

AI in Biomedicine: Arrhythmia Heartbeat Classification using Medical Image Processing and Deep Learning

Kira Sam¹, Shah Nawaz², Dr. Suresh Kumar³, Raja Vavekanand⁴

¹Università del Salento, Italy

²Benazir Bhutto Shaheed University Lyari, Karachi

³Aga Khan University, Karachi

⁴Datalink Research and Technology Lab

Abstract - Heart arrhythmias, characterized by irregular heartbeats, pose serious health risks. Wearable technologies like Apple Watch enable continuous heart activity monitoring, allowing machine learning algorithms to detect abnormal heartbeats and potentially prevent fatalities. This study focuses on classifying individual heartbeats in ECG signals using two approaches: classic machine learning with handcrafted features and deep learning with medical image processing. The LightGBM model, enhanced with handcrafted features and optimized through parameter tuning and SMOTE for data balancing, outperformed other classifiers, achieving precision of 0.94, recall of 0.94, accuracy of 0.99, and F1 score of 0.94. This study highlights the potential of integrating machine learning and deep learning techniques in medical diagnostics, offering a robust approach for accurate heartbeat classification and arrhythmia detection.

Keywords: Arrhythmia, Heartbeat, Biomedicine Medical Imaging, Deep Learning

I. INTRODUCTION

A heart arrhythmia is an irregular heartbeat that occurs when the electrical signals that coordinate the heart's beats don't work properly. Sometimes these irregularities are considered normal and harmless. However, some heart arrhythmias may cause even life-threatening signs and symptoms. Today's technologies such as wearable products (e.g. Apple Watch) allow us to monitor our heart activity continuously and from home. There is a great possibility that machine learning algorithms could detect abnormal heart beats and even prevent death. The scope of this assignment was to classify individual heartbeats in a sample ECG signal. We experimented with two possible approaches - a classic machine learning model with handcrafted features extracted from the provided samples, and an approach with generated images, fed to an image classifier. The results are summarized in the report below.

II. MATERIALS AND METHODS

A. Pre Processing

In the beginning, we decided to explore the given dataset and perform some basic pre-processing techniques. Each signal from our dataset has three files where one contains the 2lead ECG signal and the other two are representing the R-peak positions and the beat annotations. The first step in the preprocessing was to down sample the signal with higher sampling frequency, i.e 250 Hz so as to unify all the signals to the same sampling frequency (180 Hz). The second step was to filter the signal keeping only the frequency range between 0.5 and 35 Hz. Since our problem was to label ECG beats, we decided to split the signals into fixed-size beats, each containing 70 samples i.e. 35 samples before and after the R peak location. The next pre-processing step was to normalize all the beats so that they have the amplitude

between -1 and 1. [8] Finally, we had the dataset that contains separated beats and their labels.

B. 1st Approach

Based on the feature extractions approaches in literature we added additional hand-crafted features to the data we fed to the classifier. We suspected that the average HRV, its median and variance, along with average signal amplitude might provide useful information to the model and the first tests confirmed the assumptions. Later, two more features (logarithm of the current and the next RR distance) were added [8]. All of the aforementioned features were then concatenated with the 70sample beats.

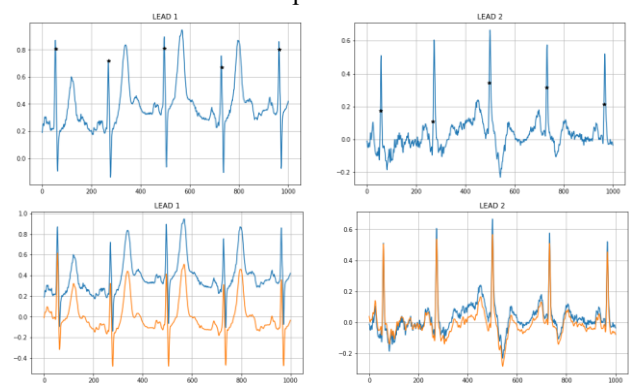


Fig. 1: Original signal from both leads together with R-peak annotations and the signal after pre-processing steps

Having the pre-processed dataset, we started by researching existing approaches to the task and decided on two very different approaches. The main idea was to test both of them and to choose the one that would be more suitable for our classification problem.

