Arrhythmia Detection Using Deep Learning

Mathankumar A[1] and Privietha P [2]

[1]*Student, Department of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India.*
[2]*Assistant Professor, Department of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India.*
[1]*matmathan2001@gmail.com,* [2]*priviethaprabhakar@gmail.com*

Abstract. This project is entitled as "Arrhythmia Detection with Deep Learning" is a dataset analysis. Cardiac Arrhythmia is a life-threatening disease, causing serious health issues in patients, when left untreated. An existing diagnosis of arrhythmias would be helpful in save millions of lives. This data set contains large amount of feature dimensions which are reduced using dimensionality reduction techniques. Kernelized SVM are employed over original data to identify the presence and absence of arrhythmia diseases. The accuracies are then improved by using Principal Component Analysis (PCA) over the original dataset. The models are then evaluated and compared using their accuracy and recall values. The results showed that on applying PCA over the data, Kernelized SVM outperforms the other classifiers with an accuracy rate.

Keywords: Arrhymia detection, PCA, kernel SVM

1. Introduction

A premature heartbeat may feel like heart skipped a beat. These additional beats are generally not concerning, they seldom mean you have a much dangerous condition. Still, a premature beat will trigger a longer-lasting arrhythmia, especially in people with heart disease. Its, very frequent premature beats that last for several years may cause a weak heart. [2]

Premature heartbeats can occur when resting. Sometimes premature heartbeats are caused by stress, strenuous or stimulants, such as caffeine.

Symptoms

The general symptoms of arrhythmia disease;

- A fluttering in the chest
- A racing heartbeat (tachycardia)
- A slow heartbeat (bradycardia)
- Chest pain

When to see a doctor, if anyone feel like your heart is beating too fast or too slowly, or it skipping a beat, make a visit to see the doctor. Seek immediate medical help if you have shortness of breath, weakness, dizziness, or near fainting, and chest pain or discomforts. [1]

2. Related Work

K-nearest neighbors algorithm uses to predict the values of new data points which further means that new data point will be allotted a value based on how closely it similar the points in the training set. The K parameter decides how many of the nearest neighbors have to be considered to determine the property of our unknown point. [3] KNN use a similarity metric to decide the nearest neighbors [4].

This similarity metric is routine the Euclidean distance between the unknown point and the other points in the dataset.

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

The major objective of SVM is to find the optimal hyperplane that linearly break the data points in two components by maximizing the margin.

3. Tools and Techniques

A Convolution Neural Network is a special type of deep learning algorithm which uses a set of filters and convolution operators to less the number of parameters. Here in this project we will use 1D CNN instead of 2D CNN as 2D CNN is used for pictures. Here 2D CNN is not require as we are going to alter and lower the parameter for training the dataset replace the images to measure the performance.

Figure 1: Diagram of Neural Networks

Figure 1 and 2 represents the pictorial representation of neural network with dense layers. In this project since Keras CNN is used we reshape our data just a bit for training the models and getting the required results. kernel size and other necessary requirements is used to determine the performance. Like Dense neural network this model makes use of Sequential type of model for better processing of input data. After the data is reshaped, the reshaped data is provided as an input while adding the parameter for convolution network for training purpose. The metrics used for this model is also accuracy same as dense neural network in order to check which among the two can provide better performance in terms of measuring accuracy.

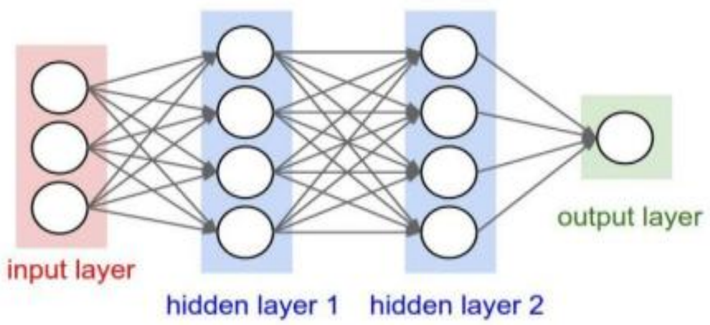

Figure 2: Working procedure of Dense Neural Networks.

