Documenting Monitor

Minoo

2022-11-30

The Monitor data has been received on 18.08.2022 as zip file called Minoo, and is saved in this pathway C:\Users\PhysioUser\Desktop\PhD\Monitor A copy of the raw signals has been saved in Monitor\raw signals. then the signals are preprocessed and saved in 3 subfolders in Monitor\pre-processed signals:

* Filtered signals
* Nirs containing transients
* Transients

The code used to pre-process the raw NIRS signals is Monitor.m. It has been modified for each participant based on individual differences. the outcome is saved as mydata.txt.

Then transient\_extraction.m has been used to extract the transients and saved files as transient and filtered.

## Transients

### Extracting features

the transient\_feature\_extraction.m file has been used to extract the features from the transients. The outcome is saved as transient\_features\_MN. calculating\_nirs\_duration.m has been used to have the duration of NIRS in each participant. my\_code\_put\_together\_allfeatures.m has been used to gather all the features extracted from transients in a single .csv file called all\_transient\_features.csv.

### Optimization for xgboost parameters\_MRI

61 infants were recruited and NIRS was measured for them, when the NIRS had been pre-processed, because some files had to be separated to two, we had 65 files. MN0052 was excluded since he had the diagnosis of sepsis, and we could not say for definite that HIE was the cause of their encephalopathy. MN0030 and MN0041 were mild that had been cooled, so they do not have MRI outcome but could be included in the HIE group. For MRI outcome, we have 58 infants, and 62 files, since 4 babies had 2 files. We have averaged the values except for the time spent below 63%, which was added, and we have 58 rows in total.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. The transient features are saved as *transient for xgb\_MRI.csv* which is colab-friendly! The result is saved in optimization\_transient\_xgb.csv. The outcome used is *Composite outcome of barkovich MRI abnormality +/- death (0=normal, 1=abnormal)* which has been in the file called MRI outcome for xgb\_58ver.csv. The optimized values are highlighted in yellow.

### leave one out xgboost\_MRI

The code called transient\_xgboost.IPYNB is used for the leave one out xgboost method.the files used for this purpose are MRI outcome for xgb\_58ver.csv and transient for xgb\_MRI.csv. Then the matlab code auc\_CI\_sen\_spec.m has been used to calculate the 95% confidence interval of AUC, accuracy, sensitivity and specificity:

The AUC for transients with MRI outcome

| ID | prob\_MRI (0.01\_3\_0.7\_1\_2\_100) |
| --- | --- |
| AUC | 0.49166666666666597 |
| 95% CI | 0.3244 0.6560 |
| Accuracy | 0.62070000000000003 |
| Senistivity | 0.22220000000000001 |
| Specificity | 0.8 |

### MRI watershed outcome

We used the same parameters as MRI, the files transient for xgb\_MRI.csv, watershed outcome for xgb.58ver.csv, and transient\_xgboost.IPYNB, and the AUC was:

The AUC for transients with MRI\_watershed outcome

| ID | prob\_watershed (0.01\_3\_0.7\_1\_2\_100) |
| --- | --- |
| AUC | 0.40602582496413198 |
| 95% CI | 0.2500 0.5711 |
| Accuracy | 0.53449999999999998 |
| Senistivity | 0 |
| Specificity | 0.75609999999999999 |

### MRI gray injury outcome

We used the same parameters as MRI, the files transient for xgb\_MRI.csv, gray injury outcome for xgb.58ver.csv, and transient\_xgboost.IPYNB, and the AUC was:

The AUC for transients with MRI\_gray injury outcome

| ID | prob\_gray matter (0.01\_3\_0.7\_1\_2\_100) |
| --- | --- |
| AUC | 0.56410256410256399 |
| 95% CI | 0.1991 0.8909 |
| Accuracy | 0.89659999999999995 |
| Senistivity | 0 |
| Specificity | 1 |