One of the problems that we met here was the the unbalanced dataset. Namely, the majority of beats were labeled as normal beat, i.e. classified with the label 'N'. Therefore, we agreed on using the SMOTE technique to prevent this, instead of cutting of the huge amount of normal beats. The SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. We decided to do the oversampling so that we have 100k ventricular (V) and super-ventricular (S) beats, and 300k normal (N) beats. It was applied only to the training data, while the test data remained unchanged so that it correctly represents the original data.

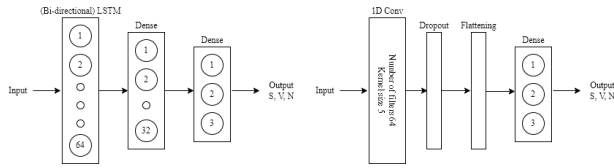


Fig. 2: Models with LSTM, Bi-directional LSTM and 1D Convolutional layers respectively

After a thorough investigation of the dataset, we developed models capable of classifying the beats based on the input data. In the first approach, we tested the SVM, Random Forest, AdaBoost, Convolutional Neural Network using 1D convolutional layers, Recurrent Neural Networks using LSTM and Bi-directional LSTM layers, and finally, the ones that produced the best results, Light GBM models. All the model parameters were fine-tuned using Grid Search with a 3-fold cross-validation, which allowed us to quickly verify in which direction to move in when developing the classifiers.

C. 2nd Approach. Images from ECG

The main idea is to exploit the state-of-the-art performances of 2d CNN in our classification problem. We need to transform each heartbeat in an image [6]. We want a way to encode the temporal information of our signal into the processed images. To obtain our 2d matrix picture, three different methodologies were used: Gramian angular field (GAF), Markov transition field (MTF), and recurrence plots (RP). All of these methods were proven to be effective on time series regardless of the specific domain [9]. I will briefly describe each methodology.

In GAF we change the representation of our signal by using the Polar Coordinate system. After normalization, we calculate the phase of each value in our time series using the following equation.

$$\begin{cases} \phi_i = \arccos(\tilde{x}_i) \\ r_i = \frac{t_i}{T}, t_i \in \mathbb{N} \end{cases}$$

Where x_i is the normalized value of a single data point in our time series. t_i is the timestamp of that data point. T is constant factor.

The final matrix contains the sum of the cosine of the 2 angles.

$$\begin{pmatrix} \cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cos(\phi_n + \phi_2) & \dots & \cos(\phi_n + \phi_n) \end{pmatrix}$$

It is a bijective function so we can recover the signal from the image. It represents the temporal correlation between the values of the time-series data. It maintains the temporal dependency. In MTF an image represents a field of transition probabilities for a discretized time series. It's useful for representing the dynamics of the time series. The high-level idea is to divide the values of our data $\{x_1, \dots, x_m\}$ into N quantile bins. From this, we can easily compute the Markov transition matrix. This matrix contains the probabilities of going from one quantile bin i to another quantile bin j. The assumption is to consider our time series as a Markov chain and compute the frequencies. This matrix does not preserve the temporal dependencies and for this reason, we need to perform another transformation that creates the Markov transition field. The difference is that we create and put data on the quantile bins considering the exact timestamp(at k-th timestamp). In this case, we will preserve the temporal dependency by having the probability of going from bin i of timestamp k to bin j of timestamp l. We cannot recover the original signal given that we have a matrix with probabilities. RP shows the collection of pairs of times at which the values of our time series have the same value(or similar values). They are a graphical representation of the matrix

$$R_{i,j} = \theta(\epsilon - ||x_i - x_j||)$$

where x_i stands for the value at time i, ϵ is a predefined threshold and θ is the Heaviside function. One assigns a “black” dot to the value one and a “white” dot to the value zero. In our case we perform the recurrence plot even without using a predefined threshold. []

