Arrhythmia Prediction and Diagnosis using Data Analysis

Mangalnathan Vijayagopal
mvijaya2@ncsu.edu
NC State University
Raleigh 27606

Shreyas Muralidhara
schikkb@ncsu.edu
NC State University
Raleigh 27606

Nischal Kashyap
nkashya@ncsu.edu
NC State University
Raleigh 27606

Pawandeep Mendiratta
psmendir@ncsu.edu
NC State University
Raleigh 27606

ABSTRACT

To detect and predict the type of arrhythmia based on Electrocardiogram (ECG) tool using machine learning models and algorithms. We will be training a model using a given dataset and then use test data to classify instances with unknown class labels.

KEYWORDS

datasets, support vector machines, neural networks, classifiers, accuracy

1 BACKGROUND AND INTRODUCTION

Problem Statement

Most cardiac disorders cause irregularities in heartbeat. These irregular patterns in rhythm of heartbeat is called Arrhythmia. Electrocardiogram (ECG) [1] is the most preferred tool used by clinical practitioners to capture heartbeat. ECG is renowned to be cost-effective, easy to use and noninvasive to the human body. However, Physicians may not interpret Electrocardiogram for large data sets effectively as it is time consuming and can also cause miss-classification of beats. They also cannot effectively identify the normalities and abnormalities in the heartbeat.

Although single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. Therefore, automatic detection of arrhythmia beats from ECG signals is a significant task in the field of cardiology. To eradicate the complexity and possibility of human error in diagnosing, we leverage the computational power and indefatigability of machine learning models[1][2].

UCI Machine learning repository transformed the ECG signals into QRS complexes(column 10) based on the R-peak(column 19) of 17 different types of beats. Our approach includes Weighted class models - Proportionate sampling for all the 17 classes of beats because of a high standard deviation in the number of samples produced per class. QRS complex extraction based on annotated files with making R-peak as the centre and of constant size. Designing Machine learning model including deep neural network and calibrate hyperparameters to obtain the best results.

The data set will be split accordingly (70/30 rule) into training and testing data. We will be using 1D-CNN, MLP, SVM and KNN with k-fold cross validation models to achieve the objectives.

2 RELATED WORK

We have referred the paper Automated Screening of Arrhythmia Using Wavelet Based Machine Learning Techniques[3]

The paper summarizes that different techniques have been used to extract R point in the ECG delineation. Most of the methods use compression of time domain features using Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA).

PCA or any other feature compression technique should condense the features better than the time domain counterpart method. Classification using 3 different algorithm - Support vector machine with various kernels(SVM), error back propagation Neural network. Gaussian mixture models(EBPNN) and Gaussian Mixture models(GMM).
We infer that PCA would be better for feature selection than LDA from further analysis. Implement Random forest in contrast with the PCA.

Block diagram of the proposed scheme was implemented with the suggested models - Support vector Machine along with various kernels. Neural network classifier models - 1 Dimension Convolution Neural Network and Back-propagation implementation using Simple network of Multi layer perceptron (MLP) approach. As novelty we have implemented kFold kNN even though it was not suggested by block diagram. Hence we achieved the lowest accuracy for the kNN

model.

We have also referred Machine Intelligent Diagnosis of ECG for Arrhythmia Classification Using DWT, ICA and SVM techniques [4].
In the paper, it is mentioned that Arrhythmia class can be grouped in 5 major classes Non-ectopic (N), Supraventricular ectopic(S), Ventricular ectopic (V), Fusion (F) and Unknown (U) for MIT-BIH arrhythmia dataset de-noised ECG R–peak is detected using Pan–Tompkins algorithm. R–peak detected signal is segmented, such that each segment consists of 99 samples before R–peak and 100 samples after R-peak. Each of these 200 samples of cardiac beats of five arrhythmia classes are used in this study. SVM separates the Binary labeled training features with maximum margin from the hyperplane.
We infer that most of the Classes of cardiac arrhythmia can be classified by linear separation, but when Linear separation is not possible we can use non linear kernel transformations for non linear mapping to higher dimensional feature space.

