BM4152 - Biosignal Processing

Fonseka K. W. T. S. — 200173J

Herath H. M. E. A. C. — 200212F

University of Moratuwa — December 12, 2024

Department of Electronic and Telecommunication Engineering

Semester 7 (Intake 2020)

Paper Implementation - Project Report

This document is submitted as a partial fulfillment of the module BM4152 - Biosignal Processing.

- Paper Title -

Classification of ECG Heartbeats Using Nonlinear Decomposition Methods and Support Vector Machine [1]

Contents

1 Introduction and Background

The electrocardiogram (ECG) is a crucial diagnostic tool in cardiology, enabling the detection of various cardiac conditions through the analysis of the heart's electrical activity. However, manually identifying arrhythmias is challenging due to the complexity of ECG signals and subtle morphological differences between heartbeats. Automated approaches combining advanced signal processing and machine learning can be effective in overcoming these challenges.

This report documents the implementation of the signal processing pipeline described in the paper "*Classification of ECG Heartbeats Using Nonlinear Decomposition Methods and Support Vector Machine*" by K. N. Rajesh and R. Dhuli [1]. The paper proposes a novel approach to classify 5 types of ECG heartbeats using non-linear decomposition methods such as Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) for feature extraction and support vector machines for classification.

The purpose of this project was to replicate the methodology described in the paper and assess its feasibility and performance using the publicly available MIT-BIH arrhythmia database [2]. The report provides an overview of the implementation process, including data pre-processing, signal decomposition, feature extraction, and classification, and highlights the challenges encountered and solutions devised during the replication.

2 Overview of the Paper

2.1 Dataset

The work in this paper is mainly studied using the MIT-BIH (Massachusetts Institute of Technology-Beth Israel Hospital) arrhythmia database [2]. Later, to assess the effectiveness of the approach, they applied it to the INCART (Institute of Cardiological Technics St. Petersburg) database [3]. In this study, they have considered five types of heartbeats.

- Normal (N)
- Premature Ventricular Contraction (PVC)
- Atrial Premature Contraction (APC)
- Left Bundle Branch Block (LBBB)
- Right Bundle Branch Block (RBBB)

2.1.1 MIT-BIH Arrhythmia Database

In this database, several types of heartbeats obtained from male and female patients are stored in 48 records. Each record duration is 30:06 min sampled at 360 Hz. Includes detailed annotations and labels that enable accurate identification and classification of arrhythmias. For this work, 2000 beats from each type (N, PVC, APC, LBBB, and RBBB) were selected.

2.1.2 INCART Database

This database is primarily used for testing purposes and includes data categorized into four types. It comprises 75 annotated ECG recordings (half-hour), all sampled at 360 Hz.

2.2 Objectives of the Research

The objective of the research presented in the paper, is to develop and evaluate an automated classification system for ECG heartbeats, addressing the challenges posed by their non-linearity and variability. The study aims to enhance arrhythmia detection by integrating non-linear decomposition techniques with machine learning, thus improving the accuracy and reliability of classification.

2.3 Challenges in ECG Classification

Classifying ECG heartbeats for arrhythmia detection presents several challenges due to the complex and dynamic nature of the ECG signal.

- **Signal Non-linearity and Non-stationarity**: ECG signals exhibit inherent nonlinearity and non-stationarity. Standard decomposition techniques often fail to adapt to this dynamic nature.
- **Uncertain ECG Morphology**: ECG morphology can vary even within the same individual under different conditions. But, it's possible to observe similar morphologies in different types of ECG beats.
- **Presence of Noise and Artifacts**: ECG recordings are often contaminated by high-frequency noise (e.g., power line interference) and low-frequency noise (e.g., baseline wander caused by respiration).
- **Feature Selection for Robust Classification**: Identifying features that accurately represent ECG dynamics and differentiate between arrhythmia classes is challenging.
- **Imbalanced Datasets**: ECG databases are often dominated by normal beats. This imbalance can bias classification models, reducing sensitivity to rare conditions.

2.4 Main Contributions

The key contributions of the paper can be identified as follows.

1. **Application of EEMD and EMD for ECG classification**: The paper introduces the use of EMD and EEMD to extract IMFs from ECG signals, providing a robust method for analyzing their nonlinear and non-stationary characteristics.
2. **Feature Extraction Based on IMFs**: It proposes a comprehensive set of features derived from IMFs.
3. **Integration with SMO-SVM Classifier**: The research employs a Sequential Minimal Optimization-based Support Vector Machine (SMO-SVM), demonstrating high sensitivity, specificity, and accuracy.

3 Methods Description

The methodology they have used in the paper is discussed in this section.

3.1 Dataset Extraction

To create the dataset for classification, they used the following records from the MIT-BIH arrhythmia database for each beat type. A summary of the records used is shown in Figure 1

Type of ECG beat	Number of beats	Record name
Normal (N)	2000	100,101,108,112
Premature Ventricular Contraction (PVC)	2000	106,107,200,201
Atrial Premature Contraction (APC)	2000	100,101,103,108,112,114,116, 118,121,124,200,201,202,205, 207,209,213,215,219,220,222, 223,228,231,232,233
Left Bundle Branch Block (LBBB)	2000	109,111,207,214
Right Bundle Branch Block (RBBB)	2000	118,207,212

Figure 1: Number of beats and records used in the paper

3.2 Denoising

The denoising technique consists of four steps.

