Analysis of Heart Rate Variability in to Predict Sepsis Preterm Neonates

Aidan O'Keeffe

UGS 320K: Undergraduate Research Experience

Dr. Joshua Chang

January 19, 2023

Abstract

Sepsis is a leading cause of death for preterm infants. Researchers have sought to develop methods for the early detection of sepsis using vital sign measurements. One area that has shown promise is heart rate variability. This project attempts to answer the question: Do heart rate variability metrics show signatures that indicate sepsis? This project's answer to the question is negative, but there is much room for improvement.

The intended purpose of this audience consists of two groups: Dr. Chang, his students, and his collaborators who may wish to understand my work, and potential employers in data science who want to see a demonstration of my skills.

Acknowledgements

I would like to thank Joshua Chang MD PhD for enabling this project with data and programming advice, and for providing me with career development pointers. I would also like to thank Alan Groves MD for providing research direction. Both of these individuals are associated with Dell Medical School at the University of Texas at Austin.

This work was completed for the course UGS 320K: Undergrad Research Experience at the University of Texas at Austin in the Fall of 2022.

Code

Python code in the form of Jupyter Notebooks for this project can be found at https://github.com/aidanlokeeffe/nicu_hrv_analysis. It is recommended to use Jupyter Notebooks in Visual Studio Code for the best experience.

Introduction

Sepsis is one of the leading causes of death for preterm infants. In part, this is because sepsis can progress from no symptoms to death in as little as 12 hours (Ring, 2018). The surest way to diagnose sepsis is with a *blood culture*, yet according to Dr. Groves, these tests require no less than 2 to 4 hours to produce results, in addition to expertise and laboratory resources. Preterm infants also do not have a large volume of blood because they are so small, so it is undesirable to draw blood from them. These hours can mean the difference between life and death, so it is common practice for an infant's attending physician to order antibiotic treatment as soon as sepsis is suspected. This practice is in the patient's best interest, but it also carries all of the costs associated with excess use of antibiotics.

This unsatisfactory situation has incentivized researchers to develop tools for the noninvasive early detection of sepsis in preterm infants. One set of vital sign measurements that have shown promise in this regard are metrics of *heart rate variability* (HRV): the beat to beat variation of heart rate. One line of reasoning for the possibile effectiveness of these measurements is that HRV is linked to the nervous system's control of heart rate, which in turn is related to the body's immune response to sepsis (Fairchild and Aschner, 2012).

The most popular tool currently in use for this application is the HeRO score; see Fairchild and Aschner for details (2012). Nonetheless, there remains room for improvement, hence the niche for this project and others (Masino et al, 2019).

The guiding question of this project was: Do heart rate variability metrics show signatures that indicate sepsis? The answer is negative, but still, there remains much room for improvement.

This article is laid out as follows: first, the programming tools and dataset are described. Then, a great deal of attention is given to the data processing pipeline; such a presentation is done with the intended audience in mind. Visualizations of the results are given, and a general discussion follows.

Materials

Programming Tools

This analysis was carried out in Python, making heavy use of the numpy, pandas, matplotlib, and waveform-database libraries.

Data

Table 1: Summary of Infants

Infant	*Became Septic*	*Gestational Age at Birth*	*Gestational Age at Start of Recording*
1	Yes	27w 4d	29w 1d
2	No	26w 0d	27w 4d
3	No	24w 5d	33w 4d
4	No	25w 1d	34w 0d
5	Yes	24w 6d	34w 5d
6	No	28w 0d	29w 4d
7	Yes	24w 0d	27w 5d

The dataset for this study consisted of vital sign recordings from 7 preterm infants. Infants 1, 5, and 7 developed sepsis, while infants 2, 3, 4, and 6 did not. Table 1 provides an overview of these 7 infants.

The septic infants were selected on the criteria that their incidents of sepsis were “very severe” and that they did not experience other major issues during the recording period. The non-septic infants were selected based on the criteria that they were close in age to one of the

septic infants and had “relatively uncomplicated” experiences during the recording period. The evaluations of “very severe” and “relatively uncomplicated” were made by Dr. Groves.

The original purpose for the collection of this data was to monitor the infants’ status during their stays at the Dell Medical School NICU. The data is now housed securely at Dell Medical School and can only be accessed by students, staff, and faculty thereof. Dr. Chang pulled the data, deidentified it to protect the infants’ privacy, and stored it on Box so it could be accessed for this project.

The data for each infant consisted of a csv with columns for time, anywhere from 1 to 3 *ECG* signals (sampled at 250 Hz), respiratory impedance (125 Hz), and SPO2 (62.5 Hz). The files were too large to be stored in the memory allocated to Python, so Dr. Chang split each of them into 5 files. This resulted in a total of 35 files, 5 for each of the 7 infants, each around 3.5 gigabytes in size. The files were named by infant number and part of the data; for example, the data for infant 1 was broken into files raw_waves_data_1a.csv, raw_waves_data_1b.csv, and so on up to raw_waves_data_1e.csv. For brevity, these files will be referred to as 1a, 1b, *et cetera*.

The original plan was to incorporate the respiratory data and SPO2, but that point in the project was not reached, so those columns were dropped when the data was loaded into Python.

