I will represent my frequency downsampling step. (With Valentina's help, I get started to use center-TBI data and I identified patients with the arterial blood pressure signal, and their patient IDs are shown like this.

then I preprocessed ABP signal of one of the patients.)

the plot shows arterial blood pressure raw signals, which is valid since it shows diastolic pressure and systolic pressure clearly.

Then after rescaling, handling outliers, and smoothing process, the ABP signal looks like this pattern. After that, I tried 2 methods to down sampled the signal from 100 hertz to about 1 hertz. The blue line shows downsampling utilizing interp1d function and the orange line used resample function.

From the graph, we can see that the signal can be valid and smooth even when the sample rate is down to 13 Hz. However, if we down sample it to 7 hertz, the signal has many small unnatural spikes and obviously loses some information. In the lower frequency plots like 3 hertz and 1 hertz, we have two different downsampling results because they are totally different mathematical methods.

Then I will briefly explain what those methods are mathematically doing. The interp1d function did linear interpolation, at the beginning original signal is sampled 100 per second and we fit a line with every two samples. Then we can sample from the line with lower sampling rates say 10 hertz or even 1 hertz.

I want to briefly explain what is the discrete Fourier transform, you might be familiar with F series which takes any periodic signal as the sum of sinusoids, each of these sinusoids is defined by their frequency k. By Euler's formula, sinusoids can be written in exponential form. So how do we calculate the coefficient at each particular frequency? The answer is to use DFT. DFT is the same thing as calculating correlations. We are multiplying a function or in our case a signal

by an analyzing function which in our case are sinusoids. Wherever function and analyzing function are similar they will multiply and sum to a large coefficient and wherever these two functions are dissimilar they will multiply and sum to a small coefficient. Now we can represent the sinusoids as a complex exponential and the result is one complex coefficient per frequency which shows the amplitude of that frequency signal contained in our original signal. That’s how we transform signals to the frequency domain and the next step I think is to do downsampling to other signals like ECG.

I drew the specific method of downsampling based on Fourier transform: first, convert the time-domain plot into a frequency domain plot, and then delete samples with high frequency each time. In this way, the frequency of the signal after downsampling is getting lower and lower, and finally, there are only a few low-frequency signals left, which is the reason for the 3hz and 1Hz diagram.

This is the number of samples of different frequencies in ecg abp and icp signals.

Then I also downsampled the other two signals. Like the ABP signal, the icp and ECG signals also lose a lot of information in some downsampling stages. As shown in the figure, the ECG signals lose important information from 20 Hz to 13 Hz, while the icp loses important information from 10 Hz to 7 Hz.

In order to predict the risk of mortality, quantita/ei/tive features have been computed based on the time series signals. Each signal is described in terms of 9 statistical and signal-based features which were extracted from the patient’s ECG signal. Maximum, minimum, and range can demonstrate the spectrum in which the distribution lies. The skewness indicates whether the distribution is symmetric or skewed. The kur’tosis measures the thickness of the tails of the distribution and the standard deviation shows how the data samples scatter around the mean.

Next step maybe I should explore more features that have to be extracted, like energy spectral density ( I have no idea what it is but I have seen some papers extract it)

I recalculated how many samples of the signal were at each frequency rate because the original result was a calculation after a signal was segmented into several signals. Now it seems that the samples have become much fewer.

Next, I checked whether the signals were all from the first day the patient was admitted, and then I found the patient IDs list for all the signals from the first day. This is about the ECG signal. And this is about the icp signal.

After extracting the features, I extracted these nine features at various sampling frequencies on the ECG signal of the test file. It can be seen that the sampling frequency does not change much until 20 Hz, indicating that there is not much information loss. So next I decided to extract features on all the ECG signals from the admission day, we obtained earlier, and take the mean of the nine features. The result is that except for the mean, none of the other features change much as the sampling frequency decreases.

The project is to use data from Center TBI to construct deep learning and traditional models and to understand how the frequency of physiological signals affects TBI prediction accuracy. First, we have all patients admitted to the ICU with a TBI diagnosis. The time series signals are filtered, cleaned and downsampled for the purposes of our sampling rate study. The ultimate goal of this project is to find reasonable sampling frequency of these physiological time series signals which allows deep learning models as well as non-DL models to predict the clinical outcomes of patients with TBI. To achieve these objectives, now I have to build models and this is my modeling plan.

