Arrhythmia recognition of ECG signals on Raspberry Pi by Artificial Neural Networks

Tuguldur Gerelmaa
Institute of Physics, University of Debrecen, Debrecen, Hungary

November 18, 2022

Abstract

ECG signals are one of the most important sources of diagnostic information. Interpretation of ECG signals is a labor intensive task and requires domain expertise. In this research, we have trained a Residual Network on a recently released large scale 12-lead ECG recordings of 40,258 patients. The ECG signals were filtered by a Butterworth Low-Pass and Butterworth Notch filters. The trained model achieved an accuracy of 98.6% on classifying normal and abnormal ECG signals. We have deployed this model on a Raspberry PI after converting it into a TensorFlowLite model. The resulting model occupied 1 MB space while requiring 7.8 MB memory.

Supervisor: Dr. Trencsényi Réka

1 Introduction

In past decades, a class of machine learning algorithms, known as deep learning, has achieved significant success in solving problems which are normally difficult for traditional computer programs. One such problem is the interpretation of ECG signals. Electrocardiogram (ECG) signals are one of the main sources for diagnostic information. Interpreting ECG signals require, in most cases, experts with extensive domain knowledge and it can be labor intensive, especially for clinical cases where constant monitoring is required.

There are numerous IoT devices in the health care sector that can be used to monitor ECG and body vitals such as smart watches, heart rate monitors, and pulse-oximeter. Most of such devices are only used for monitoring and uploading the data to a remote server or to a computer where it can be analyzed by an expert or by a computer.

This research aims to integrate the state-of-the-art deep learning algorithms and the IoT devices. These AI embedded devices can be used to detect abnormalities in the ECG signals of a patient, providing a real-time detection of arrhythmia.

In the literature, ECG signals have been extensively studied by a wide range of methods. Traditional signal processing methods such as wavelets (Nazarahari, Namin, Markazi, & Anaraki, 2015) or certain deep learning algorithms such as Convolutional Neural Networks (CNN) have achieved an accuracy of 80% in classifying ECG signals (Sharma & Eskicioglu, 2022). However, as pointed out by (Sharma & Eskicioglu, 2022), a combination of Long-Short Term Memory Networks (LSTM) and CNN model has achieved high accuracy of over 97% (Irfan et al., 2022).

Moreover, the model used in this research was based on the top performing type of CNN in the PhysioNet/Computing in Cardiology 2020 challenge (Nejedly et al., 2021). We have also utilized a recently released large scale database of 12-lead ECG signal recordings of 40,258 patients (Zheng, Guo, & Chu, 2022).

2 Methodology

Filtering

The original ECG signals had lots of recordings affected by noise. The main types of noises were: the baseline drift and the electromagnetic noise.
The baseline drift is a type of low frequency noise which is caused by the movement of body of the patient including breathing. Typical frequency of such noise is below 1 Hz (Gacek & Pedrycz, 2012).

Electromagnetic noise is a type of high frequency noise which is caused by power line interference such as AC power supply, or high power devices such as diathermy equipment, Roentgen machines, and tomography devices (Gacek & Pedrycz, 2012).

As the original authors of the database suggested (Zheng et al., 2020), Butterworth low-pass filter was utilized to remove the electromagnetic noise. For removing the baseline drift, the Butterworth notch filter was used. A simple digital filter algorithm was developed from the Signal Processing library (Scipy) in Python. The Butterworth low-pass filter was of order 3 with the cutoff frequency of 45 Hz, while the Notch filter had a rejection frequency of 0.05 Hz.

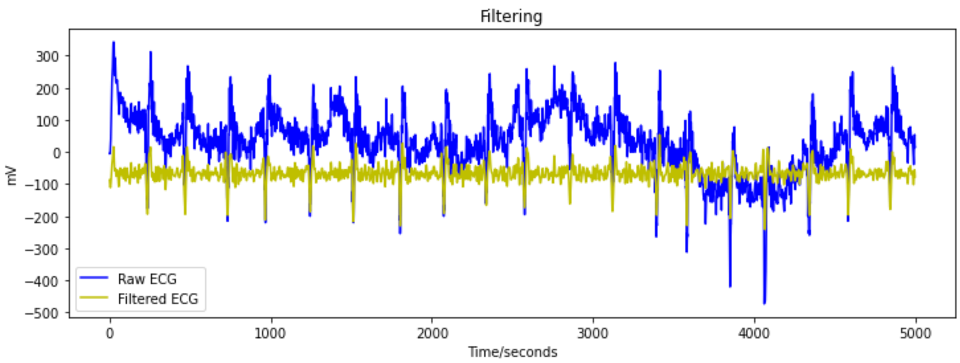
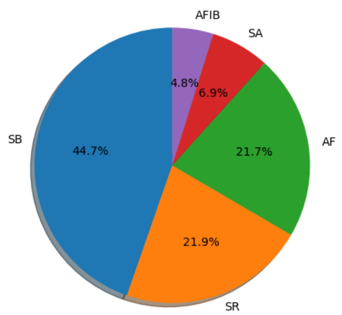


