Heart Disease Prediction from heart beat audio signals using Machine Learning and Network Analysis

Ligari D. • Alberti A. [1]

[1] *Department of Computer Engineering, Data Science, University of Pavia, Italy*
Course of Advanced Biomedical Machine Learning
Github page: https://github.com/DavideLigari01/advanced-biomedical-project

Date: June 28, 2024

Abstract — Heart disease remains one of the leading causes of mortality worldwide, making early diagnosis crucial. This study aims to predict heart diseases by analyzing heartbeat audio signals using machine learning and network analysis. We utilized a dataset from the PASCAL Classifying Heart Sounds Challenge 2011, which includes normal heart sounds, murmurs, extra heart sounds, extra systoles, and artifacts. Various preprocessing techniques such as noise reduction, resampling, and segmentation were applied to ensure data quality. Features were extracted using methods like Mel-Frequency Cepstral Coefficients (MFCC), Chroma, RMS, ZCR, and spectral features. Multiple machine learning models including LightGBM, XGBoost, CatBoost, Random Forest, and Multilayer Perceptron were trained and evaluated. The best performing model achieved high accuracy in distinguishing between different heart sound categories. This research highlights the potential of machine learning in cardiac diagnostics and provides a foundation for future advancements in the field.

Keywords ——TO BE DEFINED—

Contents

1. Introduction

1.1. Problem Domain

1.2. Research Question

1.3. Previous Research

2. Methods

2.1. Source of Data

The dataset for this project was obtained from a Kaggle repository titled *Dangerous Heartbeat Dataset (DHD)* [1], which in turn sources its data from the PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) [2]. This dataset comprises audio recordings of heartbeats, categorized into different types of heart sounds. Specifically, the dataset consists of 5 types of recordings: Normal Heart Sounds, Murmur Sounds, Extra Heart Sounds, Extrasystole Sounds, and Artifacts. Data has been gathered from the general public via the iStethoscope Pro iPhone app and from a clinic trial in hospitals using the digital stethoscope DigiScope.

2.1.1. Type of sources

2.1.2. Classes

2.1.3. Data Distribution

2.2. Data Preprocessing

To prepare the data several preprocessing operations were performed:

Noise Reduction: the audio data was already provided in a clipped format to minimize noise and irrelevant information.

Normalization: the audio are loaded using the *torchaudio.load()* function, which normalized the audio signals in the range [-1, 1].

Removal of Corrupted Files: corrupted files were identified and removed from the dataset to ensure data quality.

Outlier Detection and Removal: we investigated the average duration of each class and found the 'artifact' class to have a significantly larger average duration. This was due to a few long lasting audio recordings (see Figure 1). A large number of samples from the same audio might not be as informative, thereby we used IQR to detect and remove outliers.

Resampling: we evaluated two sampling rates to determine the optimal rate for heartbeat sounds and all audio files were resampled to a common frequency of 4000 Hz (see Section 2.4.2).

Segmentation: the audio data was segmented into 1-second intervals, identified as the optimal extraction interval (see Section 2.4.3), as it offered both good performance and dataset size increasing.

Hop and Window Size: the hop size determines the number of samples between successive windows, while the window size determines the number of samples considered. Each feature was extracted using the same window length and hop length facilitating a fair assessment of each feature's contribution to the classification task.

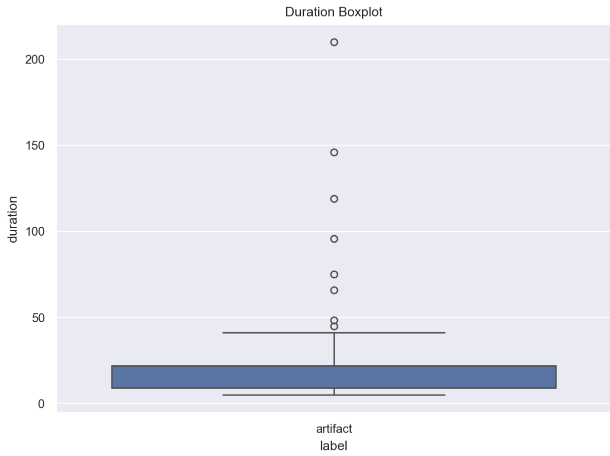

Fig. 1: Outliers in the Artifacts class.