### Optimization for xgboost parameters\_HIE

61 infants were recruited and NIRS was measured for them, when the NIRS had been pre-processed, because some files had to be separated to two, we had 65 files. MN0052 was excluded since he had the diagnosis of sepsis, and we could not say for definite that HIE was the cause of their encephalopathy. MN0030 and MN0041 were mild that had been cooled, and were included in the HIE group with non-injury label. For HIE outcome, we have 60 infants in total, some files are separated,so we have 64 files, but at the end the features are added and averaged for babies with more than one file and we have 60 rows at the end.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. The transient features are saved as *transient for xgb\_HIE.csv* which is colab-friendly! The result is saved in optimization\_transient\_HIE\_xgb.csv. The outcome used is *Clinical Grade of HIE (0=mild, 1=moderate or severe)* which has been in the file called HIE outcome for xgb\_60ver.csv. The optimized values are highlighted in yellow.

### leave one out xgboost\_HIE

The code called transient\_xgboost.IPYNB is used for the leave one out xgboost method.the files used for this purpose are HIE outcome for xgb\_60ver.csv and and transient for xgb\_HIE.csv. the AUC is significant:

The AUC for transients with HIE outcome

| ID | prob HIE (0.1\_4\_0.7\_0\_1\_100) |
| --- | --- |
| AUC | 0.79888888888888898 |
| 95% CI | 0.6725 0.9071 |
| Accuracy | 0.75 |
| Senistivity | 0.73329999999999995 |
| Specificity | 0.76670000000000005 |

In the next level, we were wondering if the features extracted from the transients could predict the HIE level, so we used the same files, just in the HIE outcome for xgb\_60ver, instead of column 2, we used the HIE levels in column 3. for optimization and training the model, we used multiclass\_xgboost.ipynb python file. The optimized parameters are saved in optimization\_transient\_HIE\_grading.xlsx and the chosen parameters are highlighted in yellow.

Then, using the R code called “multiclass\_auc&ci”, and the probabilities saved in the file multiclass\_prob.xlsx, we calculated the AUC and 95% confidence interval.

The AUC and 95% CI for transient with HIE grading outcome

| model | transient\_HIE(0.1\_6\_0.8\_0\_2\_100) |
| --- | --- |
| AUC | 0.7374 |
| low CI | 0.6105 |
| high CI | 0.9133 |

### transient first 6 hours

The transients were separated for the first 6 hours after starting of recording, and the features were extracted from them. They had been saved in transient for xgb\_MRI\_6h.csv and transient for xgb\_HIE\_6h.csv, and then using MRI outcome for xgb\_58ver.csv and HIE outcome for xgb\_60ver.csv, they had been optimized. The optimizations are saved as optimization\_transient\_6h\_MRI, optimization\_transient\_6h\_HIE\_xgb and optimization\_transient\_6h\_HIE\_grading, and the optimized values are highlighted in yellow. The results are as followed:

The AUC for transients in first 6 hours

| ID | prob\_MRI\_6h(0.01\_3\_1\_0\_2\_100) | prob HIE\_6h (0.05\_3\_1\_1\_1\_100) |
| --- | --- | --- |
| AUC | 0.47222222222222199 | 0.77666666666666595 |
| 95% CI | 0.3000 0.6513 | 0.6457 0.8891 |
| Accuracy | 0.6552 | 0.7 |
| Senistivity | 0.22220000000000001 | 0.66669999999999996 |
| Specificity | 0.85 | 0.73329999999999995 |

The AUC and 95% CI for transient in first 6 hours

| model | transient\_HIE\_6h |
| --- | --- |
| AUC | 0.7207 |
| low CI | 0.5841 |
| high CI | 0.9051 |

## NIRS signal containing tarnsients

### Extracting features

the nirs\_feature\_extraction.m file has been used to decompose the signal into 5 bandwidths, divide them into epochs of 4 hours with 2 hours of overlapping and extract the features from each epoch in each bandwidth of the NIRS signal that still contains transients. The outcome is saved as nirs\_features\_MN. The extracted features are:

* mean envelope (\*5)
* standard deviation envelope (\*5)
* signal kurtosis (\*5)
* signal skewness (\*5)
* 5th percentile of envelope (\*5)
* 95th percentile of envelope (\*5)
* mean instantaneous frequency (\*5)
* standard deviation instantaneous frequency (\*5)
* kurtosis instantaneous frequency (\*5)
* skewness instantaneous frequency (\*5)
* 5th percentile instantaneous frequency (\*5)
* 95th percentile instantaneous frequency (\*5)
* fractal dimension (\*5)
* the energy of signal in 5th percentile of the signal duration
* the energy of signal in 95th percentile of the signal duration

This gives us 67 features in total, but we will not include the energy of the signal in the 5th and 95th percentile of the signal duration.

my\_code\_put\_together\_allfeatures\_nirs.m has been used to gather all the features extracted from NIRS in a single .csv file called all\_nirs\_features.csv.

### Optimization for NIRS xgboost parameters\_MRI

Same as what was explained in *optimization for xgboost parameters\_MRI*, we had 62 files at the end.Since each NIRS recording has different duration, we had different number of epochs for each infant. the colab friendly file containing the info for these 62 files is saved as *nirs with MRI outcome for xgb*. it is a 1505\*66 matrix. The first column contains the IDs and columns 2-66 correspond to the 65 features for each epoch. The outcome is saved as MRI outcome for xgb\_nirs.csv.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. MRI outcome for xgb\_58ver.csv has been used for getting AUC. The optimized values are saved in optimization\_nirs\_MRI\_xgb.xlsx and the chosen values are highlighted in yellow.

### leave one out xgboost\_MRI\_NIRS

The code called transient\_xgboost.IPYNB is used for the leave one out xgboost method.the files used for this purpose are nirs with MRI outcome for xgb.csv, MRI outcome for xgb\_nirs.csv and MRI outcome for xgb\_58ver.csv (the last one used for getting AUC). The AUC is significant:

The AUC for NIRS with MRI outcome

| ID | prob\_MRI\_NIRS(0.1\_3\_0.9\_0\_1\_100) |
| --- | --- |
| AUC | 0.69861111111111096 |
| 95% CI | 0.5511 0.8236 |
| Accuracy | 0.6724 |
| Senistivity | 0.1111 |
| Specificity | 0.92500000000000004 |

### MRI watershed outcome

with not optimized parameters, the files nirs with MRI ourcome for xgb\_MRI.csv, watershed outcome for xgb\_nirs.csv, and transient\_xgboost.IPYNB, the AUC was:

The AUC for NIRS with MRI\_watershed outcome

| ID | prob\_watershed\_nirs(0.05\_3\_0.8\_0\_1\_500) |
| --- | --- |
| AUC | 0.67718794835007101 |
| 95% CI | 0.5305 0.8057 |
| Accuracy | 0.6724 |
| Senistivity | 5.8799999999999998E-2 |
| Specificity | 0.92679999999999996 |

### MRI gray injury outcome

with not optimized parameters, the files nirs with MRI ourcome for xgb\_MRI.csv, gray injury outcome for xgb\_nirs.csv, and transient\_xgboost.IPYNB, the AUC was:

The AUC for NIRS with MRI\_gray injury outcome

| ID | prob\_gray matter\_nirs(0.05\_3\_0.8\_0\_1\_500) |
| --- | --- |
| AUC | 0.88461538461538403 |
| 95% CI | 0.7356 0.9887 |
| Accuracy | 0.91379999999999995 |
| Senistivity | 0.16669999999999999 |
| Specificity | 1 |

### Optimization for NIRS xgboost parameters\_HIE

Same as what was explained in *optimization for xgboost parameters\_HIE*, we had 64 files at the end.Since each NIRS recording has different duration, we had different number of epochs for each infant. the colab friendly file containing the info for these 64 files is saved as *nirs with HIE outcome for xgb*. it is a 1549\*66 matrix. The first column contains the IDs and columns 2-66 correspond to the 65 features for each epoch. The outcome is saved as HIE outcome for xgb\_nirs.csv.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. HIE outcome for xgb\_60ver.csv has been used for getting AUC. The optimized values are saved as optimization\_nirs\_HIE\_xgb.xlsx and chosen parameters are highlighted in yellow.