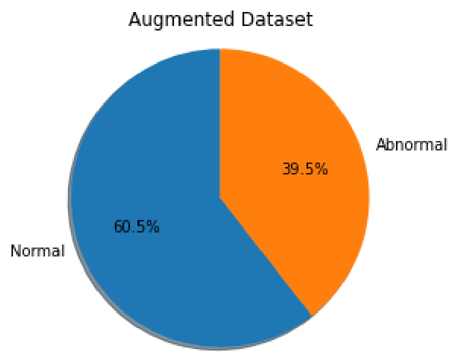
Figure 1: Effect of filtering the ECG signal by Butterworth Low-Pass and Notch filters.

Data Preprocessing

The original dataset contained 40,258 recordings of 12-lead ECG signals. It contained 51 different classifications of ECG with varying distributions. Therefore, to make sure the model has abundant training data, we have selected the Sinus Bradycardia (SB), Sinus Rhythm (SR) and Sinus Arrhythmia (SA) recordings as a "Normal" ECG signal and Atrial Fibrillation (AFIB) and Atrial Flutter (AF) as a "Abnormal" ECG signal (Figure 2). The total number of recordings after the selection was 36754.



(a) Original selected dataset



(b) After augmentation and combination

Figure 2: Preprocessing the dataset

However, as seen from Figure 2, only 4.8% of the dataset belonged to AFIB and 21.7% was AF. Machine learning models must have a uniformly distributed dataset to prevent the model developing a bias. To deal with this issue, AFIB and AF recordings were over-sampled and SB recordings were under-sampled. For the resulting augmented dataset, class weights were used.

Finally, the dataset was normalized in order for the model to learn effectively. The z scores methods was utilized.

Model Architecture

We have chosen to use the top performing models in the PhysioNet/Computing in Cardiology 2020 challenge (Nejedly et al., 2021), (Singstad & Tronstad, 2020). Nejedly et al. (2021) has achieved the best results by implementing a complex type of CNNs known as Residual Networks (ResNet) which works in the same principal as CNN-LSTM network. They both improve the gradient propagation training and significantly raise the accuracy (Nejedly et al., 2021). The CNN network learns the spatial while the LSTM network learns the temporal information in the ECG signal. The implementation of the model can be found on the repository.

3 Results

Model evaluation

We have used 80% of the data for training the model and the rest was used for testing the model's accuracy. The model was trained over 75 epochs on the Google Colab Pro server, the total duration was roughly 10 minutes. The confusion matrix of the model on the testing data can be seen from Figure 3. The model has achieved 98.60% accuracy.

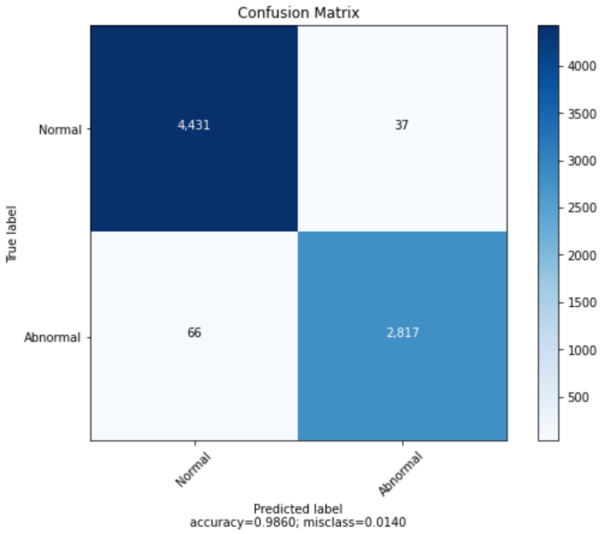


Figure 3: Confusion Matrix of the model.

