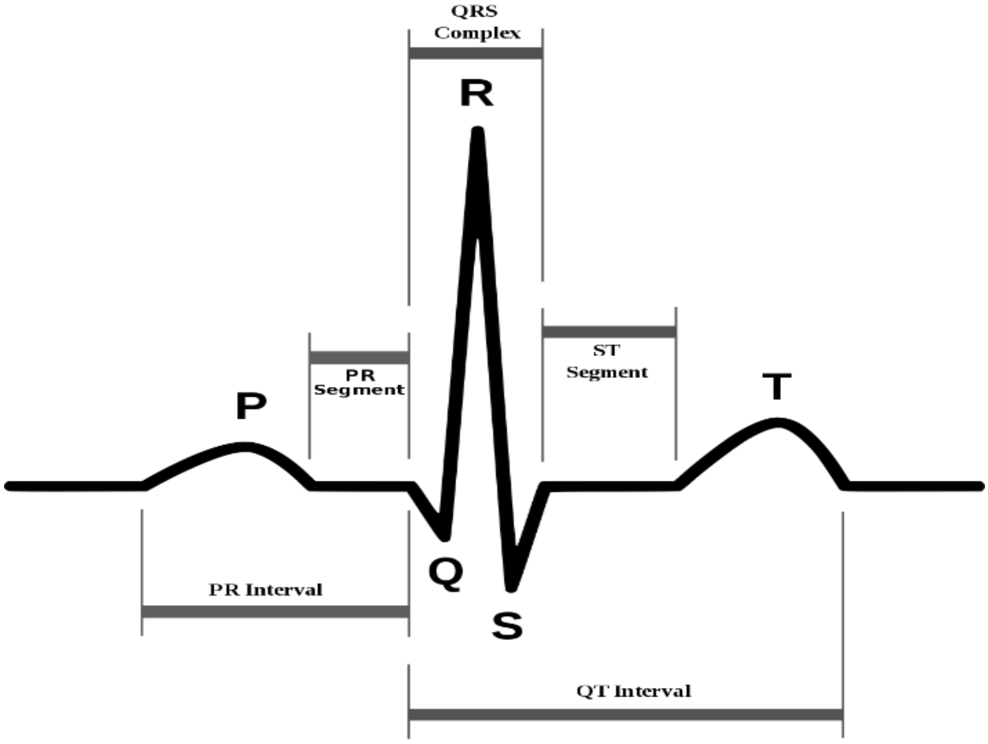


Fig 1: ECG diagram with QRS complex

Arrhythmia refers to a condition in which the normal smooth rhythm of the heart is disrupted. The heart's electrical signal becomes slow, fast, irregular or broken. The two main types of arrhythmia are i) Bradycardia causes the heart to beat very slow (as low as 60bpm), ii) Tachycardia causes the heart to beat as fast as 100 bpm.

In this paper we have proposed the classification of the ECG as normal or having arrhythmia and then classifying it for the type of arrhythmia with several different approaches like SVM, kNN, and CNN.

2. Methodology

We take the ECG data from the dataset available and extract the important features to be used in classification using techniques such as principal component analysis.

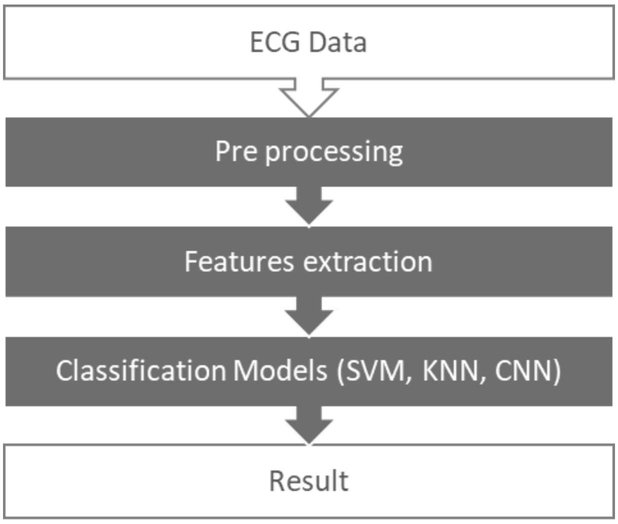


Fig 2: Steps involved to classify ECG

Before training the data with our machine learning models, we pre-process the data for data cleaning (removing missing values, handling outliers, etc.).