In the paper, An integrated ECG feature extraction scheme using PCA and wavelet transform[5], it is mentioned that Novel feature extraction on ECG using discrete random wavelet provides many features in time. Using PCA for feature selection compresses the features in time domain. Hence wavelet features contribute more significant than time domain features for arrhythmia class detection.
We found out that 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.As novelty we have implemented Random forest classifier as Feature selection which ic not found in any of references and it performs equally well in SVM model and In fact better in K fold - kNN models.

Finally in the paper, Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy[6], we inferred that ventricular response analysis is based on the predictability of the inter-beat timing ('RR intervals') of the QRS complexes in the ECG. RR intervals are derived from the most obvious large amplitude feature in the ECG, the R-peak, the detection of which can be far more noise resistant. This approach may therefore be more suitable for automatic, real-time AF detection.

Therefore we conclude that, the QRS complexes along with R peak detect the presence of cardiac arrhythmia. They also classify the majority of cardiac arrhythmia Ischemic arrhythmia, Ventricular, Super Ventricular and Artrio Ventricular arrhythmia

3 METHODS

Approach

As noted in the introduction, the machine learning algorithms used for training the prediction model include Support Vector Machines (SVM), 1 Dimensional Convoluted Neural Network, Multilayer Perceptron (MLP), and K-Nearest Neighbour(KNN) to achieve our objectives.

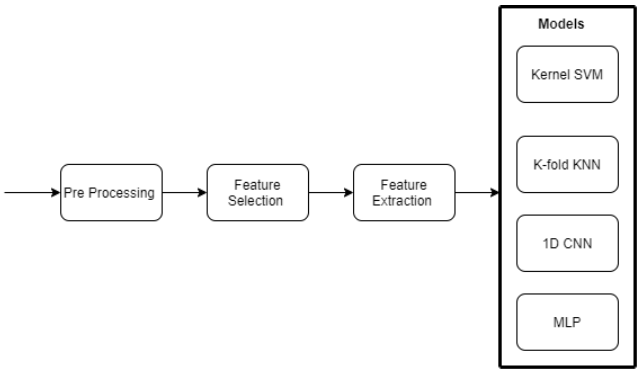

Figure 1: Arrhythmia Model Architecture

Before training, the first step in exploratory data analysis is data pre-processing. Data pre-processing involves cleaning the data to ensure that the data set fed to the model is consistent, clean and meaningful. This involves handling missing values, scaling the values of the attributes of the data set, and selecting only those attributes which contribute towards predictions.

This document will explain the complete steps used for arrhythmia prediction and classification which will contribute valuable information to medical institutes for patient diagnosis

Data Preprocessing

Data Preprocessing is a crucial step in exploratory data analysis. It involves ensuring that clean data is fed to the machine learning models so that it can clearly use this information to provide accurate predictions.

We have performed the following steps for data preprocessing in the following order:

- **Remove unwanted columns** - Deleting the attributes having more than 40% missing values as they do not contribute effectively for predictions.
- **Replace missing values** - Impute the missing values by replacing them with the attribute median. Median is chosen over mean because the mean is more susceptible to outliers.
- **Attribute Scaling** - Normalize the attributes so that values are scaled up by processing. Scaling is necessary

because there might be values which might be too large compared to others.

We found out that there was only column (feature 14) which was eligible to be removed because it had more than 40% missing values across rows.

We used SimpleImputer[7] which is a function in python sklearn[8] package to replace all the remaining missing values with the median of values in the column where the missing value was present.

Similarly, we used the StandardScaler[9] (sklearn) functionality to scale all the values to a similar range and at the same time retain their proportional differences.

Data Splitting

After data preprocessing the next step is to split the data set into training and testing data. We have followed the 70-30 rule for splitting. 70% of the entire data set is used for training the model. The remaining 30% was used for testing the accuracy of the trained model. We are using random data split with stratified sampling to ensure equal distribution of class labels in training and testing data respectively. In other words, all the class labels have an equal chance to be considered for either training data or testing data.

Feature Selection

Here we select features which predominantly contributes to the prediction of class labels. First, we remove the categorical features for which 95% of all the values were either completely 0s or completely 1s. We primarily use two techniques for feature selection.