In figure 3 we see all the heartbeat classes along with each transformation. All the transformations were achieved by using the python library pyts. All the transformations produced images of 32x32 pixels. The choice is done given that the maximum number of pixel that we can have is 70 (70 is the total of timestamps in our signal)

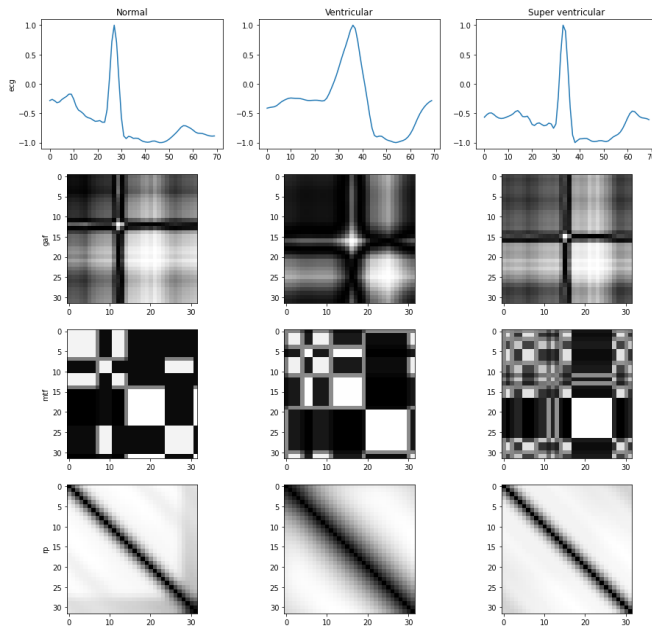


Fig. 3: ECG image transformation

Every transformation has its own peculiarities and it focuses on different parts of the signal. The idea of using all of them comes from the fact that multi-modal architectures often performs better with respect to using a single modality. In our case we concatenate them in order to have as input a 3 channel image (the standard representation for rgb images). We use a convolutional neural network to extract the features and classify each beat. Different architectures were tried such as VGG, ResNet, Inception like architectures. The best model was a custom Inception base CNN that has 5 million parameters. Very similar performances were also achieved by simple VGG like architectures. The following parameters were used in the best configuration. We set the learning rate to be $0.3e-4$, with batch size of 512. The loss function chosen was the categorical cross entropy. The dataset used is the same described in the previous chapter with the exception that no hand crafted feature was used, just the original ECG of the single heartbeat. The input are the images representing the single heartbeats.

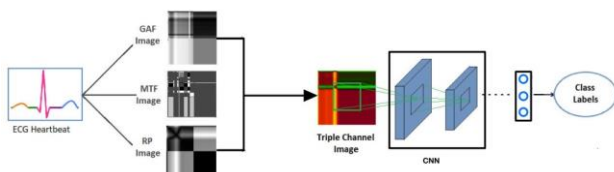


Fig. 4: Architecture

III. RESULTS

Below, we report a selected number of metrics that we observed in training the models. The metrics were the basis for the final model choice. The reported values are the macro averages of the metrics per label.

Model	Precision	Recall	Accuracy	F1 score
-------	-----------	--------	----------	----------

Images approach

Random Forest

SVM

LightGBM

LSTM

AdaBoost

Bi-directional

LSTM

1D CNN

0.83	0.87	0.97	0.85
0.93	0.93	0.99	0.93
0.59	0.58	0.58	0.58
0.94	0.94	0.99	0.94
0.86	0.95	0.98	0.90
0.43	0.72	0.70	0.44
0.91	0.94	0.98	0.93
0.43	0.39	0.92	0.39

IV. DISCUSSION

Based on the model scores, we immediately discarded the SVM, feed-forward neural network, and AdaBoost models. Their performances could have possibly been further improved, however, they were significantly lower than other models', so we preferred to focus on the more powerful ones. Furthermore, the SVM model took a long time to train (approximately 4 hours), which meant investing more resources into an ineffective training. We were very satisfied with the performance of the Random Forest and LightGBM models. They are both ensemble techniques, which train smaller decision trees and then use either bagging (Random Forest) or boosting (LightGBM) to classify the samples. The advantage of the LightGBM is its fast optimization algorithm, which allowed us to train the model in approximately 6 minutes, leaving us more time to run a larger number of experiments.

