**Evaluation of performance of classifiers on Arrythmia data**

*Indumathi Desabathina*

*School of Computing, University of Utah*

**Problem Statement:** Detection and classification of **Arrythmia** based on ECG recordings.

The main aim of the study is to build a system that can classify ECG records into normal and diseased class, i.e., is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups.

Arrhythmia is a condition in which the heart beats in an irregular fashion, the heart rate can be too fast or too slow. Arrythmia that are not due to structured heart disease account for 5 to 10% of sudden cardiac arrests**. Early detection of arrythmia can help minimize the human loss**. ECG conveys information about the structure and function of heart electrical conduction system. It is a graph between the voltage versus time of the electrical activity of the heart. The main components of ECG are 3 waves: P wave, QRS wave, and T wave. The Objective of the study is to build different machine learning models and deep learning models to analyze the recordings to detect arrythmia.

The dataset used in this study is “**Cardiac Arrythmia Dataset**” from UCI Machine Learning Repository. The dataset is ECG recordings of 452 patients. This database contains 279 attributes, 206 of which are linear valued, and the rest are nominal. The basic features are age, gender, height, and weight of the patient. Since the data is about ECG recordings, it contains large number of features like presence of waves like shape, duration and relation between the three different wavs of the ECG. The target variable, the variable that need to be predicted has 16 classes.

|  |  |
| --- | --- |
| **Class Code** | **Class** |
| 01 | Normal |
| 02 | Ischemic changes |
| 03 | Old Anterior Myocardial Infarction |
| 04 | Old Inferior Myocardial Infarction |
| 05 | Sinus tachycardia |
| 06 | Sinus bradycardia |
| 07 | Ventricular Premature Contraction |
| 08 | Supraventricular Premature Contraction |
| 09 | Left bundle branch block |
| 10 | Right bundle branch block |
| 11 | 1. degree Atrioventricular block |
| 12 | 2. degree AV block |
| 13 | 3. degree AV block |
| 14 | Left ventricular hypertrophy |
| 15 | Atrial Fibrillation or Flutter |
| 16 | Others |

**Literature Review:**

ECG recordings capture captures a plethora of observations of the heart. For the dataset that is being used in this study, the number of features is 278. Among the 278 attributes include features like age, gender, height, and weight of the patient, the rest of the attributes include existence of certain waves and their measurement. All features are not necessary in model building.

**“Multiclass Classification of Cardiac Arrhythmia Using Improved Feature Selection and SVM Invariants”** explores the result of reducing the dimension or features. This study has proved that for this dataset, reducing the number of features increased the accuracy of the classifiers. Techniques used for dimensionality reduction or feature selection is Wrapper Method.

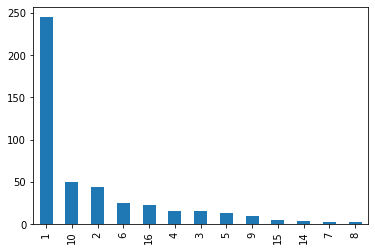
**Classification of ECG arrhythmia using learning vector quantization neural networks.** The abstract of the above study is:

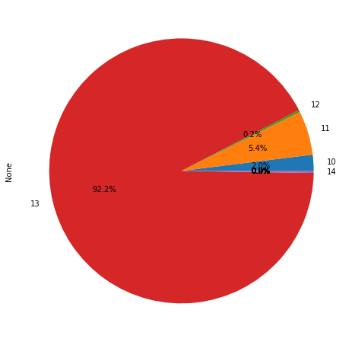
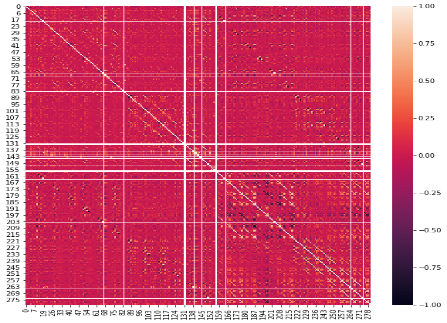
“The aim of this study is to apply learning vector quantization (LVQ) neural networks to classify arrhythmia from the Electrocardiogram (ECG) dataset. LVQ classification algorithms do not approximate density functions of class samples but directly define class boundaries based on prototypes, a nearest-neighbor rule and a winner-takes-it-all paradigm. It has a superior performance over back-propagation (BP) method in the sense of minimizing the classification errors while maintaining rapid convergence.”