3.1 Dataset

This database contains 279 attributes, 207 of which are linear valued and the rest are nominal. Concerning the study of H. Altay. The aim is distinguishing between the presence and absence of cardiac arrhythmia diseases and to classify it in one of 16 groups. Class 1 describe to normal ECG classes 2 to 15 refers to different classes of arrhythmia and class 16 describe to the rest of unclassified ones. For the time being, there older a computer program that makes such a classification. However, the variation between the cardiology and the programs classifications. Taking the cardiology as a gold standard we aim to minimize this variation by means of machine learning tools. The data is modelled using various algorithms.

3.2 Network Architecture

In order to exact determine the performance of neural networks, the best method is to determine it to use AUC Curve. Figure 3 shows the graphical representation as AUC Curve. The Receiver Operating Characteristic curve is a graphical plot that allow us to assess the performance of binary classifier. Area Under Curve score is used when we have imbalanced dataset as in case of this project which is validate from ROC and is a very helpful metric for measuring performance in case of imbalanced datasets.

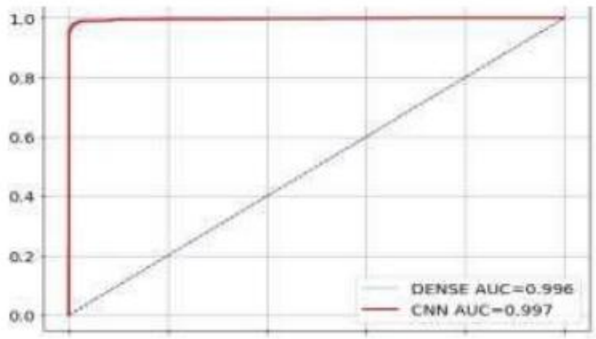

Figure 3: AUC curve Diagram

In order to reduce the dimensionality of dataset consisting of large inter-related variables, while retaining as much as possible of the variation present in the dataset, we use PCA. The features are transformed into new set of variables called principal components, which are uncorrelated. It gives a measure of how changes in one-dimension affect changes in the other

An eigenvector vector of a linear transformation, is a nonzero vector that changes at much by scalar factor when that linear transformation is applied to it. The relevant eigenvalue is the factor by which the eigenvector is scaled.

Every Layer in the Neural Network contains neurons, which compute the weighted average of its input and this weighted average is passed through a non-linear function, called as an “activation function”.

The dropout used to drop out the nodes (input and hidden layer) in a neural network.

Activation (ReLU), dropout (0.5), dense (4), activation (softmax) layers are executed again to get better accuracy and performance. The softmax function is often called in the final stage because it changes the output to 0 and 1 with 1 as each total probability sum. There are no parameters required for the softmax function.

3.4 Training and testing

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization. Figure 4 and 5 shows the Jupiter notebook code of model preparation.

3. Modeling

```
In [70]: # will store result of each model.

         result = pd.DataFrame(columns=['Model','Train Accuracy','Test Accuracy'])
```

Kernelized SVM

```
In [71]: from sklearn import svm
         KSVM_clf = svm.SVC(kernel='sigmoid',C=10,gamma=0.001)
         KSVM_clf.fit(X_train, y_train)
         y_pred_train = KSVM_clf.predict(X_train)
         y_pred_test = KSVM_clf.predict(X_test)
```

```
In [72]: ksvm_train_recall = recall_score(y_train,y_pred_train,average="weighted" )
         ksvm_test_recall = recall_score(y_test, y_pred_test,average="weighted")

         ksvm_train_accuracy = accuracy_score(y_train, y_pred_train)
         ksvm_test_accuracy = accuracy_score(y_test, y_pred_test)
```

```
In [73]: print('Train Recall score: {}'
               .format(ksvm_train_recall))
         print('Test Recall score: {}'
               .format(ksvm_test_recall))
         confusion_matrix(y_test, y_pred_test)

         Train Recall score: 0.850415512465374
         Test Recall score: 0.7912087912087912
```

Figure 4: Trained and Test Model of kernelized SVM.