### leave one out xgboost\_HIE\_NIRS

The code called transient\_xgboost.IPYNB is used for the leave one out xgboost method.the files used for this purpose are nirs with HIE outcome for xgb.csv, HIE outcome for xgb\_nirs.csv and HIE outcome for xgb\_60ver.csv (the last one used for getting AUC). The AUC is significant:

The AUC for NIRS with HIE outcome

| ID | prob\_HIE\_NIRS(0.1\_5\_0.7\_0\_1\_100) |
| --- | --- |
| AUC | 0.818888888888888 |
| 95% CI | 0.7025 0.9237 |
| Accuracy | 0.6 |
| Senistivity | 0.9667 |
| Specificity | 0.23330000000000001 |

In the next level, we were wondering if the features extracted from the NIRS could predict the HIE level, so we used the same files, just in the HIE outcome for xgb\_nirs, instead of column 2, we used the HIE levels in column 3. for optimization and training the model, we used multiclass\_xgboost.ipynb python file. The optimized parameters are saved in optimization\_nirs\_HIE\_grading.xlsx and the chosen parameters are highlighted in yellow.

Then, using the R code called “multiclass\_auc&ci”, and the probabilities saved in the file multiclass\_prob.xlsx, we calculated the AUC and 95% confidence interval.

The AUC and 95% CI for nirs with HIE grading outcome

| model | nirs\_HIE(0.05\_6\_1\_0\_1\_500) |
| --- | --- |
| AUC | 0.7626 |
| low CI | 0.6279 |
| high CI | 0.8934 |

### NIRS first 6 hours

The NIRS were separated for the first 6 hours after starting of recording (the first 2 epochs). They had been saved in nirs with MRI outcome for xgb\_6h.csv and nirs with HIE outcome for xgb\_6h.csv, and then using MRI outcome for xgb\_58ver.csv, HIE outcome for xgb\_60ver.csv, MRI outcome for xgb\_nirs\_6h, and HIE outcome for xgb\_nirs\_6h, they had been optimized. The optimizations are saved as optimization\_nirs\_6h\_MRI, optimization\_nirs\_6h\_HIE\_xgb and optimization\_nirs\_6h\_HIE\_grading, and the optimized values are highlighted in yellow. The results are as followed:

The AUC for nirs in first 6 hours

| ID | prob\_MRI\_NIRS\_6h(0.01\_4\_0.7\_1\_1\_100) | prob\_HIE\_NIRS\_6h(0.05\_3\_0.8\_0\_2\_100) |
| --- | --- | --- |
| AUC | 0.48611111111111099 | 0.66555555555555501 |
| 95% CI | 0.3165 0.6691 | 0.5204 0.8084 |
| Accuracy | 0.6552 | 0.6 |
| Senistivity | 0.16669999999999999 | 0.6 |
| Specificity | 0.875 | 0.6 |

The AUC and 95% CI for nirs in first 6 hours

| model | nirs\_HIE\_6h |
| --- | --- |
| AUC | 0.6548 |
| low CI | 0.5414 |
| high CI | 0.7615 |

## Filtered NIRS signal

### Extracting features

the nirs\_feature\_extraction.m file has been used to decompose the signal into 5 bandwidths, divide them into epochs of 4 hours with 2 hours of overlapping and extract the features from each epoch in each bandwidth of the Filtered signal. The outcome is saved as filtered\_features\_MN.txt. The extracted features same as extracted features from the NIRS signals containing transients. The code my\_code\_put\_together\_allfeatures\_nirs.m has been used to put all the features in a single file called all\_filtered\_features.csv.