1. Eliminate the mean bias from the noisy ECG signal.
2. 5th order moving average filter (cut-off frequency = 24 Hz) is used for eliminating the high-frequency components due to power line interference and muscle noise.
3. A high-pass filter with a cut-off frequency of 1 Hz is used for baseline wander suppression.
4. A low-pass filter with a cut-off frequency of 45 Hz is used to further suppress any left-out high-frequency artifacts.

3.3 Segmentation

The MIT-BIH arrhythmia database includes detailed annotations and labels provided by the experts. To obtain a heartbeat, a window of length 300 is applied on ECG signal. This corresponds to a segment size of 830 ms in the time domain.

3.4 Signal Decomposition

Two main methods are used for decomposing the signal.

1. Empirical mode decomposition (EMD)
2. Ensemble empirical mode decomposition (EEMD)

3.4.1 Empirical mode decomposition (EMD)

The EMD process generates IMFs using the sifting process described in [4]. This operation is continued until the final IMF component becomes the residue of the signal. In this method, the first 5 IMFs are extracted from each ECG segment.

3.4.2 Ensemble empirical mode decomposition (EEMD)

EEMD gives the true IMF components. Computing true IMFs using EEMD is described in [5]. They have considered the first 7 IMFs in the EEMD analysis. EMD toolbox [6] has been used for implementing these algorithms.

3.5 Feature Extraction

Various parameters are extracted from ECG to generate feature vectors for the classification mechanism. These parameters are calculated from IMF1 - IMF5 for EMD and IMF1 - IMF7 for EEMD. The parameters used here are,

1. Sample entropy (SEN) - A measure of regularity of a time series.
2. Coefficient of variation (CV) - Compares the degree of dispersion from normal ECG beats to abnormal ECG beats.
3. Singular values (SV) - Describe the energy distribution of different modes.
4. Band power (BP) - Represents the average power of each IMF.

All parameters are extracted from each of the selected IMFs of length 300. Therefore, the feature vector length when using EMD is 20, and when using EEMD, is 28.

3.6 Classification

In this paper, they have used a sequential minimal optimization-support vector machine (SMO-SVM) for discriminating five types of heartbeats. Training SVM with a large number of examples leads to a very large quadratic programming (QP) optimization problem. SMO solves very large QP optimization problems analytically. It decomposes the given QP optimization problem into small QP problems.

They have also used the Kernel method to transform the patterns into higher dimensional space where patterns can be linearly separable. They have tested it out with **linear, gaussian, and cubic kernels** for classification.

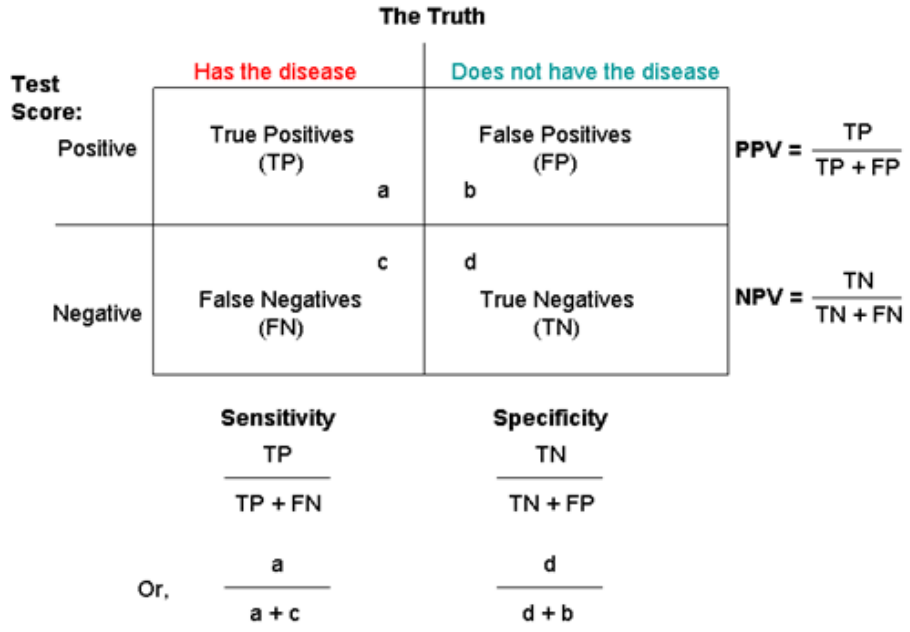

Figure 2: Sensitivity and specificity [7] of a classification

Sensitivity, specificity, and accuracy are the performance measures used in this work. The 10-fold cross-validation method is employed to calculate these measures. In 10-fold cross-validation, the entire dataset is divided into 10 subsets. Each time, 1 subset is used for testing, while the remaining 9 subsets are used for training. This process is repeated 10 times. Finally, the average of the measures across all folds is computed.

4 Implementation Details

Based on the methods they have explained in the paper, our implementation details and the challenges we faced are discussed in this section.

4.1 Dataset Extraction

Based on the data records they have given in Figure 1, we tried to create a dataset of 2000 beats for each type of beat.

4.1.1 Challenges Faced

For a single data type, by using only the records mentioned in Figure 1, we could not obtain 2000 beats. Some data types had a lesser number of ECG beats.

For example; the paper mentions obtaining 2000 beats for Premature Ventricular Contraction (PVC) using records 106, 107, 200, and 201. However, according to the MIT-BIH Arrhythmia Database Directory, the total number of PVC beats in these records appears to be 1603 (520 + 59 + 826 + 198).

