Phonocardiogram Classification

Author: Jake Dumbauld

March – April, 2022

Contents

1 Problem

"Congenital heart diseases affect about 1% of newborns, representing and important morbidity and mortality factor for several severe conditions, including advanced heart failure," (Reyna, et al., n.d.). Auscultation, the action of listening to the sounds of the heart and other organs, provides early and easily accessible information regarding congenital and acquired cardiac diseases, particularly in children. Around 10% of adults, and 30% of children have a harmless heart murmur (Harvard Health Publishing, 2022). While less than 1% of murmurs are pathological, (Doshi, 2018) heart sounds remain an important indicator of potential underlying cardiac disease. Phonocardiograms are recordings of these heart sounds.

Worldwide, there is a significant delta between the available experts who are certified to auscultate and evaluate patients for the presence of a murmur and the number of experts needed to satisfy this demand. This disparity is exaggerated in developing countries (Reyna, et al., n.d.).

Machine learning is a promising solution to this problem. A machine learning model trained to identify the presence of cardiac murmurs from phonocardiograms would off-load the burden of the initial screening process, allowing the experts to complete further work-up on the patients who truly need it.

In this project, I initially set out to create a machine learning model capable of classifying phonocardiograms. Over time, my goal evolved to evaluating the impact of patient information on the performance of models I had trained to this problem.

2 An Area of Ongoing Research

In my research, I frequently referred to "Deep Learning Methods for Heart Sounds Classification: A Systematic Review" (Chen, et al., 2021). In their review, they identified a myriad of approaches with different model artchitectures, input features, optimizers, and the use of segmentation data. Segmentation separates the heart sounds into their principle components, and can be used to classify murmurs into different types, and to segment data for LSTM training. Most commonly, other research teams used a 2D convolutional neural network (CNN) architecture. Input features were varied: 1D time series signals were the most common, with some derivative features such as mel frequency cepstral coefficients (MFCC) or spectrograms. A few research teams tried to employ a recurrent neural network (RNN) architecture on both MFCC and 1D time series data.

Absent from consideration in the literature was the inclusion of patient information in the training of these models.

3 The Data

3.1 Sourcing & Basic Description

In this year's George B. Moody PhysioNet Challenge, a dataset was published that included 3163 phonocardiogram recordings as .wav files. An accompanying .csv included patient information for the 947 patients in the study. There were additional columns qualifying and grading the murmurs present in the set. Finally, segmentation data was provided for each of the files. Due to time constraints on the project, I did not utilize this segmentation data. These data were gathered across two separate campaigns in 2014 and 2015, and 14% of total sample of of phonocardiograms were from patients who attended both campaigns. A full data dictionary is available at https://moody-challenge.physionet.org/2022/.

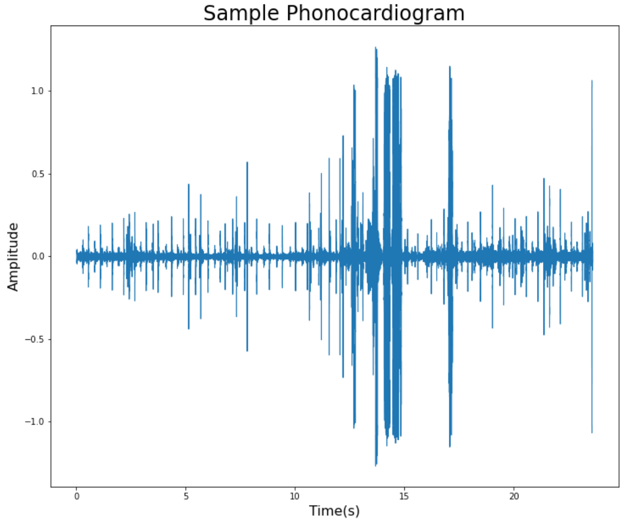
3.2 Preprocessing & Feature Engineering

3.2.0 A Brief Note

Preprocessing and feature engineering was not a linear process. However, below it is laid out linearly to provide clarity in the goals and progression. The code that produced these data was assembled in an iterative and exploratory manner as the aim of the project evolved, so please forgive any inconsistencies between the below text and the ordering of notebooks & code.

3.2.1 .wav to Uniform Amplitude Arrays

The first step in making the data machine readable was converting the .wav files into amplitude data with the librosa library.



Choosing a sampling rate is a crucial decision in this process. A sampling rate twice the highest frequency in an audio sample is sufficient to capture all the available information (Nyquist–Shannon sampling theorem, 2022). Human hearing ranges from 20 Hz - 20 kHz, and for this reason most audio is sampled at 40 kHz.