Methods

Figure 1: Project Pipeline

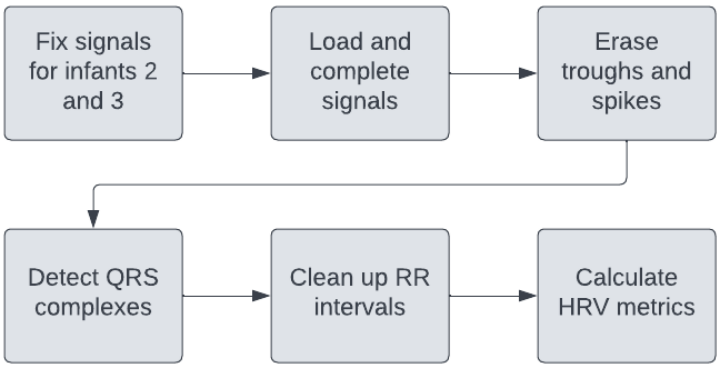
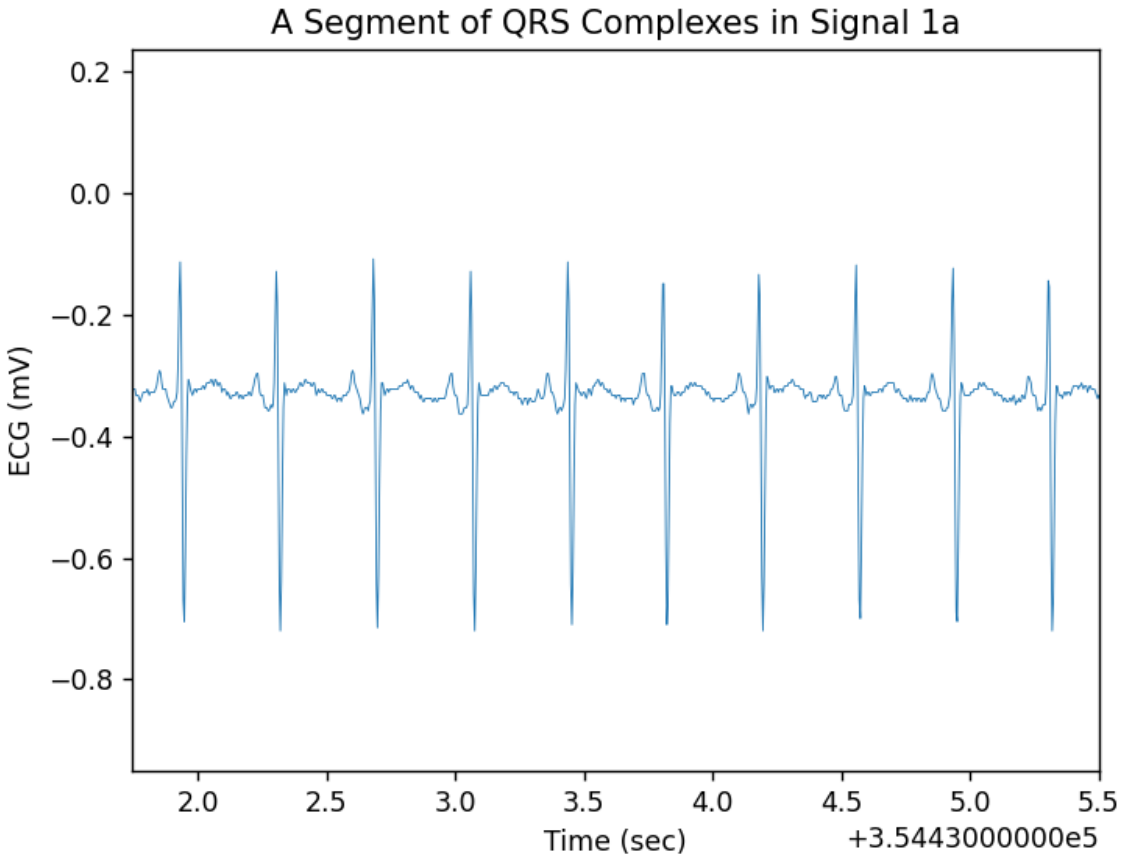


Figure 1 is a flow chart providing a high-level view of the data pipeline in this project.

Figure 2: QRS complexes

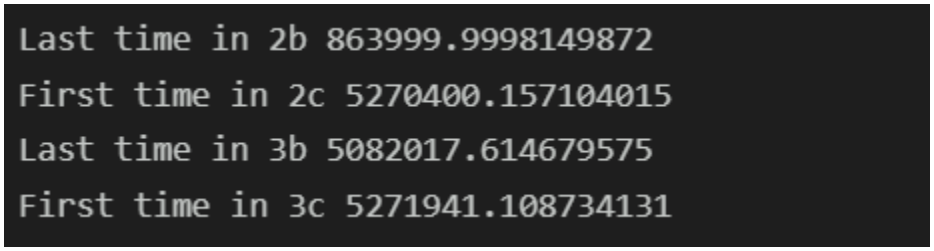


In order to calculate HRV metrics, one first needs a clean, complete ECG signal. Disturbances to the signal can obscure *QRS complexes*, the waveform of a heartbeat in an ECG

as illustrated in Figure 2. From the ECG, an algorithm called a *QRS detector* can label the location of each QRS complex, called the *R peak.* From the R peaks, *RR intervals*—the amount of time between each R peak—can be calculated. *HRV metrics*—statistics measuring the variation in the duration of the RR intervals—can then be calculated.

Step 1: Fix signals for infants 2 and 3

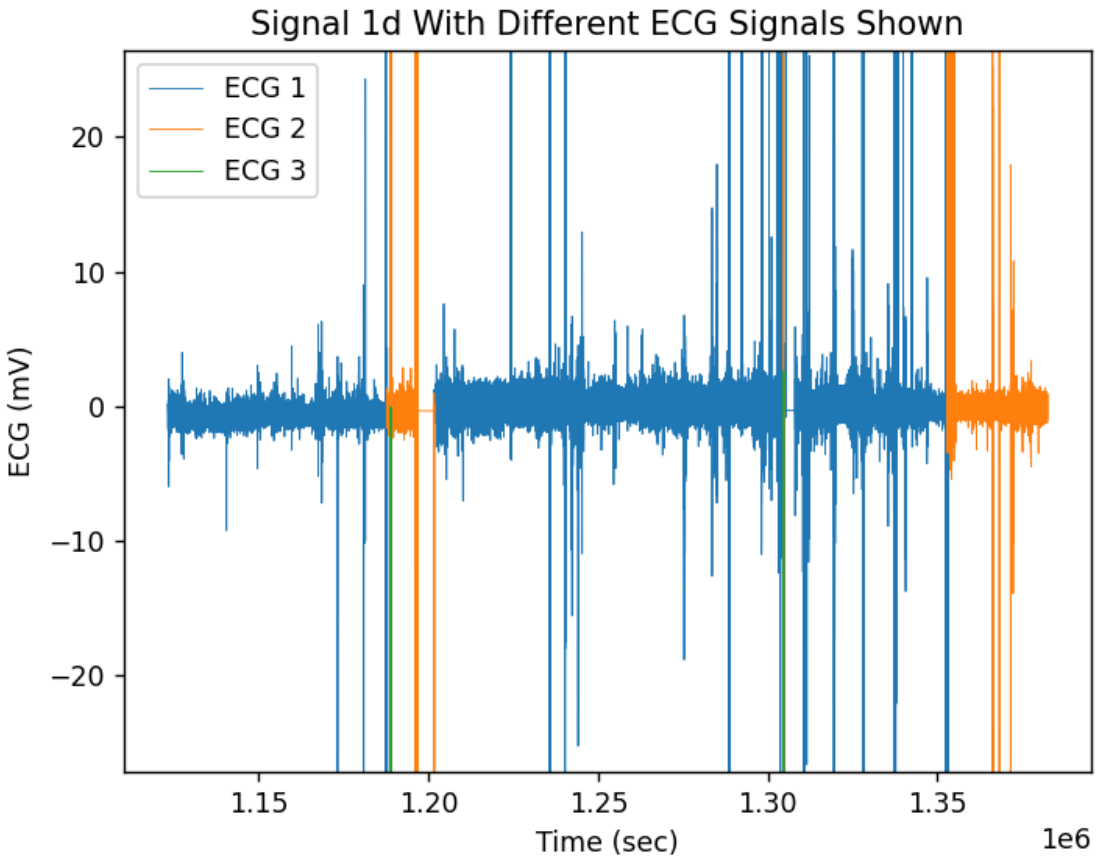
Figure 3: Gaps in the Time Index for Infants 2 and 3



While inspecting the data, it was noticed that the time index for file 2b was discontinuous with that of 2c, and likewise for 3b and 3c. To resolve this, the time index in files 2a and 2b were shifted forward in time to line up with 2c, and likewise for 3a and 3b to 3c.