4.1.2 Solutions Implemented

We sent an email to the authors asking them to explain the process they have used for ECG segmenting to obtain 2000 beats for each type. Since we haven't got any reply for that we used the whole dataset for obtaining those ECG segments in each type. This resulted in 100017 ECG segments with the following distribution as shown in Figure 3.

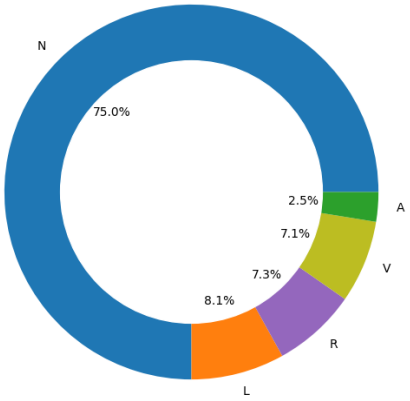

Figure 3: Dataset distribution after segmentation

Numerically, the data distribution contains N - 75016, V - 7129, A - 2546, L - 8071, and R - 8071 ECG segments. Since we used only the MIT-BIH arrhythmia database for our implementation, 2500 beats were selected randomly where 2000 were used as the training set and 500 were used as the test set from each class.

4.2 Denoising

We have implemented a similar denoising technique (mean bias correction, moving average filter of order 5) as mentioned in the paper but with some slight modifications.

4.2.1 Additional Improvements Done

For the moving average filter also we compensated for the phase shift by using the group delay of the filter. We also used the `edge` mode padding to repeat the edge values of the the signal.

```
# Calculating phase shift using filter order
phase_shift = math.ceil((window_size - 1) / 2)
# Compensating for the phase shift by padding the filtered signal
compensated_signal = np.pad(filtered_signal, (phase_shift, 0), mode='
    edge')
```

For HPF and LPF we have used 6th order Butterworth Filters with a Forward-Backward filtering technique and have cut-off frequencies of 1 Hz and 45 Hz respectively.

```
b, a = sg.butter(order, cutoff, btype = 'high') # Butterworth HPF
b, a = sg.butter(order, cutoff, btype = 'low') # Butterworth LPF
filtered_signal = sg.filtfilt(b, a, signal) # Forward Backward filtering
```

4.3 Segmentation

ECG beats with 300 data points which corresponds to 830 ms were extracted from the records.

4.3.1 Challenges Faced

They haven't mentioned how they have selected a reference point for getting an ECG segment.

4.3.2 Solutions Implemented

Therefore we used the annotation of the R peaks to segment the 300-point window. So that the R peak annotation will be centered (150th point) in the extracted ECG segment. Figure 4 below shows such a denoised signal in which the R peak annotation is centered in the window.

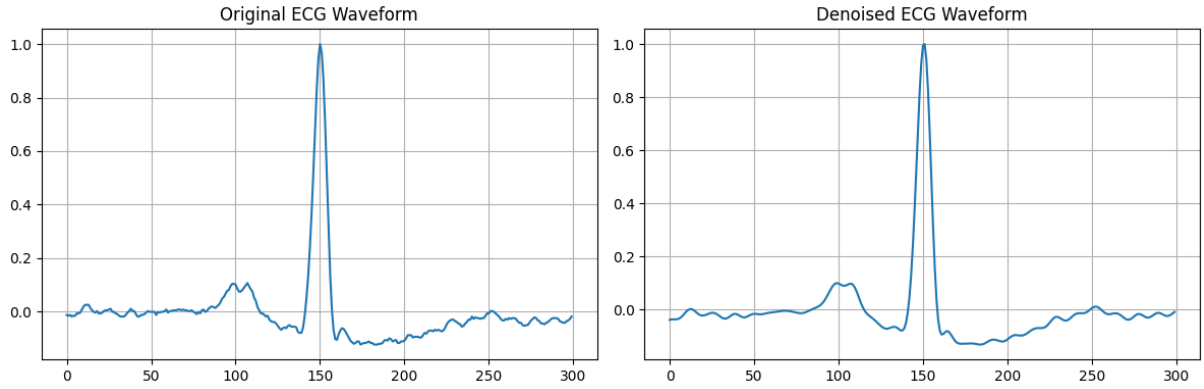

Figure 4: Original ECG segment vs denoised ECG segment of the same record

4.3.3 Additional Improvements Done

We then normalized all the ECG segments using z-score normalizing and scaled it to the range of `-1` to `+1` to avoid any biases that would occur due to the amplitudes of the signal during classification.

```
pos = annotation.sample[i] # Position of the R-peak (sample index)
# To make sure we don't go out of bounds when extracting the beat
if window_size // 2 <= pos < len(signals) - window_size // 2:
    beat = signals[pos - window_size // 2 : pos + window_size // 2]
    beat = stats.zscore(beat) # Z-score normalization
    beat = beat / np.max(np.abs(beat)) # Unity scaling
```

4.4 Signal Decomposition

PyEMD library is used for EMD and EEMD.

4.4.1 EMD

- For `extrema_detection` out of `simple` and `parabol` methods we chose the `simple` method.
- And for the envelope `spline_kind`, we chose the `cubic` spline.

The following stop criteria were used for the decomposition of IMFs. `std_thr` = 0.3, `total_power_thr` = 0.0005, `range_thr` = 0.0001 which corresponds to variance change between siftings, how much of energy is solved, and whether the difference is tiny respectively.