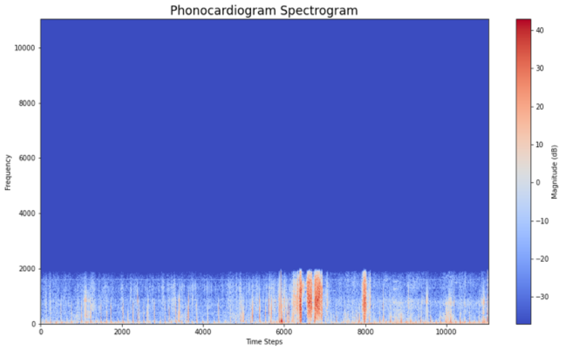
The fundamental heart sounds range in frequency between 20 Hz and 500 Hz (McGee, MD, 2018). However, in review of the data I found frequencies ranging up to 2 kHz. I was unsure if the data above 500 Hz would be of use, but I decided to create signal data with two different sampling rates to investigate this.

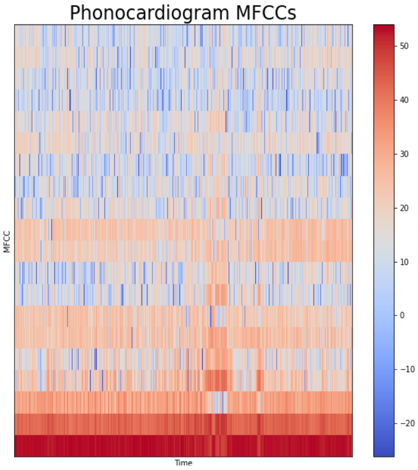
I chose 1024 Hz and 4096 Hz as my sampling rates to process the audio signal data. These sampling rates captured the information of interest while providing a distinct benefit: they reduced amount of data future models would be trained on by 20 and four-fold respectively over a standard 20 kHz sampling rate. This reduction greatly improved memory efficiency and training time of future models.

Using the librosa library, I looped over the .wav files in the dataset to generate an array with 3163 signals of varying length. I then trimmed and padded these arrays to 12 second clips for the 4 kHz signal data, and 6 seconds for the 1 kHz signal data. I chose 12 seconds because ~90% of the sample was at least that length, minimizing the amount of zero-padding necessary. The decision to trim the 1 kHz signals to 6 seconds ocurred when trying to optimize an LSTM approach which will be discussed later.

3.2.2 Amplitude to MFCCs

Mel Frequency Cepstral Coefficients are a mathematical derivation of audio data that represent frequencies in a binned fashion. This more closely approximates the human auditory system's response as compared to a linearly spaced spectrogram, whose granular data bears little resemblance to how we percieve sound. The following two plots represent this idea visually; the same audio file was used to generate the below spectra and the above waveform.
Notice how the 'loudness' in the graphical representation, indicated by dark red, is clustered towards the bottom bins of the MFCCs and appears in lower frequencies of the spectrogram.





MFCCs are well suited to input in CNN architectures due to their similarities to photographic data and their information density. A signal comprised of 49,152 time steps reduces to an array of shape (20,97) when converted to MFCCs.

Looping over my trimmed and padded 4 kHz signal data, I created MFCCs for each signal in the sample.

3.2.3 Patient Information to Signals

The patient information training_data.csv was given in in a mix of floats, integers, and strings representing categorical variables. Of the full data dictionary linked above, I kept patient age, sex, height, weight, pregnancy status, and murmur location. I converted age from a categorical string to ordinal integers ranging from one to five; unknowns were assigned to zero. I re-cast pregnancy staus from a boolean to an integer, and mapped sex to a binary variable. Height & weight were encoded as floats, so no transformation was necessary. However, there were nan values in these columns. I imputed these with the means of the patients' age & sex group, if this was available for the patient. Otherwise, I imputed with the means of the overall sample.

I then reshaped each vector into an array of static signals, congruent with the data they were concatenated to.

I then binarized the target variable, presence of a murmur. I dropped 156 audio samples; the murmur status of these samples was labelled as 'unknown' which does not fit into the binary label structure of the future models.

3.2.4 Final Data Arrays

In total, I generated 5 arrays in this process as well as one vector encoding my binary target. The arrays are summarized below.

Array Data	Patient Information?	Shape
Amplitude, sampling rate of 4 kHz.	No	(3007, 49152)
Amplitude, sampling rate of 1 kHz	No	(3007, 6144)
Amplitude, sampling rate of 1 kHz	Yes	(3007, 11, 6144)
MFCCs, sampling rate of 4 kHz	No	(3007, 20, 97)
MFCCs, sampling rate of 4 kHz	Yes	(3007, 30, 97)

4 Modelling

4.1 Simple Statistical Modelling

I initially chose to evaluate the first array with simple statistical models, as this was early in my exploration of machine learning. I had yet to learn the limitations of these models when applied to complex sequence data such as this. I used the scikit-Learn library to instantiate, fit, and evaluate linear regression, K-Nearest Neighbors (KNN), and Support Vector Machine models. Performance across all these models was poor, but KNN provided the most promising surface-level results. To improve these results, I used a scikit-Learn's implementation of GridSearch to optimize a pipeline. This included scaling, dimensionality reduction with principal component analysis, and hyperparameter tuning within the KNN function. The resulting best estimator predictably yielded no improvement in test accuracy.