Step 2: Load and complete signals

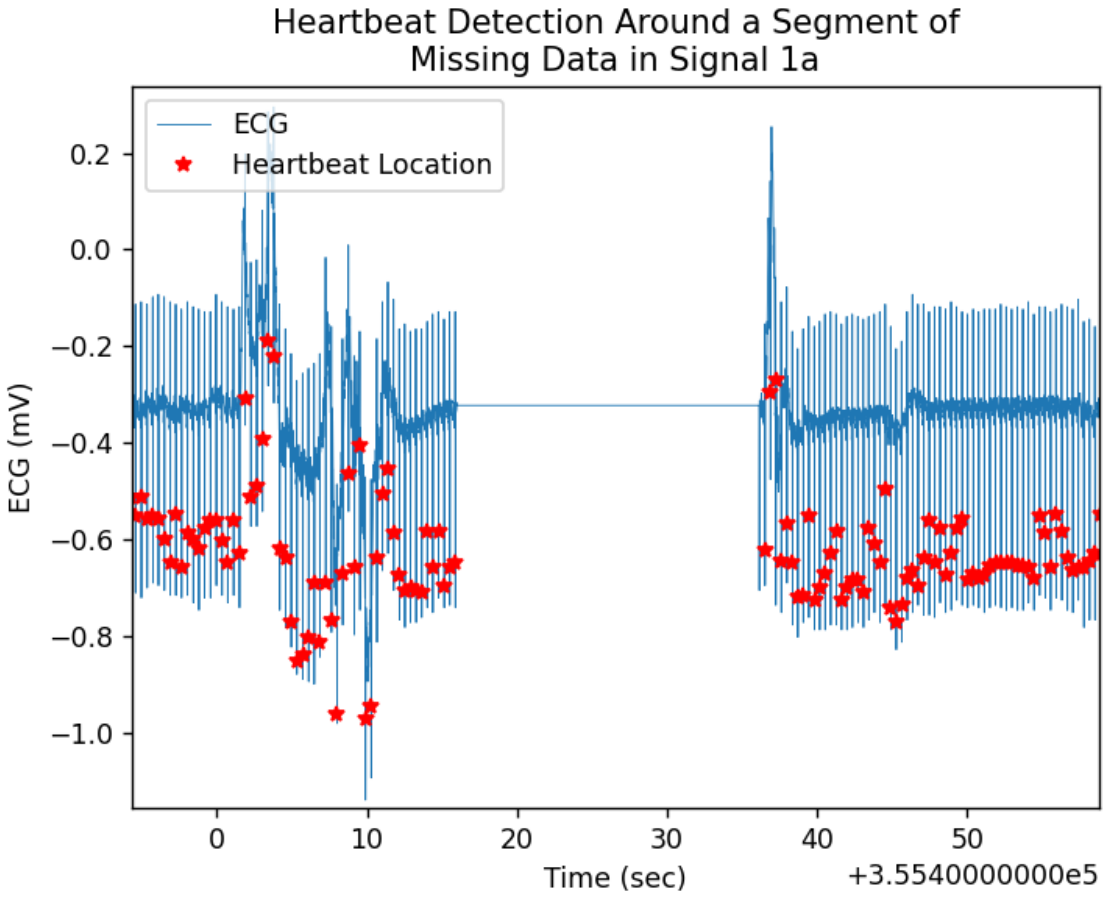
Figure 4: The different ECG signals



Each file could have columns for anywhere from 1 to 3 ECG signals, termed ECG 1, ECG 2, and ECG 3. This is because the bedside monitor requires 3 electrodes to measure the ECG. However, for almost all times, only one ECG signal is actually saved, as can be seen in Figure 4. Since all of the ECG signals range between the same values (also visible in Figure 4), it would be redundant to record all 3 of them at once. Causes for the saved signal to switch are varied.

To form one complete ECG signal, ECG 1 was taken first, and then ECG 2 and ECG 3, in that order, were used to fill in as much missing data as possible.

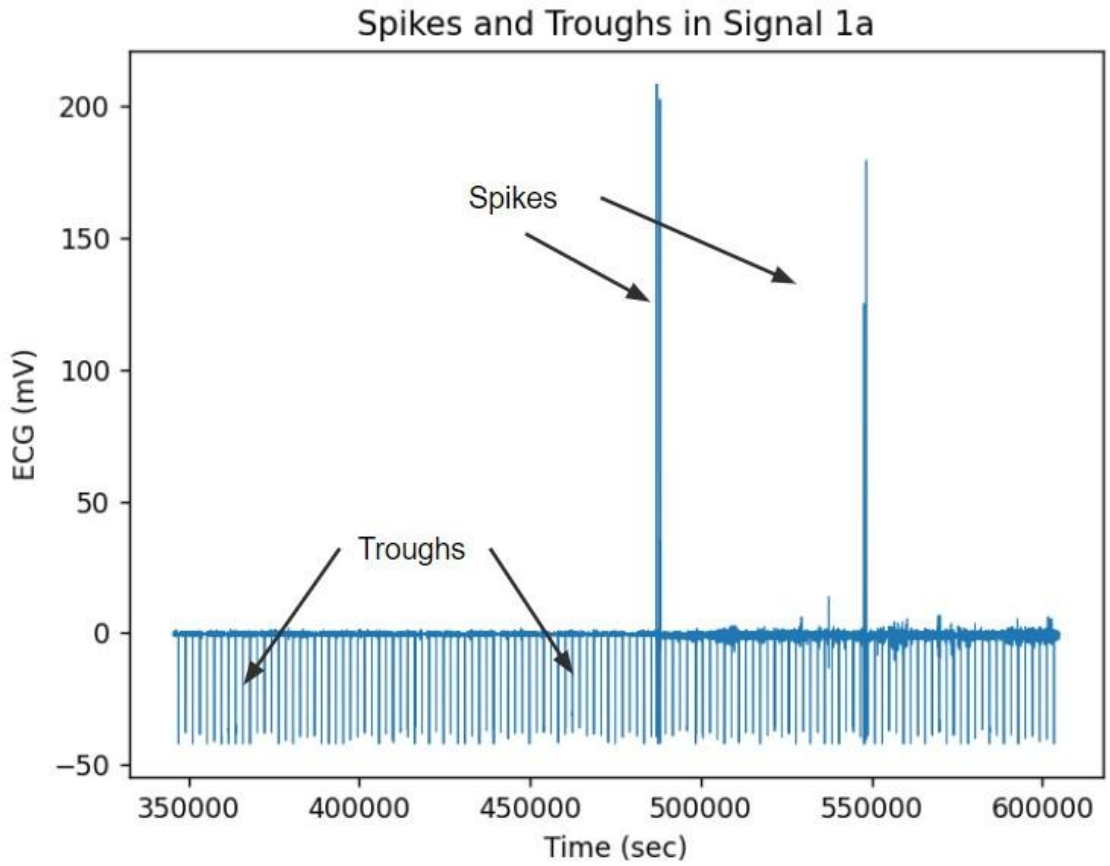
Figure 5: Heartbeat detection around a segment of missing data