Challenges Faced

When we implemented 5 IMFs, we noticed that some of the segments had only less than 5 IMFs. Thereby reducing the dataset from 12500 to 4094 (training set - 3279, testing set - 815). Also, the dataset was highly unbalanced resulting in a distribution of N - 1138, V - 264, A - 1510, L - 322, and R - 860 (where each should be 2500).

Solutions Implemented

Therefore we reduced the number of IMFs computed to 3 instead of 5. Now the dataset was reduced only from 12500 to 12464 (training set - 9973, testing set - 2491). The dataset was better than before, resulting in a distribution of N - 2500, V - 2467, A - 2497, L - 2500, and R - 2500. These 3 IMF datasets were used to generate the feature extraction vector and then for classification.

```
emd = EMD(spline_kind = "cubic", std_thr = 0.3, total_power_thr =
    0.0005, range_thr = 0.0001)
IMFs = emd(X_i, max_imf = EMD_IMFS)
if (len(IMFs) > EMD_IMFS):
    IMFs = IMFs[0 : EMD_IMFS]
```

4.4.2 EEMD

- 100 `trials` were used to get the EMD performance with added noise.
- Standard deviation of AWGN is set by `noise_width = 0.05`.

7 IMFs were obtained from each trial and averaged across the trials.

```
eemd = EEMD()
```

Challenges Faced

One of the main challenges we faced in running the EEMD algorithm was the computational power it required and the time taken to run the decomposition algorithm for the whole dataset.

Solutions Implemented

We mostly used Google Colab for our implementation. So to run this part of the code we used Kaggle because it offers to run the code on the server even though the PC is off.

4.5 Feature Extraction

For each IMF we calculated the four parameters Sample entropy (SEN), Coefficient of variation (CV), Singular values (SV), and Band power (BP) to create the feature vector. The following code segment was used to calculate them.

```
# Perform SVD and extract singular values (SV)
singular_value = svd(imf.reshape(-1, 1), compute_uv=False)[0]

# Calculating Coefficient of Variation (CV)
mean_imf = np.mean(imf)
std_imf = np.std(imf)
CV = (std_imf / mean_imf) ** 2

# Calculating Sample Entropy (SEN)
sampEn = ant.sample_entropy(imf)

# Calculating Band Power (BP)
bp_value = np.mean(np.square(imf))
```

For EMD since only 3 IMFs are extracted and there are 4 features for each IMF, a feature vector of length 12 was generated for an individual ECG segment. For EEMD since we extract 7 IMFs, a feature vector of length 28 was generated for an individual ECG segment. These feature vectors were then used for classification.

4.5.1 Additional Improvements Done

To prevent biases towards certain features, we standardized the features by removing the mean and scaling them to unit variance. Statistics (median and interquartile range) of these feature vectors (in EMD method with 5 IMFs) for each class are presented in Figure 5.

	Features	Normal	PVC	APC	LBBB	RBBB
0	Class	0.00000 ± 0.00000	1.00000 ± 0.00000	2.00000 ± 0.00000	3.00000 ± 0.00000	4.00000 ± 0.00000
1	SV 1	0.05089 ± 0.02900	0.10905 ± 0.05707	0.10800 ± 0.05504	0.06914 ± 0.01787	0.05185 ± 0.02695
2	CV 1	118.03880 ± 276.44496	97.15311 ± 216.34062	168.74910 ± 344.25535	54.34901 ± 88.00730	208.46506 ± 367.94844
3	SEN 1	2.25398 ± 0.48159	2.36608 ± 0.90599	2.20296 ± 0.35158	2.77913 ± 0.76851	2.56959 ± 0.78599
4	BP 1	0.01693 ± 0.00730	0.01866 ± 0.01522	0.01618 ± 0.00537	0.02575 ± 0.01365	0.02201 ± 0.01352
5	SV 2	0.13836 ± 0.03645	0.13822 ± 0.04158	0.20190 ± 0.04442	0.11981 ± 0.02427	0.13351 ± 0.03482
6	CV 2	126.53466 ± 250.60584	43.89344 ± 107.86725	269.42422 ± 546.14917	64.11625 ± 111.13538	135.15813 ± 266.98180
7	SEN 2	2.66169 ± 0.57599	4.51172 ± 1.41158	2.80197 ± 0.54473	4.22164 ± 1.13531	3.09875 ± 0.71577
8	BP 2	0.02362 ± 0.01088	0.06785 ± 0.04211	0.02617 ± 0.01064	0.05941 ± 0.02994	0.03201 ± 0.01475
9	SV 3	0.11978 ± 0.03777	0.06832 ± 0.02516	0.13797 ± 0.03231	0.06548 ± 0.02005	0.11069 ± 0.02796
10	CV 3	37.07178 ± 76.66323	13.76464 ± 32.90011	28.18380 ± 82.98832	12.38100 ± 17.83627	40.89680 ± 86.77326
11	SEN 3	2.22955 ± 0.54296	3.17498 ± 0.98543	2.55662 ± 0.60321	2.40384 ± 0.84124	2.29719 ± 0.68511
12	BP 3	0.01657 ± 0.00807	0.03360 ± 0.02099	0.02179 ± 0.01010	0.01926 ± 0.01392	0.01759 ± 0.01045
13	SV 4	0.04432 ± 0.02240	0.00565 ± 0.01322	0.06462 ± 0.03068	0.01321 ± 0.01508	0.02796 ± 0.02107
14	CV 4	9.18045 ± 12.03861	0.63642 ± 2.33704	16.03715 ± 30.51351	4.99390 ± 7.83703	10.23459 ± 14.72090
15	SEN 4	1.11257 ± 0.47544	1.07103 ± 0.95790	0.94626 ± 0.43586	1.40144 ± 0.45700	1.31043 ± 0.54085
16	BP 4	0.00413 ± 0.00361	0.00382 ± 0.00612	0.00298 ± 0.00319	0.00655 ± 0.00432	0.00572 ± 0.00491
17	SV 5	0.00000 ± 0.00201	0.00000 ± 0.00000	0.00246 ± 0.00602	0.00000 ± 0.00000	0.00000 ± 0.00112
18	CV 5	0.00000 ± 0.35791	0.00000 ± 0.00000	0.12744 ± 0.69882	0.00000 ± 0.00000	0.00000 ± 0.15508
19	SEN 5	0.00000 ± 0.23928	0.00000 ± 0.00000	0.30137 ± 0.34953	0.00000 ± 0.00000	0.00000 ± 0.25342
20	BP 5	0.00000 ± 0.00038	0.00000 ± 0.00000	0.00030 ± 0.00081	0.00000 ± 0.00000	0.00000 ± 0.00043

