Brain-computer interface for mental workload tracking

Data science report

Rei Masuda

**Problem statement/competitor analysis**

The prevalence of working and studying from home has increased drastically over the past few years, often resulting in an increase in the amount of time we spend on the computer. Despite the advantages of these flexible arrangements, some individuals report decreased productivity and deteriorated mental health. There are many tools and apps available that can be implemented at home for improving productivity and mental well-being. However, the majority of these solutions fail to leverage the most central component of productivity and mental health – the brain. A potential solution here is to record brain activity to detect increased “mental workload” during computer-based work/study, and adjust the computer interface to reduce this load in real-time (e.g. by reducing the amount of clutter on the screen). The recorded data can also be saved and used as a “tracker”, to provide the user with insights about their mental workload during certain times or tasks.

Non-invasive (i.e. no surgery or implantation required) brain recording devices that can be used at home have only become commercially available in the last couple of years. Technological advancements of these devices have focused on the quality and quantity of sensors that can be implemented into a single device, which have made many of the current market-leading products very expensive. For example, the best “helmet-like” devices range from $1200 US (Sens.ai) to $100,000 US (Kernel Flow2). There have also been parallel developments of smaller and more affordable “headband-like” devices, which are more accessible for the everyday user. The best of these devices in the market currently range between $129 US (NeuroSky) and $400 US (Muse). The lower spatiotemporal resolution and scale of smaller devices make them less flexible in their applications in the productivity and mental well-being space. Furthermore, none of these smaller devices have an in-built capability to track mental workload in their product packages.

A tech startup is looking to create a product that combines a small device with a comprehensive software to track mental workload. They would like to use a particular type of sensor technology known as functional near-infrared spectroscopy (fNIRS), as it is the cheapest and least invasive. They do not have a prototype device yet and are seeking consultation on the feasibility of using a small fNIRS device for this purpose, as well as the software requirements for the algorithm.

**Stakeholders**

The startup has hired me as an external data scientist with expertise in neuroscience. The key stakeholders are:

1. Executive team, who would like to assess the overall feasibility of the project and how it aligns with the company’s strategic goals.
2. Product development team (particularly software developers), who will want to know the technical details of software implementation and integration with the wearable device.
3. Finance department, who would like to assess the cost of development, potential revenue, and return on investment.
4. Marketing team, who will want to understand the market potential for the product and how it can be positioned to appeal to consumers.

**Business question**

The startup is currently considering two software packages. The first option is to deliver a software with a pre-trained algorithm. The second option is to deliver a software that requires algorithm training on the user end. They would like to perform a cost-benefit analysis for each option after assessing the algorithm accuracy of each software option.

**Data question**

The main data question is “are predictions of mental work load for a given user more accurate if algorithms are trained using data from other users, as opposed to only using that user’s own data?”.

**Data source**

Since they do not have data of their own yet, the data was sourced from the largest open-source fNIRS dataset available, from researchers in the Human-Computer Interaction Lab at Tufts University, Boston. Since the dataset contains brain recordings from small fNIRS devices while participants performed mental workload tasks, it is relevant for assessing the feasibility and requirements of the startup’s product. **8 metrics from 2 sensors** were collected from **68 participants who each performed 16 blocks** of a memory task. Each task block consisted of 40 presentations of a letter (trials), and participants had to press a button in response to the letter if the same letter was presented “n” trials ago. Task blocks differed in terms of this number “n”, where higher numbers are thought to induce greater mental workload. The data acquisition paradigm, and task structure for each mental workload task are shown in the schematics below (green boxes indicate when a participant should press the button on each task). **All participants performed 4 blocks of each mental workload task**.