The following are the steps involved in our prediction architecture:

A. Pre-processing

Data preprocessing is a very crucial step before using the data for machine learning algorithms to predict accurate values. In this research we employ these techniques:

i. Removing data columns where there are too many missing values,

ii. Replacing empty entries with relevant data (we used SimpleImputer to substitute all the missing values with the median value for that column/attribute),

iii. Scaling/Normalizing the data to make it more uniform (we used StandardScaler function to normalize the data).

B. Features extraction

Feature extraction is the process of selecting important features and reducing the dimensions of the data to make it more manageable. It has the benefit of making the computation process more efficient and less time consuming.

We use the following features extraction techniques:

1. Principal Component Analysis

Principal Component Analysis or PCA is a statistical procedure used for feature selection and feature extraction. We try to map various principal components with respect to the given data points and find out which components cover or represent the variance in the correlated variables. In simple terms, PCA is mainly used to highlight and quantify the similarities and differences between features in the data set. The number of components is always less than or equal to the count of attributes in the data set. The first principal component always captures the largest variance in the data set followed by subsequent orthogonal principal components. In our dataset applying PCA to capture the components with maximum variance will remove time domain features and retain the wavelet features as majority of wavelet features are captured when we have 95% cumulative variance.

We have decided the number of components based on the plots of the following.

- Plot of Eigen values v/s No of components.

- Plot of Percent of cumulative variance v/s No of components.

Based on the plots, we select the components whose cumulative variance captures 95% of variability.

By performing principal component analysis on the given data set, we have obtained information which are compiled into the plot in Figure

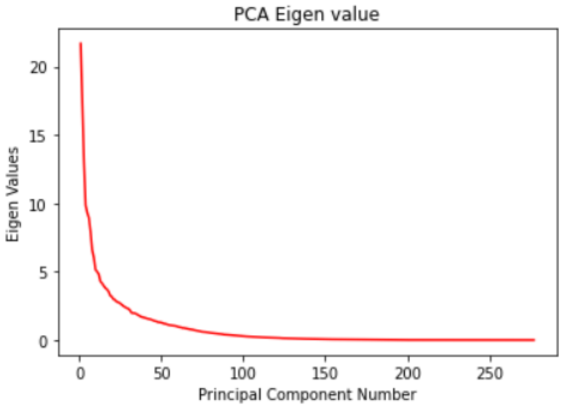


Fig 3: Principal Component vs Eigen values

By inspecting the plots, we estimate that approximately 88 principal components have cumulative variance captures 95%variability. Therefore, we select these 88 features to predict the class labels.

2. Random Forest

This model works on a portion of the data set by continuously sampling with replacement and then fitting a decision tree to the model. Each decision tree is a sequence of yes-no questions based on a single or combination of features. All the features are not considered by the tree, which confirms that individual decision trees are not co-related. Hence the classifier less prone to over-fitting.

The measure of impurity is by Gini index. By implementing Random forests, we are selecting desired attributes which does not cause model over-fitting.

The Hyperparameter for Random forest is the number of classifiers considered. We have set this value to 20. This is done to ensure that there is no bias in selecting the attributes from the dataset. If a feature is selected, then we can be sure that it was selected by a majority of decision trees generated by Random forests.

C. Models

Machine learning models are algorithms which are used to predict the values of unknown input depending on the patterns in the input output combination of previously known data. The following machine learning models have been implemented for this research paper:

1. Support Vector Machine (SVM)

Support vector machines or SVM is a supervised machine learning model used for classification and regression analysis. Support vector machine creates a hyperplane in N dimensions which is used to classify the data points. These hyperplanes act as decision boundaries which help to classify the data points. Points lying on opposite sides are classified to be from different classes. The dimensions of the hyperplane depend upon the count of features that are inputted. If the number of input features are n, then this hyperplane has a dimension of n-1. For example, if the count of input features is 2, then the hyperplane is a line.

We have used the sklearn library for implementing the SVM Model. SVM uses kernels which are a combination of mathematical functions. A kernel function changes input data into a required form. There are 9 kernel functions available. We have implemented the following kernels:

- Linear
- Polynomial
- Gaussian Radial Basis Function (RBF)
- Sigmoid function

Linear kernel is best suited for our model as the data set is linearly separated.

In this project, SVM has been implemented using linear kernel, polynomial kernel and radial basis kernel using features selected by principal component analysis and random forest.

