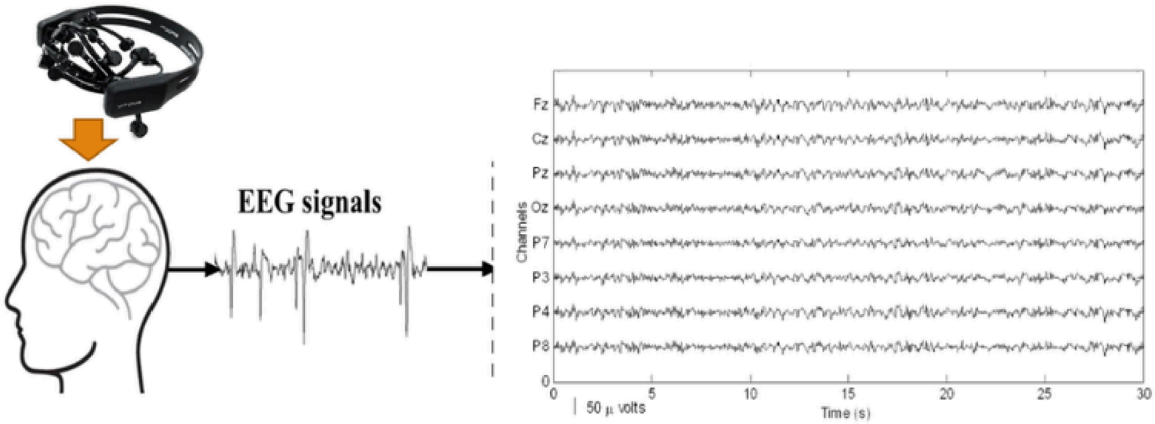
Autoregressive model on our data



Based on the paper "*Multimodal evidence suggests the linearity of brain dynamics at the macroscale*"

We want to apply an AR (Autoregression) model to our data and after applying this method of forecasting we aim to report 3 measurements to see how well our AR model did the prediction:

1-the accuracy of their one-step and multistep ahead predictions by cross-validated regional R_i^2:

$$R_i^2 = 1 - \frac{\sum_t \varepsilon_i(t)^2}{\sum_t (y_i(t) - \bar{y}_i)^2}$$

and the latter is often assessed via a χ^2 test of whiteness (Methods), for each channel $i = 1,\ldots,n$. In equation (3), $\varepsilon(t)$ is the same one-step-ahead prediction error in equation (2) and $\bar{y}_i$ is the temporal average of $y_i(t)$ (often equal to zero due to mean centreing) and corresponds to a constant predictor which always predicts $\mathbf{y}(t)$ equal to its average $\bar{\mathbf{y}}$. Therefore, it is clear that R_i^2 is always ≤1 but 'can be negative'. A value of $R_i^2 = 1$ indicates a perfect model (for channel/region i), $R_i^2 = 0$ indicates a model as good as the constant predictor, and $R_i^2 < 0$ indicates a model worse than the constant predictor.

R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

2-amount of unexplained residual dynamics (chi-squared test of whiteness)

Multivariate Test of Whiteness

The multivariate test of whiteness is a standard measure used to evaluate the goodness of fit in the prediction error method. This test assesses whether the residuals (prediction errors) from a model are "white." Essentially, "white" residuals mean that all temporal structure or dynamics in the data have been effectively captured by the model.

Definition of Whiteness in Multivariate Time Series A multivariate time series e(t) is considered "white" if it shows no statistical dependence across time. This means that $e(s)$ and $e(t)$ should be independent for $s \neq t$.

Specific Formula and Implementation

The original definition of Q used is:

$$Q = (N - M) \sum_{i=1}^{M} \operatorname{tr} \left(\hat{R}_e(i)^T \hat{R}_e(0)^{-1} \hat{R}_e(i) \hat{R}_e(0)^{-1} \right)$$

Where:

- N is the number of test samples.
- M is the number of cross-correlation lags.
- $\hat{R}_e(i)$ is the finite-sample estimate of the cross-correlation matrix between channels of $e(t)$ at lag i.

The cross-correlation matrix $\hat{R}_e(i)$ is defined as:

$$\hat{R}_e(i) = \frac{1}{N-M} \sum_{t=0}^{N-M-1} e(t+i) e(t)^T, \quad i = 0, 1, \ldots, M$$

Since $\hat{R}_e(0)$ can be singular or nearly singular, the pseudo-inverse of $\hat{R}_e(0)$ is used instead of its inverse when computing Q.

3-computational complexity (central processing unit (CPU) time).

For all models, the continuous-time dynamics in the equation was first discretized.

With a slight abuse of notation, we also represent the discretized dynamics as:

$$\mathbf{x}(t) - \mathbf{x}(t-1) = f(\mathbf{x}(t-1)) + \mathbf{e}_1(t),$$

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0, \qquad t = 1, \dots, N \\ \mathbf{y}(t) &= h(\mathbf{x}(t)) + \mathbf{e}_2(t), \qquad t = 0, \dots, N \end{aligned}$$

Linear_Autoregression:

Extend above equation to:

$$\begin{aligned} \mathbf{y}(t) - \mathbf{y}(t-1) = {} & W\mathbf{y}(t-1) + D_2\mathbf{y}(t-2) \\ & + D_3\mathbf{y}(t-3) + \cdots \\ & + D_d\mathbf{y}(t-d) + \mathbf{e}(t) \end{aligned}$$

Linear AR models ('AR-2 (sparse)', 'VAR-2 (sparse)', 'AR-3 (sparse)', 'VAR-3 (sparse)', 'AR-100 (sparse)', 'AR-100 (scalar)'). Motivated by the long history of AR models in neuroscience[24,72,73], here we extend equation (6) to

$$\begin{aligned} \mathbf{y}(t) - \mathbf{y}(t-1) = {} & W\mathbf{y}(t-1) + D_2\mathbf{y}(t-2) \\ & + D_3\mathbf{y}(t-3) + \cdots \\ & + D_d\mathbf{y}(t-d) + \mathbf{e}(t) \end{aligned} \tag{7}$$

for an 'AR-d' model. The number of lags d was tuned separately for fMRI and iEEG, and the matrix W is either made sparse using LASSO or enforced to be diagonal. Note that the latter results in n scalar AR models at each node, which are completely decoupled from each other. We restricted the matrices $D_2, D_3, \dots$ to be diagonal in 'AR' models but not so in full vector autoregressive ('VAR') models. In both cases, we used LASSO regularization to promote sparsity in the regressors, signified by the '(sparse)' suffix in method identifiers, with the regularization hyperparameter λ chosen optimally and separately for each model (cf. 'Hyperparameter selection' below). In general, we found that λ is a moderately sensitive parameter, more so for the whiteness of residuals than for R^2 (see Supplementary Fig. 3 for an example).

Results of Meeting with Dr.Hajipour

Most important thing!

We have to ensure our model works well and can fit to any Y we want.

- [x] ~~Plot~~ $\cancel{Y\ and\ \widehat{Y}}$ ~~to visually see how well our prediction and regression is~~
- [x] ~~Build an artificial EEG signal with simulation (this fake signal can be linear in a confident way) and then fit the regression model (should be perfectly fit to the signal). Then we have to add some noise and see its effects.~~
- [x] ~~Plotting and visualizing box plot to:~~
  1. To compare subjects in each condition and visually see the accuracies.
  2. To see the robustness of our model in different subjects and also trials.
  3. To compare different conditions (sham stim7 stim8).
  4. To compare with the results of the paper.

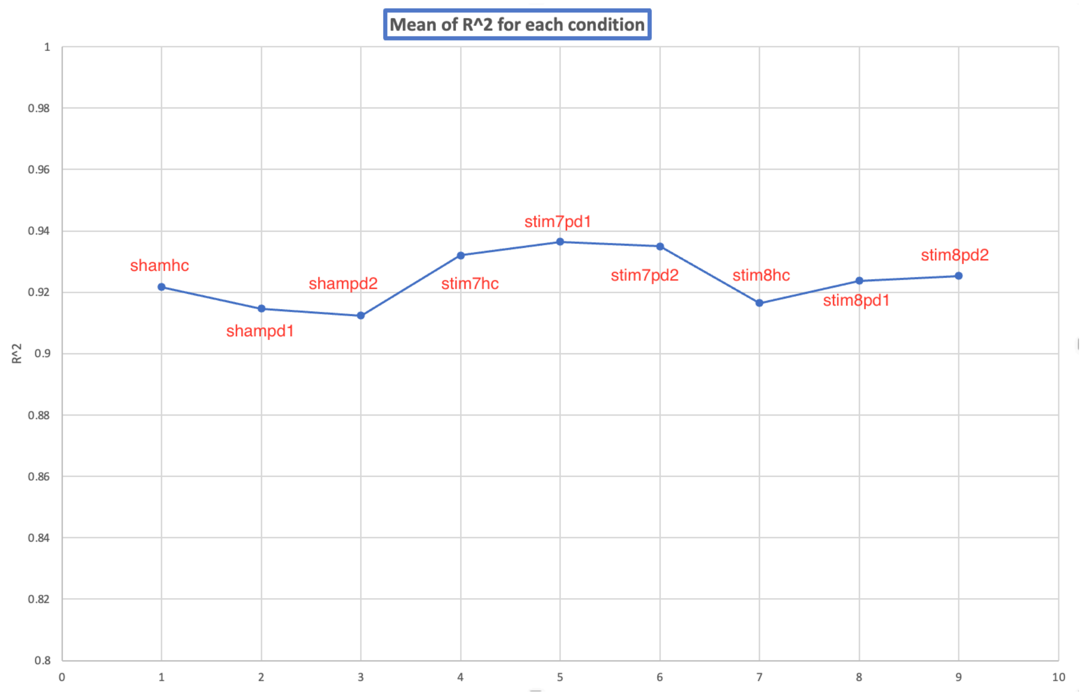
Model Results – to raw data in this link: [Row Data1]

First we want to talk about the R2 metric which is a crucial sign of how well the model fitted to our data. We give the average of trials in different subjects for each electrode so we'll have $27 * 22/20$ matrices which show the R^2 metric for each electrode(rows) for each subject(columns) . Here is an example of this result (for "*shamhceeg*" Data set)