- Random Forests
- Principal Component Analysis

Principal Component Analysis

Principal Component Analysis[11] 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 number 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 1

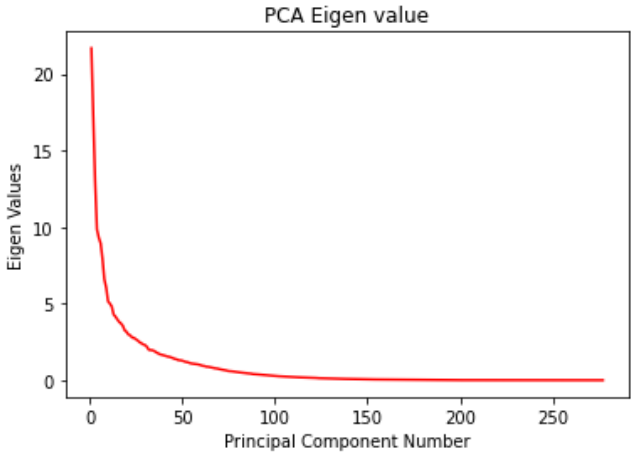

Figure 2: Principal Component vs Eigen Values

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.

Random Forests

This model[10] 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.

All the wavelet features were better extracted by Random forest than PCA. Hence resulting in slightly more features than PCA. Hence the Novelty idea of random forest seems to work better.

By implementing Random Forests on the given data set, we have found that only 99 features (columns) out of 279 contribute predominantly to the prediction of the class label.

4 MODELS

Machine learning models are functions or algorithms which are used to predict output for an unknown input based on previous patterns of input-output combinations.

Support Vector Machines

Support vector machines[12] or SVM is a supervised machine learning model used for classification and regression analysis.

The objective of a support vector machine is to create a hyperplane in an N dimensional space that distinctly classifies the data points. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. The dimensions of the hyperplane depends upon the number of input features. For example, if the number of input features is 2, then the hyperplane is a line. If the number of input features is 3, then the hyperplane is a 2D plane.

According to SVM, firstly, we must find the points that lie closest to all the other data points. These points are called support vectors. Next, we find the distance between the dividing plane and the support vectors. This distance is called as margin. The main goal is to maximize this margin to obtain an optimal dividing plane also known as the hyper-plane.

For our project, we have made use of the sklearn library for implementing the SVM Model.

SVM uses a set of mathematical functions known as kernels[13] . A kernel function transforms input data into a required form. There are 9 kernel functions available. We have implemented the following kernels:

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

Linear kernel is best suited for our model as the data set is linearly separated. For novelty, we are comparing linear kernel function with polynomial kernel and radial basis kernel to verify that the accuracies are lower with polynomial and radial basis kernel functions.

Our SVM implementation includes regularization of hyperparameters using critical factors ranging from 10^{-3} to 10^{3}.

We have implemented SVM using linear kernel, polynomial kernel, and radial basis function kernel for the features selected by PCA and Random forests. We can ascertain that the accuracy scores from Random Forest features and PCA features are similar.

k-Nearest Neighbors

The K Fold k-nearest neighbors (KNN)[14] algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems.

All the wavelet features were better extracted by Random forest than PCA. Hence resulting in slightly more features than PCA. Hence the Novelty idea of random forest seems to work better.

The basic idea of a K Fold KNN algorithm is to find the k nearest points in a training data set to a test data point and predict the class of the test data point based on the nearest data entities.Here the global data set is divided into 5 folds. Every 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. The nearest points are decided on the basis of Euclidean distance or Minkowski distance calculations.

According to our KNN algorithm, we initially calculate the Euclidean distance between every data point in the training data set with the test data point. Gradually we select the k nearest neighbors from the training model which has the least euclidean distance with respect to the test data point. 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.

The dataset Cardiac Arrhythmia is more of an imbalanced dataset which has 16 class labels with disproportionate class values(with class labels missing for 11,12 and 13). It is important for us to cross validate every data point in order to obtain higher efficiency. This is where K Fold KNN[16] Algorithm comes into place. With the help of this algorithm, we test every data point and classify them to their respective class labels.