***Data acquisition blocks (numbers represent task difficulty)***

***Task 3***

***Task 2***

***Task 1***

***Task 0***

*trial 40*

C

C

B

A

A

A number on a white background

Description automatically generated with medium confidence

D

A

C

B

A

D

C

A

BBB

A

*trial 5*

*16 blocks*

*Subject 2*

*Subject 1*

*Subject 68*

E

D

C

B

A

*trial 4*

*trial 3*

*trial 2*

*trial 1*

**Exploratory data analysis**

Below is an example of what the sensor data looks like for one task block (80 seconds time series). The name of the sensor measurement describes the sensor location (AB or CD), measurement type (I = intensity, PHI = phase), and the sensor channel (O = oxygenated blood, DO = deoxygenated blood). All measurements were acquired at a sampling rate of 5.2Hz, and were band-pass filtered between 0.001-0.2Hz to remove signal artifacts that are not related to brain activity.

A graph of different types of graphs

Description automatically generated with medium confidence

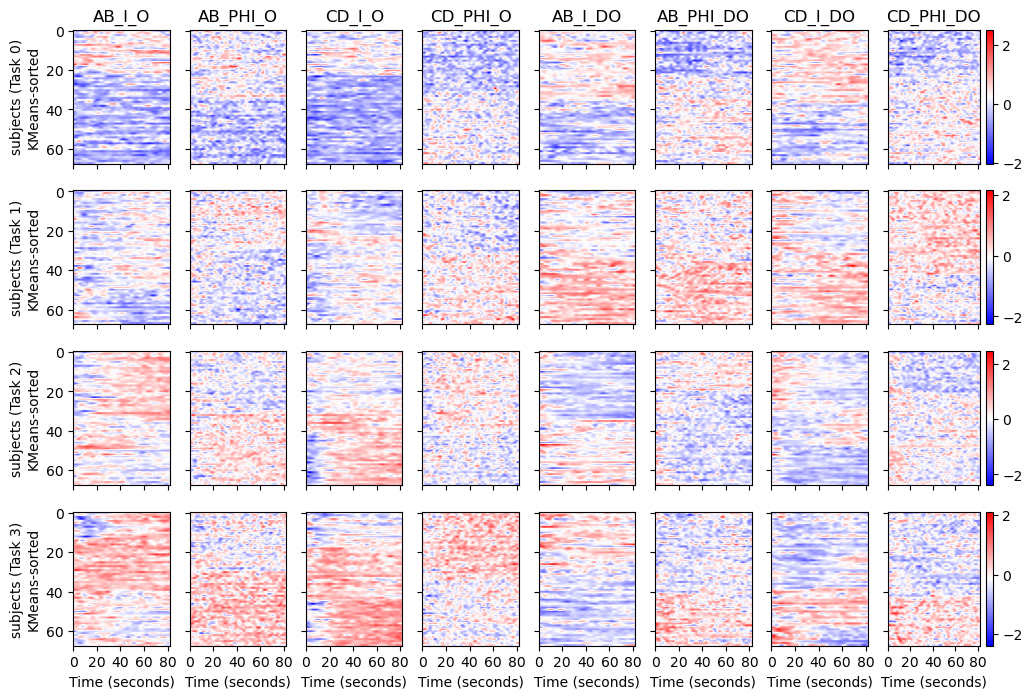
The end goal of product is to provide a software that can accurately predict mental workload from these sensor readings. The average time series across all task blocks and subjects was visualized and overlaid for each level of mental workload, to see if we can see any obvious differences. There seems to be a clear separation between lower (Task 0 and Task 1) and higher (Task 2 and Task 3) mental workload on most sensor measurements. The sensor readings have been **scaled** from this point onwards.

A graph of different colored lines

Description automatically generated with medium confidence

To visualize between-subject variability, the average timeseries across the 4 blocks of each task level was plotted on a subject-by-subject basis after KMeans clustering. Focusing in on the top left heatmap (Channel AB\_I\_O for task level 0), we can see that the majority of subjects have relatively low (blue) activity throughout the block, but there are 20 or so subjects that have relatively high activity. Furthermore, if we focus on the heatmap on the 3rd row and 3rd column (Channel CD\_I\_O on for task level 2), there is cluster of subjects at the bottom that have low activity across the first 15 seconds, which is the followed by high activity for the remainder of the block. Therefore, there is considerable variability between subjects in terms of both the level of activity, as well as the time at which high and low activity is observed across a task block.

