Extracting Baby's Heartbeat Signal From Mother's ECG

Serban-Alexandru Tonie (S4721586), Aras Aniulis (S4694996), Amr Abdou (S4678753), Alexandru-Mihai Chirita (S4740661)

December 2024

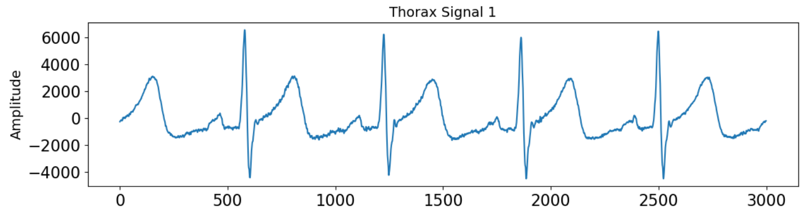
This project focuses on the challenging task of isolating and identifying a fetal heartbeat signal (ECG) from the dominant maternal ECG signal in recordings from a pregnant mother's thorax and abdomen. The method leverages noise cancellation techniques, where the mother's ECG signal is predicted and subtracted from the abdominal channel with the most prominent fetal ECG trace. A high-pass filter is applied as a preprocessing step to eliminate low-frequency baseline wander in the raw signals. Linear regression is employed to model the mother's ECG signal using the thorax channels as inputs. Post-processing steps, including signal smoothing and noise reduction, are performed to address distortions introduced during subtraction, ensuring the extracted signal is realistic and clean. The final fetal ECG signal is evaluated by confirming its pulse rate, which is expected to be approximately twice that of the mother's heartbeat, and its overall consistency.

1 Introduction

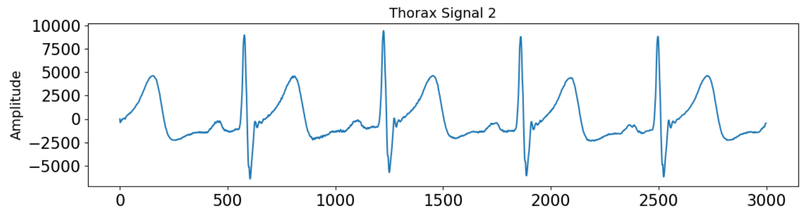
Clinical efforts have been invested in the detection of heartbeats of unborn children. The signals of a child's electrocardiogram (ECG) are overshadowed by their mother's because of the strength of the mother's signals. This makes the task of extracting a child's ECG signals from signals that contain both mother and child's ECG signals an increasingly complex task. This project aims to explore an approach to the clinically relevant task of extracting a child's heartbeat from a mother's in a signal that contains both heartbeats. As many techniques and data sets were discussed in another paper[7], we plan to use a lightweight data set, originally found here[1]. Success was defined by the ability to filter a signal that exhibited pulses corresponding to a fetal heart rate roughly twice that of the mother's heartbeat, reflecting a physiologically plausible result.

2 Data

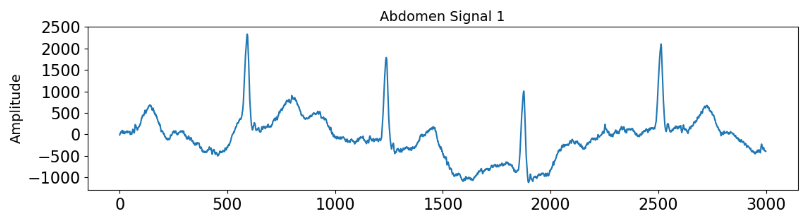
To achieve our objective, we use a dataset presented by Lee Cheng [1]. This lightweight data set consists of 5 raw ECG channels "thorax1" "thorax2", " abdomen1", "abdomen2", "abdomen3", each having 20,000 datapoints sampled at 1000Hz. These channels were recorded on a mother to be able to obtain (after pre-processing) a clean signal from the thorax channels. This clean signal can then be used to obtain the signal of the child's heartbeat. Figure 1 shows a segment of 3000 steps of these 5 channels.