4.2 Neural Networks

4.2.1 Architecture Selection

I opted to use the keras and tensorflow libraries for neural network construction due to the approachability of the code, as well as its wonderful documentation. My initial efforts in designing neural network architectures could be characterized by the phrase, "stumbling around in the dark."

In this misguided initial approach, I learned two important lessons: I needed a systematic way of creating network architectures, and I needed to constrain randomness.

Eventually, my first lesson led me to the kerastune library, which I used to search a space of potential network architectures that I determined had some efficacy in my initial approach.

The second lesson led me to setting identical seeds across the notebooks I used to search for optimal models. This ensured reproducibility of the results in my 'model search' portion of the experiment.

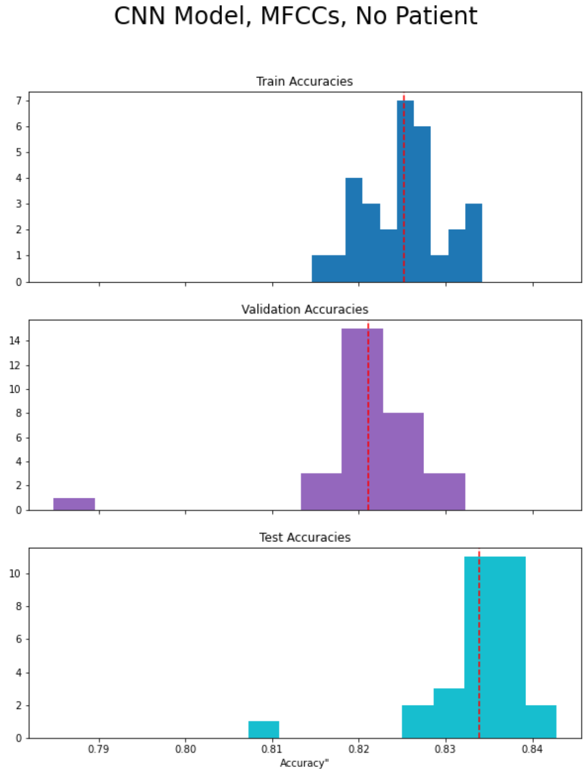
In total, I created 9 models optimized over 50 trials, each trained for 100 epochs with an early stopping callback monitoring validation loss.

Architecture	Input Data	Patient Info?
Simple Sequential	4 kHz amplitude data (12s)	No
Simple Sequential	1 kHz amplitude data (12s)	No
Simple Sequential	1 kHz amplitude data (12s)	Yes
Simple Sequential	MFCCs	No
Simple Sequential	MFCCs	Yes
CNN	MFCCs	No
CNN	MFCCs	Yes
RNN	1 kHz amplitude data (6s)	No
RNN	1 kHz amplitude data (6s)	Yes

The code for each model was written in a separate notebook so that they could be run in parallel on Google Collab. However, in an unfortunate mishap, my session training the RNN models timed out after training for 24+ hours. When I tried restarting the session, I inadvertently overwrote the directory containing the back-ups with all the old models & their scores. The result was the exclusion of the RNN models from evaluation; I had insufficient time to retrain and optimize the models for comparison.

4.2.2 Model Evaluation

With optimized (within constraints) architectures for each model, I trained models of these architectures on their datasets across 30 trials in an environment with unconstrained randomness. I kept track of the model training, validation, and test accuracies for each iteration, as well as a running confusion matrix across all trials. I plotted the distribution of each model's train, validation, and test accuracies on a shared x-axis with a vertical line representing the sample mean, as seen below.


CNN Model, MFCCs, No Patient

Train Accuracies

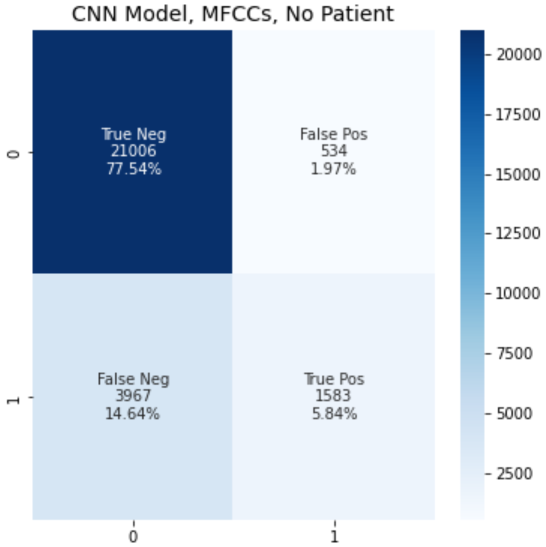
Validation Accuracies

Test Accuracies

A printout of model precision, recall, and F1 score was generated alongside the confusion matrices.

```
CNN, MFCCs, No Pt
----------------------------------
Precision: 0.747756258856872
Recall: 0.285225225225225

F1 Score: 0.4129385678883526
```



I also performed an independent t-test on the test accuracies of models with the same base architecture and training information, differing only in inclusion of patient information.