To predict the long-term outcome, mortality, I will **implement several traditional machine learning approaches, like** logistic regression, random forest, K-NN, and Gaussian SVM, etc. Before training, I have already extracted **statistical and signal-based features like mean, median, variance, etc. Recently, I read a paper in which the author found 2 features are significant to do prediction, and I have extracted them too.**

**To predict short-term outcomes, like changes in ICP, we will start out using LSTM models which are based on RNN structures. I have written a sample, a simple LSTM structure as follows, with a single hidden layer. Based on this structure, we** can define a new **Stacked LSTM** model with more hidden layers.

After that, we can try CNN model in a hybrid model with an LSTM backend in which I can utilize [TimeDistributed wrapper](https://machinelearningmastery.com/timedistributed-layer-for-long-short-term-memory-networks-in-python/) to reuse the same CNN model when reading different sub-sequence of data.

1.

Traumatic brain injury (TBI) is defined as the disruption of normal brain function caused by an insult to the head. We predict mortality in patients with TBI based on the first day of admission signal data, using the dataset Center-TBI.

The ultimate goal of this project is to find reasonable sampling frequencies of these physiological time series signals which allow deep learning models as well as non-DL models to predict the clinical outcomes of patients with TBI.

（reasonable sampling frequency for these physiological time series signals that is low enough to reduce monitoring costs while containing enough information to make predictions.）

2.

First, we have all patients admitted to the ICU with a TBI diagnosis. The time series signals (ABP, ECG, ICP) are cleaned, denoised, smoothed, and downsampled for the purposes of our sampling rate study.

3.

I downsampled these signals from 100 Hz to 1 Hz with 2 methods. the orange line represents Fourier Transform method and the blue line represents Interpolation method.

到11

All of these signals lose a lot of information in some downsampling stages. (the ECG signals lose important information from 20 Hz to 13 Hz, while the icp loses important information from 10 Hz to 7 Hz.)

177 patients have signals on the first day of admission, the mortality rate is 18%, and we train models using 10 fold cross-validation.

Before training, I have already extracted 9 **statistical features: maximum, minimum, mean, median, mode, variance, range, kurtosis, skewness.**

**4.**

To predict the mortality, before using deep learning, I **implemented several traditional machine learning approaches, like** logistic regression, random forest, Gaussian SVM, etc~~. Features used to train models here are~~ **~~statistical features of the original signals.~~** On the original data, random forest and SVM performed the best, with an AUC above 0.69.

5. Then, I trained RF, SVM models on 100Hz, 75Hz, 50Hz, 25Hz, 20Hz, 15Hz, 10Hz, 5Hz and 2Hz signal data. Even with a sampling frequency of 2hz, the model's AUC score doesn't decrease that much. However, according to the downsampling procedure figures we drew earlier, these signals lose a lot of signal-based information when they are downsampled to 7hz. The reason may be that the **features extracted** for traditional machine learning cannot effectively reflect signal-based information, that is, even when the fluctuation and shape of the signal have changed, these features have not changed much

6. NN

NN models might better capture signal-based information so the next step I can **Implement dl models.** I have already segmented each signal into several chunks and constructed an LSTM baseline. The next step is to train the LSTM model with different sampling rate signals.

…… AUC score decreases from 0.673 to 0.5834. In the same way, I build a model based on RNN structure, the performance of NN decreases from 0.7407 to 0.5556. Even with a sampling frequency of 3hz, the model's AUC score doesn't decrease that much.

Yes, I agree. This work will be my next step.

Even with a sampling frequency of 3hz, the model's accuracy score doesn't decrease that much. However, according to the downsampling procedure figures we drew earlier, these signals lose a lot of signal-based information when they are downsampled to 7hz. The reason may be that the features we extracted for traditional machine learning cannot effectively reflect signal-based information, that is, even when the fluctuation and shape of the signal have changed, these features have not changed much.