The optimal number of clusters determined by the silhouette score is shown below the heatmaps, indicating that there is almost always 2 clusters of subjects (tested a range of clusters between 2 and 10).

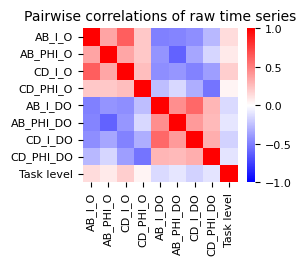


A table with numbers and letters

Description automatically generated

A graph of red squares and blue squares

Description automatically generatedOxygenated (O) and deoxygenated (DO) channels are anti-correlated with each other, which is the expected relationship for fNIRS sensors. Correlations between O channels were positive, as were the correlations between DO channels. These correlations can be removed by transforming channel values into linear combinations of them using techniques such as PCA (right). Implementation of PCA will be tested during model building to assess the influence of correlations between channels.



*Key findings from EDA:*

1. *The average activity across subjects can generally discriminate low and high mental workload, but not necessarily between all levels of the task.*
2. *There is noticeable variability in activity between subjects (grouped into 2 or 3 clusters), which suggest that the same model may not work for different subjects.*
3. *Correlations between channels are present, and should be accounted for.*

**Model paradigms**

There are two distinct paradigms, which relate to how the final software will be deployed. “Subject-specific models” will tell us how an algorithm would perform if there was no pre-training, and was instead trained on the user end. “Generic models” will inform us about how pre-training a model before deployment (with no training on user end) would perform. **Training and validation splits were always 75% and 25%, respectively.**

A screenshot of a computer

Description automatically generated

***Generic models (pre-train before deployment)***

***Subject-specific models (train on user-end after deployment)***

Since the total sample size would only be 16 blocks per subject, each 80 second task block was further split into smaller “chunks” to increase the sample size. For example, breaking down each block into 8 non-overlapping chunks would result in a total sample size of 128 (16 x 8). Sampling with overlapping chunks was also tested. Given the slow dynamics of fNIRS signals, **sample sizes < 5 seconds are not recommended.**

*\*\*The exact partitioning of test data was changed across multiple iterations to find the best partition sizes. The exact size of sample “chunks” were also changed across multiple iterations to find the best size \*\**

**Sample sizes tested for generic models**

|  |  |  |
| --- | --- | --- |
| **Train/test split** | **Chunk size / overlap size** | **Final train/test sample size** |
| 75% / 25% | 20s / 0s | 3264 train / 1088 test |
| 75% / 25% | 10s / 0s | 6528 train / 2176 test |
| 94% / 6% | 20s / 0s | 4096 train / 256 test |

**Sample sizes tested for subject-specific models**

|  |  |  |
| --- | --- | --- |
| **Train/test split** | **Chunk size / overlap size** | **Final train/test sample size** |
| 75% / 25% | 5s / 0s | 192 train 64 test |
| 75% / 25% | 10s / 0s | 96 train / 32 test |
| 75% / 25% | 20s / 10s | 96 train / 32 test |

**Model architectures**

There are three main classes of model architectures that were applied to each model paradigm: Machine learning classifiers, Deep learning with 1-dimensional inputs, and Deep learning with 2-dimensional inputs. The pipelines used for each model architecture are describe below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model architecture** | Machine learning | Deep learning (1D) | Deep learning (2D) |
| **Feature engineering**  (applied separately on each channel) | Manually extract features by taking 7 summary metrics across each time series sample (e.g. mean) | In-built feature extraction from raw time series data within deep learning architecture. | Manually extract scalogram images from each time-series sample, using continuous wavelet transforms (CWT). |
| **Input dimensionality** | 56 scalar features | 8 vector features (1-dimensional) | 8 image features (2-dimensional) |
| **Feature scaling** | * Z-scoring | * Z-scoring | * Z-scoring |
| **Dimensionality reductions techniques tested** | * PCA * Feature Agglomeration | * None | * None * PCA (on scalogram images) |
| **Estimators tested** | * Support vector classifier * Random forest classifier * K-nearest neighbors classifier | * 1-dimensional convolutional neural network | * 2-dimensional convolutional neural network |
| **Labels to predict** | * All labels (0,1,2,3) * Low (0 or 1) vs. High (2 or 3) | * All labels (0,1,2,3) * Low (0 or 1) vs. High (2 or 3) | * All labels (0,1,2,3) * Low (0 or 1) vs. High (2 or 3) |