The parameter which improved the results the most in the LightGBM model, was surprisingly the learning rate. By increasing it, one would normally risk not minimizing the gradient and reaching the minimum. However, a model with a relatively high learning rate was able to classify even the marginal labels better, possibly because of high penalization rates of L1 and L2 regularization. The ratio between undersampling and oversampling also proved to be of value when trying to increase the scores.

Concatenating the two signal leads could further improve the results and add additional information to feed into the model. Research suggests that using the logarithms of a larger number of features might also render them more understandable for the models. Furthermore, we discussed creating an LSTM model, which would take into account 4 previous beats and classify the fifth one, possibly finding the relationships between subsequent beats. Due to the limited time, we were unable to explore these approaches now, but it would be interesting to research it further in the future.

V. CONCLUSION

In the end, the model that performed best was the LightGBM with the following parameters: learning rate of 0.5, max depth of trees 10, using *multiclass logloss* as metric, minimum data in leaves of 10, and 1000 estimators. We also set the alpha regularization to 0.5 and lambda regularization to 0.7327. The imbalanced dataset was handled using the SMOTE technique, undersampling the signal 300000 times and then oversampling it 100000 times.

We believe that the first approach worked best for the task because the hand-crafted features were able to capture the

whole signal variation, whereas the images created contain information only on the single beat.

REFERENCES

- [1] A. Rajkumar, M. Ganesan and R. Lavanya, "Arrhythmia classification on ECG using Deep Learning," 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS), 2019, pp. 365-369, doi: 10.1109/ICACCS.2019.8728362.
- [2] Ullah, Amin, et al. "Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation." *Remote Sensing* 12.10 (2020): 1685.
- [3] Vavekanand, R. (2024). A Machine Learning Approach for Imputing ECG Missing Healthcare Data. *Available at SSRN 4822530*.
- [4] P. de Chazal and R. B. Reilly, "A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features," in *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535-2543, Dec. 2006, doi: 10.1109/TBME.2006.883802.
- [5] U. Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, Arkadiusz Gertych, Ru San Tan, A deep convolutional neural network model to classify heartbeats, *Computers in Biology and Medicine*, Volume 89, 2017, Pages 389-396, ISSN 0010-4825, <https://doi.org/10.1016/j.compbiomed.2017.08.022>.
- [6] Carlos A. Moreno-Camacho, Jairo R. Montoya-Torres, Anicia Jaegler, Natacha Gondran, Sustainability metrics for real case applications of the supply chain network design problem: A systematic literature review, *Journal of Cleaner Production*, Volume 231, 2019, Pages 600-618, ISSN 0959-6526, <https://doi.org/10.1016/j.jclepro.2019.05.278>.
- [7] Ahmad, Zeeshan, et al. "ECG heartbeat classification using multimodal fusion." *IEEE Access* 9 (2021): 100615-100626.
- [8] Nurmaini, S.; Umi Partan, R.; Caesarendra, W.; Dewi, T.; Naufal Rahmatullah, M.; Darmawahyuni, A.; Bhayyu, V.; Firdaus, F. An Automated ECG Beat Classification System Using Deep Neural Networks with an Unsupervised Feature Extraction Technique. *Appl. Sci.* 2019, 9, 2921. <https://doi.org/10.3390/app9142921>
- [9] Alfaras, Miquel and Soriano, Miguel and Ortín, Silvia. (2019). A Fast Machine Learning Model for ECG-Based Heartbeat Classification and Arrhythmia Detection. *Frontiers in Physics*. 7. 10.3389/fphy.2019.00103.
- [10] Zhiguang Wang and Tim Oates. Imaging Time-Series to Improve Classification and Imputation(2015).arXiv.1506.00327
- [11] Vavekanand, Raja, A Machine Learning Approach for Imputing ECG Missing Healthcare Data (May 9, 2024). Available at SSRN: <https://ssrn.com/abstract=4822530>