2. K-Nearest Neighbors or kNN

The k Nearest Neighbors or kNN is an unsupervised machine learning algorithm which can be used for classification and regression problems. In k Nearest Neighbors classification the classification is done using the Euclidean distance of each test point with the training data and then the classification is done using the value of k. For k=1, the test data point is classified same as the point in training data having the least distance from the concerned test point. In K Fold KNN algorithm the complete data is divided in some number of folds. Each fold is considered as a test data set at least once and prediction is made on it based on the other folds which are aggregated as a training data set. According to this algorithm, we calculate the Euclidean distance between every data point in the training data set with the test data point. The value of k is then decided for which the test accuracy is maximized. The class variable of the test data point is predicted using the class variables of the k nearest neighbors.

KNN is desirable in areas where there is less information about the data set. For example, there may be outliers in the data set or redundancy for which we may want to incorporate other rules to queries that don't fit well in the dimensional space in which the KNN algorithm runs in.

In this project we have used k Fold k Nearest Neighbors algorithm for classifying the data because the UCI database contains highly unbalanced data with 16 class labels having disproportionate values. With the K fold KNN we cross validate all the data which is important to get better accuracy.

3. 1-D Convolutional Neural Network

We have used convolutional neural network because it works quite well for classification problems. We have used keras.layers.Conv1D and fully connected layers for building the model. This approach involves making a block of convolutions and then flattening as required by the connected layers to create class labels. Since Conv1D model is in tabular form we use it here. To deal with imbalanced dataset we have used weighted loss function.

More weights are allocated to the labels which occur less in the dataset using sklearn compute_class_weights.

To get the final output we have used Adam optimizer with a softmax layer.

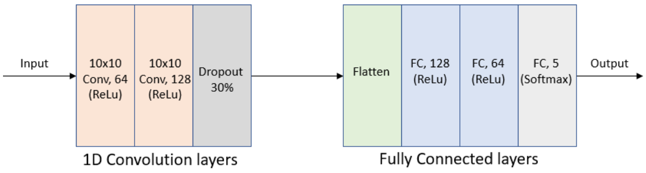


Fig 4: 1D CNN Layer Architecture

3. Experiments and Results

The dataset used in UCI Arrhythmia dataset (https://archive.ics.uci.edu/ml/datasets/Arrhythmia) which consists of arrhythmia data for 452 (rows) patients each of which contain 279 attributes (columns). The patients in the dataset are classified in one out of the 16 classes as mentioned in table 1.

Some of the important features from the dataset include age, gender, PQRST wave signal, height and channel signal information etc. The following table contains the Arrhythmia class names and the key factors for those classes as well.

Arrhythmia class and Key factors:	**No. of Instances**
1. **Normal** - QRS complexes related to R-peak	245
2. **Ischemic changes** - QRS related to R-peak	44
3. **Old Anterior Myocardial Infarction**- RR interval model & PR interval variability with P similarity	15

4. **Old Inferior Myocardial Infarction** - interval model & PR interval variability with P similarity	15
5. **Sinus tachycardia -** RR interval irregularity	13
6. **Sinus bradycardia -** RR interval irregularity	25
7. **Ventricular Premature Contraction (PVC)** - QRS complexes related to R-peak	3
8. **Supraventricular Premature Contraction** - QRS complexes related to R-peak	2
9. **Left bundle branch block -** QRS complexes related to R-peak	9
10. **Right bundle branch block -** QRS complexes related to R-peak	50
11.**1. degree Atrio Ventricular block** - QRS complexes related to R-peak	0
12**.2. degree AV block** - QRS complexes related to R-peak	0
13**.3. degree AV block** - QRS complexes related to R-peak	0
14.**Left ventricule hypertrophy** - QRS complexes related to R-peak	4
15. **Atrial Fibrillation or Flutter** - Absence of P waves, presence of f-waves in TQ interval	5
16. **Others**	22

Table 1: Arrhythmia dataset class distribution

1. Support Vector Machine (SVM)

Support Vector Machines algorithm was applied along with the feature selection approaches PCA and Random Forest. With the kernels and the critical factors as the hyperparameters, we obtained a maximum accuracy of 73.53% with linear kernel (Critical factor, c = 0.01(PCA),c = 0.1(Random Forest Classifier)) for both the feature selection methods. The graph plot in Figure 5 and Figure 6 shows the comparisons between the accuracies of the kernel functions for corresponding critical factors for both the feature selection approaches. Figure 7 displays the classification report for the best model.

