Project Update

Ruixuan Dong

May 11, 2023

---

1 Check the Changes

I tried to find a good way to fit models with raw data set, but there was still bad performance. Based on that, I started to put more focus on data processing under the guidance of Prof. Ke.

First of all, I tried to check the time series plots when the sleeping posture changes. The Figure 1 shows some examples.

In the left figure, I have selected the first five samples and the last five samples where the posture changed. I have shown their time series plots separately, totaling ten plots. The time period where the change occurred is from 2020-10-02-13:34:39 to 2020-10-02-13:34:49.

In the right figure, I have selected the first five samples and the last five samples where the posture changed. I have shown their time series plots separately, totaling ten plots. The time period where the change occurred is from 2020-10-10-12:33:29 to 2020-10-10-12:33:39.

Besides, the data set used in plots have been standardized.

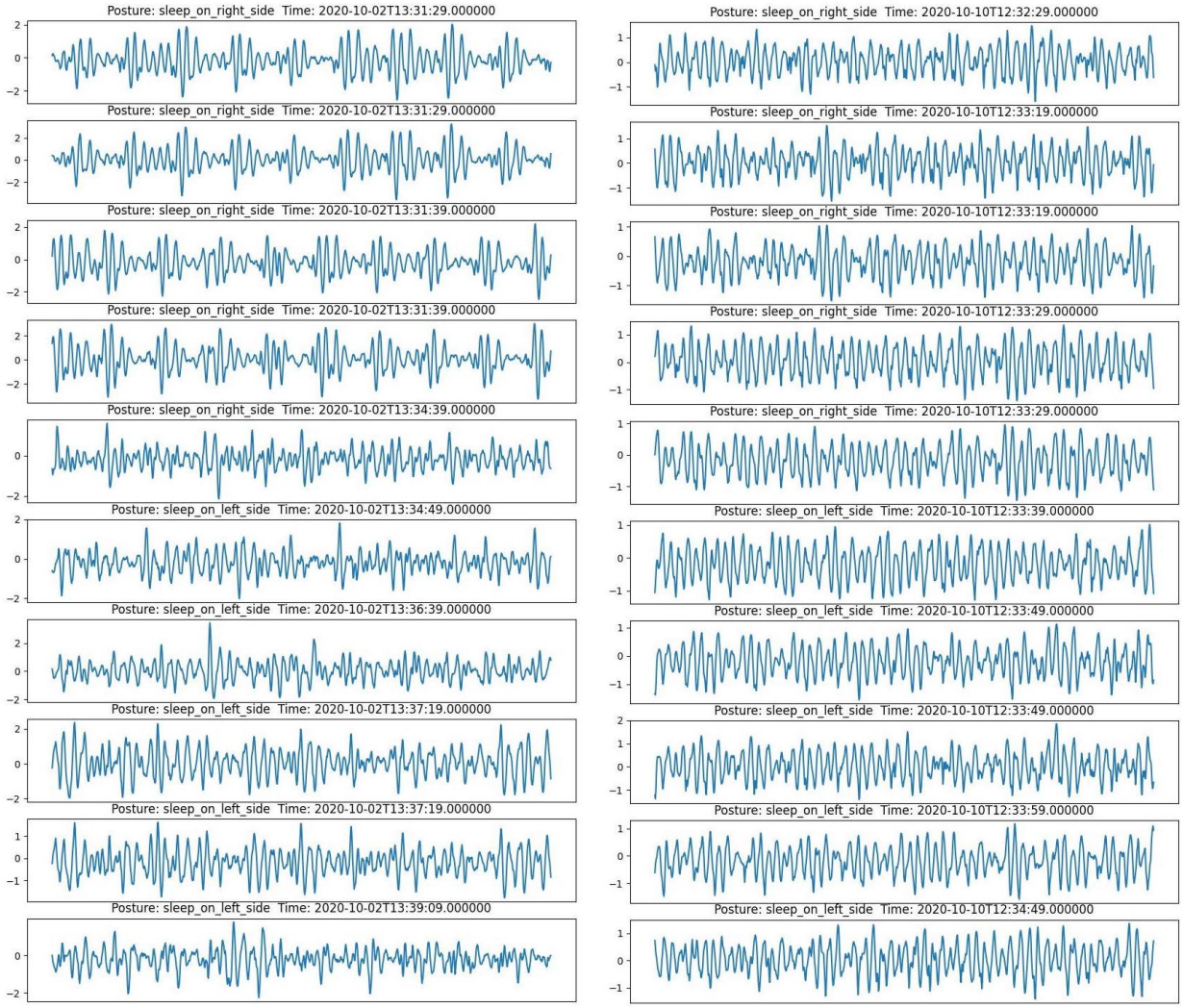


Figure 1: Left: From posture "sleep on right side" to "sleep on left side", the first five samples are sleeping on right side in time order, and the last five samples are sleeping on left side in time order. Right: From posture "sleep on right side" to "sleep on left side", the first five samples are sleeping on right side in time order, and the last five samples are sleeping on left side in time order.

It seems that the plots do not exhibit periodicity, such as heart rate or respiratory rate, what would happen if we apply noise reduction techniques to this data?

2 Reduce noise

2.1 Wavelet Method

There are three steps for wavelet method:

1. Decomposition
2. Thresholding
3. Reconstruction

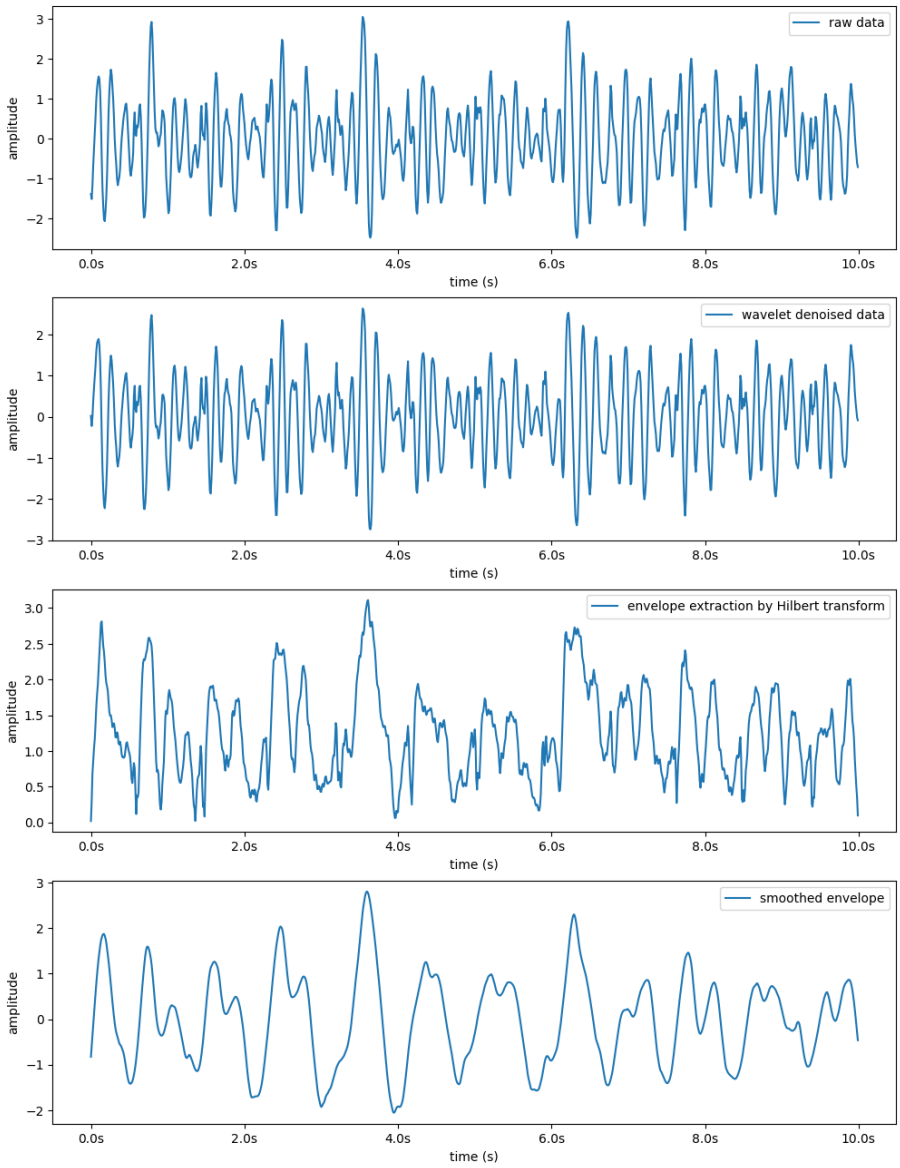
2.2 Envelope extraction by Hilbert transform

To extract the envelope of the signal, we use a Hilbert transform. The Hilbert transform assumes that the amplitude of the signal is modulated by its instantaneous amplitude.