Figure 5: Median ± interquartile range of features extracted in the EMD method.

4.6 Classification

For our work, we have only implemented the SVM without the SMO technique. We tested the SVM with linear, gaussian, and cubic kernels for classification. The regularization parameter (C) varies from 0.1 to 100 for each kernel to find optimal mode parameters (Grid search).

```
# Range for the C parameter
C_range = np.logspace(-1, 2, 10)  # Logarithmic scale from 0.1 to 100
# Kernels to test
kernels = ['linear', 'poly', 'rbf']
```

5 Implementation Results

5.1 Classification Results Using EMD

Since the data extraction methods (from the data records) they used was not clearly mentioned in the paper, we tried out different techniques to see which would result in the best classification. The results of each of those techniques are discussed in the following sections. Performance measures were calculated as mentioned before in Section 3.6.

5.1.1 Using 5 IMFs - Replacing NaN with Zeroes

In the first method for EMD, we use 5 IMFs as mentioned in the paper. Since some of the ECG beats do not decompose into 5 IMFs, the features related to them give a NaN value. So, to proceed further we replaced those NaN values with zeroes. No data rows were removed in this. Therefore the training set remained at 10000 datapoints (2000 each) and the testing set remained at 2000 datapoints (500 each). The classification results are given in Figures 6 and 7.

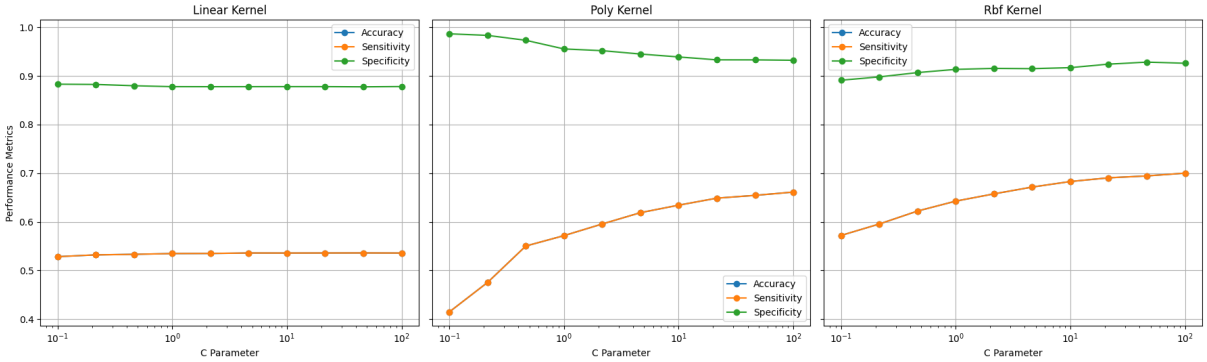

Figure 6: Performance metrics vs C parameter for different kernels.

The optimal SVM model parameters are `kernel = 'rbf'` and `C = 100`. The plots in Figure 6 were for 10-fold cross-validation. When the optimal parameters are used with the testing data the performance metrics were as follows. Accuracy - 67.48%, Sensitivity - 67.50%, Specificity - 91.87%.

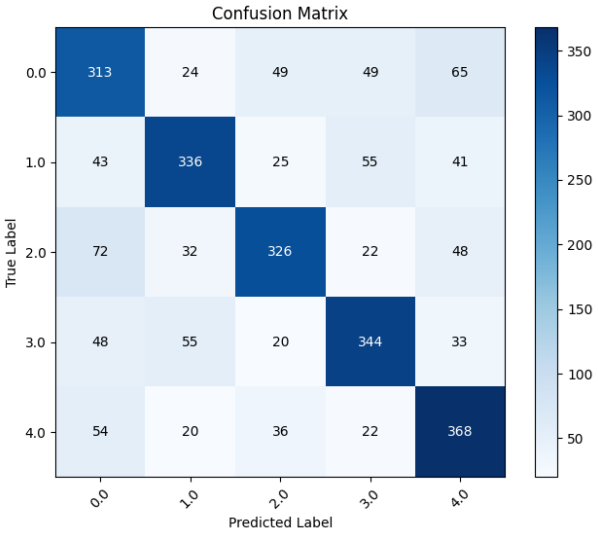

Figure 7: Confusion matrix for classification with optimal parameters.