1-D Convoluted Neural Network

Since this is a classification problem, the convolutional model works the best as a classifier. We are using keras.layers.Conv1D library along with fully connected layers for building the model. The approach includes creating a single block of convolutions and flattening the results as required by the connected layers to generate the class labels.
We are using Conv1D model because it is tabular data. Since

our data is imbalanced (the distribution of class labels is not uniform across the dataset), we will be using ***weighted loss function*** to penalize misclassification of labels. The labels which occur less in dataset cannot afford to be missclassified. Therefore higher weights are assigned to such classes using sklearn's compute_class_weights package.
We are using ***Adam*** optimizer with a softmax layer in order to obtain the final predictions.

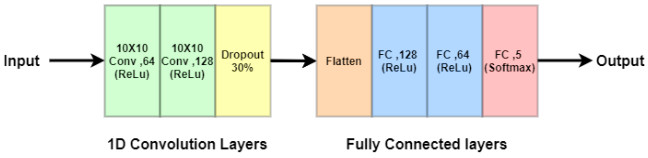

Figure 3: 1D-CNN Layer Architecture

Multi Layer Perceptron

Multi Layer Perceptron or MLP, is a feed-forward neural network which are fully connected with multiple single neurons. The input layers in the model are fully connected to the hidden layer and the output layer is fully connected to the hidden layer. Such an implementation is called as Deep Neural Network. For this model, we will be implementing the ***weighted loss function*** to compensate the imbalanced classes in the training data set.[1]

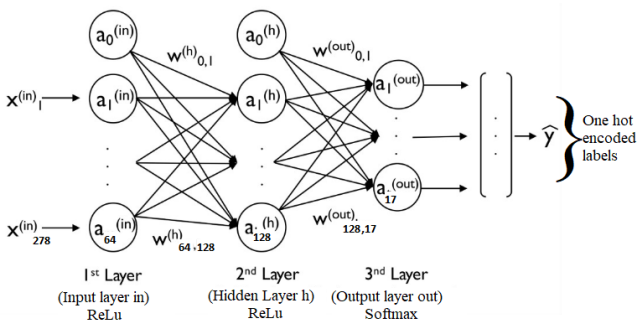

Figure 4: Multi Layer Perceptron Architecture

Rationale

We implemented SVM because it is the most reliable model for the stratified arrhythmia data set for classifying and regression.
We implemented KNN because this model requires no training before making predictions. Therefore, new data can be seamlessly added to the model without impacting the accuracy scores.
Neural Networks are famous for high classification accuracy. This is because the deep layers of a neural network represents hierarchical and non-linear combination of features and patterns detected from the input. Therefore, instead of hand coding essential features, neural network autonomously chooses features resulting in high classification accuracy. This justifies our choice of neural networks (MLP and 1D-CNN) for arrhythmia classification.
Multi Layer Perceptrons (MLP) are universal approximators which can be used to create mathematical models using regression analysis. Since classification is a form of regression when the class labels are categorical, MLPs make good classifier algorithms.
1D-CNN is generally more preferred for image classification, but flattened images are equivalent to 1D-CNN with multiple attributes. Since the waveform is converted into attributes in the data set, 1D-CNN is also a good choice as a machine learning model for this data set.

5 EXPERIMENTS AND RESULTS

Dataset

We are using the UCI Machine Learning repository (http://archive.ics.uci.edu/ml/datasets/Arrhythmia) which comprises of a data set containing arrhythmia data for 452 (rows) patients each of which contain 279 attributes (columns). These patients are classified into 1 out of 16 types of Arrhythmia (class labels). Our feature space of 279 attributes includes patient information such as gender, age, PQRST wave signal, height and channel signal information[2].

Class names key factors and the distribution of the for UCI Arrhythmia dataset is described in [Table 1].

Hypothesis

After thorough analysis of the dataset and diligent review of related work, we have come up with the following hypothesis:
(1) Predicting the common types of arrhythmia from ECG data by detecting patterns in QRS complexes based on the R-peak values which is derived from the dataset. If the ECG waveform is between 30ms to 60 ms or beyond that i.e RR peak width is beyond 60ms then the data is considered as probability of arrhythmia. Our main goal is to predict the non - arrhythmia results as accurately as possible.