After combining the signals, there was often still some missing data. It had to be filled in somehow because the QRS detector would not work on an incomplete signal. It was decided to use fill-forward—where the most recent data point is written over missing data—because the QRS detector could handle these flat segments of data with minimal loss of R peaks, as illustrated in Figure 5.

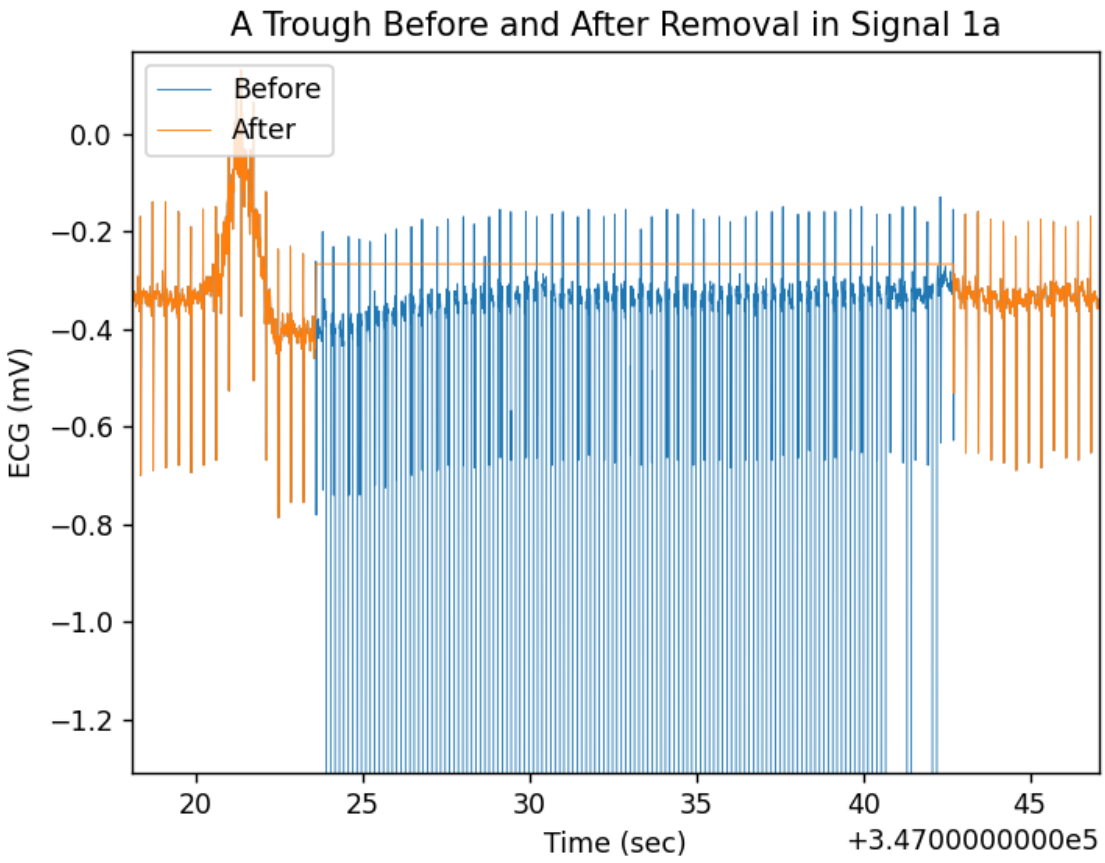
Step 3: Erase troughs and spikes

Figure 6: A signal with troughs and spikes



After the signal was completed, it needed to be cleaned. There were 2 kinds of large deviations in these ECG signals. The first were spikes, sporadically distributed deviations which could be positive or negative. The second were troughs, periodically distributed deviations in the negative direction. Both kinds of deviations are shown in Figure 6. Both kinds of deviations were found to trip up the QRS detector, so they had to be removed.

Figure 7: Zoomed in view of a trough before and after erasing



The blue curve in Figure 7 shows a trough. Observe that a trough is actually a rapid succession of negative spikes with a duration of about 15 seconds. This duration was a problem because it was long enough to obscure some heartbeats. We had originally hoped to salvage some of the obscured beats, but we found that detecting the still-visible QRS complexes in a trough artificially inflated the RR intervals; therefore, we decided to erase troughs entirely. Spikes were erased too, which was acceptable because they were often short in duration. Once the deviations were erased, fill-forward was used to re-complete the signal; the orange curve in Figure 7 shows the result.