27x22 double

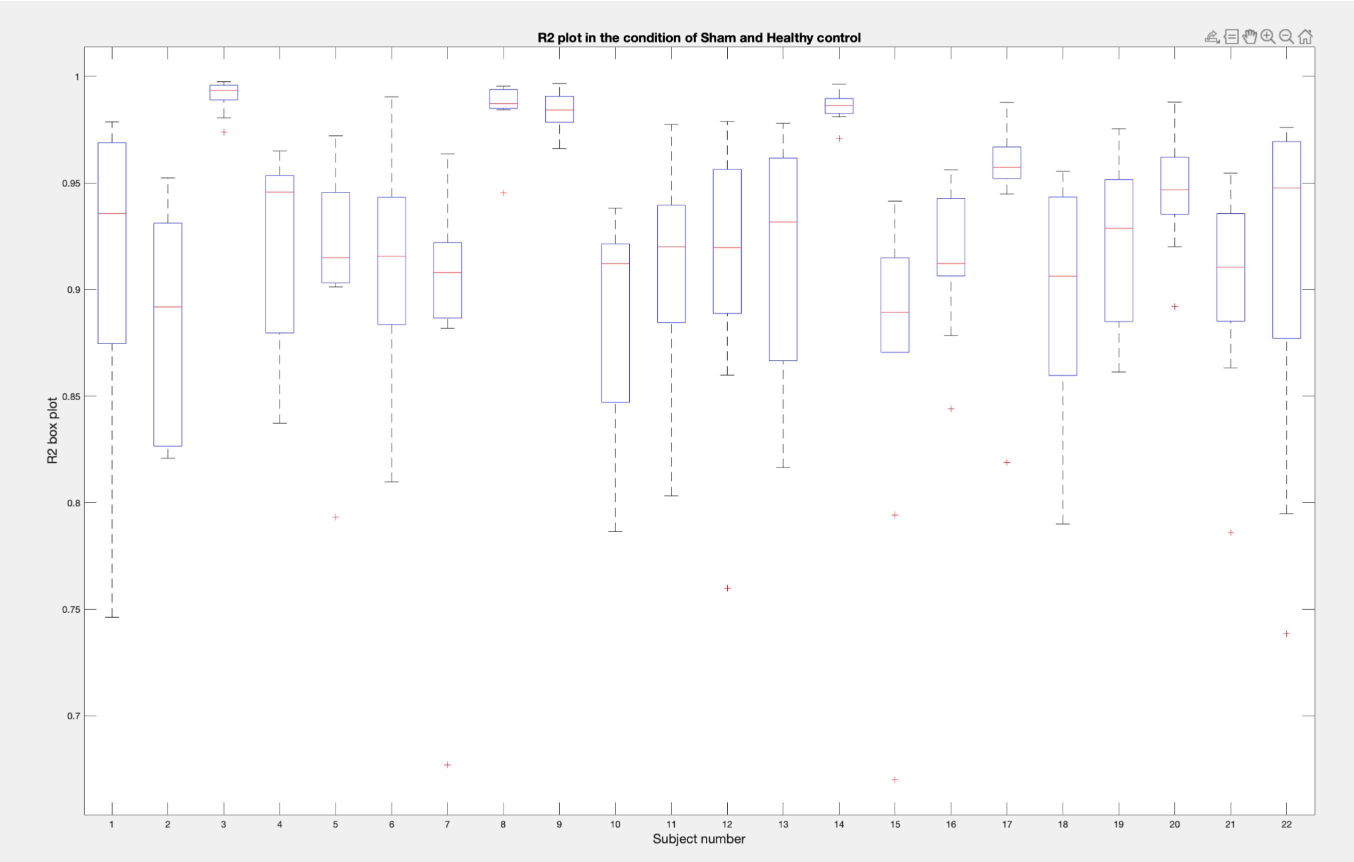
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0.9613	0.9136	0.9819	0.8990	0.9506	0.8650	0.8483	...	...	0.9172	0.8213	0.9167	0.9083	0.9934	0.8527	0.8611	0.9629	0.9257	0.8671	0.9429	0.9308	0.8842
0.9128	0.9083	0.9924	0.9236	0.9587	0.8604	0.8540	...	...	0.8827	0.8658	0.9380	0.9430	0.9874	0.7997	0.9273	0.9078	0.8559	0.8793	0.9518	0.9324	0.9175
0.9149	0.9274	0.9951	0.9044	0.9418	0.8614	0.8461	...	...	0.9392	0.8895	0.9748	0.9915	0.9831	0.9057	0.9407	0.9614	0.8859	0.9864	0.9605	0.9275	0.9054
0.9643	0.9456	0.9942	0.9387	0.8871	0.9854	0.9074	...	...	0.9182	0.7719	0.9460	0.9080	0.9934	0.8939	0.8798	0.9837	0.9231	0.8596	0.9884	0.9292	0.9188
0.9662	0.9036	0.9793	0.9095	0.8299	0.9191	0.8854	...	...	0.8269	0.8873	0.9465	0.8704	0.9926	0.7912	0.8509	0.8504	0.9247	0.8591	0.9759	0.8637	0.9279
0.9740	0.9018	0.9930	0.9081	0.8408	0.9183	0.9069	...	...	0.9504	0.9407	0.9532	0.9542	0.9636	0.8680	0.8715	0.9108	0.9453	0.8694	0.9584	0.8759	0.9173
0.9626	0.9556	0.9974	0.9400	0.8795	0.8922	0.8962	...	...	0.9014	0.9415	0.9900	0.9392	0.9936	0.8427	0.9301	0.9569	0.9217	0.9774	0.9552	0.9045	0.9075
0.9338	0.8994	0.9950	0.8830	0.9458	0.8465	0.8953	...	...	0.8656	0.8919	0.9886	0.9188	0.9941	0.9806	0.7443	0.9782	0.9290	0.9756	0.9578	0.8977	0.9245
0.9495	0.9554	0.9888	0.9135	0.9803	0.9422	0.8776	...	...	0.8921	0.8392	0.9467	0.9240	0.9902	0.9349	0.8954	0.9708	0.8279	0.8182	0.9938	0.9194	0.8632
0.9445	0.9297	0.9981	0.9270	0.8961	0.8857	0.8813	...	...	0.9041	0.8605	0.9886	0.9092	0.9910	0.8565	0.8888	0.9641	0.9131	0.9605	0.9911	0.8975	0.8867
0.8836	0.9701	0.9899	0.9363	0.9028	0.9118	0.9275	...	...	0.8710	0.8215	0.9508	0.9138	0.9967	0.9301	0.8799	0.9938	0.8349	0.9308	0.9375	0.9533	0.7808
0.9540	0.9154	0.9921	0.8838	0.9922	0.9327	0.8929	...	...	0.8908	0.8630	0.9952	0.9122	0.9937	0.7687	0.8787	0.9558	0.8149	0.8624	0.9619	0.9509	0.8323
0.9593	0.7843	0.9813	0.8922	0.8897	0.9205	0.8605	...	...	0.9537	0.9263	0.9083	0.8608	0.9913	0.9141	0.9341	0.9191	0.8990	0.8368	0.8963	0.8987	0.8963
0.9305	0.9235	0.9973	0.9185	0.9900	0.9213	0.9363	...	...	0.8990	0.9372	0.9970	0.8706	0.9931	0.8561	0.8992	0.9569	0.9082	0.8538	0.9496	0.8295	0.8919
0.9370	0.8716	0.9940	0.9150	0.9098	0.8931	0.8627	...	...	0.8160	0.8810	0.7713	0.8691	0.9871	0.8721	0.9478	0.9910	0.8822	0.9020	0.9925	0.8700	0.8785
0.9623	0.8396	0.9889	0.9355	0.9961	0.9811	0.9209	...	...	0.9061	0.9381	0.9930	0.9279	0.9963	0.7943	0.9929	0.9760	0.8568	0.9185	0.9531	0.8878	0.9238
0.8792	0.9166	0.9893	0.9217	0.9901	0.9689	0.8367	...	...	0.8823	0.9870	0.7785	0.9446	0.9864	0.9975	0.9870	0.9920	0.9222	0.9061	0.8918	0.8844	0.9662
0.9446	0.8178	0.9844	0.9406	0.9948	0.9544	0.9251	...	...	0.8763	0.9746	0.9234	0.9662	0.9869	0.8360	0.9664	0.9913	0.8567	0.9080	0.9687	0.9016	0.9624
0.9031	0.7827	0.9875	0.9152	0.9118	0.9804	0.9533	...	...	0.8405	0.9913	0.8749	0.9281	0.9750	0.7574	0.9306	0.9413	0.8397	0.9206	0.8461	0.9107	0.8087
0.8631	0.8684	0.9883	0.9326	0.8323	0.8807	0.9525	...	...	0.8571	0.9003	0.8653	0.8939	0.9869	0.8485	0.9332	0.9574	0.9183	0.9502	0.9351	0.8980	0.8226
0.8547	0.9069	0.9921	0.8655	0.7836	0.9120	0.8660	...	...	0.8806	0.9530	0.8162	0.8757	0.9834	0.9200	0.9078	0.9667	0.9309	0.9323	0.9306	0.8862	0.7921
0.8516	0.8899	0.9852	0.9272	0.8322	0.9839	0.8294	...	...	0.8748	0.9702	0.7785	0.9779	0.9868	0.8907	0.9472	0.9144	0.9710	0.9865	0.9820	0.9179	0.9936
0.9135	0.7278	0.9926	0.9754	0.9219	0.8964	0.9417	...	...	0.8840	0.9536	0.8865	0.9236	0.9692	0.8170	0.9921	0.9588	0.8828	0.9950	0.8916	0.8982	0.9662
0.8046	0.9088	0.9967	0.9491	0.8760	0.8797	0.8521	...	...	0.8485	0.9142	0.8081	0.8726	0.9942	0.8677	0.9021	0.9426	0.9195	0.9560	0.9149	0.8700	0.9942
0.8958	0.7095	0.9946	0.9909	0.9282	0.8403	0.9226	...	...	0.8835	0.9828	0.8685	0.9215	0.9448	0.8935	0.9696	0.9533	0.8558	0.9846	0.9098	0.9011	0.9833
0.8545	0.8307	0.9852	0.9908	0.9058	0.8408	0.9062	...	...	0.8553	0.9721	0.8723	0.9082	0.9697	0.8511	0.9870	0.8974	0.8791	0.9943	0.9396	0.9063	0.9964
0.7764	0.9078	0.9937	0.8592	0.8933	0.8647	0.8984	...	...	0.8281	0.9511	0.8230	0.8842	0.9917	0.8658	0.8818	0.8625	0.9144	0.9944	0.9515	0.8906	0.9961

Here we have plotted mean of R^2 for each condition to see overall results of model in our data set:

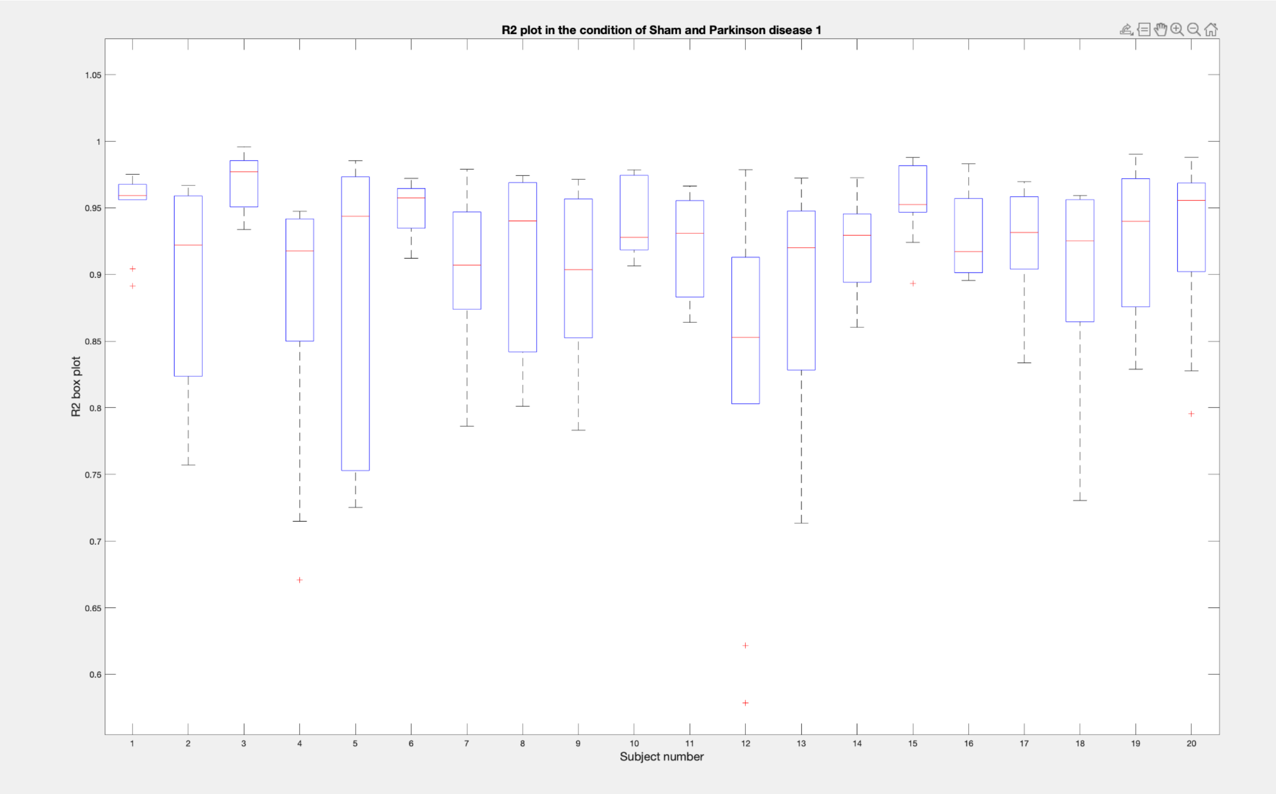


Box plots

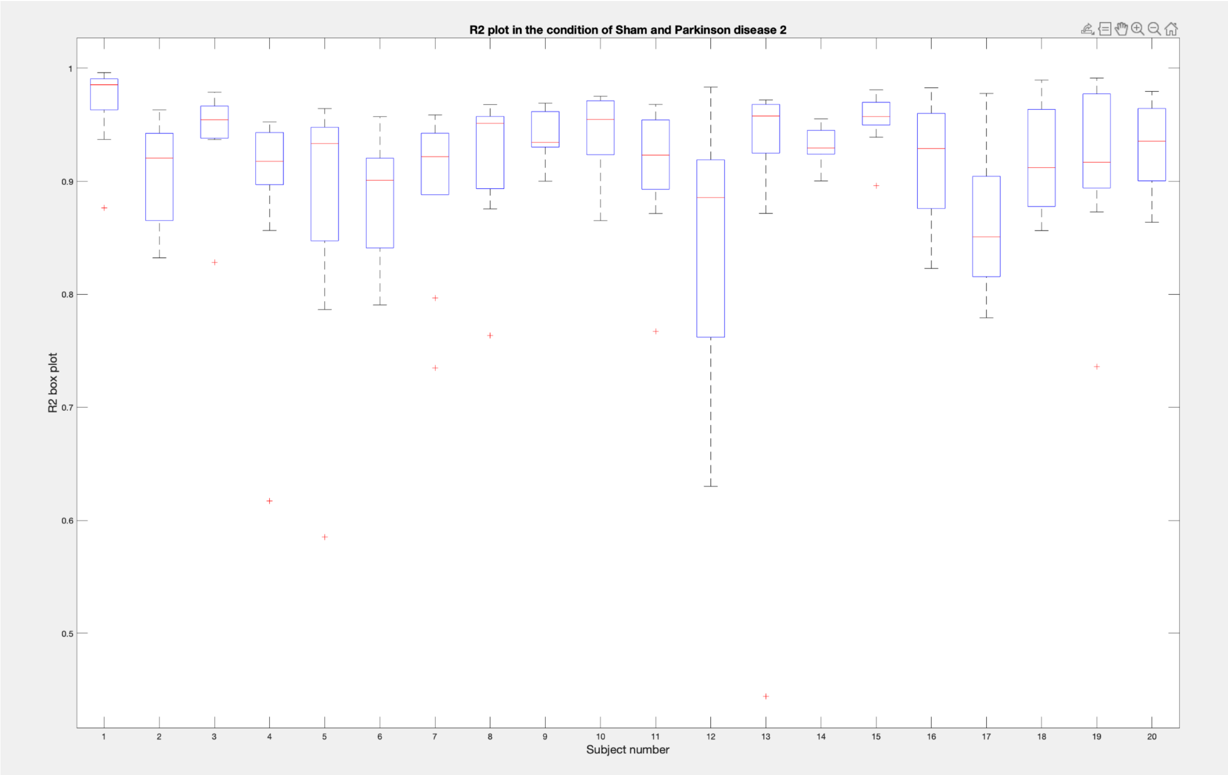
shamhc



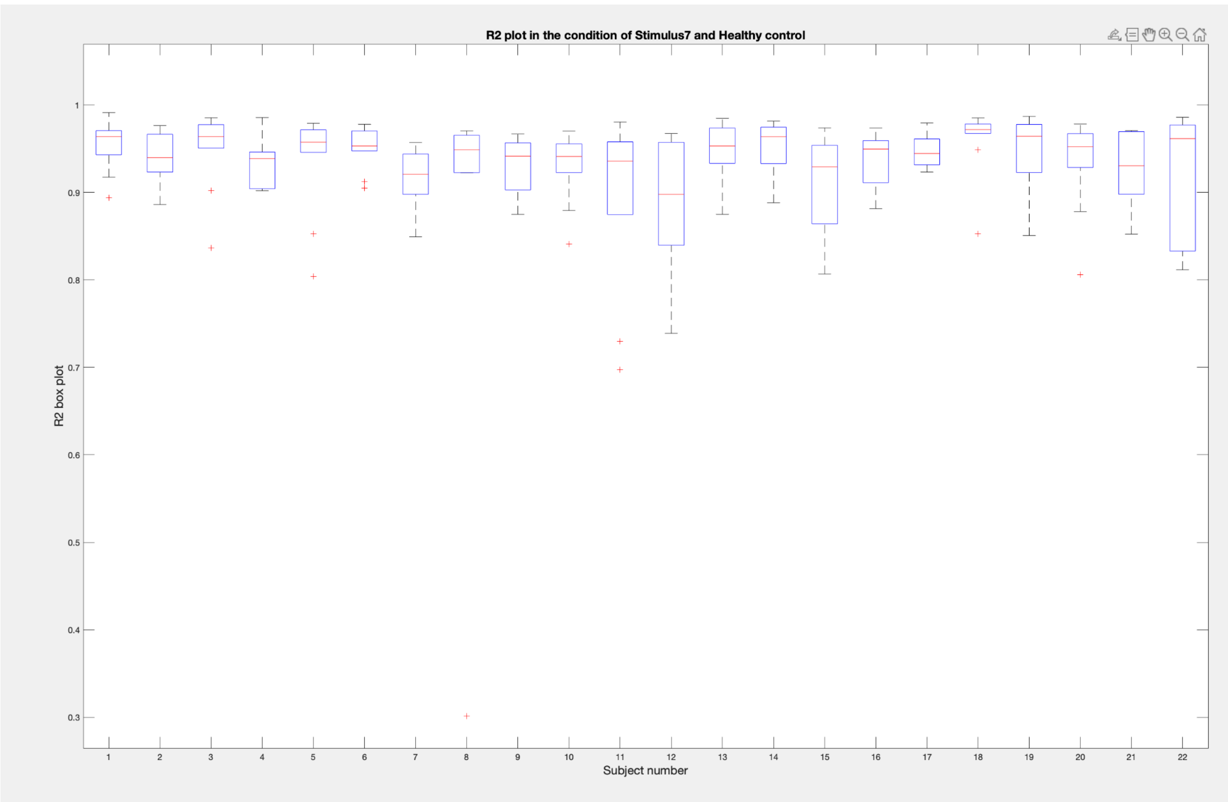
shampd1



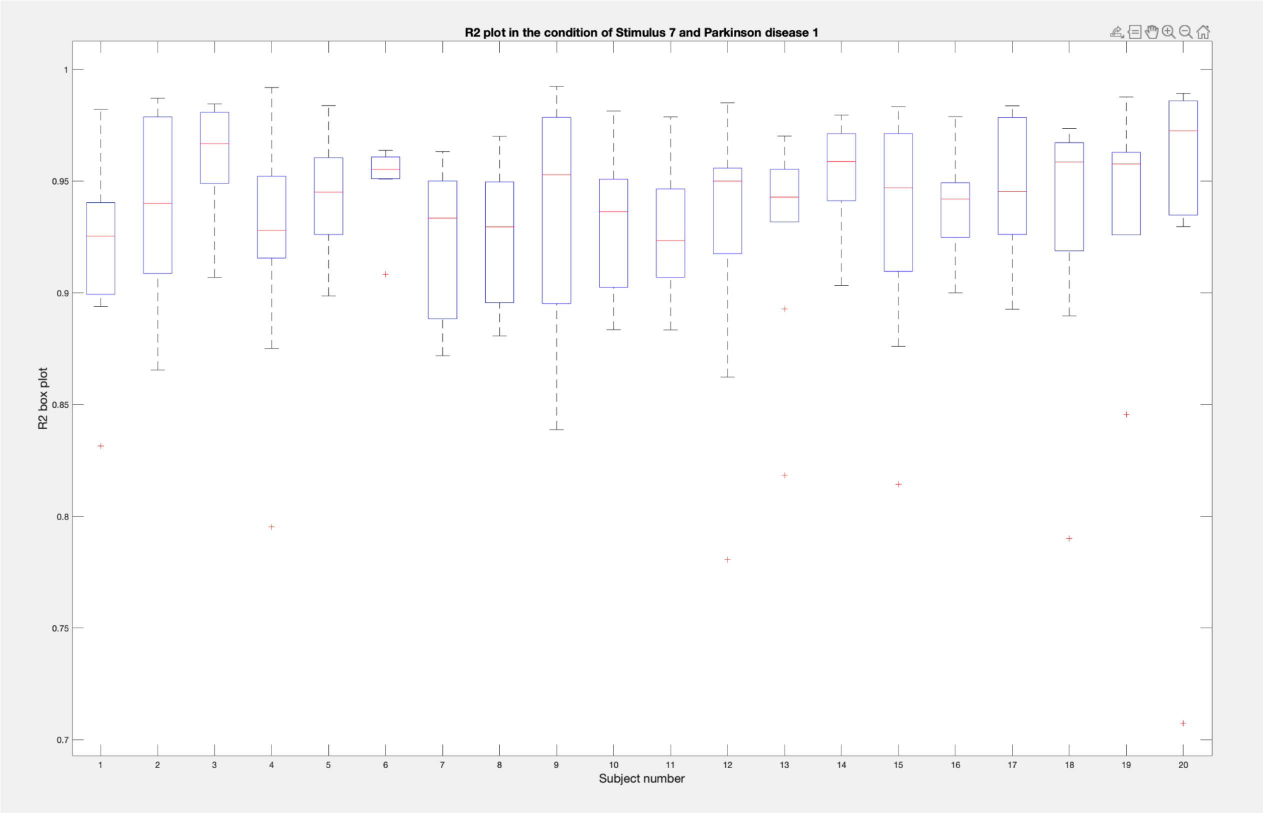
Shampd2



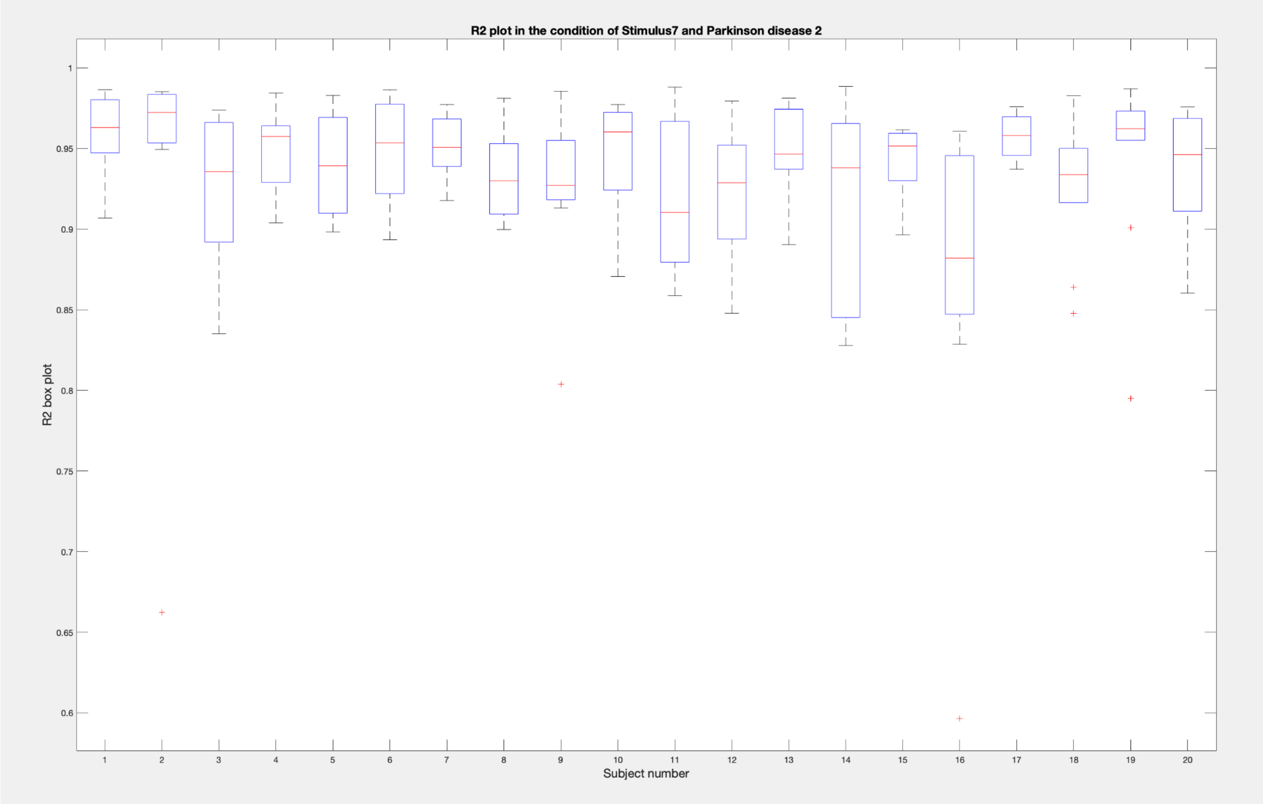
Stim7hc



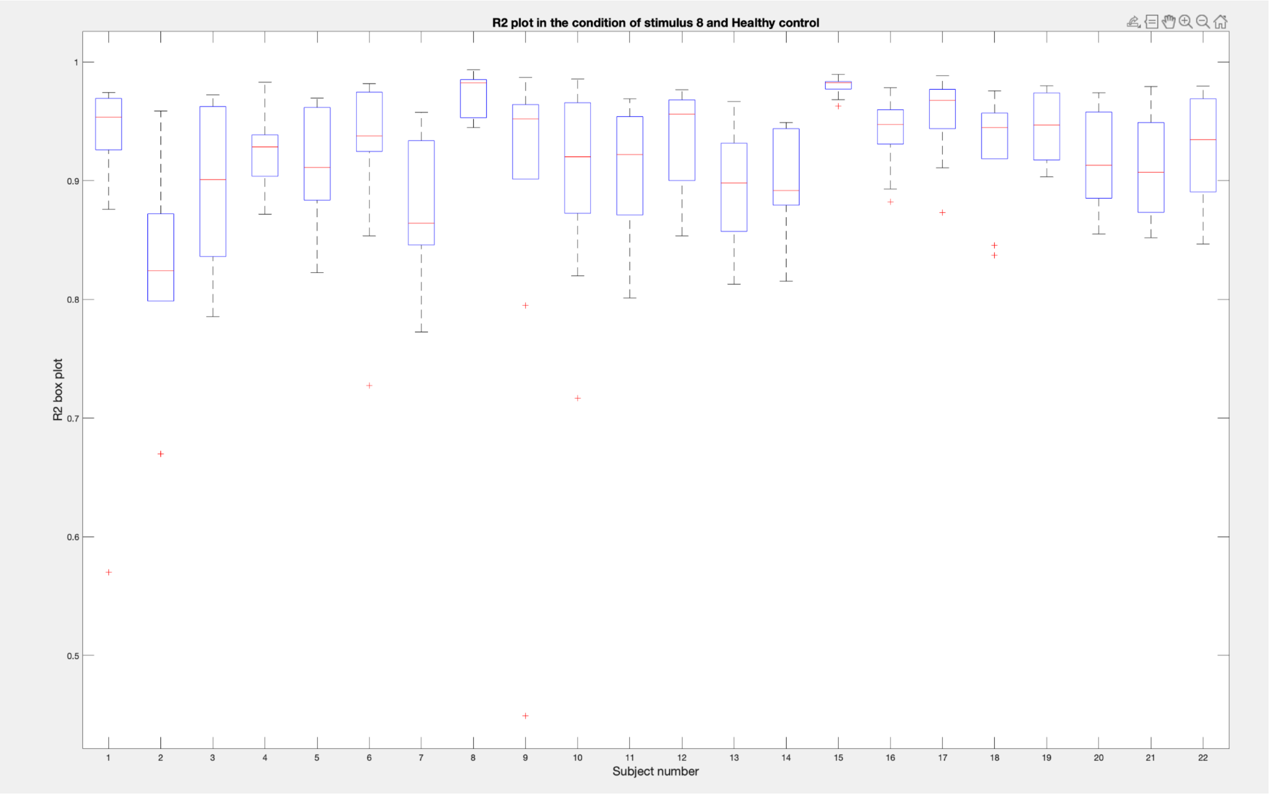
Stim7pd1



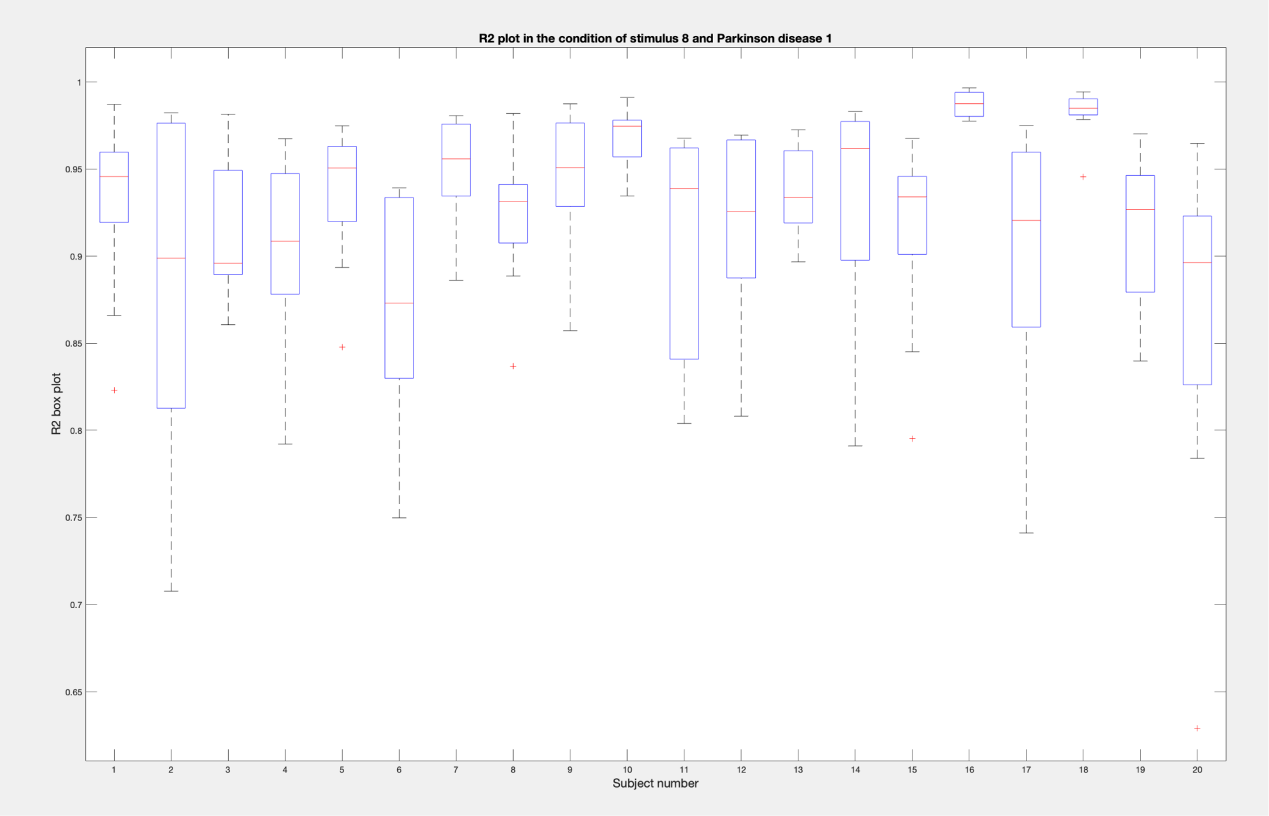