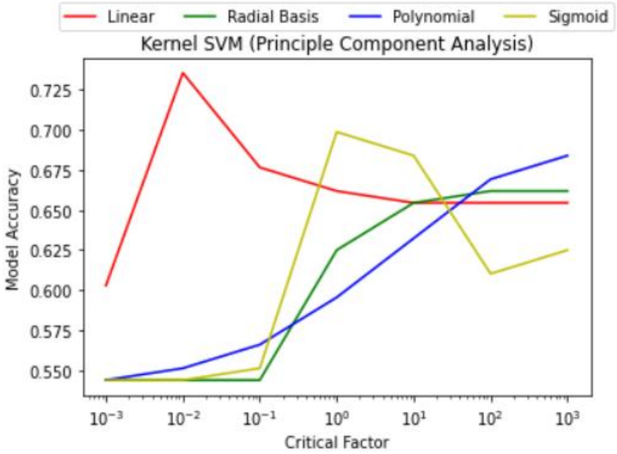


Fig 5: Critical factor vs model accuracy for linear, radial basis, polynomial and sigmoid kernel functions (PCA)

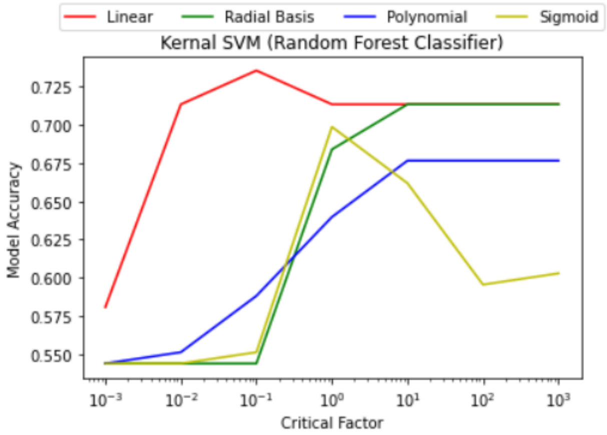
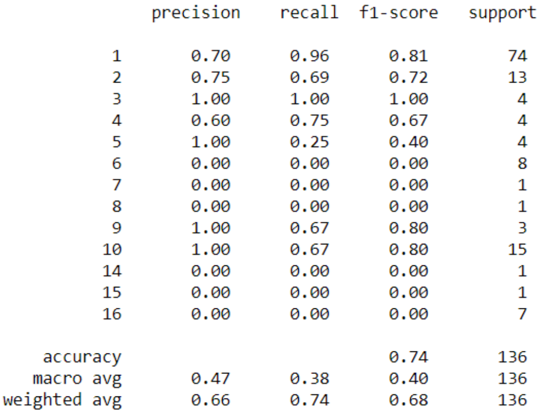


Fig 6: Critical factor vs model accuracy for linear, radial basis, polynomial and sigmoid kernel functions (Random Forest)



```
              precision    recall  f1-score   support

           1       0.70      0.96      0.81        74
           2       0.75      0.69      0.72        13
           3       1.00      1.00      1.00         4
           4       0.60      0.75      0.67         4
           5       1.00      0.25      0.40         4
           6       0.00      0.00      0.00         8
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         1
           9       1.00      0.67      0.80         3
          10       1.00      0.67      0.80        15
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         7

    accuracy                           0.74       136
   macro avg       0.47      0.38      0.40       136
weighted avg       0.66      0.74      0.68       136
```

Fig 7: Classification report for linear SVM

2. k Nearest Neighbors (kNN)

We have generated accuracy scores of the model for multiple values of k for both the feature selection methods. We have found that for value of k = 5, maximum accuracy is 61.27% for PCA and for value of k = 3, a maximum accuracy of 65.49%. The data set was split into 7 folds initially and K Fold cross validation was performed for various values of k. We also observed that the accuracy was slightly more for random forest than for PCA.

The graph plots for the accuracy scores mentioned are present in fig 8 and fig 9 respectively.

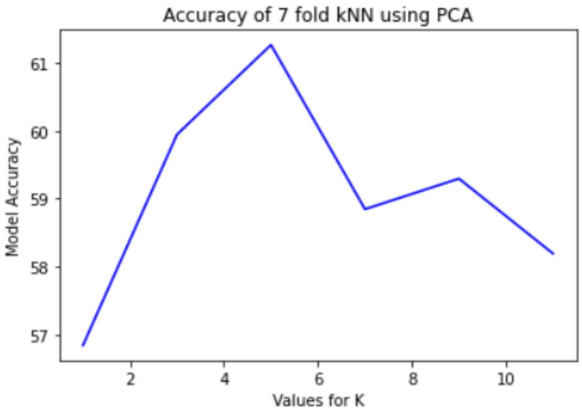


Fig 8: k vs Accuracy (PCA)