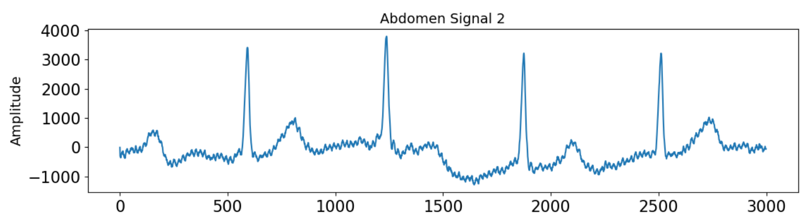
(a) Raw Thorax Signal 1



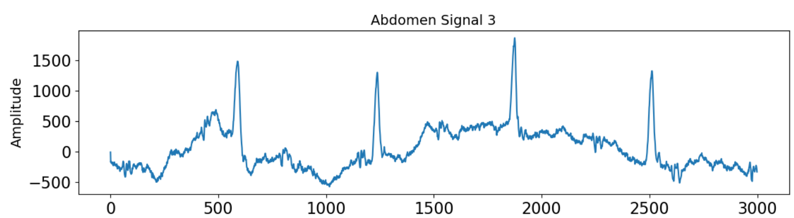
(b) Raw Thorax Signal 2



(c) Raw Abdomen Signal 1



(d) Raw Abdomen Signal 2



(e) Raw Abdomen Signal 3

Figure 1: Unprocessed signals

3 Methods and Experiments

This section discusses our approach to solving the child & mother ECG problem, with a focus on our process of finding the best fit techniques for each step.

3.1 Strategy

Our initial idea was very similar to the *classical* way of solving this problem proposed by the concept of noise canceling; we would aim to predict the mother's ECG footprint and subtract it from data where the child ECG is more pronounced. Figure 2 illustrates the general approach we decided upon to isolate the child's ECG footprint $\hat{s}$. Our input signal $s + v_0$ would be the ECG channel with the most pronounced child signal; abdomen 3. v_0 would be the sum of any noise present and the mother's ECG. Signal y would ideally be equivalent to v_0 and thereby cancel out every signal except the child's ECG.

This initial plan made it clear that our model would serve the same purpose as the *denoising filter* in Figure 2. It also made it clear that to receive any potential stand-in for noise and the mother's ECG, we would need an input channel which would isolate both of those signals. Our initial observations suggested that the thorax channels best isolated them both.

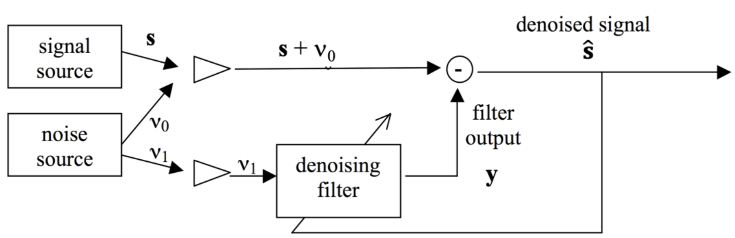


Figure 2: This figure shows a de-noising filter commonly used in signal processing. Our approach takes inspiration from this filter. This figure was taken from [2].

3.2 Pre-Processing

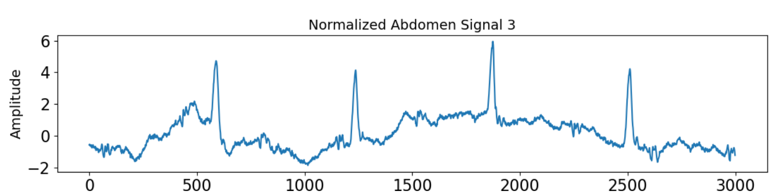


Figure 3: Normalized Abdomen Signal 3 to unit variance and zero mean

Signals often have a DC offset, which is a constant baseline value added to the signal. For example, in ECG signals, this offset could come from sensor drift or electrode imbalance. Standardizing to unit variance also ensures equal weighting, such that when we are subtracting signals from one another, one does not dominate the other simply based on its larger amplitude. Figure 3 how normalizing affects the amplitude of the signal, when comparing it to its raw variant in Figure 1e.