Stim7pd2



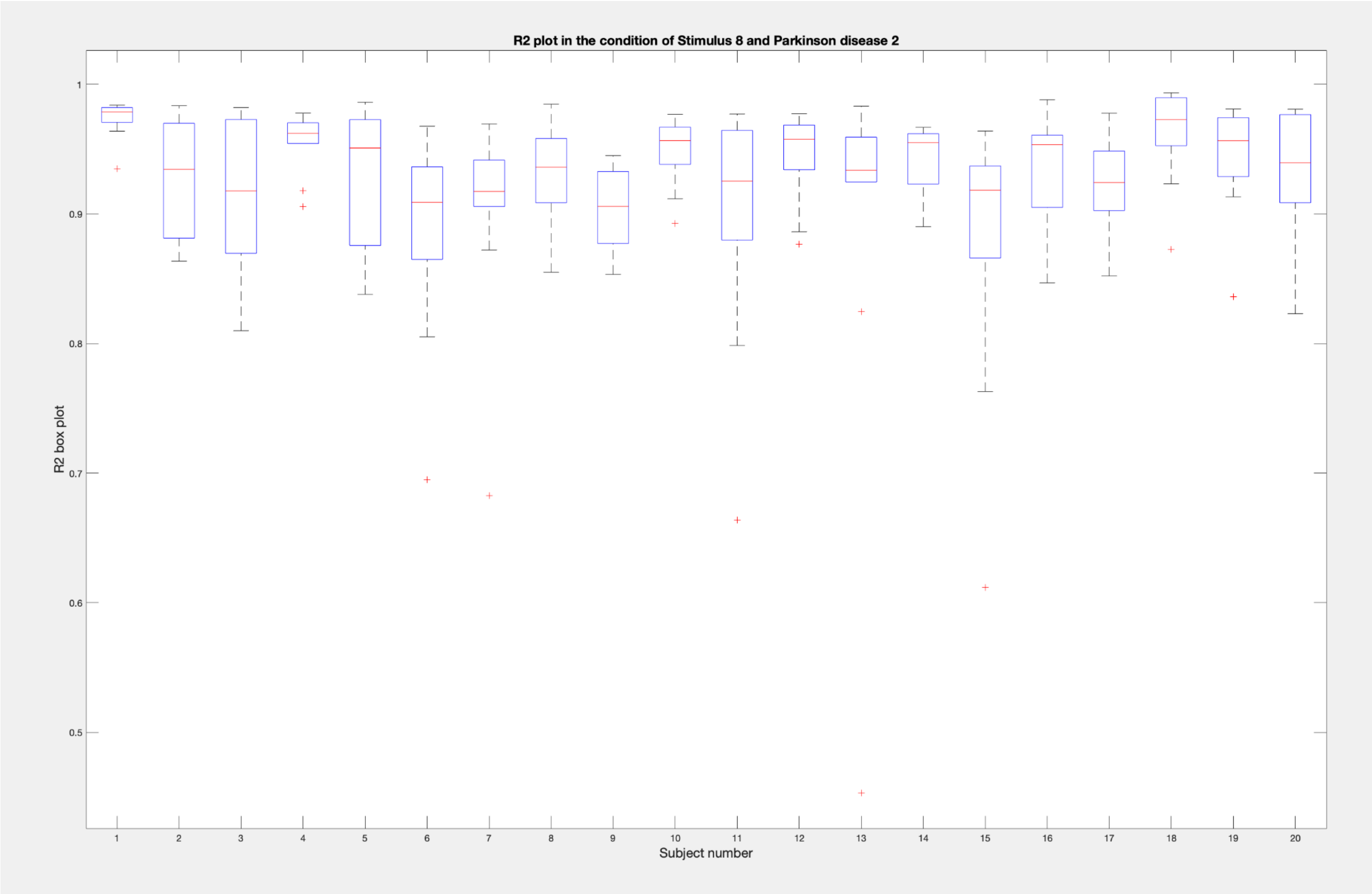
Stim8hc



Stim8pd1

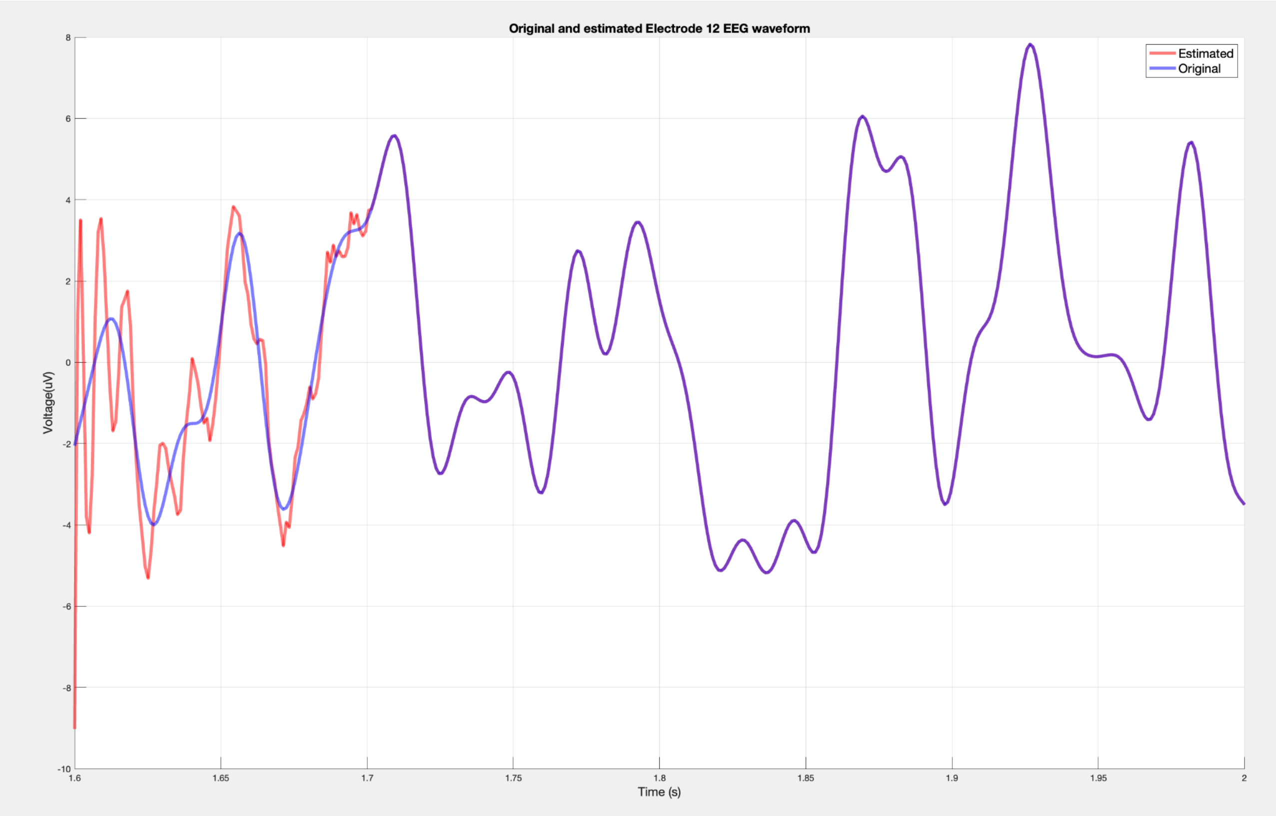


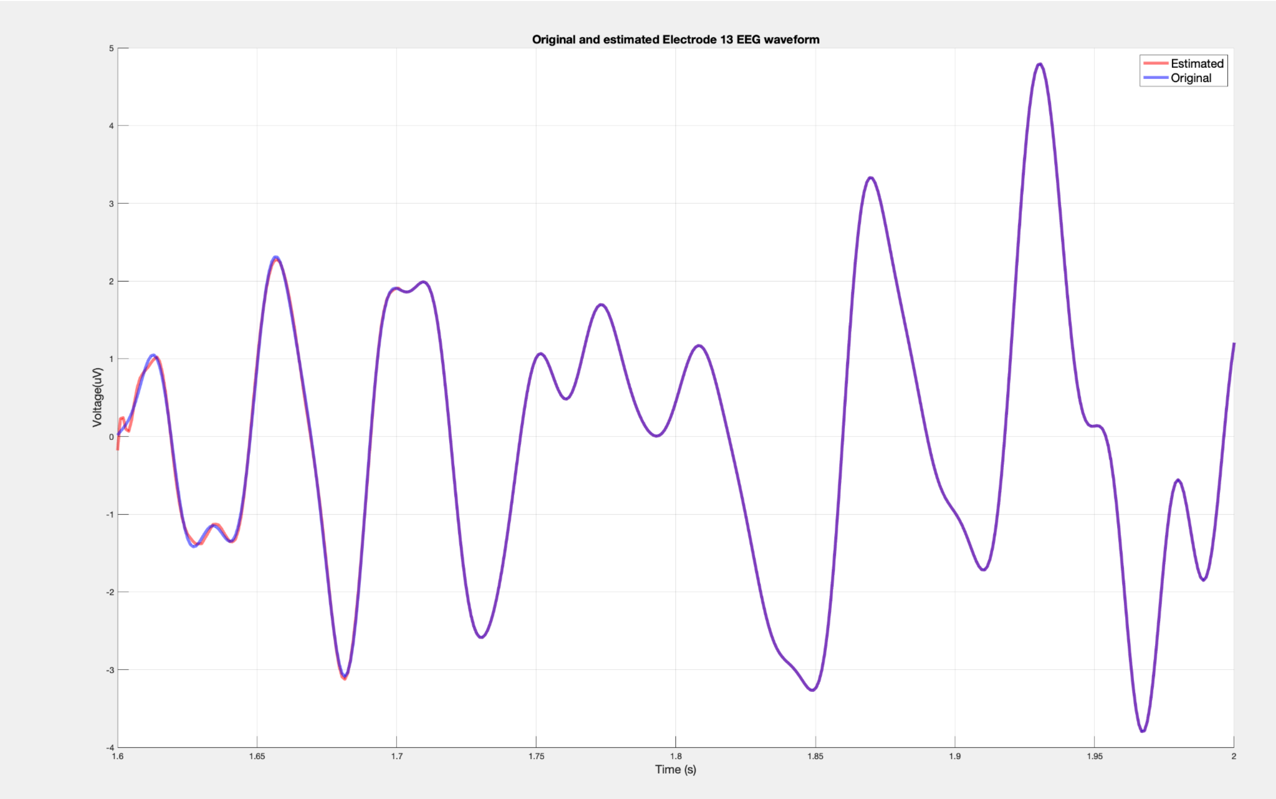
Stim8pd2

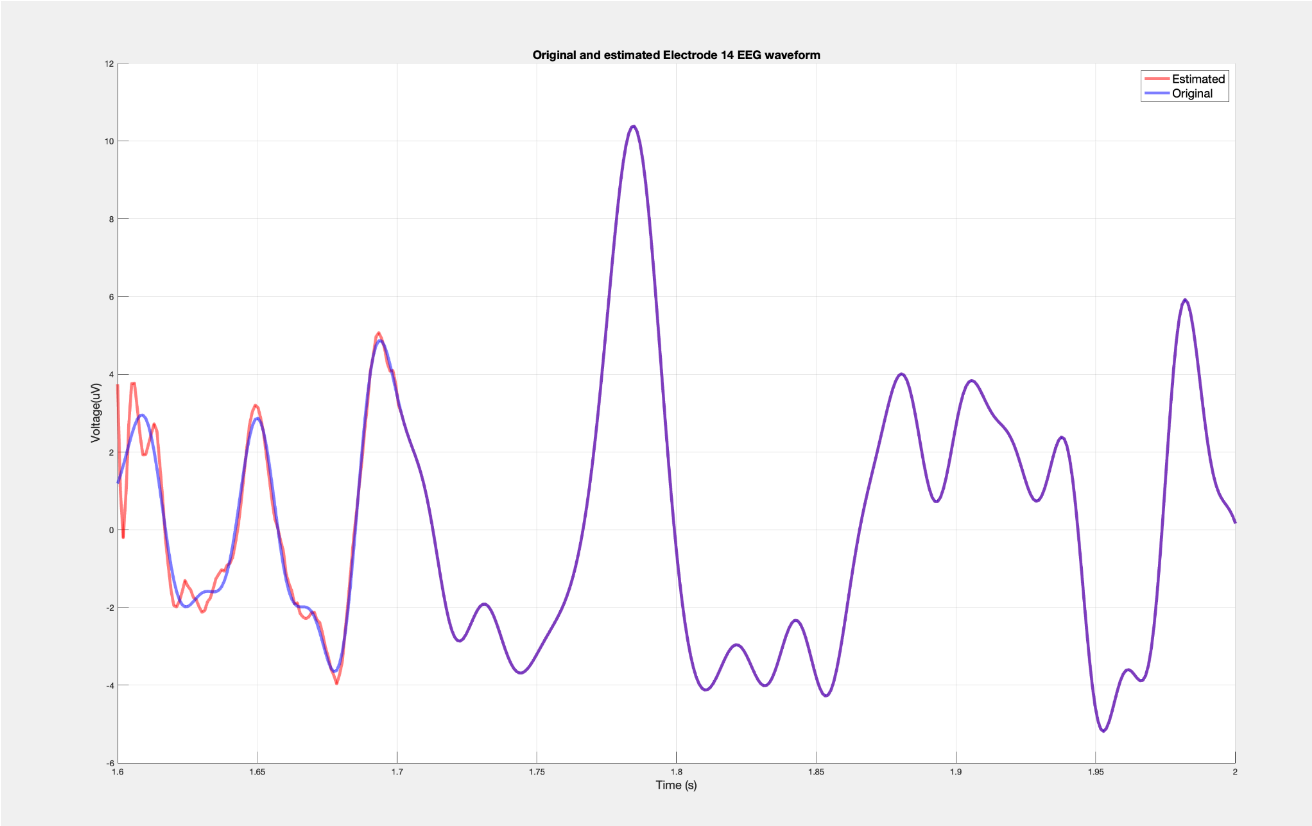


R2 plot in the condition of Stimulus 8 and Parkinson disease 2
R2 box plot
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Subject number