Step 4: Detect QRS complexes

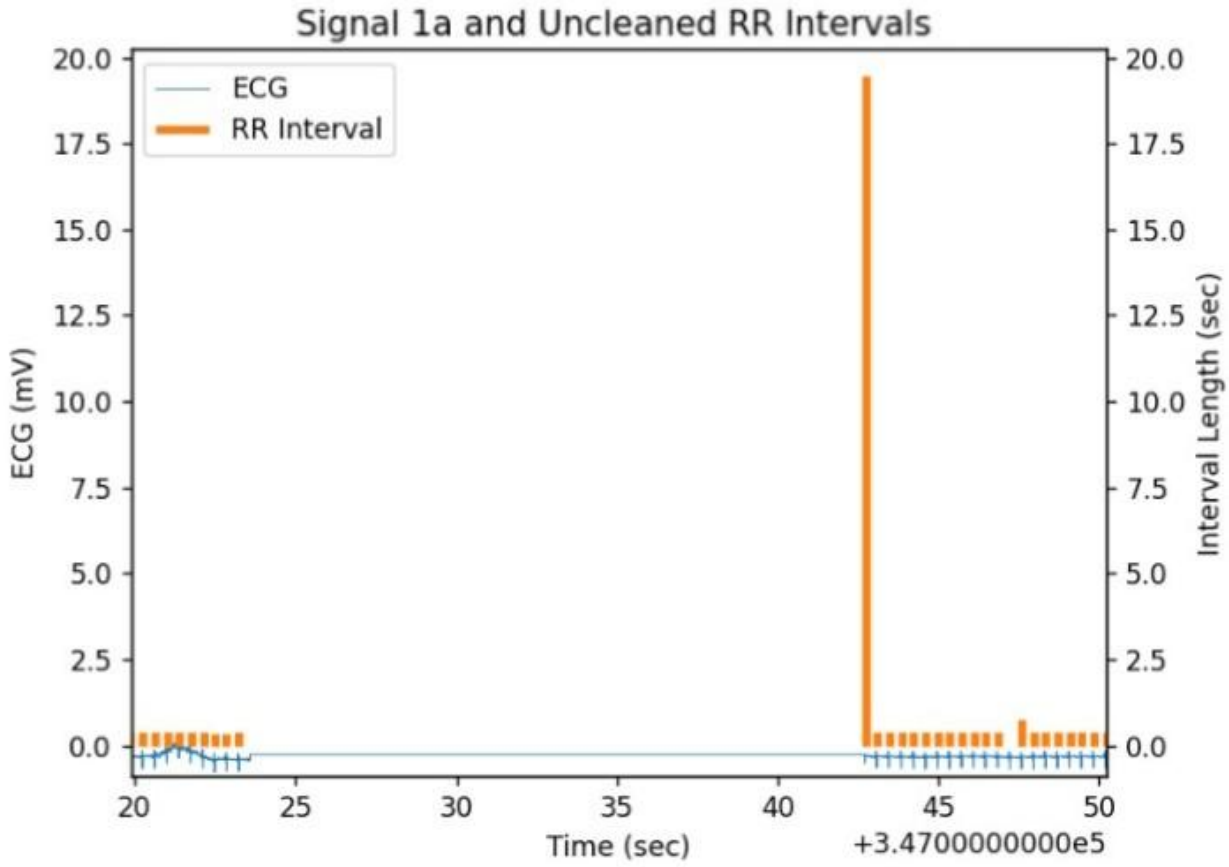
With the amount of data involved in this project, annotating QRS complexes by hand would be an impossible task, hence a QRS detector was required.

There are many QRS detectors available. For this project, we decided to use XQRS from the waveform-database library. There were two reasons for this. The first was that one study found that XQRS places QRS complexes accurately in time while also keeping other types of error low (Eilers, Chromik, and Arnrich, 2021). The second was that XQRS is readily available in Python.

It was found that XQRS was unacceptably slow if it was fed the whole ECG all at once, but fast enough if it was fed the signal in chunks. Thus, the signal from each file was broken into chunks of 10,000 measurements in time, QRS complexes were detected in each of these, and then the results were combined and stored.

Step 5: Clean up RR intervals

Figure 8: Examples of RR intervals that need to be cleaned



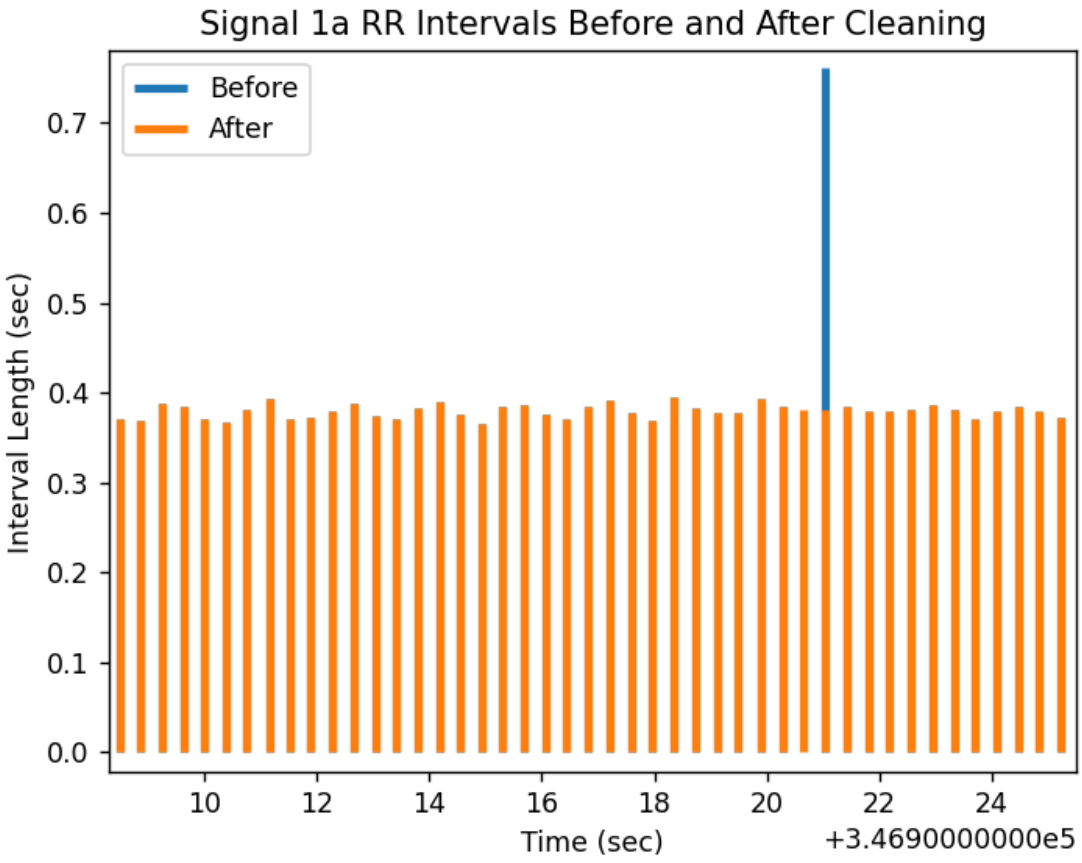
In Figure 8, the ECG signal has been plotted in blue, and the RR intervals have been plotted in orange. The RR intervals are visualized as vertical bars. The height of each bar is the amount of time that has passed since the previous beat, and the location of the bar on the x-axis corresponds to the current beat.

Now, note that there are some RR intervals that are unrealistically short or unrealistically long. Figure 8 shows long intervals (but not short ones), and also illustrates long intervals may be due to either missing data or undetected beats. Here, long RR intervals due to missing data are termed missing intervals, and long RR intervals due to undetected beats are termed multiple intervals.

Dr. Groves provided a physiological lower bound of 0.25 second for an RR interval, so the RR intervals were recalculated with beats corresponding to these short intervals ignored.

Cleaning long RR intervals was not a simple because it is difficult to distinguish between missing intervals and multiple intervals; sometimes, many beats in a row may be undetected, while other times, a segment of missing data may be quite short. We believe that developing an algorithm to distinguish these cases could be an interesting research project in its own right. For this project, a suboptimal solution was implemented. First, any RR interval longer than 5 seconds was assumed to be a missing interval and thrown out. Any multiple interval less than 5 seconds was assumed to be a missing interval and broken into a number of equally sized pieces. The results of this process are shown in Figure 9. Note that the orange and blue intervals correspond almost exactly, except for where there is a multiple interval.