TensorFlowLite model conversion

The table below shows the results of different methods of converting the model into TensorFlowLite model, which is suitable for running on Raspberry Pi. Quantization refers to converting the parameters of the model (weights) into a suitable datatype, reducing the size in process.

Table 1: TensorFlowLite model conversion results

Technique	Datatype	Size
Default	float32	2 MB
Float16 quantization	float16	1 MB
Dynamic range quantization	float16/int8	539 KB

The Float16 quantization is the preferred technique as it has reduced by 50% of the original size while having insignificant loss in accuracy. The peak memory usage of the model was 7.68 MB.

4 Conclusion

Applying Residual Networks, as suggested by the literature, resulted in a model with a high accuracy of 98.60%, the other models developed in the literature also has achieved similar high accuracy levels up to 99% (Irfan et al., 2022). Therefore, this research supports the use of ResNet and CNN-LSTM model architecture for classifying

ECG signals.

By using TensorFlowLite optimization, the size of the model can be reduced to 539 KB, 1MB, and 2MB depending on the preferred embedded device and accuracy. Thus, most Raspberry PI generations can be utilized for real-time detection of arrhytmia in ECG signals.

However, the model is limited by the fact that it was trained to classify between simple classes of sinus rhythms and atrial fibrillation. Due to the lack of availability of certain signals such as Right Bundle Branch Block or Supraventricular Tachycardia, many other important ECG signals were discarded, and thus the accuracy does not reflect the true performance of the model in real life applications.

For future works, the model can be made even smaller and suitable for microcontroller implementations by converting the weights into a full integer type.

References

Gacek, A., & Pedrycz, W. (Eds.). (2012). *ECG signal processing, classification and interpretation.* Springer London. Retrieved from https://doi.org/10.1007/978-0-85729-868-3 DOI: 10.1007/978-0-85729-868-3

Irfan, S., Anjum, N., Althobaiti, T., Alotaibi, A. A., Siddiqui, A. B., & Ramzan, N. (2022, July). Heartbeat classification and arrhythmia detection using a multi-model deep-learning technique. *Sensors*, *22*(15), 5606. Retrieved from https://doi.org/10.3390/s22155606 DOI: 10.3390/s22155606

Nazarahari, M., Namin, S. G., Markazi, A. H. D., & Anaraki, A. K. (2015, July). A multi-wavelet optimization approach using similarity measures for electrocardiogram signal classification. *Biomedical Signal Processing and Control*, *20*, 142–151. Retrieved from https://doi.org/10.1016/j.bspc.2015.04.010 DOI: 10.1016/j.bspc.2015.04.010

Nejedly, P., Ivora, A., Smisek, R., Viscor, I., Koscova, Z., Jurak, P., & Plesinger, F. (2021). Classification of ecg using ensemble of residual cnns with attention mechanism. In *2021 computing in cardiology (cinc)* (Vol. 48, p. 1-4). DOI: 10.23919/CinC53138.2021.9662723

Sharma, K., & Eskicioglu, R. (2022, July). Deep learning-based ECG classification on raspberry PI using a tensorflow lite model based on PTB-XL dataset. *International Journal of Artificial Intelligence &amp Applications*, *13*(4), 55–66. Retrieved from https://doi.org/10.5121/ijaia.2022.13404 DOI: 10.5121/ijaia.2022.13404

Singstad, B.-J., & Tronstad, C. (2020). Convolutional neural network and rule-based algorithms for classifying 12-lead ecgs. , *47*, 1-4. DOI: 10.22489/CinC.2020.227

Zheng, J., Chu, H., Struppa, D., Zhang, J., Yacoub, S. M., El-Askary, H., ... Rakovski, C. (2020, February). Optimal multi-stage arrhythmia classification approach. *Scientific Reports*, *10*(1). Retrieved from https://doi.org/10.1038/s41598-020-59821-7 DOI: 10.1038/s41598-020-59821-7

Zheng, J., Guo, H., & Chu, H. (2022). *A large scale 12-lead electrocardiogram database for arrhythmia study.* PhysioNet. Retrieved from https://physionet.org/content/ecg-arrhythmia/1.0.0/ DOI: 10.13026/WGEX-ER52

Repository location https://github.com/8Nero/ArrythmiaRasp