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. The code transient\_feature\_extraction\_below63.m is used to calculate the time rcSO2 signal is spending below 63%, and transient\_features\_extraction\_below63\_transient.m has been used to calculate the time transient signal is spending below 63%. 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.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:

The optimized parameters for XGBoost with MRI outcome

| AUC | learning rate | max\_depth | colsample | gamma | min child | n est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.4917 | 0.01 | 3 | 0.7 | 1 | 2 | 100 |

### 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 |

To be sure about the used parameters, we take another set of them to see if there is any difference in AUC or not, so we have:

The optimized parameters for XGBoost with MRI outcome

| AUC | learning rate | max\_depth | colsample | gamma | min child | n est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.4764 | 0.05 | 3 | 1 | 0 | 2 | 500 |

Using different parameters, but the same code and files, still the AUC was not significant:

The AUC for transients with MRI outcome

| ID | prob\_MRI (0.05\_3\_1\_0\_2\_500) |
| --- | --- |
| AUC | 0.47638888888888797 |
| 95% CI | 0.3063 0.6482 |
| Accuracy | 0.58620000000000005 |
| Senistivity | 0.27779999999999999 |
| Specificity | 0.72499999999999998 |

### 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:

The optimized parameters for XGBoost with HIE outcome

| AUC | learning rate | max depth | col sam | gamma | min child | n est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.8078 | 0.1 | 6 | 0.7 | 0 | 1 | 100 |

### 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\_6\_0.7\_0\_1\_100) |
| --- | --- |
| AUC | 0.80777777777777704 |
| 95% CI | 0.6838 0.9110 |
| Accuracy | 0.7833 |
| Senistivity | 0.76670000000000005 |
| Specificity | 0.8 |

To be sure about the used parameters, we take another set of them to see if there is any difference in AUC or not, so we have:

The optimized parameters for XGBoost with HIE outcome

| AUC | learning rate | max depth | col sam | gamma | min child | n est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.7889 | 0.05 | 3 | 1 | 0 | 1 | 500 |

Using different parameters, but the same code and files, still the AUC was significant:

The AUC for transients with HIE outcome

| ID | prob HIE (0.05\_3\_1\_0\_1\_500) |
| --- | --- |
| AUC | 0.78888888888888897 |
| 95% CI | 0.6577 0.8948 |
| Accuracy | 0.75 |
| Senistivity | 0.76670000000000005 |
| Specificity | 0.73329999999999995 |

## 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.m has been used to gather all the features extracted from transients 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:

The optimized parameters for XGBoost with MRI outcome in nirs

| AUC | eta | max\_depth | col\_sample | gamma | min\_child\_weight | n\_est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.7042 | 0.05 | 3 | 0.8 | 0 | 1 | 500 |

### 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 not significant:

The AUC for NIRS with MRI outcome

| ID | prob\_MRI\_NIRS(0.05\_3\_0.8\_0\_1\_500) |
| --- | --- |
| AUC | 0.70416666666666605 |
| 95% CI | 0.5626 0.8326 |
| Accuracy | 0.6552 |
| Senistivity | 0.1111 |
| Specificity | 0.9 |

To be sure about the used parameters, we take another set of them to see if there is any difference in AUC or not, so we have:

The optimized parameters for XGBoost with MRI outcome in NIRS

| AUC | eta | max\_depth | col\_sample | gamma | min\_child\_weight | n\_est |
| --- | --- | --- | --- | --- | --- | --- |
| 0.6972 | 0.1 | 4 | 1 | 1 | 1 | 100 |

Using different parameters, but the same code and files, still the AUC was not significant:

The AUC for NIRS with MRI outcome

| ID | prob\_MRI\_NIRS(0.1\_4\_1\_1\_1\_100) |
| --- | --- |
| AUC | 0.69722222222222197 |
| 95% CI | 0.5575 0.8268 |
| Accuracy | 0.70689999999999997 |
| Senistivity | 0.1111 |
| Specificity | 0.97499999999999998 |

### 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:

The optimized parameters for XGBoost with HIE outcome in nirs

| AUC | eta | max\_depth | col\_sam | gamma | min\_child\_weight | n\_estimator |
| --- | --- | --- | --- | --- | --- | --- |
| 0.8156 | 0.1 | 6 | 0.8 | 0 | 1 | 100 |

### 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 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\_6\_0.8\_0\_1\_100) |
| --- | --- |
| AUC | 0.81555555555555503 |
| 95% CI | 0.6862 0.9250 |
| Accuracy | 0.61670000000000003 |
| Senistivity | 0.9667 |
| Specificity | 0.26669999999999999 |

To be sure about the used parameters, we take another set of them to see if there is any difference in AUC or not, so we have:

The optimized parameters for XGBoost with HIE outcome in NIRS

| AUC | eta | max\_depth | col\_sam | gamma | min\_child\_weight | n\_estimator |
| --- | --- | --- | --- | --- | --- | --- |
| 0.81 | 0.01 | 6 | 0.9 | 1 | 1 | 500 |

Using different parameters, but the same code and files, still the AUC was significant:

The AUC for NIRS with HIE outcome

| ID | prob\_HIE\_NIRS(0.01\_6\_0.9\_1\_1\_500) |
| --- | --- |
| AUC | 0.81 |
| 95% CI | 0.6866 0.9163 |
| Accuracy | 0.66669999999999996 |
| Senistivity | 0.9667 |
| Specificity | 0.36670000000000003 |