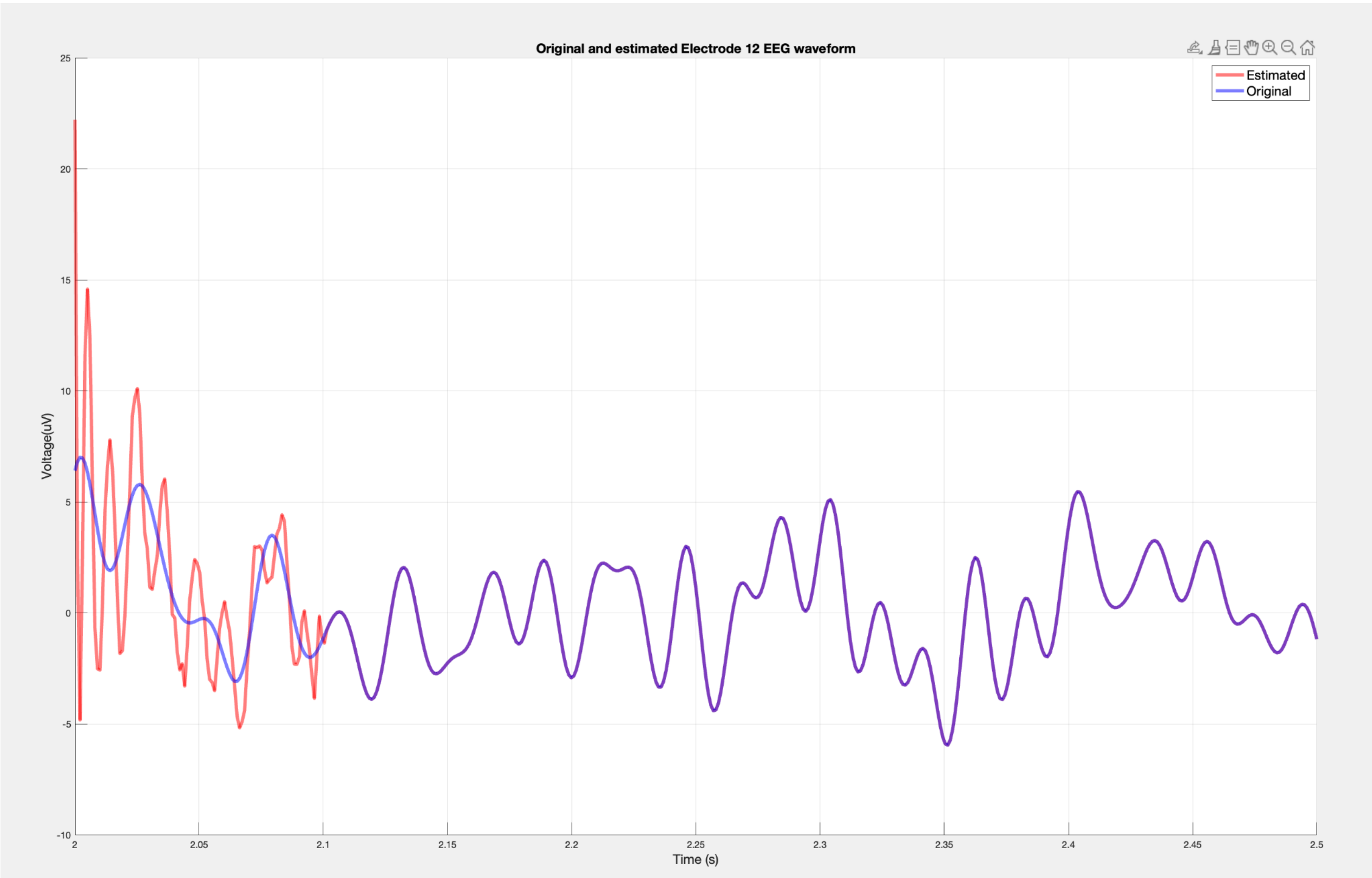
To visually see how estimated Y $(\widehat{Y})$ and original Y can fit together, We plot each of them for electrodes 12 13 14. **For Subject 1 - shamData and Healthy - Trial 1**