5.1.2 Using 5 IMFs - Removing Vectors with NaNs

Rather than replacing NaN values with zeroes, we removed the rows which contain at least one NaN value. Therefore the training set was reduced to 3279 datapoints and the testing set was reduced to 815 datapoints. This resulted in an unbalanced dataset as mentioned before. The classification results are given in Figures 8 and 9.

The optimal SVM model parameters are `kernel = 'rbf'` and `C = 21.54`. The plots in Figure 8 were for 10-fold cross-validation. When the optimal parameters are used with the testing data the performance metrics were as follows. Accuracy - 71.41%, Sensitivity - 61.26%, Specificity - 92.09%.

Unlike in Figure 7 the confusion matrix in Figure 9 does not show an even distribution between classes. That may be because our dataset is highly unbalanced after removing the rows with NaN values. Therefore

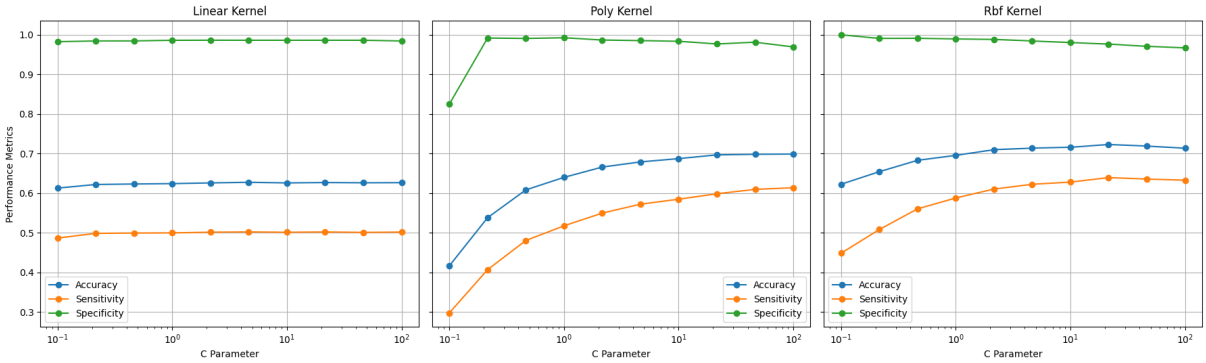

Figure 8: Performance metrics vs C parameter for different kernels.

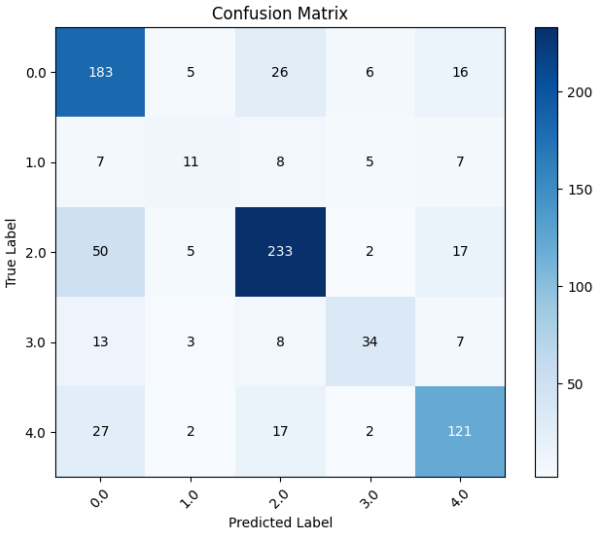

Figure 9: Confusion matrix for classification with optimal parameters.

even though the accuracy is better than the previous method, we cannot conclude that the SVM model trained with removing vectors having NaNs performs better.

5.1.3 Using 3 IMFs - Removing Vectors with NaNs

As mentioned before to avoid the issue of lack of IMFs in all ECG segments, we tried our implementation with 3 IMFs. Since the dataset is not as unbalanced as before, we tested the 3 IMFs method only by removing the rows containing NaN values. The classification results are given in Figures 10 and 11.

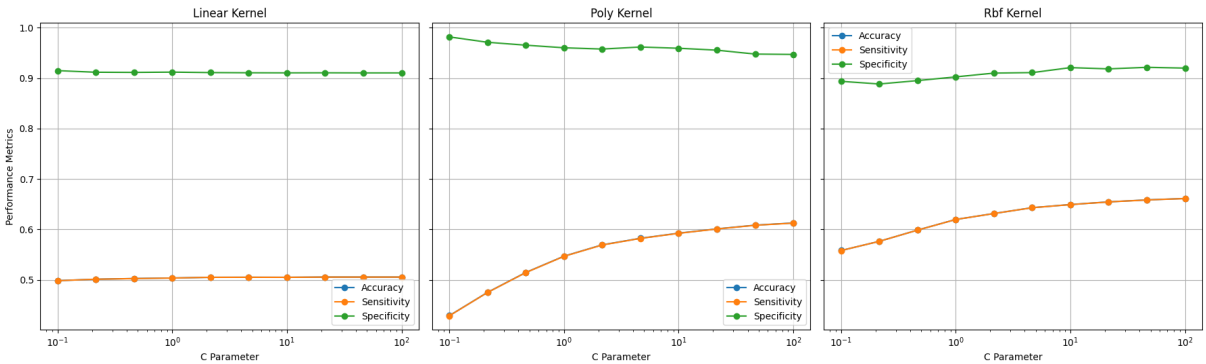

Figure 10: Performance metrics vs C parameter for different kernels.