Kernal svm with PCA

```
In [78]: from sklearn import svm
         KSVM_clf = svm.SVC(kernel='sigmoid',C=10,gamma=0.001)

         KSVM_clf.fit(X_train_pca, y_train)
         y_pred_train = KSVM_clf.predict(X_train_pca)
         y_pred_test = KSVM_clf.predict(X_test_pca)
```

```
In [79]: ksvmp_train_recall = recall_score(y_train, y_pred_train, average='weighted')
         ksvmp_test_recall = recall_score(y_test, y_pred_test, average='weighted')
         ksvmp_train_accuracy = accuracy_score(y_train, y_pred_train,)
         ksvmp_test_accuracy = accuracy_score(y_test, y_pred_test)
```

```
In [80]: e=accuracy_score(y_test, y_pred_test)
         print('Train Recall score: {}'
               .format(ksvmp_train_recall))
         print('Test Recall score: {}'
               .format(ksvmp_test_recall))

         confusion_matrix(y_test, y_pred_test)

         Train Recall score: 0.8393351800554016
         Test Recall score: 0.8021978021978022
```

Figure 5: Trained and Test Model of kernelized SVM with PCA

3.5 Results

When trained over original data, Kernelized SVM proved to be the best among other classifiers in terms of recall value, with an accuracy percent of 79.12%. After performing Principal Component Analysis, when data is trained our models, let Kernelized SVM model with an accuracy of 80.21%. Figure 6 represents the graphical representation of the comparison models.

Result

```
In [84]: ax=result.plot(kind="bar",figsize=(20,5))
         ax.set_xticks(result.index)
         ax.set_xticklabels(result.Model,rotation=90)
         plt.title('Train & Test Scores of Classifiers')
         plt.xlabel('Models')
         plt.ylabel('Accuracy Scores')
         plt.legend(loc=4 , bbox_to_anchor=(1.2, 0))
         plt.show();
```

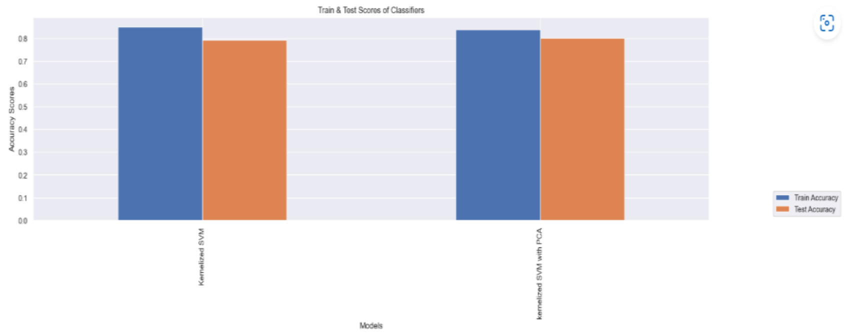

Figure 6: Visualize our final results via Bar chart.

4. Future Work

In future a plan is there to create an application to sense a heartbeat through smartwatch and regularly monitor Arrhythmia Disease

If Arrhythmia exist message will be sent to the emergeny contacts along with the display in mobile and smart watch.

Data set can be replaced with the recent updates of parameters for better accuracy.

Planned to create a self-designed CNN model for Arrhythmia detection

5. Conclusion

This study proposes a method for classification of arrhythmia using ECG data by implementing various machine learning techniques. To improve the accuracy of the model, Principal Component Analysis (PCA) is performed over the data to reduce its dimensions and then the data is modelled. The results show that Kernelized SVM outperformed other classifiers when data with reduced features is trained, as PCA reduced the complexity of the original data. The model predicts the absence or presence of cardiac arrhythmia and classifies it into one of the 16 classes with an accuracy of 80.21%. Our results suggest that Kernelized SVM model can be used to diagnose cardio-vascular diseases like arrhythmia in hospitals

References

1. V. Mahesh, A. Kandaswami, C. Vimal, and B. Sathish (2009) ECG Arrhythmia Classification Based the Logistic Model Trees
2. Anam Mustaqeem, Syed Muhammad Anwar, and Muahammad Majid Multi-Class Classification of Cardiac Arrhythmia Using Feature Selection and SVM Invariants
3. Babak Mohammadzadeh, Setarehdan, & Maryam Mohebbi (2009) Support Vector Machine – based arrhythmia classification using reduce features of heart rate variability signals
4. Jalal A. Nasiri, Mahmoud Naghibzadeh, H. Sadoghi Yazdi, and Bahram Naghibzadeh (2008) ECG Arrhythmia