2.3. Data Preprocessing

2.4. Feature Extraction

s demonstrated by [4] and [3], MFCCs are highly effective features for heartbeat classification. In addition to MFCCs, we incorporated other features to capture various characteristics of heart sounds, enhancing the classification accuracy. The features used are explained in the following section.

2.4.1. Features Type

MFCC
Mel-Frequency Cepstral Coefficients (MFCCs) are representations of the short-term power spectrum of sound. They are derived by taking the Fourier transform of a signal, mapping the powers of the spectrum onto the mel scale, taking the logarithm, and then performing a discrete cosine transform. MFCCs are effective in capturing the timbral texture of audio and are widely used in speech and audio processing due to their ability to represent the envelope of the time power spectrum. In heartbeat classification, MFFCs can reflect the different perceived quality of heart sounds, such as the presence of murmurs or other anomalies.

Chroma STFT
Chroma features represent the 12 different pitch classes of music (e.g., C, C#, D, etc.). They are particularly useful for capturing harmonic and melodic characteristics in music. By mapping audio signals onto the chroma scale, these features can identify pitches regardless of the octave, making them useful for analyzing harmonic content in heart sounds.

RMS
Root Mean Square (RMS) measures the magnitude of varying quantities, in this case, the amplitude of an audio signal. It is a straightforward way to compute the energy of the signal over a given time frame. RMS is useful in audio analysis for detecting volume changes and can help identify different types of heartbeats based on their energy levels. For example, in a given timeframe the RMS may be altered

by the presence of a murmur with respect to a normal heart sound.

ZCR
Zero-Crossing Rate (ZCR) is the rate at which a signal changes sign, indicating how often the signal crosses the zero amplitude line. It is particularly useful for detecting the noisiness and the temporal structure of the signal. In heartbeat classification, ZCR can help differentiate between normal and abnormal sounds by highlighting changes in signal periodicity.

CQT
Constant-Q Transform (CQT) is a time-frequency representation with a logarithmic frequency scale, making it suitable for musical applications. Since it captures more detail at lower frequencies, it may be useful for analyzing the low-frequency components of heart sounds.

Spectral Centroid
The spectral centroid indicates the center of mass of the spectrum and is often perceived as the brightness of a sound. It is calculated as the weighted mean of the frequencies present in the signal, with their magnitudes as weights. In heart sound analysis, a higher spectral centroid can indicate sharper, more pronounced sounds, while a lower centroid suggests smoother sounds.

Spectral Bandwidth
Spectral bandwidth measures the width of the spectrum around the centroid, providing an indication of the range of frequencies present. It is calculated as the square root of the variance of the spectrum. This feature helps in understanding the spread of the frequency components in the heart sounds, which can be indicative of different heart conditions.

Spectral Roll-off Spectral roll-off is the frequency below which a certain percentage of the total spectral energy lies. It is typically set at 85% and helps distinguish between harmonic and non-harmonic content. In heartbeat classification, spectral roll-off can be used to differentiate between sounds with a concentrated energy distribution and those with more dispersed energy.

2.4.2. Sampling Rate Selection

The sampling rate of the data were heterogeneous, ranging from 4000 Hz to 44100 Hz, with a majority of the data being sampled at 4000 Hz. To assess the impact of the sampling rate on the classification performance, we trained different models on different features, extracted at different sampling rates and from various intervals. Each model is then evaluated using different metrics, taking into account the class imbalance issue. We also consider a possible dependency between the sampling rate and the extraction interval, as shown in Algorithm 1.
The results, reported in Figure 2 showed no evident advantage to using a mix of sampling frequencies over a fixed resampled sample rate. Moreover, employing a fixed sample rate of 4000 Hz reduces the risk of introducing bias, enhances efficiency, and permits the use of a broader range of features and models.