Figure 9: RR intervals before and after multiple interval reduction



Step 6: Calculate heart rate variability metrics

After all of the previous steps, the data was finally in a form from which HRV metrics could be calculated. The mean RR interval, *meanRR*, was calculated; this is not a heart rate variability metric, but it was so easy to compute that it was included. Dr. Groves was interested

in seeing four heart rate variability metrics: the standard deviation of the RR intervals, the root mean square of successive differences, the probability ot two consecutive RR intervals differing by more than 50 ms, and the probability of two consecutive RR intervals differing by more than 20 ms, abbreviated as *SDRR*, *RMSSD*, *pNN50*, and *pNN20*, respectively. Detailed discussions of these metrics' definitions and their relation to the nervous system can be found in work by Shaffer and Ginsberg (2017).

These statistics were calculated in a rolling window of 12 hours in length, so that for each of the plots below, for each point in time, the value of the metric at that time point is the metric calculated over the previous 12 hours.

Results

Figures 11 through 15 show the values of the HRV metrics for the 7 infants. Recall from Table 1 that infants 1, 5, and 7 developed sepsis, whereas infants 2, 3, 4, and 6 did not. The vertical red line denotes the time of sepsis for the septic infants, and the center of the recording period for the other infants. Plots zoomed in to this region have not been produced, but inspecting the following plots closely reveals that **there are no visual indicators in the leadup to sepsis**.