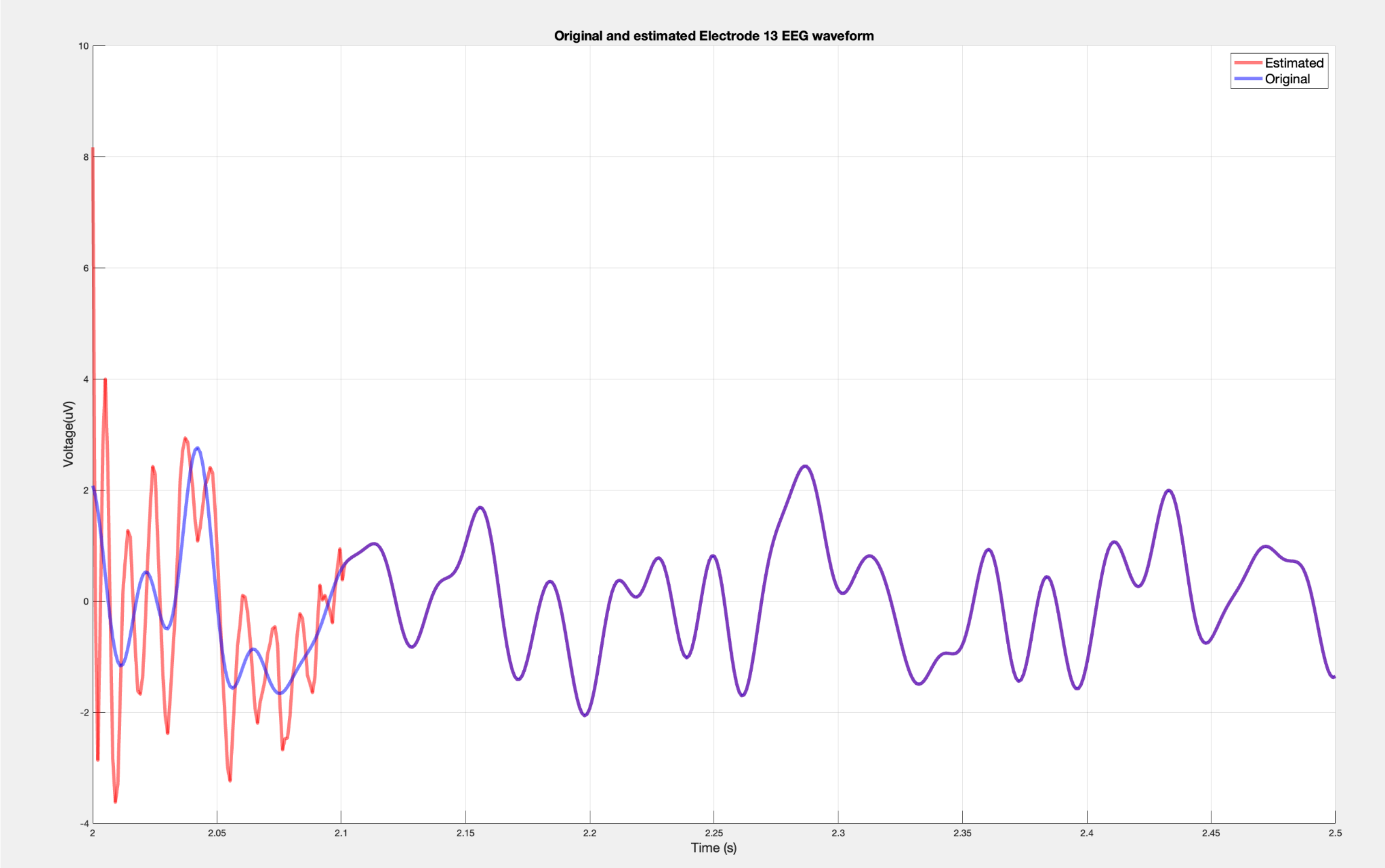


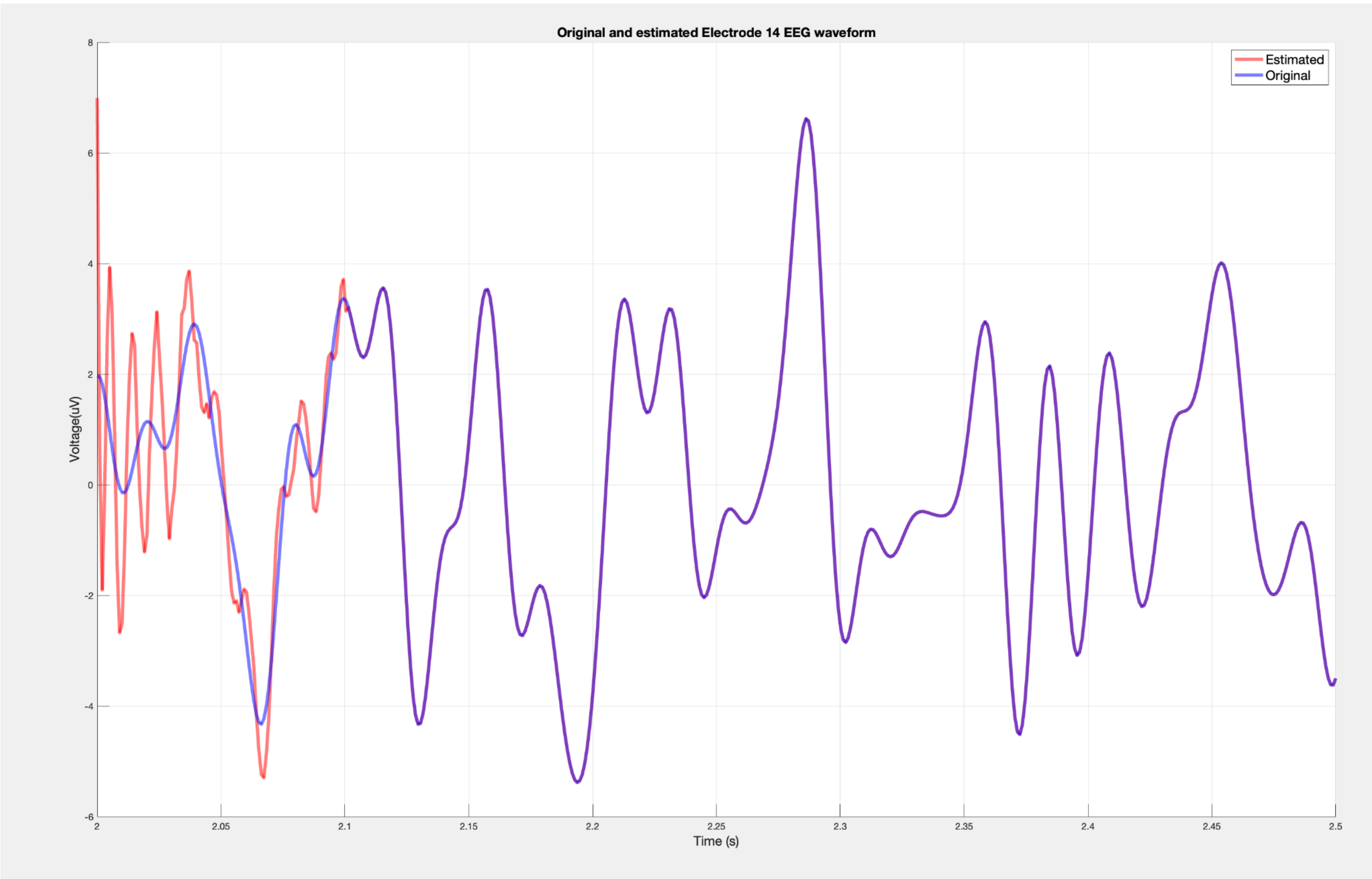




And Also for Stim8 - Subj5 - pd2 - trial 3

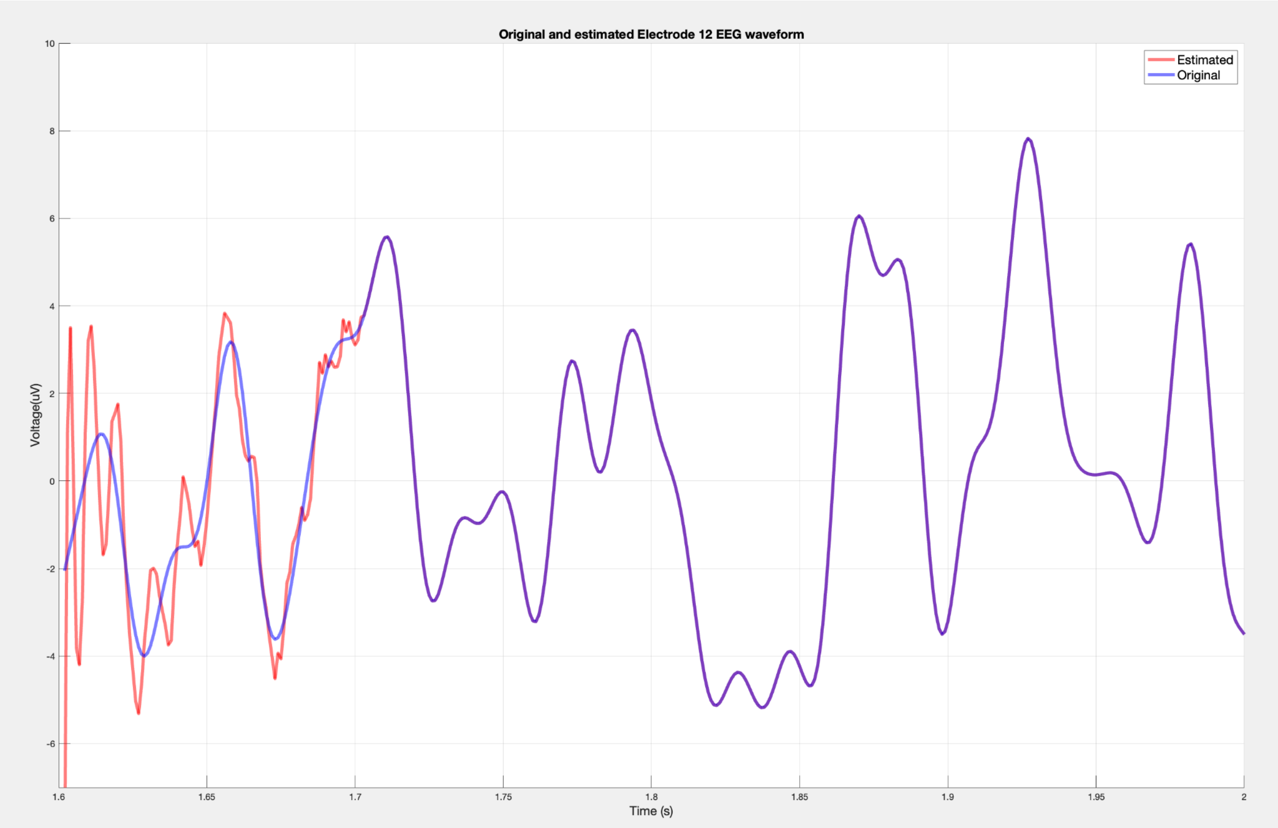



Original and estimated Electrode 13 EEG waveform
Estimated
Original
Voltage(uV)
Time (s)
10
8
6
4
2
0
-2
-4
2
2.05
2.1
2.15
2.2
2.25
2.3
2.35
2.4
2.45
2.5

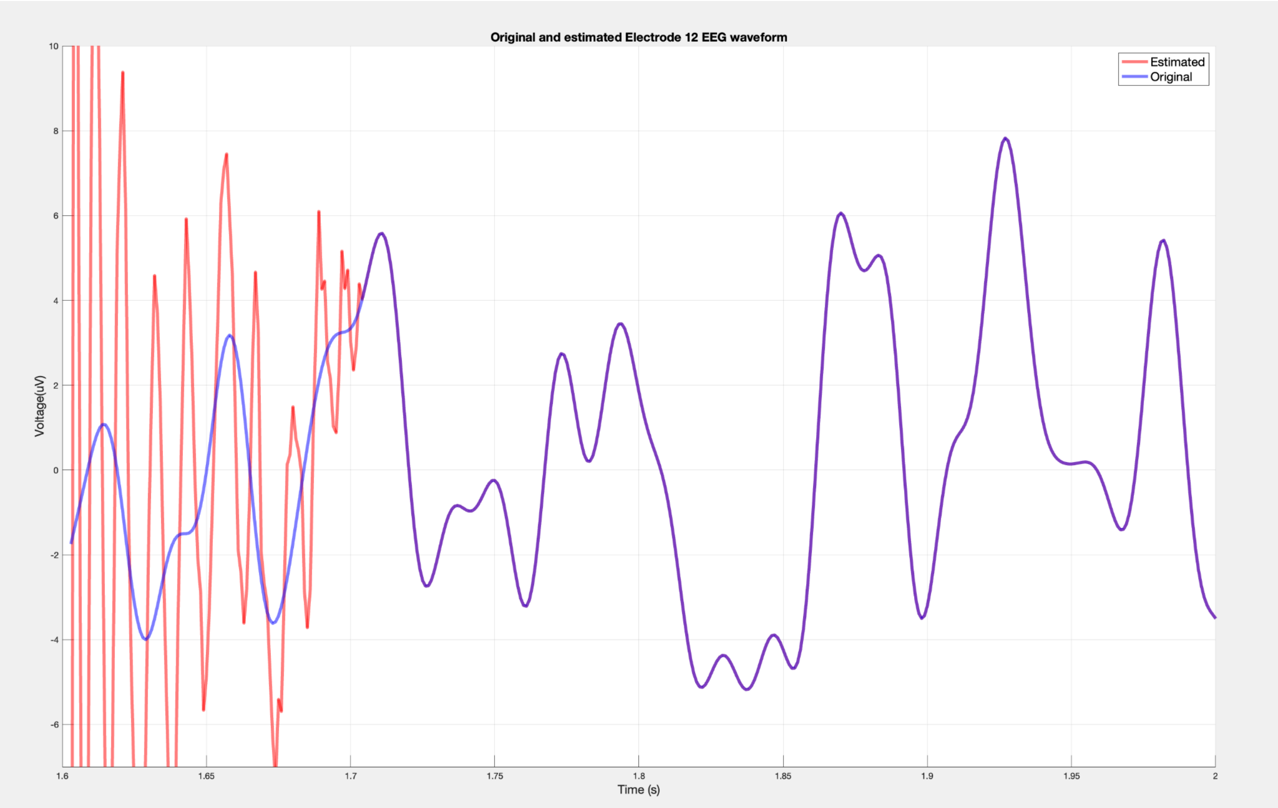

Original and estimated Electrode 14 EEG waveform
Estimated
Original
Voltage(uV)
Time (s)
8
6
4
2
0
-2
-4
-6
2
2.05
2.1
2.15
2.2
2.25
2.3
2.35
2.4
2.45
2.5

Multi Step Prediction

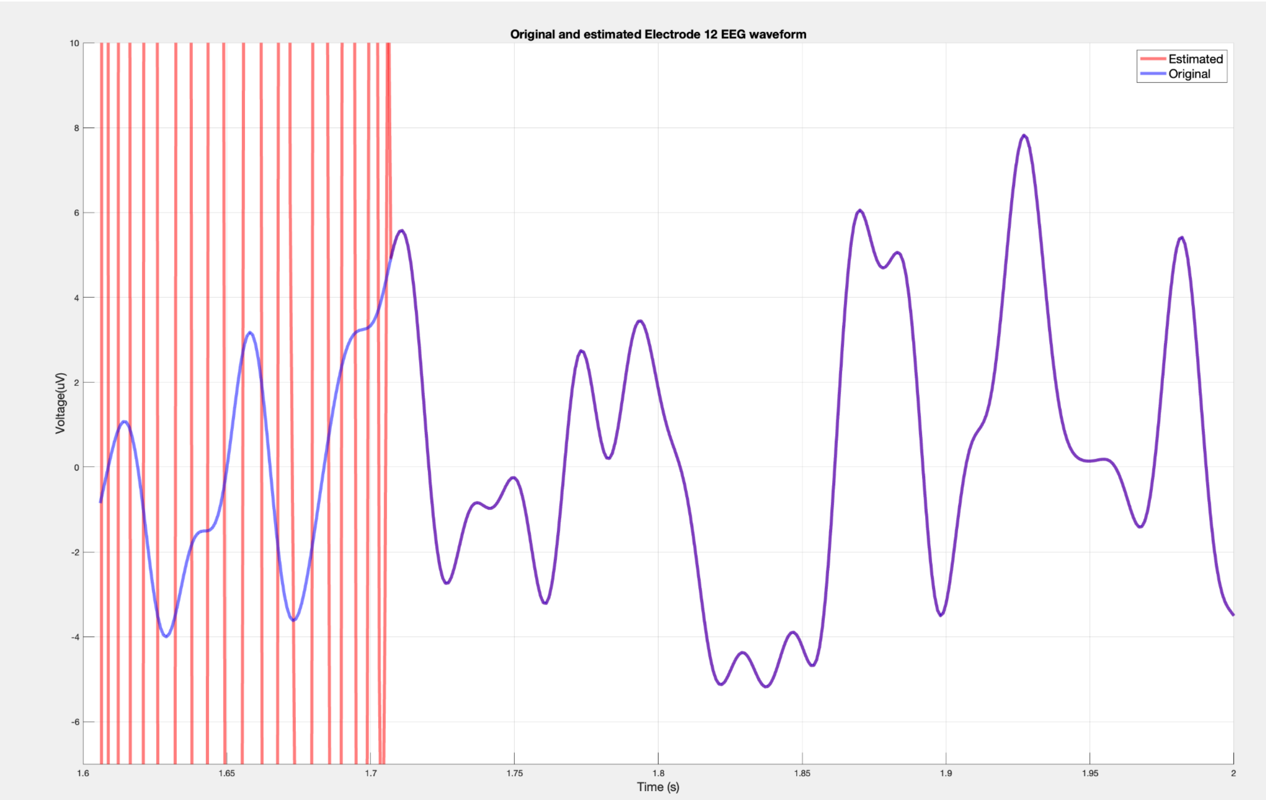
1 step ahead prediction:



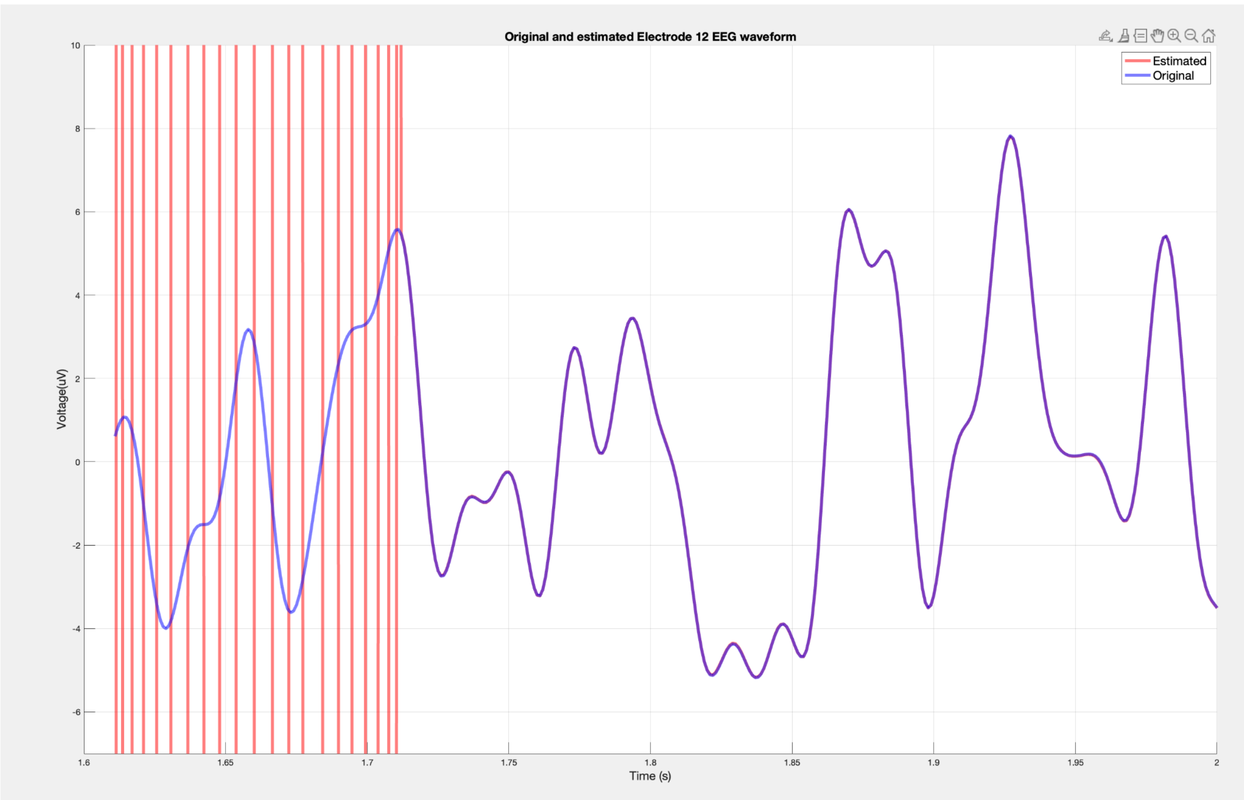
2 step ahead prediction:



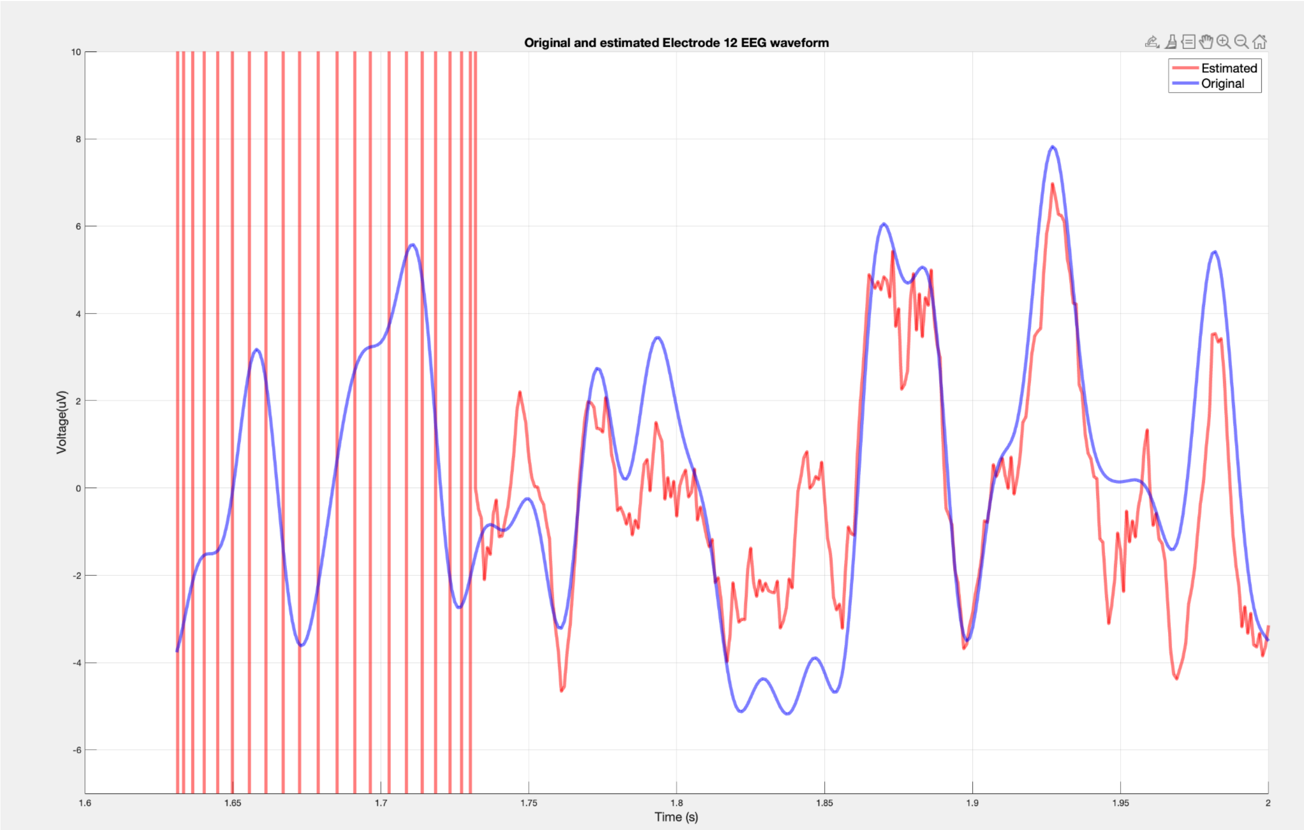
5 Step ahead:



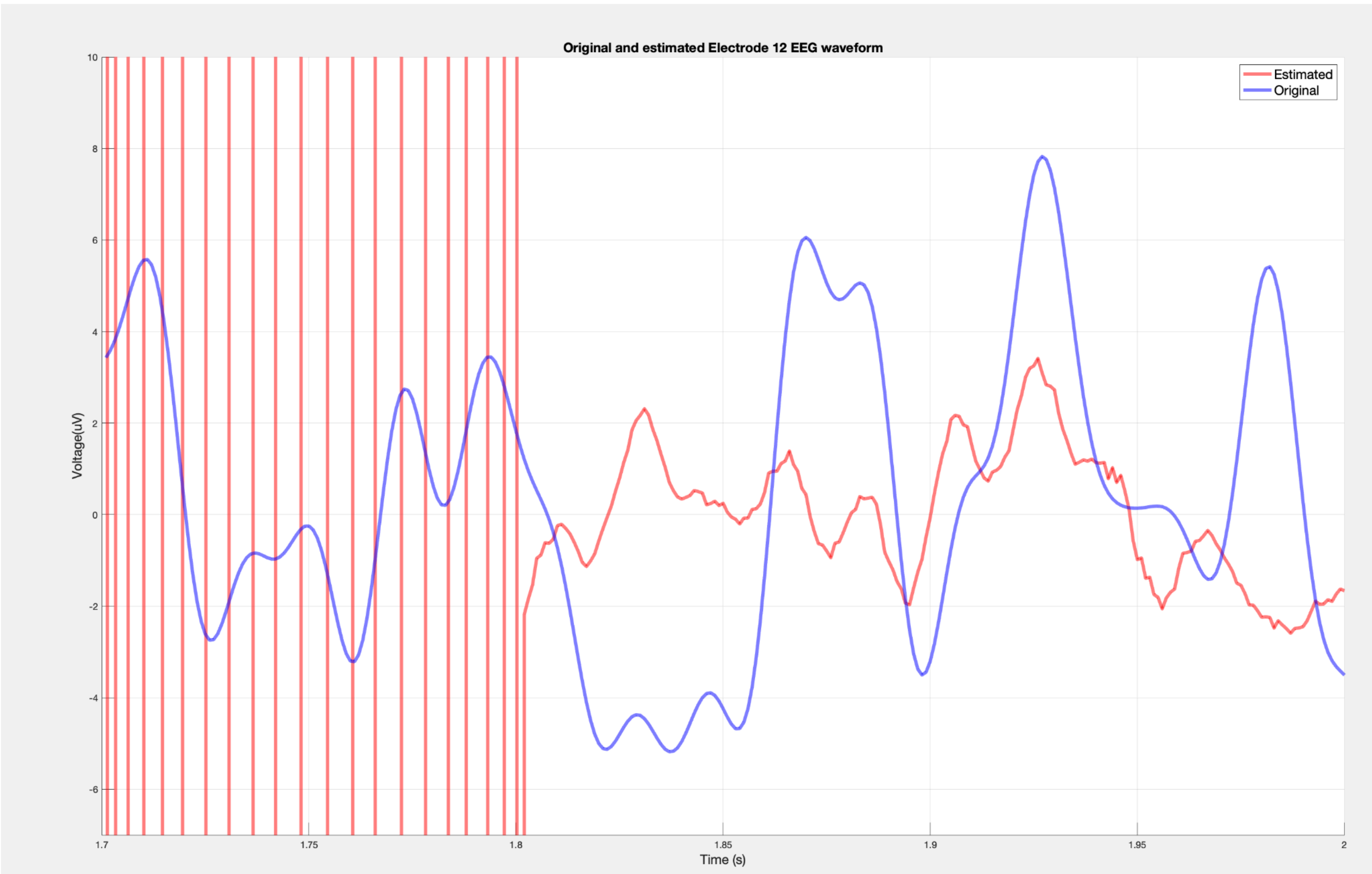
10 step ahead:

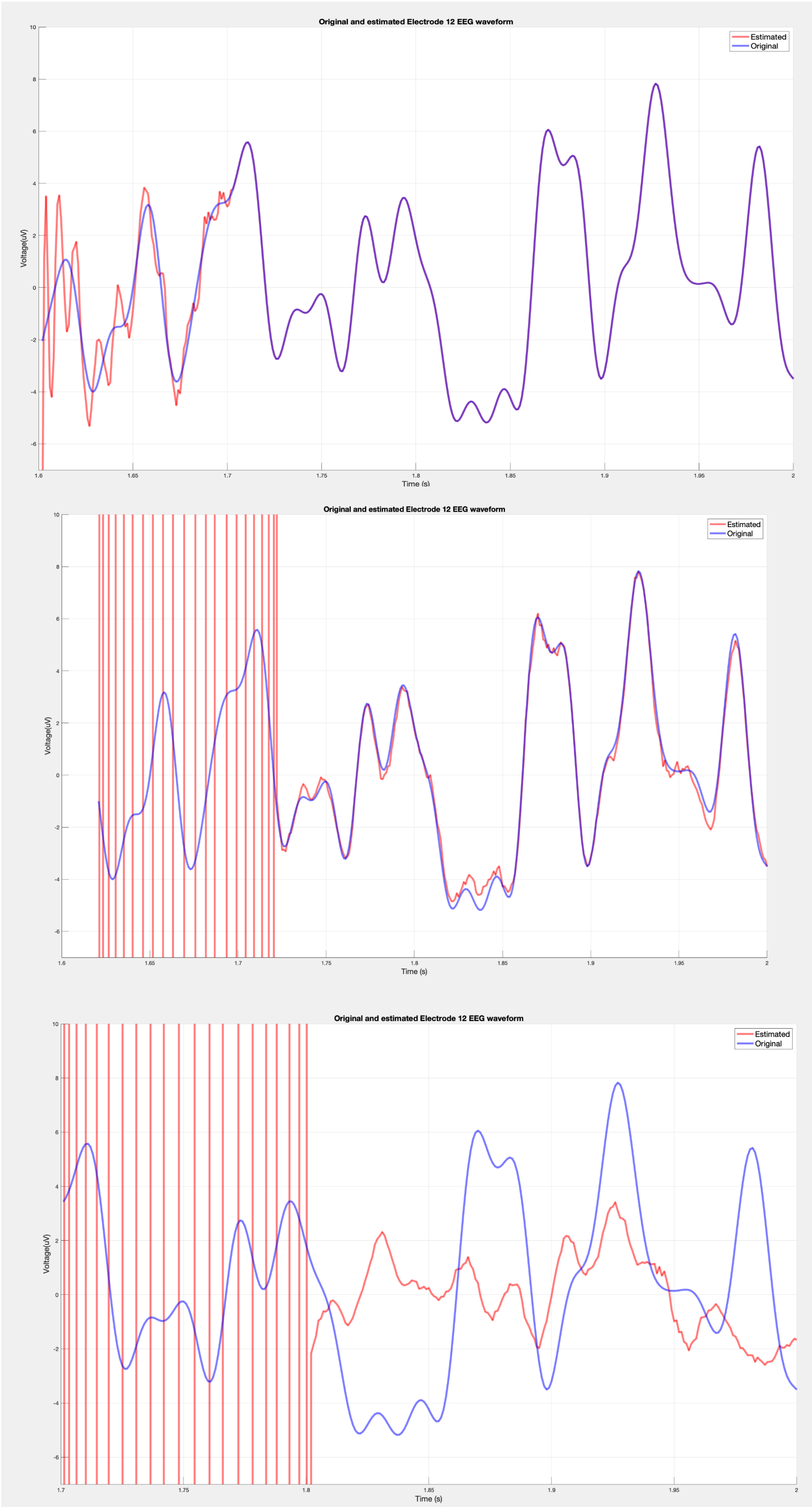


30 step ahead:



100 step ahead




Original and estimated Electrode 12 EEG waveform
Estimated
Original
Voltage(uV)
Time (s)
Original and estimated Electrode 12 EEG waveform
Estimated
Original
Voltage(uV)
Time (s)
Original and estimated Electrode 12 EEG waveform
Estimated
Original
Voltage(uV)
Time (s)

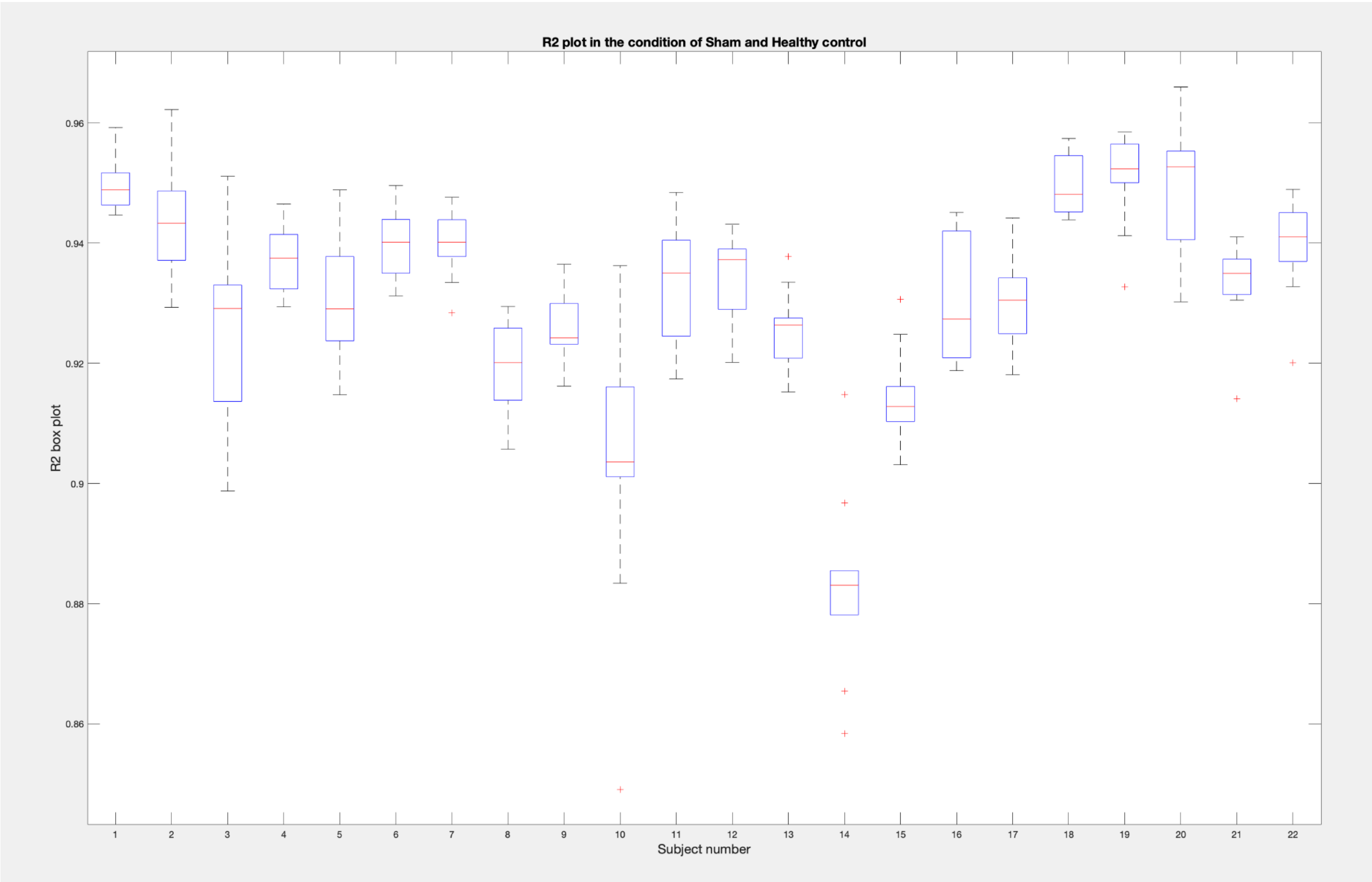
There is a trade off between High number of lags and low numbers. When the number of lag is high from the beginning of the test section to Number_of_lags our material for prediction (which is lagged time series) we have a great error which is because our model put zero in the incomplete part of the lagged series which makes our prediction with bug error. By the way, after the steps equal to Number of lags our predictor does an EEG forecast in a perfect way.