**Dropout: A Simple Way to Prevent Neural Networks from Overfitting.** The article explores the idea of dropout and its role in preventing the overfit of the neural network.

There are many studies in literature that have extensively explored machine learning in determining the presence of cardiac arrythmia.

**Methods and Results:**



****

**Figure 1 Figure 2 Figure 3**

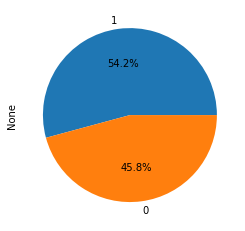
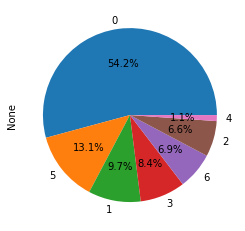
Distribution of Features: 279 attributes, 206 of which are linear valued, and the rest are nominal. Figure1 depicts the distribution of the target variable.

Correlation b/w features: Figure2 depicts the correlation among the independent variables.

Missing Values: The features which have missing values are features like Vector angles in degrees on front plane of QRS, T, P, QRST, J. These features are linear.

Data Preprocessing:

1. Missing Values: In this dataset, the missing values are represented by ‘?’. From preliminary inspection, the features containing missing values are linear. The appropriate way to replace missing values is to find the mean of the feature and replace missing value with mean.
2. Scaling & Encoding: The linear features are normalized before the data is used for training and validation. The categorical variables should be one-hot encoded. However, in out data, all the categorical features are binary valued., i.e. 0 and 1.
3. Data Imbalance: From figure 1, the dataset is imbalanced in terms of target variable. In this study imbalance is handled using two techniques: (i) Merge all the disease classes into one class, (ii) combine disease class into smaller sub-groups based on medical terminology. Figure 3 and Figure 4 show the distribution of target class after implementing (i) and (ii).

****

**Figure 4 Figure 5**

**Models:**

Machine Learning:

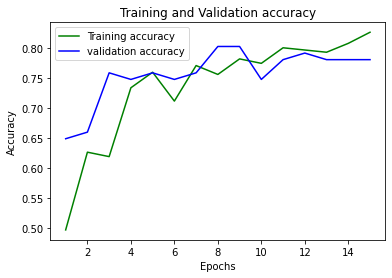
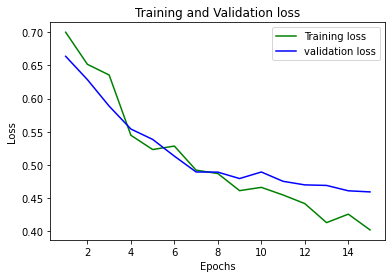
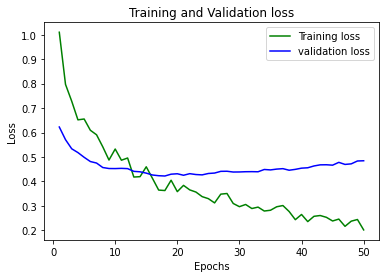
|  |  |
| --- | --- |
| Model | Hyper-Parameters |
| SVC | C (1e-10, 1e-10) |
| Random Forest | Depth (2, 32) Estimators (2, 128) |
| Logistic Regression | C (1e-10, 1e-10) |
| ElasticNet | Ratio (0,1) |

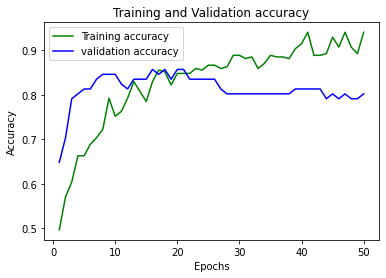
Approach1:

**Best Classifier:** Random Forest **Best Hyper-Parameters:** 18 and 64 **F1-score :** 78.8

Deep Learning:

|  |  |  |  |
| --- | --- | --- | --- |
| Neural Network | Training Accuracy | Validation Accuracy | F1-Score |
| Fully-Connected | 90.69 | 79.12 | 76.45 |
| Convolutional | 82.39 | 78.02 | 70.99 |