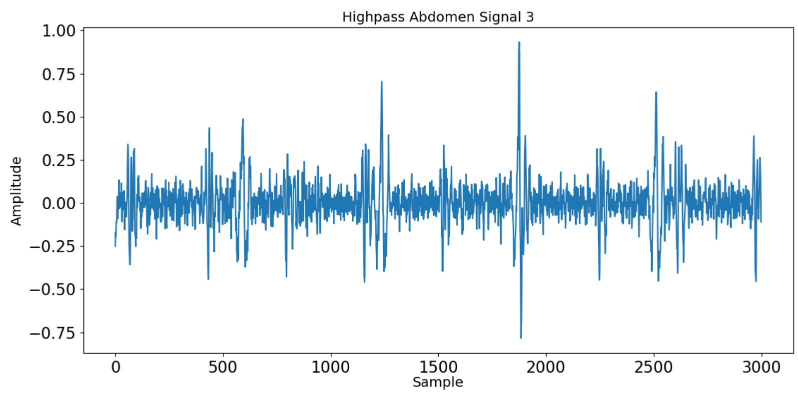


Figure 4: Highpass filtered Abdomen Signal 3

The next step in our pre-processing is removing the baseline wandering from the signal. Baseline wandering occurs in ECG signals when the isoelectric line moves up and down, not being stable. This instability may be caused by a multitude of factors such as respiration, body movement and sweating [3]. In the abdomen signals (Figures 1c, 1d, 1e), the wandering can be clearly observed. We must remove this feature from the signal because it can lead to misinterpretations; the model might "learn" certain exaggerated aspects of the signal or overlook other essential ones. The highpass filter helps remove this baseline wandering and keep the actual ECG signal intact, by blocking all frequencies below a certain set threshold. In our experiment, we decided to use different parameters for the highpass filters for the thorax and abdomen channels. Since the abdomen has considerably more baseline wandering, we decided to use a cutoff of 30, and for the thorax channels, a cutoff of 0.5 as they do not present such large wandering. It is important to note here that, in order to extract the filter coefficients, we normalized this cutoff frequency by dividing it with half the sampling frequency. This value of 30 was chosen by simply trying out multiple values and observing the results. In the end we kept this value because it eliminated enough low frequency from the signal where one can still pinpoint the mother's heartbeat (the larger spikes) and also the child's heartbeat (the smaller spikes). Figure 4 shows a portion of the abdomen 3 signal after the highpass filter has been applied to it. We have also padded the signal before filtering such that the start and end do not get distorted by the filter. We chose the value of 15 for the pad length because, after trying out multiple values, this seemed sufficient such that our signal ends looked good. As for the high pass filtering, we rely on the Butterworth filter. The Butterworth filter is a type of signal processing filter designed to allow a certain frequency range to pass with minimal distortion. We rely on SciPy's Butterworth implementation [5], where it takes as parameters among others the cutoff frequency in Hz, the order of the filter, which determines the speed of attenuation of frequencies at the cost of computational complexity and possible phase distortions, the type of filter, which could be high, low or bandpass, as well as whether the signal is digital and continuous or analog and discrete. For our use case, we relied on the parameters above, which where set as follows:

- **cutoff**: As described above, we use 2 different cutoff frequency values, one value for the thorax channels and one for the to ensure the baseline wander has been removed.
- **order=4**: Through empirical observations, and we found that an order of 4 perfectly balances the attenuation speed and the computational cost, with minimal phase shift and complexity.
- **btype='high'**: This is the setting for high pass filters, which we needed.
- **analog='False'**: This is the setting for discrete signals, which is in essence how our signal has been obtained (sampled at a rate of 1000Hz per second)

Giving those parameters to Butterworth, we obtain the numerator and denominator polynomials for the actual filter, **a** and **b**. We then apply the actual filter, relying on SciPy's **filtfilt** function [6], where it takes as parameters among others **a** and **b**, which are the numerator and denominator polynomials of the filter, and the padded signal. We use **filtfilt** due to its application of the filter twice, once backwards, and once forwards. This type of filtering results in what is called *zero-phase filtering*, meaning the signal is not shifted in time after being filtered.