### Optimization for filteted xgboost parameters\_MRI

Same as what was explained in *optimization for xgboost parameters\_MRI*, we had 62 files at the end.Since each NIRS recording has different duration, we had different number of epochs for each infant. the colab friendly file containing the info for these 62 files is saved as *filtered with MRI outcome for xgb*. it is a 1505\*66 matrix. The first column contains the IDs and columns 2-66 correspond to the 65 features for each epoch. The outcome is saved as MRI outcome for xgb\_nirs.csv.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. MRI outcome for xgb\_58ver.csv has been used for getting AUC. The optimized values are saved in optimization\_filtered\_MRI\_xgb.xlsx and the chosen values are highlighted in yellow.

### Leave one out xgboost\_MRI\_filtered

For the MRI outcome, and using the same code transient\_xgboost.IPYNB, and the files called filtered with MRI outcome for xgb.csv, MRI outcome for xgb\_nirs.csv and MRI outcome for xgb\_58ver.csv have been used for training and testing the model, and this is the result:

The AUC for Filtered with MRI outcome

| ID | prob\_MRI\_filtered(0.1\_3\_0.8\_0\_2\_100) |
| --- | --- |
| AUC | 0.72222222222222199 |
| 95% CI | 0.5795 0.8486 |
| Accuracy | 0.70689999999999997 |
| Senistivity | 0.22220000000000001 |
| Specificity | 0.92500000000000004 |

### MRI watershed outcome

with not optimized parameters, the files filtered with MRI ourcome for xgb\_MRI.csv, watershed outcome for xgb\_nirs.csv, and transient\_xgboost.IPYNB, the AUC was:

The AUC for filtered with MRI\_watershed outcome

| ID | prob\_watershed\_filtered(0.05\_3\_0.8\_0\_1\_500) |
| --- | --- |
| AUC | 0.72596843615494899 |
| 95% CI | 0.5795 0.8525 |
| Accuracy | 0.74139999999999995 |
| Senistivity | 0.17649999999999999 |
| Specificity | 0.97560000000000002 |

### MRI gray injury outcome

with not optimized parameters, the files filtered with MRI ourcome for xgb\_MRI.csv, gray injury outcome for xgb\_nirs.csv, and transient\_xgboost.IPYNB, the AUC was:

The AUC for filtered with MRI\_gray injury outcome

| ID | prob\_gray matter\_filtered(0.05\_3\_0.8\_0\_1\_500) |
| --- | --- |
| AUC | 0.90064102564102499 |
| 95% CI | 0.7500 1.0000 |
| Accuracy | 0.93100000000000005 |
| Senistivity | 0.33329999999999999 |
| Specificity | 1 |

### Optimization for filtered xgboost parameters\_HIE

Same as what was explained in *optimization for xgboost parameters\_HIE*, we had 64 files at the end.Since each NIRS recording has different duration, we had different number of epochs for each infant. the colab friendly file containing the info for these 64 files is saved as *filtered with HIE outcome for xgb*. it is a 1549\*66 matrix. The first column contains the IDs and columns 2-66 correspond to the 65 features for each epoch. The outcome is saved as HIE outcome for xgb\_nirs.csv.

Then a leave-one-out method optimization has been done to find the best values for xgboost method. the python code used is optimization.IPYNB. HIE outcome for xgb\_nirs.csv has been used for getting AUC. The optimized values are saved as optimization\_filtered\_HIE\_xgb.xlsx and chosen parameters are highlighted in yellow.

### Leave one out xgboost\_HIE\_filtered

for the HIE outcome, and using the same code transient\_xgboost.IPYNB, and the files called filtered with HIE outcome for xgb.csv, HIE outcome for xgb\_nirs.csv and HIE outcome for xgb\_60ver.csv have been used for training and testing the model, and this is the result:

The AUC for Filtered with HIE outcome

| ID | prob\_HIE\_filtered(0.1\_4\_0.9\_0\_2\_100) |
| --- | --- |
| AUC | 0.84111111111111103 |
| 95% CI | 0.7308 0.9470 |
| Accuracy | 0.7167 |
| Senistivity | 0.9667 |
| Specificity | 0.4667 |

In the next level, we were wondering if the features extracted from the filtered could predict the HIE level, so we used the same files, just in the HIE outcome for xgb\_nirs, instead of column 2, we used the HIE levels in column 3. for optimization and training the model, we used multiclass\_xgboost.ipynb python file. The optimized parameters are saved in optimization\_filtered\_HIE\_grading.xlsx and the chosen parameters are highlighted in yellow.