Figure 10: Mean RR

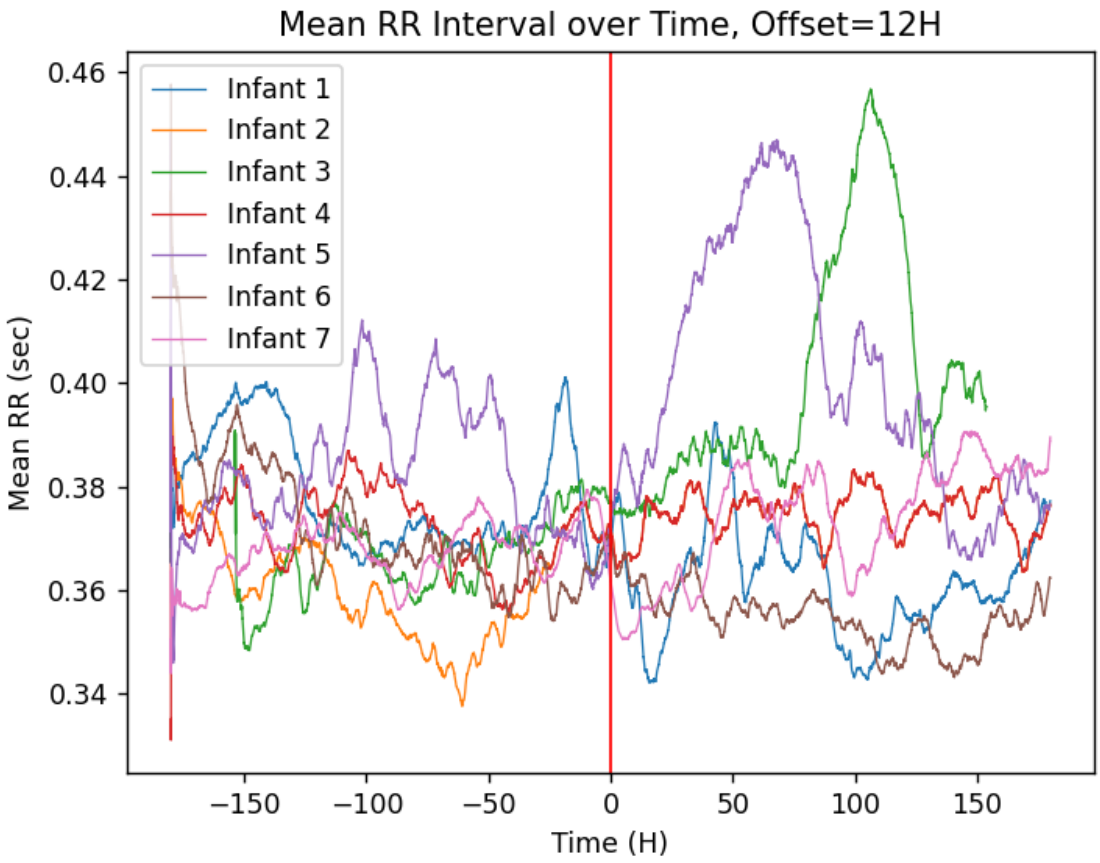


Figure 11: SDRR

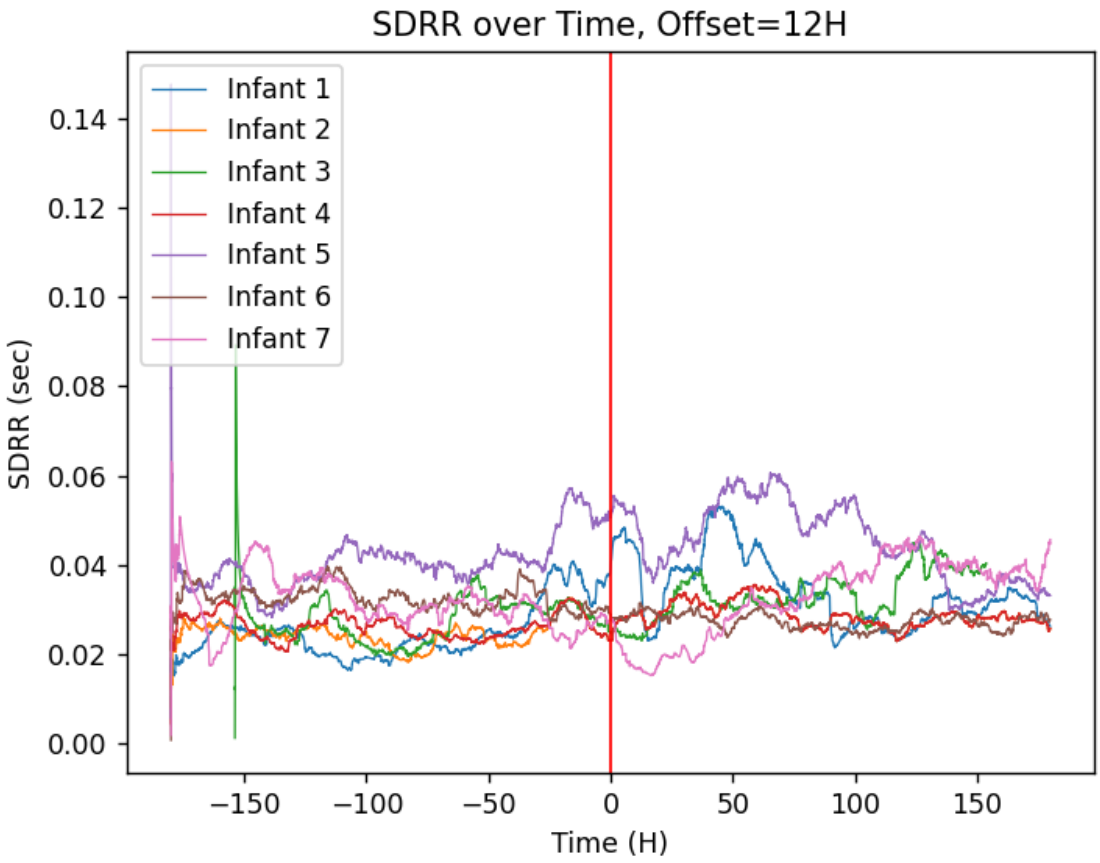


Figure 12: RMSSD

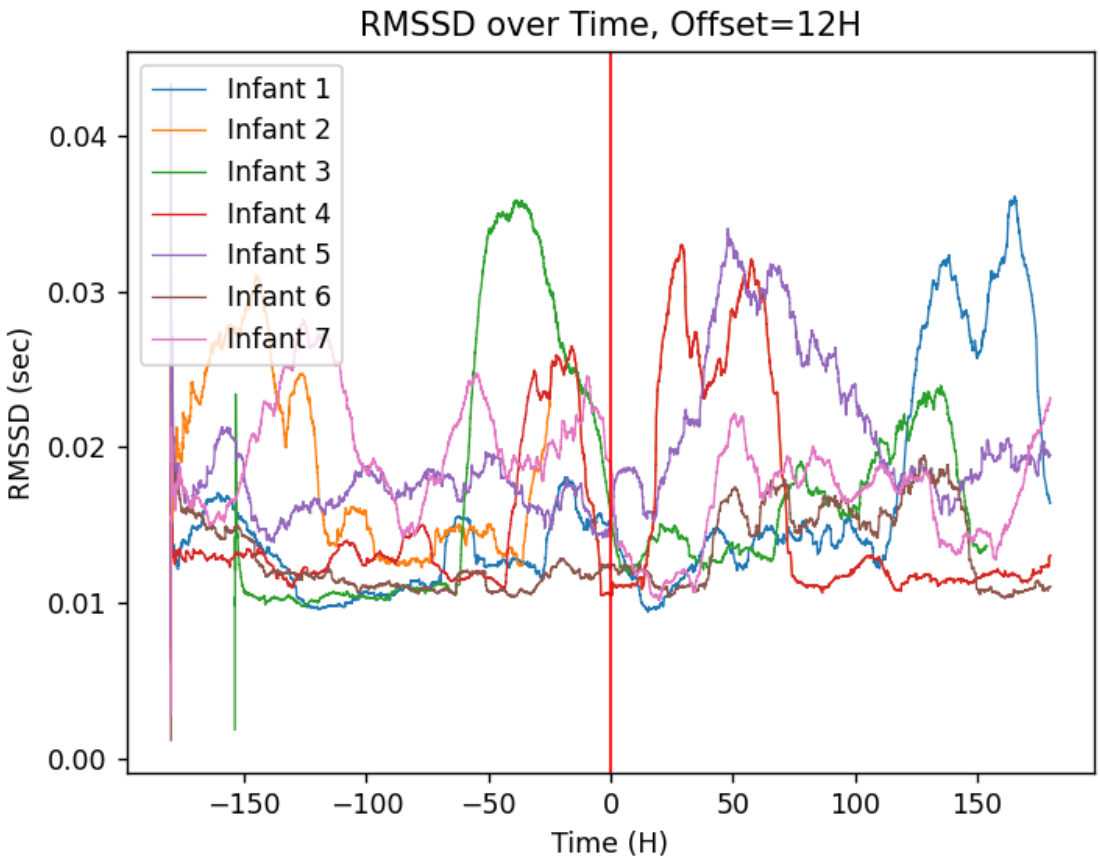


Figure 13: pNN20

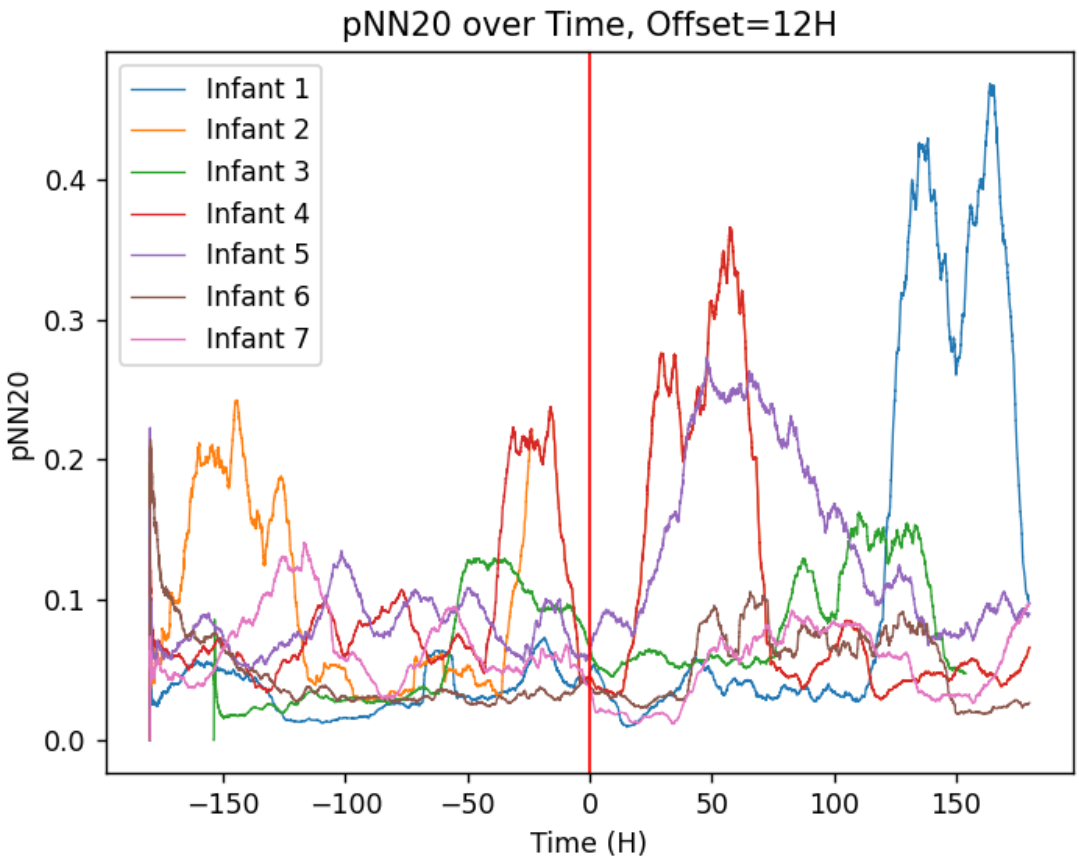
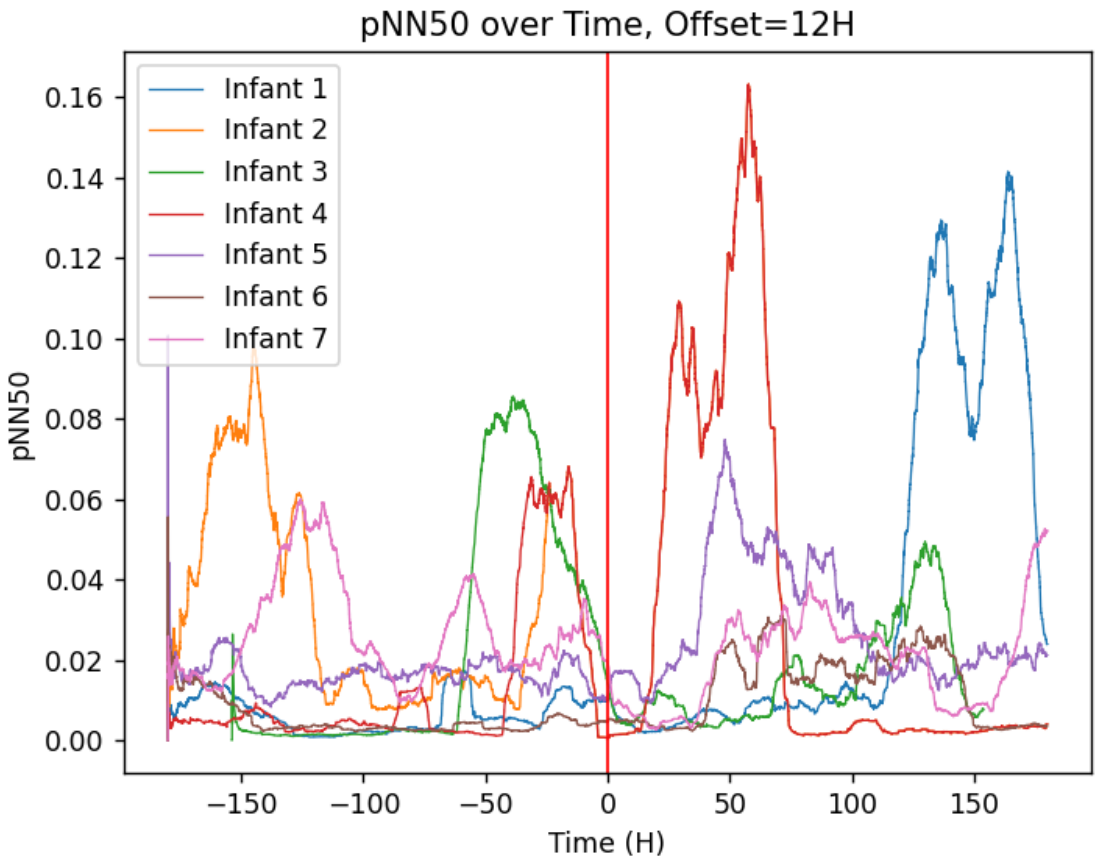


Figure 14: pNN50



Discussion

First things first: Figures 10 though 14 show that, starting around 40% of the way through the data, the metrics for infants 2 and 4 coincide in a way that cannot be coincidental. This points to an issue of data integrity. Unfortunately, the cause of this issue could be in the Python code or it could go all the way back to the pulling of the data. Investigation of this issue would be good.

Focusing just on the pipeline, there is much room for improvement. First, all the way back at the bedside monitor, the source of troughs could be investigated. These only appeared in files 1a and 1b, but it may be possible to understand what caused them and if they can be avoided in the future, as they increased data loss. Second, the method of pulling the data should be corrected to solve the time index gap problem entirely. Third, it could prove fruitful to denoise the ECG data in an effort to reduce the number of undetected beats. Fourth, the work of Eilers, Chromik, and Arnrich specifically excluded ECG data from infants, so an investigation of other QRS detectors could reveal the best one for this application (2021). Fifth, a more sophisticated approach to distinguishing between missing and multiple intervals would aid in recovering more missed beats.

The issue of breaking apart multiple intervals deserves special attention. The way we approached this—breaking them into a number of equally sized pieces—artifically reduces HRV metrics over a rolling window because there are several intervals of exactly the same length. Here, it was not a problem because the window was 12 hours long, but as the window becomes smaller and smaller, this approach causes more and more downward bias. We propose a method of iterating over the intervals, and every time an unrealistically long one