But when the number of lags is low (for example 1 or 2) then we don't have initial error because we have our lagged time series in every time points but this make some error in every time points because we have to do forecast using only 2 lagged series (y(t-1) and y(t-2)) so our prediction will be less accurate.

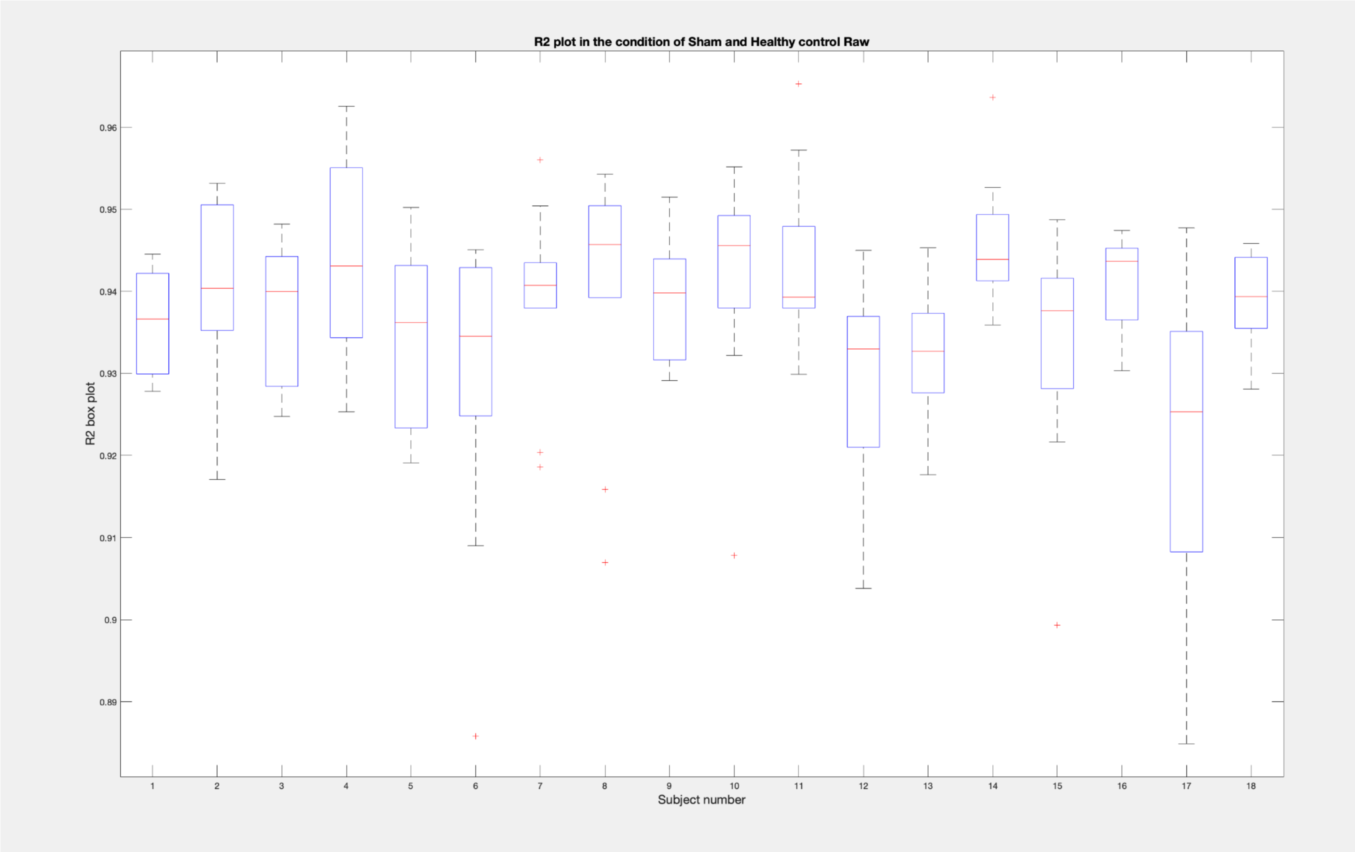
This will make us curious about which model (Based on sparsity pattern and number of lags) would be great in the purpose we want to search?

Maybe the model with 1 step ahead prediction - Dense (w_mask=full)

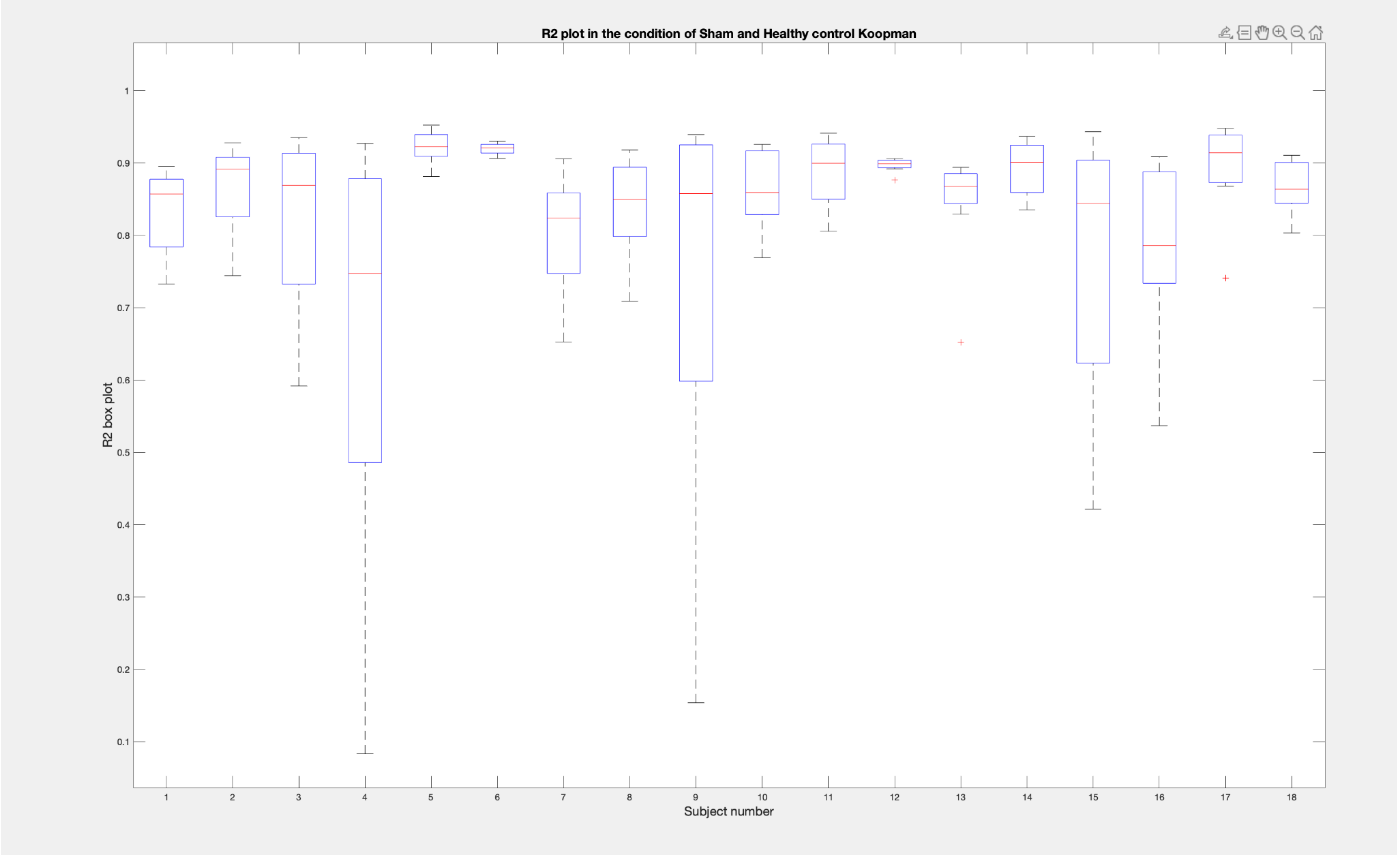
The W_mask parameter in the linear_AR function specifies the sparsity pattern for the weight matrix W W in the autoregressive (AR) model. The W_mask can take several forms, each dictating a different approach to selecting which weights to include in the model.

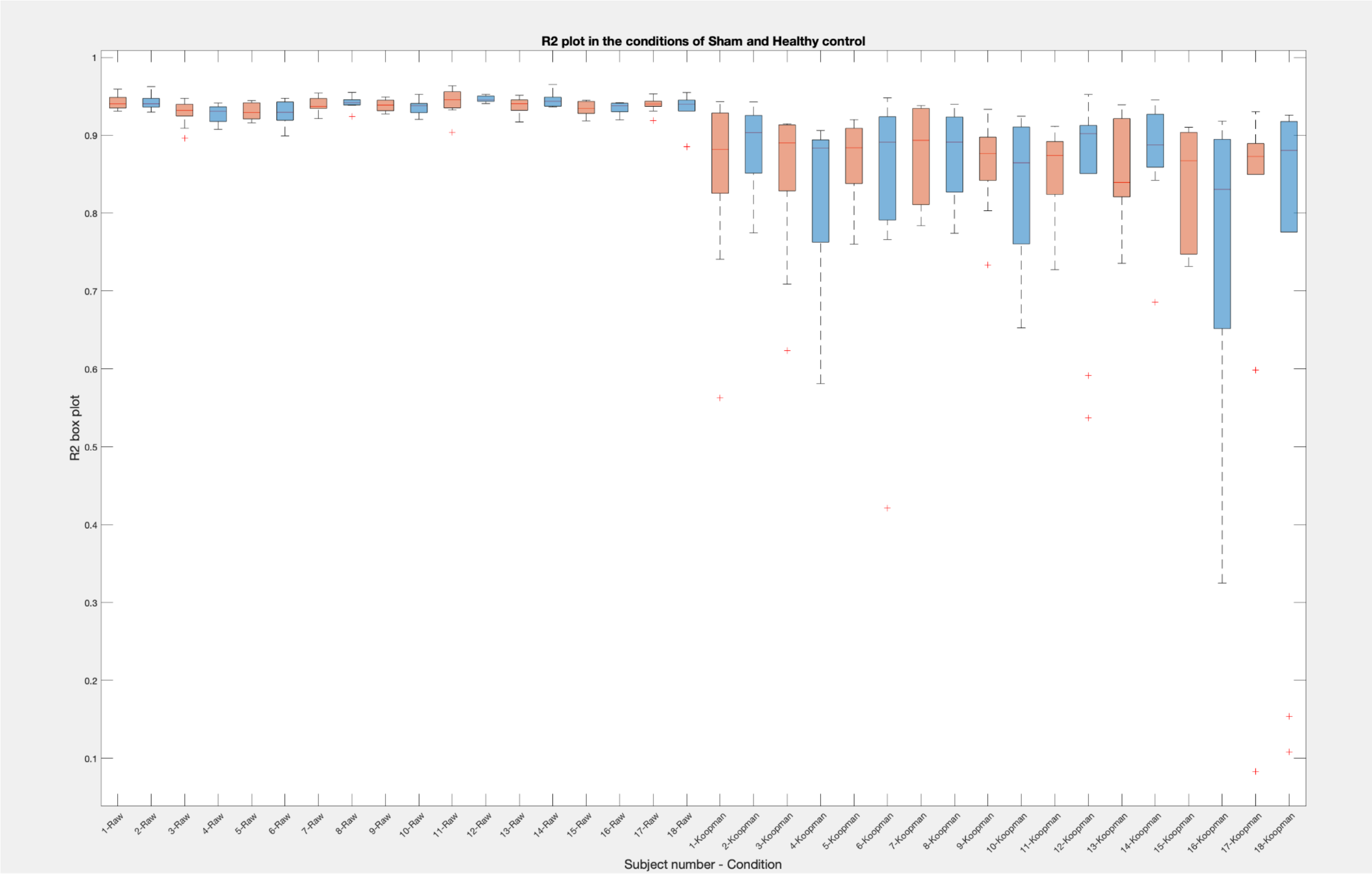


Raw


R2 plot in the condition of Sham and Healthy control Raw
R2 box plot
Subject number
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0.89
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Koopman


R2 plot in the condition of Sham and Healthy control Koopman
R2 box plot
Subject number
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18


R2 plot in the conditions of Sham and Healthy control
R2 box plot
Subject number - Condition
1-Raw
2-Raw
3-Raw
4-Raw
5-Raw
6-Raw
7-Raw
8-Raw
9-Raw
10-Raw
11-Raw
12-Raw
13-Raw
14-Raw
15-Raw
16-Raw
17-Raw
18-Raw
1-Koopman
2-Koopman
3-Koopman
4-Koopman
5-Koopman
6-Koopman
7-Koopman
8-Koopman
9-Koopman
10-Koopman
11-Koopman
12-Koopman
13-Koopman
14-Koopman
15-Koopman
16-Koopman
17-Koopman
18-Koopman