(2) Perform proportionate sampling for all 16 classes of heartbeats due to high standard deviation in the number of samples by using weighted class models.

Support Vector Machines

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

Arrhythmia class and Key factors:	No. of Instances
1. **Normal** - QRS complexes based on the R-peak	245
2. **Ischemic changes** - QRS complexes based on the 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 tachycardy** - RR interval irregularity	13
6. **Sinus bradycardy** - RR interval irregularity	25
7. **Ventricular Premature Contraction (PVC)** - QRS complexes based on the R-peak	3
8. **Supraventricular Premature Contraction** - QRS complexes based on R -peak	2
9. **Left bundle branch block** - QRS complexes based on R-peak	9
10. **Right bundle branch block** - QRS complexes based on R-peak	50
11.**1. degree AtrioVentricular block** - QRS complexes based on R-peak	0
12.**2. degree AV block** - QRS complexes based on R-peak	0
13.**3. degree AV block** - QRS complexes based on R-peak	0
14.**Left ventricule hypertrophy** - QRS complexes based on 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: Class distribution for UCI Arrhythmia data

kernel (Critical factor, c = 0.01(PCA),c = 0.1(Random Forest Classifier)) for both the feature selection methods.

The graph plot in Figure 4 and Figure 5 shows the comparisons between the accuracies of the kernel functions for corresponding critical factors for both the feature selection approaches. Figure 6 displays the classification report for the best model.

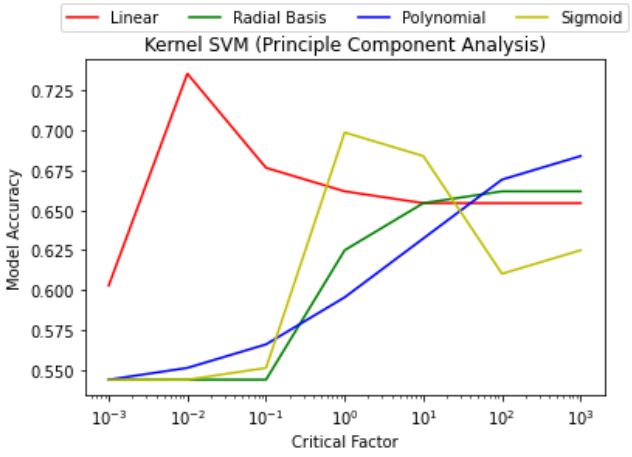

Figure 5: SVM Accuracies for PCA Features

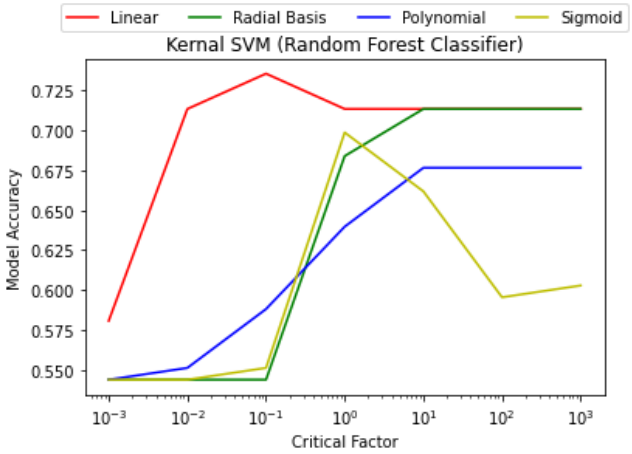

Figure 6: SVM Accuracies for Random Forest Features

k-Nearest Neighbors

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 found in Figure 4 and Figure 5 respectively.