is found, dividing it by the previous interval and round the result to determine how many pieces to break it into, fitting a normal distribution to the previous intervals, and using this normal distribution to break the multiple interval into pieces. There are problems with this approach too, such as what to do if the previous interval is also a multiple interval or how many previous interval are necessary to fit the normal distribution, but our hope is that it would recover missed beats and also preserve some of the variability.

Taking a broader perspective, the fact that this simple approach did not work points to the necessity of applying more sophisticated statistical methods to this problem. The HeRO score, for example, uses a specialized form of logistic regression to produce its results, and other approaches have used logistic regression, decision tree models, and other techniques coming from the realm of sophisticated statistical learning (Fairchild and Aschner, 2012; Masino et al, 2019). Predicting the future is a difficult task, and the problem of predicting sepsis is no exception.

Finally, while the negative result is disappointing, this was an undergraduate research project, so to the end of providing the author with real world experience in data science, it has been a major success.

Bibliography

Eilers, J., Chromik, J. and Arnrich, B. (2021). Choosing the appropriate QRS detector. Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, 4, 50–59. https://doi.org/10.5220/0010234600500059

Fairchild, K. D. and Aschner, J. L. (2012). Hero monitoring to reduce mortality in NICU patients. Research and Reports in Neonatology, 65–76. https://doi.org/10.2147/rrn.s32570

Masino, A. J. et al (2019). Machine learning models for early sepsis recognition in the neonatal intensive care unit using readily available electronic health record data. PLOS ONE, 14(2). https://doi.org/10.1371/journal.pone.0212665

Ring, E. (2018, January 25). Warning as sepsis can kill in 12 Hours. Irish Examiner. Retrieved January 17, 2023, from https://www.irishexaminer.com/news/arid-30824690.html