After filtering, we remove the padding, keeping the length of the signal the same as before.

3.3 Model Selection & Training

This section discusses our choice for the *denoising filter* as show in figure 2. We chose to explore the predictions using linear regression as the relationship between our input and output channels are likely to be linear.

3.3.1 Linear Regression

It is important to mention that linear regression models are most effective when there is a *linear* relationship between the input and output channels. In the case of the mother and child ECG, this seems like a suitable assessment to make: we assume that the ambient noise from the mother's body is near constant and the mother's heartbeat remains consistent. The only difference predicting the the mother's ECG in abdomen 1 and abdomen 2, should be the noise, and perhaps the amplitude of the ECG. This implies a linear relation ship, as there is no massive transformation happening by placing the ECG measurement devices in another area. For this reason, we assumed that a linear regression model would be suitable to identify this kind of relationship.

Equation 1 shows the prediction by a linear regression model is simply the product of the weight vector w and the value(s) of the input vector $\hat{x}$ plus a constant. This relatively simple method serves as a good foundation to show the linear relationship between the thorax and abdomen channels.

$$y = w^T \hat{x} + b \tag{1}$$

A key feature of our input data is that it is ordered by time; it is a time-series dataset. This introduced a second level of complexity when choosing the model. Our initial assumption was the linear regression is not specifically intended to be used for time-series data predictions, nonetheless after some assessment, we concluded that there is no reason which makes it *unsuitable* for it. By introducing lagged inputs for the input channel, we would be able to provide some sort of temporal ordering to the input channel and ensure that each predicted value is a result of multiple points in the past.

3.3.2 Parameters & Input

It is important to mention that linear regression, due to its basic nature does not have many hyper-parameters. Furthermore, our usage of the sklearn implementation of it did not introduce many complexities to the underlying algorithm [4].

The only parameter of note is related to the variable b as seen in Equation 1. The *fit_interceptbool* parameter determines whether a value for the b variable needs to be calculated or not. In the case of our implementation and problem solution, we decided that it was necessary to maintain this value set to true (which coincides with the default sklearn model behavior). Since our normalization and other pre-processing cannot guarantee absence of all noise and baseline drift, it was decided that there is no downside to predicting a b term for the model.

3.3.3 Training

Training the linear regression model is rather simple, it is only dependent on the input data X and the expected output value y and several a few calculations as shown by Equation 2.

$$w = (X^T X)^{-1} X y \tag{2}$$

The input for the model training included both of the thorax channels each of feature dimension $\in \mathbb{R}^1$ and length 20000 samples. To ensure that our linear model was able to utilize the temporal ordering of the data in its predictions we *lagged* the input by a factor of 600. This increased the input X size to 20000 samples of 600 features each.

The reason we chose a lag of 600 is because we found it to be the best window to capture the key features of the ECG. We observed that roughly every 600 time steps a new heart beat would be shown. This ensured that our model would have enough context to learn the whole causal relationship encapsulated in our data. Having larger lags would result in our model being fed more data than needed to make its prediction, it is fair to assume that data only of the most recent heartbeat would need to be a part of this window, instead of that which came 30 heart beats ago. In contrast, having a window is too small would result in our model not having enough context to properly estimate the relationship between our input and output.

The next step was to stack the two thorax channels resulting in the increased input dimensions of (20000, 1200). This meant that we have 20000 entries of feature size $\mathbb{R}^{1200}$ all combined into 1 input for the model with 20000 rows and 1200 columns. This has the advantage of allowing our model to estimate the heart beat without using only 1 input channels as a source and with a added temporal dependence.