1D-CNN

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

	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

Figure 7: Classification report for Linear SVM

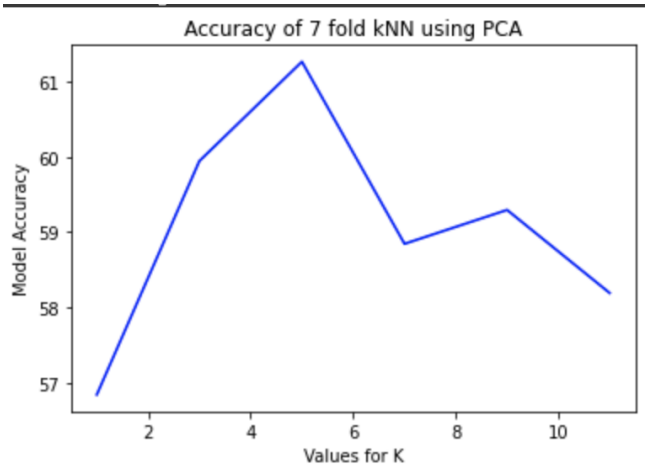

Figure 8: KNN Accuracies for PCA Features

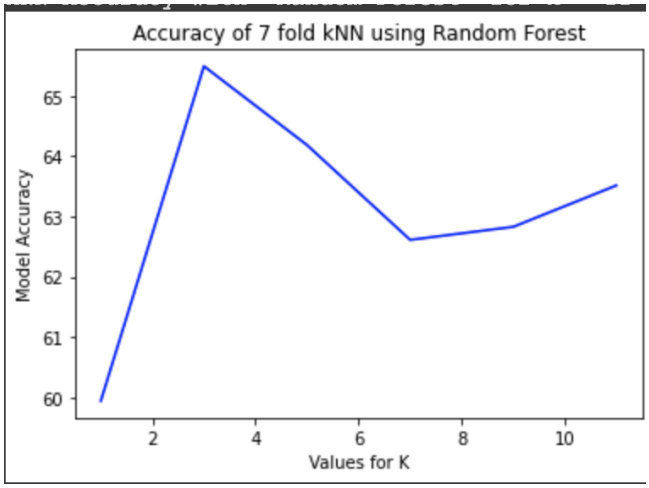

Figure 9: KNN Accuracies for Random Forest Features

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

Figure 10: Classification report for 1D-CNN

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 7.
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.

Multi Layer Perceptron

Since the hidden layers and the softmax output layer are trainable, the hyperparameter tuning is performed for all layers. The hidden layers with units 64 and 128 and RELU activation are fixed for the artificial multi layer perceptron and the results are extracted using softmax with class labels of size 17. The output layer uses Adam optimizer with a learning rate set to default of 0.001. The classification report for the Multi Layer Perceptron Model is found in Firgure 8.

Discussion

For SVM, the data is linearly separated. From our work, we conclude that linear kernel works best for the data set by comparing accuracy scores with other multi-dimensional

Classification Report for model

	precision	recall	f1-score	support
1	0.74	0.85	0.79	74
2	0.40	0.46	0.43	13
3	0.80	1.00	0.89	4
4	0.75	0.75	0.75	4
5	0.67	0.50	0.57	4
6	0.43	0.38	0.40	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	0.75	0.40	0.52	15
14	1.00	1.00	1.00	1
15	0.50	1.00	0.67	1
16	0.00	0.00	0.00	7
accuracy			0.67	136
macro avg	0.54	0.54	0.52	136
weighted avg	0.65	0.67	0.65	136

Figure 11: Classification report for MLP

kernels.
For KNN, this was a novel approach which was not implemented in prior related work. The reason we chose KNN was that it was the only model where the entire dataset is validated using k-fold KNN, making the model more robust to provide accuracy for all the classes. But the accuracies turned out to be quite less when compared to the scope of prior work suggested models - SVM, 1D-CNN and MLP.

The previous work suggested to use PCA for feature selection while we find that the Random forest classifier to be equally good at feature selection for SVM and KNN implementation. This is another novel approach.
In prior related work, the ECG data was directly fed to the models. We have used the tabular form of ECG data available at UCI Machine learning repository. Therefore, it is not insightful to compare accuracy scores with each other.
Using weighted loss functions for 1D-CNN as well as MLP, we reduce the penalty for using a balanced class models.
None of the related work referenced has implemented 1D-CNN and MLP. These are novel approaches to the dataset implemented by us to effectively solve the problem statement.
By implementing these models, we have covered the objectives specified in the hypothesis section.