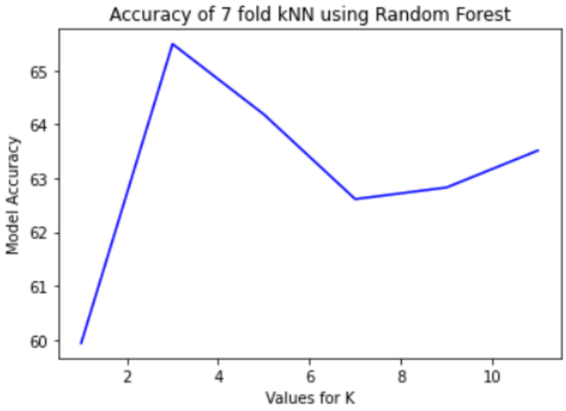
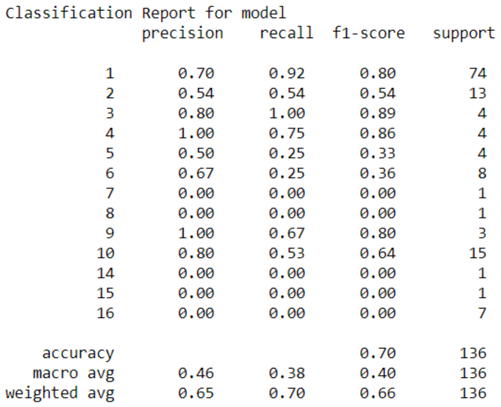


Fig 9: k vs Accuracy (Random Forest)

3. 1-D Convolutional Neural Network

The convolution layers with 64 filters and increasing 128 filters are added as a block of two respectively. The activation function for the convolutional block is RELU with the fixed kernel size of 10. In order to retain the important features from the block, we initialize a dropout of 0.3 to the convolution maxpool. The fully connected dense layers with 128 and 64 units are added in order to extract the features specific to the class and the predictions are extracted from the softmax layer with Adam optimizer having the default learning rate of 0.001. We specify the precision, recall and f1-score values for the individual arrhythmia classes along with the macro and weighted averages as shown in Figure 10.

With the above specified hyperparameters we achieved the best accuracy of 69.85 % for imbalanced test data of 136 records. Since all the layers are trainable, the hyperparameters can be tuned is performed for all layers.



```
Classification Report for model
             precision    recall  f1-score   support

           1       0.70      0.92      0.80        74
           2       0.54      0.54      0.54        13
           3       0.80      1.00      0.89         4
           4       1.00      0.75      0.86         4
           5       0.50      0.25      0.33         4
           6       0.67      0.25      0.36         8
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         1
           9       1.00      0.67      0.80         3
          10       0.80      0.53      0.64        15
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         7

    accuracy                           0.70       136
   macro avg       0.46      0.38      0.40       136
weighted avg       0.65      0.70      0.66       136
```

Fig 10: Classification report for 1D CNN

3. Conclusion

In this paper we have compared three classification techniques for the classification of ECG data from the UCI Cardiac Arrhythmia dataset. The classification has been done for one of the 16 classes available in the dataset. We observed that we obtained an accuracy of 73.53% in linear kernel SVM with both the feature selection methods.

References

[1] N. Kalkstein, Y. Kinar, M. Na'aman, N. Neumark and P. Akiva, "Using machine learning to detect problems in ECG data collection," 2011 Computing in Cardiology, Hangzhou, 2011, pp. 437-440.

[2] Principal Component Analysis - https://en.wikipedia.org/wiki/Principal_component_analysis

[3] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique Arrhythmia, Cardiac in Adults with Congenital Heart Disease". (n.d.). SpringerReference. doi:10.1007/springerreference_109215

[4] Hamza, S., & Ayed, Y. B. (2020). N Support Vector Machines - https://data-flair.training/blogs/svm-support-vector-machine-tutorial/SVM Kernel Functions - https://data-flair.training/blogs/svm-kernel-functions/

[5] Electrocardiography - https://en.wikipedia.org/wiki/Electrocardiography

[6] Novitasari, H B, Nur Hadianto, Sfenrianto, A Rahmawati, Risha Prasetyo, Jaja Miharja and Windu Gata. "K-nearest neighbor analysis to predict the accuracy of product delivery using administration of raw material model in the cosmetic industry (PT Cedefindo)." (2019).