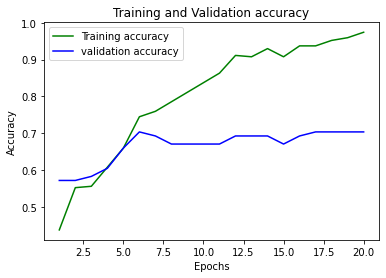
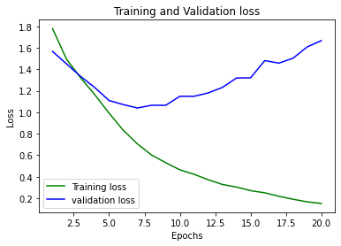
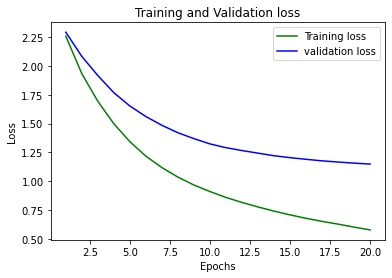
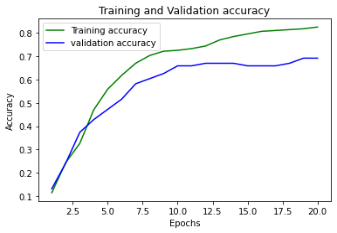
**Figure 6 Figure 7 Figure 8 Figure 9**

Approach2:

**Best Classifier:** Random Forest **Best Hyper-Parameters:** 21 and 127 **F1-score :** 70.39

Deep Learning:

|  |  |  |  |
| --- | --- | --- | --- |
| Neural Network | Training Accuracy | Validation Accuracy | F1-Score |
| Fully-Connected | 67.78 | 64.84 | 62.63 |
| Convolutional | 73.57 | 70.33 | 63.73 |

****

**Figure 10 Figure 11 Figure 12 Figure 13**

**Discussion**

Table1 has the list of machine learning classifiers with next column indicating the hyper-parameter considered while building the model. In this study, for machine learning classifiers, tuning is done with the help of library called Optuna. After running n trails, Optuna gives out the classifier with highest value and its best hyper-parameters.

Machine Learning classifiers used in the study are: Logistic Regression, Support Vector classifier, RandomForest, and ElasticNet. For both the classification task, i.e. binary and multiclass classification, Optuna picked Random Forest as the best classifier. The most probable reason would be is hat Random Forest is an ensemble method, gives better predictions than other classifiers.

The deep-learning neural networks used in the study are: Fully connected neural network and Convolutional neural network. With large number of features there is a chance of overfitting in the neural network. Hence, care is needed while modelling the neural network especially while tunning the hyperparameters.

Figures 6-9, correspond to training and validation accuracy of fully connected neural network and convolutional neural network respectively for binary classification. From **figure 6 and 7**, one can observe overfitting of the model. Used Optuna to check for better hyper-parameters but it didn’t give the results as expected. Figure 8 and 9, since CNN facilitates feature-detection, there seems to be less over-fitting as compared to fully connected neural network.

Figures 10-13, correspond to training and validation accuracy of fully connected neural network and convolutional neural network respectively for multi-class classification. For multi-class classification, fully connected layer with-dropout has better performance than compared to CNN. With multi-class, the Optuna didn’t’ t give the expected results. The accuracy and the loss graph of the model were like a parallel line to the axis.

To avoid-overfitting, one may consider the following techniques like: Oversampling the minority class, Under sampling the majority class, combing both the techniques like oversampling and under sampling for better generalization of the model. The three techniques mentioned here deals with the instances in the data but not feature.

**Conclusion & Limitations:**

1. The main limitation while dealing with this particular data is the dimensionality, with such a smaller number of instances, it is difficult to build a model that generalizes well.
2. One may encounter the problem while splitting the data is, certain classes are not sufficient to stratify.
3. Even in the case if stratification was possible in all the three sets, one may not use techniques like oversampling and under sampling the dataset. Reason: Most sampling techniques use nearest neighbor algorithm, and it requires for the certain distribution of the class variables.
4. Relying on one Optuna for hyper-parameter tuning is not a viable option. One should consider exploring techniques like Randomized Search and grid search for hyper-parameter tuning.
5. Cases such as neural network, if the model seems to generalize well while comparing the accuracies and loss of training & val. But doesn’t give better results on test. Two reasons: insufficient amount of test data, or the training is done for less number of epochs and hence the model seems to be not-overfitting.
6. Domain knowledge is required while handling the medical data. Sometimes imbalance of the data can be handled with some domain knowledge. Combining of classes based on quartile is not ideal in every case.
7. Dimensionality reduction, i.e. reducing the number of features can help build models that are more stable. Dimensionality reduction eliminates feature that are correlated and produces parameters that are stable for different samples of data.