The optimal SVM model parameters are `kernel = 'rbf'` and `C = 100`. The plots in Figure 10 were for 10-fold cross-validation. When the optimal parameters are used with the testing data the performance metrics are as follows. Accuracy - 65.72%, Sensitivity - 65.70%, Specificity - 91.43%.

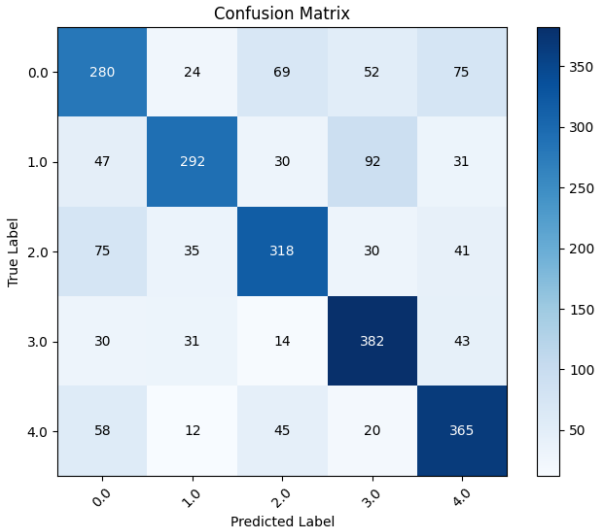

Figure 11: Confusion matrix for classification with optimal parameters.

The 5 IMFs, with NaN values replaced by zeroes, showed the best results. We assumed that the feature vector created from 3 IMFs, after removing rows with NaN values, was not complex enough to identify patterns and classify between the models, even though all the ECG segments contained the 3 IMFs.

5.1.4 Comparison Between the Results of the Paper and Our Implementation

Table 1 summarizes and compares the results for EMD between the original paper and our implementation. In the original paper, they got the best results using the `'rbf'` kernel. For our implementation also we got the best results using the `'rbf'` kernel.

	Accuracy (%)	Sensitivity (%)	Specificity (%)
Original Paper ('rbf')	96.45	85.80	98.81
Our Implementation ('rbf')	67.48	67.50	91.87

Table 1: Comparison between the results of the paper and our implementation in EMD

5.2 Classification Results Using EEMD

Similar to the original paper, we selected the first 7 IMFs generated from EEMD signal decomposition. After extracting above mentioned features we removed the signals with NaN entries. The classification results are depicted in Figure 12.

The optimal kernal for the SVM is 'rbf'. The highest specificity was achieved when c = 21.54. When the optimal parameters are used with the testing data the Accuracy was 93.87%. Confusion matrix is shown in Figure 13.

5.3 Corelation Based Feature Selection

To assess the importance of each feature in classifying ECG beats, we calculated Pearson's correlation coefficients between the class and each feature. The absolute value of these coefficients of the feature are given in Figure 14. They have been sorted in descending order.

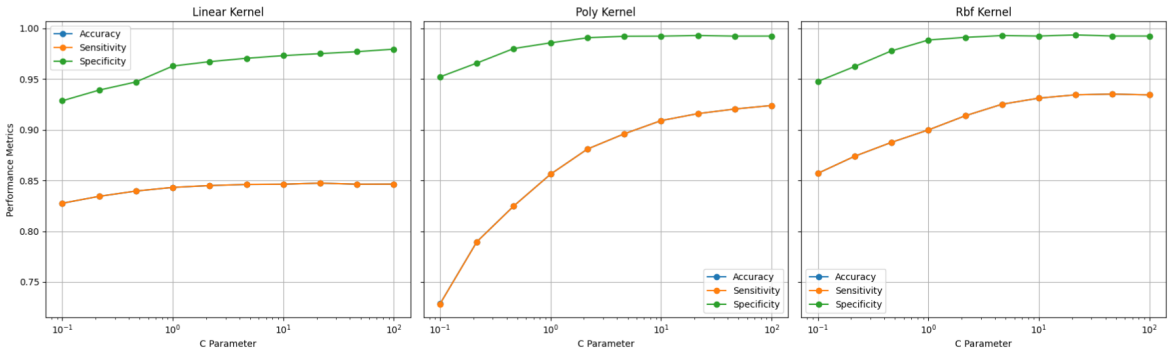

Figure 12: Performance metrics vs C parameter for different kernels.(EEMD)

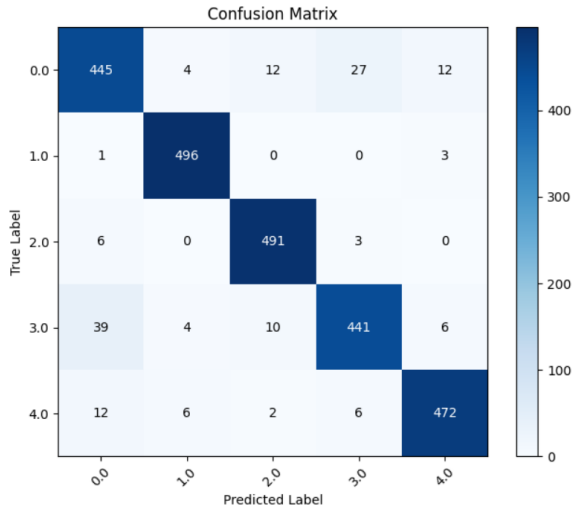

Figure 13: Confusion matrix for classification with optimal parameters.(EEMD)