Algorithm 1 Sampling rate choice

Input:
features = [`mfcc30 & 120`, `cqt30 & 70`, `chroma12`]
sampling_rates = [`mix`, `4000`]
extraction_intervals = [`0.5`, `1`, `2`, `3`]
models = [`rf`, `svm-rbf`, `lr`]
metrics = [`macrof1`, `mcc`]
for sr **in** sampling_rates **do**
  for interval **in** extraction_intervals **do**
    for feature **in** features **do**
      extract feature with interval at sr
      for model **in** models **do**
        train model with extracted feature
        for metric **in** metrics **do**
          evaluate model with metric
Output:
Given all the results, group by `model` and average the values of a specific `metric` across `features`

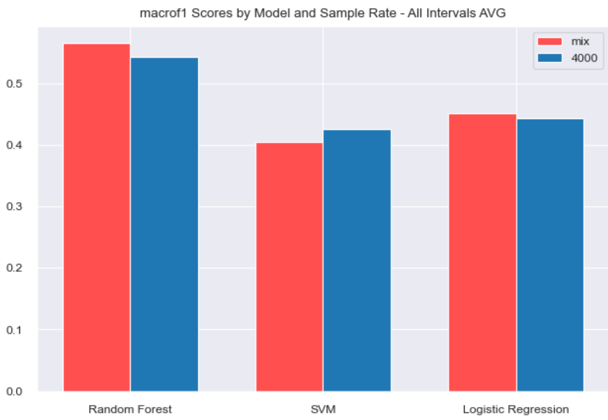

Fig. 2: Comparison of the macro F1 score for different sampling rates.

2.4.3. Extraction Interval Selection

The extraction interval refers to the duration of the audio segment from which the features are extracted. Its choice was conducted similarly to the sampling rate selection, considering only the fixed sample rate of 4000 Hz. However the interval also affects the number of samples available in each class, so it must be chosen carefully, especially considering the limited number of samples available for some classes. The results showed that a 2-second interval yielded the best performance, however it also reduced the number of samples, impeding a correct training and evaluation of the models. As a consequence, we picked a 1-second interval as a compromise.

2.4.4. Number of Features per Type

2.5. Feature Selection

2.6. Models

2.6.1. Metrics used

2.6.2. Prevention Model

2.6.3. Support Model

2.7. Tools and Software

- Scikit-learn - Numpy - Pandas - Matplotlib - Seaborn - Scipy - XGBoost - CatBoost - PyTorch - Torchaudio - Librosa - TensorFlow - Keras - Shap - Imblearn - Other Utility Libraries (e.g. joblib, os, sys, etc.)

3. Results

3.1. Prevention Model

3.2. Support Model

3.2.1. Explainability

3.3. Other Experiments

3.3.1. CNNs

3.3.2. Tiered Ensemble Model

3.3.3. Data Augmentation

4. Discussion

4.1. Positioning in Existing Research

4.2. Limitations

5. Conclusion

5.1. Overall Impression

5.2. Future Work

6. Appendix

References

[1] en. URL: https://www.kaggle.com/datasets/mersico/dangerous-heartbeat-dataset-dhd.

[2] P. Bentley et al. *The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results*. URL: http://www.peterjbentley.com/heartchallenge/index.html.

[3] Wei Chen et al. "Deep Learning Methods for Heart Sounds Classification: A Systematic Review". In: *Entropy* 23.6 (May 2021), p. 667. ISSN: 1099-4300. DOI: 10.3390/e23060667.

[4] Ali Raza et al. "Heartbeat Sound Signal Classification Using Deep Learning". en. In: *Sensors* 19.2121 (Jan. 2019), p. 4819. ISSN: 1424-8220. DOI: 10.3390/s19214819.