Finally, our target vector y is composed of the abdomen 1 and abdomen 2 channels, slightly differently from the input data. Each channel had a dimensions of (20000, 1) (which already fits the dimension criteria required by Equation 2), but to ensure that our model was not trained specifically for good performance on only one of the channels, we decided to generalize the performance by taking the mean of the sum of both of the target channels pairwise. This meant that the dimensions remained the same, however the target data was now more suitable for our model.

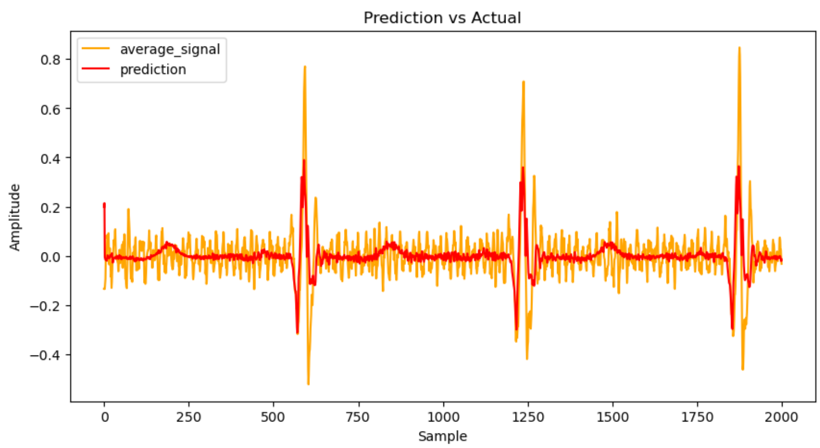
We can then see that as shown in Figure 1 the dimension of $w \in \mathbb{R}^{1200\times 1}$, and the dimension of $\hat{x} \in \mathbb{R}^{20000\times 1200}$ resulting in $y \in \mathbb{R}^{20000\times 1}$.

3.4 Post-Processing

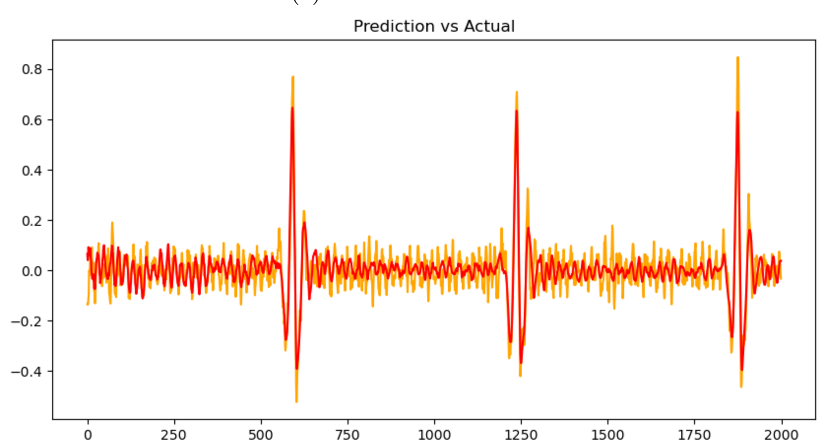
Since our prediction represents the mother's heartbeat, the first step in the post-processing would require to subtract this prediction from one of the signals that contain the child's heartbeat too, in our case abdomen 3. The result of this subtraction can be seen in Figure 6. One can see the small spikes in the signal that correspond to the child's heartbeat. Figure 7 shows the child signal alone. Now, in order to define even more clear the signal, we can get rid of the negative amplitudes by just squaring the signal, as can be seen in Figure 8. And finally, the signal can be smoothed using a moving average filter. For this filter, we used a window of 50 samples. This value was chosen in a similar matter as the cutoff value mentioned in the pre-processing. We experimented with numerous values, observed the outcomes, and ultimately chose the one that produced a signal aligning with our preferences.