6 CONCLUSIONS AND FUTURE SCOPE

As mentioned in the Discussion subsection, we were successfully able to predict classifications for all the labels which had adequate information required to train the models and then perform proportionate sampling of all the 16 classes of heartbeats.

In our current approach we have considered detecting and classifying the class-imbalanced UCI Arrhythmia tabular data based on the QRS wavelets based on the R-peak value and achieved accuracy values ranging from 60% to 70% with a maximum accuracy of 73.53% by the Neural Network Models 1D-CNN and MLP, using the Weighted loss function.
Future scope of work includes developing Transfer Learning models using Resnet, InceptionResnetv2, VGG which would be trained on the ECG sinus wave graph data for Arrhythmia detection and classification. By ensembling results from our 1D-CNN model predictions with Transfer learning model predictions, based on individual class confidence levels for both the models, We can potentially predict the Hybrid model to have an accuracy of 85% for each class of cardiac arrhythmia.

7 ACKNOWLEDGEMENTS

We thank Dr. Thomas Price, the teaching assistants and the Dept. of Computer Science at North Carolina State University for their support and guidance.

8 REFERENCES

[1] A. Das, F. Catthoor and S. Schaafsma, "Heartbeat Classification in Wearables Using Multi-layer Perceptron and Time-Frequency Joint Distribution of ECG," 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), Washington, DC, USA, 2018, pp. 69-74.
[2] 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.
[3]Martis, R.J., Krishnan, M.M.R., Chakraborty, C. et al. Automated Screening of Arrhythmia Using Wavelet Based Machine Learning Techniques. J Med Syst 36, 677–688 (2012). https://doi-org.prox.lib.ncsu.edu/10.1007/s10916-010-9535-7
[4] U. Desai, R. J. Martis, C. G. Nayak, Sarika K. and G. Seshikala, "Machine intelligent diagnosis of ECG for arrhythmia classification using DWT, ICA and SVM techniques," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-4.
[5]R. J. Martis, C. Chakraborty and A. K. Ray, "An Integrated ECG Feature Extraction Scheme Using PCA and Wavelet Transform," 2009 Annual IEEE India Conference, Gujarat, 2009, pp. 1-4.
[6] M. Carrara, L. Carozzi, T.J. Moss, M. De Pasquale, S. Cerutti, M. Ferrario, D.E. Lake, J.R. Moorman, Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy, Physiol Meas 36 (9) (2015) 1873–1888.
[7] A. M. Salem, K. Revett and E. A. El-Dahshan, "Machine learning in electrocardiogram diagnosis," 2009 International

Multiconference on Computer Science and Information Technology, Mragowo, 2009, pp. 429-433.
[8] Scikit-learn: A machine learning library for python - https://scikit-learn.org/stable/
[9] StandardScaler: Feature Scaler https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02
[10] Random Forests - https://en.wikipedia.org/wiki/Random_forest
[11] Principal Component Analysis - https://en.wikipedia.org/wiki/Principal_component_analysis
[12] Support Vector Machines - https://data-flair.training/blogs/svm-support-vector-machine-tutorial/
[13] SVM Kernel Functions - https://data-flair.training/blogs/svm-kernel-functions/
[14] 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).
[15] Cross Validation - https://en.wikipedia.org/wiki/Cross-validation_(statistics)
[16] K Fold - http://statweb.stanford.edu/ tibs/sta306bfiles/cvwrong.pdf

A MEETING SCHEDULES

Throughout the project duration, all four of us punctually attended meetings via Zoom video conference on the scheduled days given below:

(1) April 4th - 1:30pm to 3:30pm
(2) April 9th - 5:30pm to 8:30pm
(3) April 14th - 7:30pm to 10:30pm
(4) April 16th - 7:30pm to 10:30pm
(5) April 18th - 4:30pm to 6:30pm
(6) April 21st - 1:00pm to 5:00pm
(7) April 22nd - 1:00pm to 5:00pm
(8) April 23rd - 4:00pm to 7:00pm
(9) April 24th - 3:00pm to 6:00pm

B PROJECT LINK

You can find this project on the NCSU github enterprise server using the following link.
https://github.ncsu.edu/mvijaya2/ALDAProject