5 Results

5.1 Model Performance

In the analysis of my models, the best performing model was the CNN model trained on MFCCs, without patient information. It exhibited generalizability, scoring better on unseen data than on the training data. Mean training, validation, and test accuracies were 82.52%, 82.10% and 83.39% respectively. This model achieved a precision & recall of 0.74 and 0.29 respectively, showing a preference for the negative class. The F1 score of this model slightly edged out the CNN model trained on MFCCs with patient information; the former scored 0.4129 and the latter 0.4064.

5.2 Patient Information T-Tests

My null hypothesis for the T-Test was as follows:

$$H_0: \mu_{test\ acc\ without\ patient\ info} = \mu_{test\ acc\ with\ patient\ info}$$

I was able to reject the null hypothesis (p = 8.4 e -13) for the sequential 1k signal. However, the inclusion of patient information thoroughly skewed the output. Models trained on patient data had an abysmal aggregate F1 score (0.002) due to classifying nearly all samples as negative. Interestingly, this behavior was not observed in the model without patient information.

I was also able to reject the null hypothesis (p = 3.1 e -9) for the sequential MFCC models. The inclusion of patient information greatly increased randomness in model output and decreased mean test accuracy. However, it did increase model recall, which is a desirable attribute in a model designed for initial screening.

Disappointingly, I failed to reject the null hypothesis (p=0.092) in the case of my best performing models, the CNN MFCC models.

6 Conclusions

6.1 Further Considerations

Firstly, I would revisit the LSTM modelling problem. I'm particularly curious about the comparison between 4k sampling rate data vs 1k sampling rate data in the LSTM architecture. Since LSTM architectures perform best on sequences of 250-500 time steps (Brownlee, PhD, 2017), I'm inclined to believe that the 1k sampling rate would yield superior performance since the sounds we're 'listening' for are on the order of 0.25 seconds and longer. Once the LSTM data is gathered, I would pursue the best model and try and optimize its performance for the problem.

Additionally, I would redesign the model trialing experiment to track precision, recall, and F1 scores at the trial level, rather than in aggregate. This would provide more data to create distributions and allow me to perform more statistical analysis on model performance.

6.2 Final Thoughts

Over the course of this exercise, the goals and scope of the project ballooned as my understanding of the problem space deepened. When I began this project, I intended to fit a 22 kHz signal to simple statistical models and evaluate the output. This is clearly not what was produced. I am admittedly disappointed that my intuition that patient information would improve model performance was not soundly validated. However, I am confident the practical value in learning about its use in phonocardiogram classification exceeds that of which I initially set out to do.

Works Cited

(2022, Mar 15). Retrieved from Nyquist–Shannon sampling theorem: https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

Brownlee, PhD, J. (2017, Jun 26). *Techniques to Handle Very Long Sequences with LSTMs*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/handle-long-sequences-long-short-term-memory-recurrent-neural-networks/

Chen, W., Sun, Q., Chen, X., Xie, G., Wu, H., & Xu, C. (2021). Deep Learning Methods for Heart Sounds Classification: A Systematic Review. *Entropy*.

Doshi, A. R. (2018, Dec 5). Innocent Heart Murmur. *Cureus*.

Harvard Health Publishing. (2022, Jan 12). *Heart Murmur A to Z*. Retrieved from https://www.health.harvard.edu/a_to_z/heart-murmur-a-to-z

McGee, MD, S. (2018). Chapter 39 - Auscultation of the Heart: General Principles. In S. M. MD, *Evidence-Based Physical Diagnosis (Fourth Edition)* (p. 327). Elsevier.

Reyna, M. A., Elola, A., Oliveira, J., Renna, F., Gu, A., Sadr, N., . . . Clifford, G. D. (n.d.). *Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022*. Retrieved from physionet.org: https://moody-challenge.physionet.org/2022/

The CirCor DigiScope Dataset: Oliveira, J., Renna, F., Costa, P. D., Nogueira, M., Oliveira, C., Ferreira, C., … & Coimbra, M. T. (2022). The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification. IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2021.3137048.

The standard citation for the PhysioNet resource: Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., … & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215-e220.