Then, using the R code called “multiclass\_auc&ci”, and the probabilities saved in the file multiclass\_prob.xlsx, we calculated the AUC and 95% confidence interval.

The AUC and 95% CI for filtered with HIE grading outcome

| model | filtered\_HIE(0.1\_6\_0.8\_0\_1\_100) |
| --- | --- |
| AUC | 0.8110 |
| low CI | 0.7097 |
| high CI | 0.9251 |

### Filtered first 6 hours

The filtered signals were separated for the first 6 hours after starting of recording (the first 2 epochs). They had been saved in filtered with MRI outcome for xgb\_6h.csv and filtered with HIE outcome for xgb\_6h.csv, and then using MRI outcome for xgb\_58ver.csv, HIE outcome for xgb\_60ver.csv, MRI outcome for xgb\_nirs\_6h, and HIE outcome for xgb\_nirs\_6h, they had been optimized. The optimizations are saved as optimization\_filtered\_6h\_MRI, optimization\_filtered\_6h\_HIE\_xgb and optimization\_filtered\_6h\_HIE\_grading, and the optimized values are highlighted in yellow. The results are as followed:

The AUC for filtered in first 6 hours

| ID | prob\_MRI\_filtered\_6h(0.05\_3\_0.8\_0\_2\_100) | prob\_HIE\_filtered\_6h(0.01\_3\_0.8\_1\_2\_100) |
| --- | --- | --- |
| AUC | 0.50972222222222197 | 0.80999999999999905 |
| 95% CI | 0.3183 0.6862 | 0.6943 0.9124 |
| Accuracy | 0.70689999999999997 | 0.75 |
| Senistivity | 0.16669999999999999 | 0.66669999999999996 |
| Specificity | 0.95 | 0.83330000000000004 |

The AUC and 95% CI for nirs in first 6 hours

| model | filtered\_HIE\_6h |
| --- | --- |
| AUC | 0.7312 |
| low CI | 0.6198 |
| high CI | 0.8406 |

## PAS 2023 abstract

For PAS 2023 abstract, John and Gene suggested that we do not need to know if the NIRS can predict if the babies underwent cooling (0 as level I and 1 as levels II and III) or even the Hie grading. So we focused if the signals (not even the transients were focused in this abstract) were predictive of MRI abnormalities, and also if this model was working better than HIE grading, and cooling. So we made HIE grade and TH\_58ver.csv, which the first column is ID, the second is HIE grading, and the third is cooling data, and using the auc\_CI\_sen\_spec.m, we calculated the AUC of the HIE grading model, AUC = 0.6625; aauc\_ci = 0.506 - 0.807). The AUC of the cooling data was 0.608; CI = 0.475 - 0.748). The model based on NIRS and filtered NIRS were performing better: AUC = 0.699, CI = 0.551 - 0.824 and AUC = 0.722, CI = 0.579 - 0.849 respectively. Just to test, we added the HIE grading and if the infants were cooled to the NIRS and filtered models to see if the performance has improved in the files called *nirs with MRI outcome for xgb\_TH\_HIEgrade.csv* and *filtered with MRI outcome for xgb\_TH\_HIEgrade.csv* and they were optimized in the files called optimization\_nirs\_MRI\_withTH&HIEgrade.xlsx and optimization\_filtered\_MRI\_withTH&HIEgrade.xlsx. The performance did not improve at all, even the AUC decreased a little bit.

Also John suggested to not include the first 6 hours analysis, since we first should be sure that the it is 6 hours after birth, not after the starting of the nirs recordings.

outcome = data(:,2)

a = data(:,5)

[aauc,aauc\_ci]=bootstrap\_aucs(a,outcome)