4 Results

Figures 5 presents the results for the 2 iterations of the model. The performance of these two models is shown in Table 1, with the large window size performing the best. Fig. 7 represents the unfiltered child signal. Squaring the child signal gets us Fig. 8. After moving the average signal, the filtered child signal looks like Fig. 9



(a) Small window of 3.



(b) Large window of 600.

Figure 5: Prediction of the Abdomen signal compared to the signal from abdomen 2

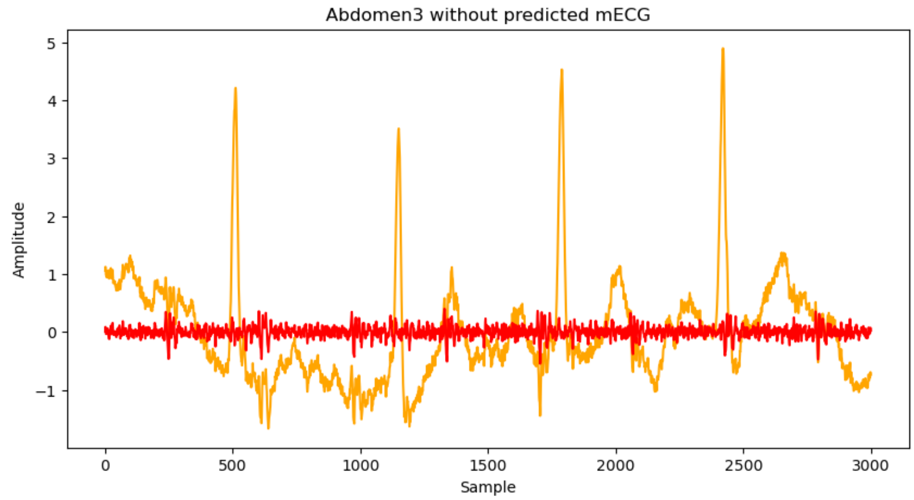


Figure 6: Abdomen 3 signal after subtracting prediction

Iteration	Mean squared error
Small window size of 3	0.0154
Large window size of 600	0.0093

Table 1: Performance of the different iterations

5 Discussion

Our experimentation yielded interesting results, however most notably we were able to successfully extract the child ECG signal through the use of the mother's abdomen and thorax channels. Firstly, our attempts at using linear regression to predict the mother's ECG were successful in several ways; figure 5 shows that even with a small lag size of 3, our model is able to learn a substantial amount of the mother's ECG features. This is further highlighted by figure 1 which shows that our initial linear regression attempts had an MSE of 0.0154, which is already rather good. Nonetheless, the reason we did not proceed with the initial model is due to its inability to capture the features of the true ambient noise. The second subfigure in figure 5 shows the prediction increasing the lags to 600. This substantially improved the models ability to fit the mother's ECG signal, as well as match the noise to a higher degree then before. This result is shown in figure 6 where we subtract the predicted signal from the abdomen 3 channel. This leaves us with a stationary signal for the child's ECG.

After obtaining this initial signal we can already identify what we perceive to be the child's ECG signal, beating at intervals roughly half the size of the mothers. This is best shown by figure 7 where we can clearly identify the child's ECG peaks. However, the issue of the noise was still present, furthermore the