2.3 Smooth Envelope

To further process the envelope of the signal, we apply a Savitzky–Golay filter.

Implement result:



3 Modeling

After implement the data processing steps above, we rebuild a neural network. In this part, we try to use a different way to create training and testing data: Each time use 43 day's of data as training and set the remaining 1 day as testing. To achieve this goal, we fit models based on each training set, i.e. we'll fit 44 neural network in total in this part. Then, we will use each day's data as testing set to assess each NN model. Each fitted model is saved in an ".h5" file. Then, the prediction results for each testing set are displayed below:

	loss	accuracy	f1_score	precision	recall
Day1	2.221324	0.584000	0.570526	0.578638	0.563039
Day2	1.284966	0.590164	0.590330	0.597500	0.583534
Day3	1.821251	0.540146	0.545803	0.564665	0.528549
Day4	4.342727	0.204545	0.194444	0.196237	0.192708
Day5	0.668681	0.787234	0.810929	0.828736	0.794792
Day6	1.264447	0.688963	0.697432	0.713961	0.682102
Day7	1.332536	0.669903	0.665774	0.673147	0.658631
Day8	1.696271	0.409091	0.377439	0.457734	0.325893
Day9	2.941081	0.459854	0.457639	0.457639	0.457639
Day10	0.785922	0.762452	0.731652	0.750717	0.714583
Day11	0.546918	0.786408	0.791093	0.814511	0.769574
Day12	2.085983	0.636792	0.640963	0.647408	0.634821
Day13	0.642458	0.725806	0.727322	0.745977	0.710417
Day14	1.664152	0.594862	0.596164	0.609581	0.583834
Day15	1.154118	0.578402	0.550385	0.580998	0.524148
Day16	1.363383	0.677621	0.679619	0.695359	0.665121
Day17	2.228790	0.576923	0.564087	0.565726	0.562500
Day18	1.692549	0.423529	0.424672	0.430444	0.419147
Day19	1.433873	0.571918	0.606442	0.613299	0.600000
Day20	1.943042	0.506061	0.512332	0.519343	0.505682
Day21	1.880222	0.588933	0.594246	0.600284	0.588497
Day22	3.695263	0.274678	0.280443	0.285123	0.276042
Day23	6.256645	0.098522	0.085253	0.085714	0.084821
Day24	5.450447	0.146199	0.134094	0.135936	0.132353
Day25	2.868031	0.185039	0.165874	0.179782	0.154167
Day26	1.157250	0.660944	0.658507	0.677882	0.640835
Day27	4.467694	0.161716	0.156437	0.158271	0.154715
Day28	4.729933	0.168627	0.166077	0.169010	0.163281
Day29	4.721951	0.135972	0.133164	0.136115	0.130592
Day30	5.345479	0.077572	0.077605	0.078117	0.077109
Day31	2.378765	0.474000	0.473766	0.479440	0.468359
Day32	5.413353	0.152778	0.115906	0.117272	0.114583
Day33	4.127620	0.071006	0.075927	0.078184	0.073864
Day34	5.358311	0.111576	0.109144	0.110397	0.107964
Day35	5.257705	0.080357	0.089894	0.091826	0.088068
Day36	3.853204	0.092647	0.072692	0.078020	0.068182
Day37	0.689533	0.751896	0.751848	0.766840	0.737827
Day38	2.444560	0.435000	0.406650	0.411770	0.401786
Day39	1.686083	0.521127	0.547251	0.575835	0.522321
Day40	2.802088	0.520325	0.528375	0.532407	0.524595
Day41	1.051902	0.675879	0.673449	0.688480	0.659643
Day42	2.631368	0.408867	0.390578	0.397097	0.384375
Day43	1.819116	0.466905	0.459368	0.470130	0.449306
Day44	1.666230	0.586466	0.595016	0.602276	0.588194
Overall	2.610619	0.445948	0.442650	0.453360	0.433368