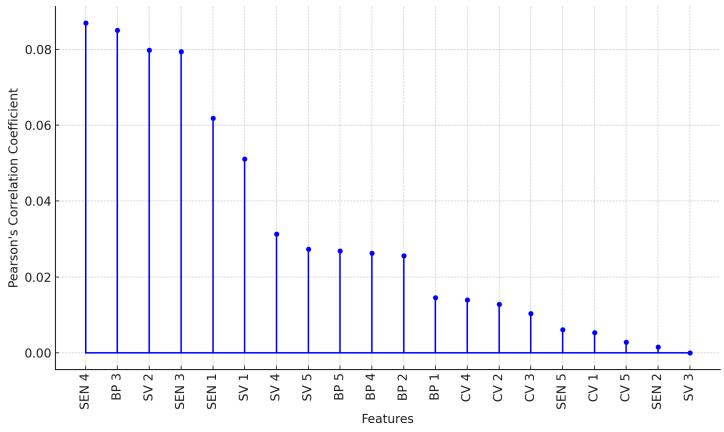

Figure 14: Features sorted in descending order based on Pearson's correlation.

Then we plotted the accuracies of models trained with each increasing feature. By Figure 15, we can see that the maximum accuracy of 67.48% is reached when all the 20 features are used. But also when the first 11 sorted features are used for training we can get an equivalent accuracy of 65.64% similar to the full model.

The correlation coefficient indicates the strength of the linear relationship between the feature and the class, giving only a rough estimate. Since these relations are often nonlinear, this is not a perfect method to assess the importance of a feature.

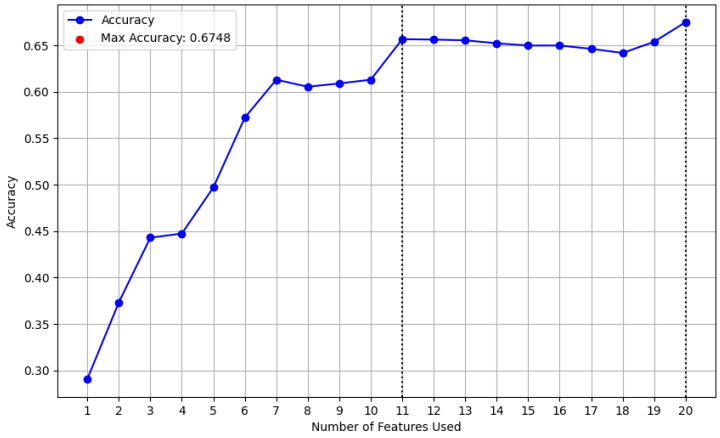

Figure 15: Overall accuracies for different numbers of features selected using correlation-based feature selection.

6 Conclusion and Possible Improvements

This report documents the replication of a study focused on classifying ECG heartbeats using non-linear decomposition methods, specifically EMD and EEMD, combined with an SVM classifier. While the implementation successfully reproduced key steps of the original methodology, it faced several challenges, including unclear dataset preparation instructions from the original paper and computational limitations with EEMD. These challenges resulted in alternate data pre-processing methods and led to incomplete replication of certain aspects of the study.

The best classification results were achieved using EMD with 5 IMFs and replacing NaN values with zeroes, yielding an overall accuracy of 67.48%. However, this fell significantly short of the original study's reported performance of 96.45%. Reducing the number of IMFs to 3 helped balance the dataset but resulted in less complex feature vectors, negatively impacting classification performance.

Understanding the exact relationship between underlying physiology and extracted features is crucial for applying these methods in real-life scenarios. The importance of each feature in classifying heartbeats can be assessed using an ablation test. Furthermore, various types and numerous beats should be incorporated to generalize the results. Instead of fixed windows, one can use adaptive methods to segment heartbeats. Addressing computational challenges, particularly with EEMD, through optimized algorithms or advanced hardware would also enable a more comprehensive replication of the original study's methods.

In conclusion, while the replication encountered limitations, it provides a solid foundation for advancing automated ECG analysis by identifying critical areas for improvement and emphasizing the importance of pre-processing and computational strategies in achieving optimal results.

References

[1] K. N. Rajesh and R. Dhuli, "Classification of ecg heartbeats using nonlinear decomposition methods and support vector machine," *Computers in Biology and Medicine*, vol. 87, pp. 271–284, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482517301701

[2] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," pp. e215–e220, 2000, [Online]. [Online]. Available: https://physionet.org/content/mitdb/1.0.0/

[3] ——, “Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals,” pp. e215–e220, 2000, [Online]. [Online]. Available: https://physionet.org/content/incartdb/1.0.0/

[4] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0193

[5] Z. WU and N. E. HUANG, “Ensemble empirical mode decomposition: A noise-assisted data analysis method,” *Advances in Adaptive Data Analysis*, vol. 01, no. 01, pp. 1–41, 2009. [Online]. Available: https://doi.org/10.1142/S1793536909000047

[6] P. Flandrin, “Emd matlab 7.1 codes with examples,” 2007.

[7] Those Nerdy Girls, “Sensitivity and Specificity,” https://thosenerdygirls.org/sensitivity-and-specificity/, 2024, accessed: 2024-12-10.