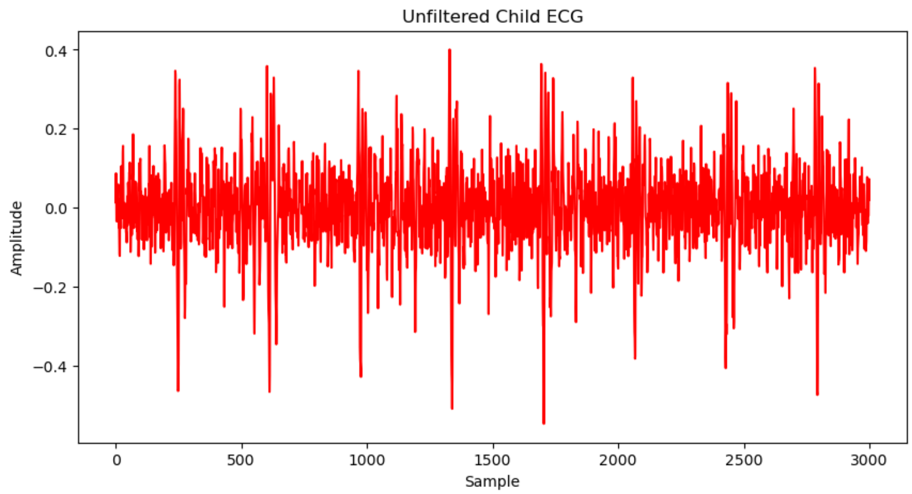


Figure 7: Unfiltered child extracted signal

shape associated with ECGs was lost. To reintroduce the shape of the ECG we chose to square each value in the result. This reintroduced the peaks only in the positive y axis direction allowing us to get closer to the child's ECG.

To remove the noise we chose to use a moving average filter. This had the desired result of reintroducing the shape associated with the ECG. On another note, moving average filtering can sometimes result in phase delay and removal of high frequency information. Nonetheless, this was less relevant for our scenario as we specifically want to avoid high frequency noise. However, in other signal processing scenarios, it could be undesirable to introduce this kind of data loss. Figure 9 shows the extracted child's ECG with post-processing against the mothers thorax and abdomen channels.

Irregardless of our success in extracting a signal akin to that of the child's ECG, there are several concerns to highlight. Firstly, by increasing the lag size so dramatically, we risk overfitting to the mother ECG. This is a slight issue that is very difficult to quantify due to our lack of data and other input channels. Since our model is trained on both of the thorax channels, we essentially use up all of the data we could use for testing and/or validation. We would have to sacrifice substantial portions of our input to get a sense of whether we may be overfitting. Furthermore, our results may be skewed in our favor, because the model is only trained on 1 person's (and child's) ECG signals. If we had access to a larger dataset, there would likely be more varied noise and ECG footprints for different people. Thereby we cannot conclude that our model would suffice for the extract of the child's ECG for all pregnant women.

Moreover, we introduce 2 stages of heavy filtering to extract features which we can use to train the model, and then modify the output signal further to get the shape we desired. This could lead to information loss or mutation in ways

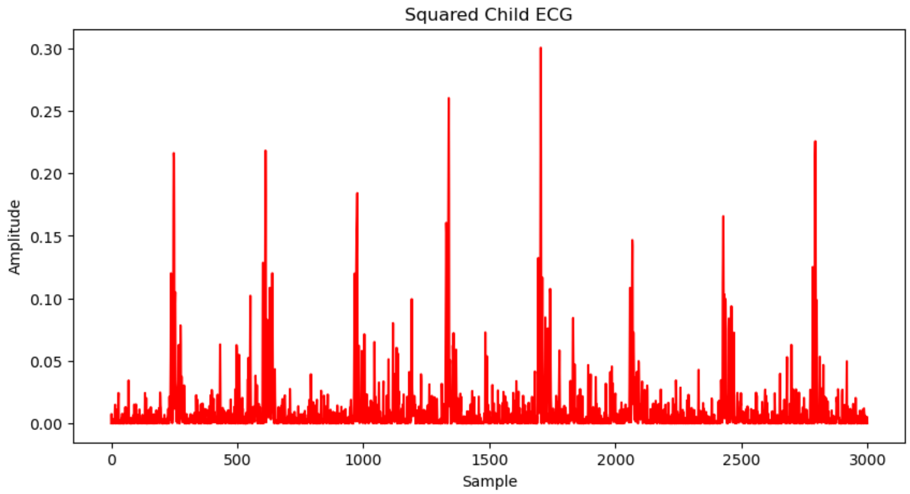
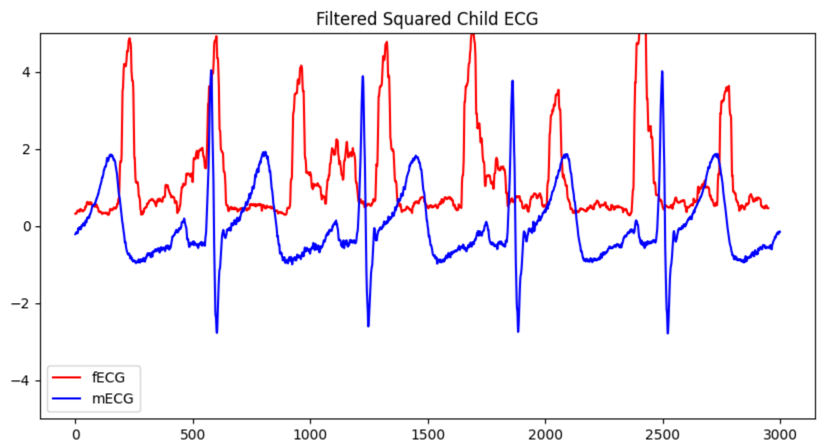
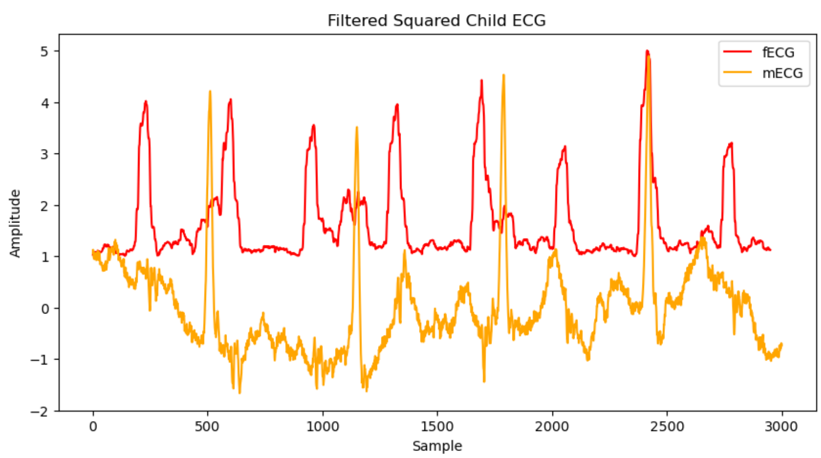


Figure 8: Squared child signal

which are not necessarily correct, rather they only appear to have a correct effect. Additionally, since we do not have any child ECG channels, there is no real way for us to determine whether our extracted signal is correct or accurate, rather we have to assume it is based off its features (mainly the frequency of the signal being twice as fast as the mothers). Nonetheless, given our assumptions, we conclude that we have successfully extracted the child's ECG signal.



(a) Child signal compared with mother thorax 1 signal



(b) Child signal compared with mother abdomen 3 signal

Figure 9: Filtered child signal

References

[1] Lee Cheng. Maternal-fetal ecg dataset, 2016. Originally available at: `http://www.masys.url.tw/AU/AU.htm` and `http://www.masys.url.tw/AU/2015SP/BMSD-D/HW/HWfinal-Maternal_Fetal_ECG/ProjectDescription.htm` (links no longer accessible).

[2] Herbert Jaeger. Lecture notes: Machine learning, 2024. Accessed: December 1, 2024.

[3] Morningdot. What is baseline wandering? https://www.crazyengineers.com/threads/what-is-baseline-wandering.37893.

[4] Scikit. Scikit-learn: Machine learning in python.

[5] SciPy. Butterworth filter documentation, 2024. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html Accessed: 2025-02-01.

[6] SciPy. filtfilt documentation, 2024. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt Accessed: 2025-02-01.

[7] Radana Vilimkova Kahankova, Radek Martinek, Rene Jaros, Khosrow Behbehani, Adam Matonia, Michal Jezewski, and Joachim Behar. A review of signal processing techniques for non-invasive fetal electrocardiography. *IEEE Reviews in Biomedical Engineering*, PP:1–1, 08 2019.