Hyperparameters were optimized for all models by randomly sampling a hyperparameter search space 15 times, using 3-fold cross-validation. To test the applicability of the model in detecting high mental workload only, we also predicted subsets or combinations of labels (described in the last row of the table above).

**Modelling results**

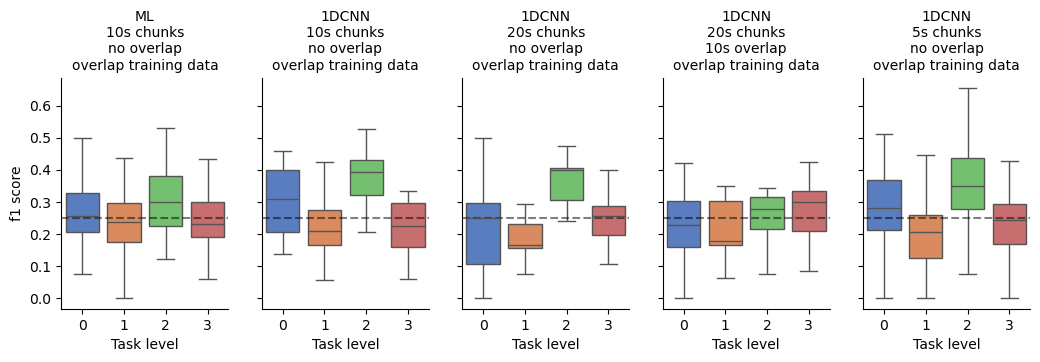
1. *Predicting all class labels with generic models:*

A graph of different colored rectangular shapes

Description automatically generated with medium confidence

The best model is outlined in grey (1D-CNN), where its ability to predict high mental workload (2 and 3) was the highest and **least varied across subjects** (smaller whiskers). Increasing the training size to 94% did not improve the model, suggesting that generic models may not benefit from larger datasets.

1. *Predicting all class labels with subject-specific models:*



The best model is outlined in black (1D-CNN), where its ability to predict high mental workload (2 and 3) was the highest and **least varied across subjects** (smaller whiskers). This model used 20 second chunks, and did slightly better than the equivalent model using 10 second chunks despite the smaller dataset size. These CNN models may therefore benefit from larger datasets, using 20 second chunks. The accuracy is much worse than the generic models on average.

1. *Predicting low vs. high labels with generic models:*

Several boxes with different colored squares

Description automatically generated with medium confidence

The best model is outlined in grey (ML), where most of the subjects (as indicated by the whiskers) are performing better than chance level at predicting high mental workload. 20 second chunks were optimal for this binary classification.

1. *Predicting low vs. high labels with subject-specific models*

Several boxes with text

Description automatically generated with medium confidence

The best model is outlined in black (1D-CNN), but is again performing much worse than the generic models on average.

*Key takeaways from modelling:*

1. *Generic models perform better on average*
2. *Subject-specific models perform better with 1D-CNN architecture*
3. *Subject-specific models may benefit from larger datasets*

**Data answer**

Predictions of mental work load for users are generally more accurate if algorithms are trained using data from other users (generic), with the current dataset size.

**Business answer**

Deploying a pre-trained algorithm will benefit from greater accuracy, and will not require algorithm training infrastructure to be deployed with the software.

**Response to stakeholders**

Deploying a pre-trained algorithm with the software is not a promising option as the accuracy is not very high. The alternative software option has **potential** to be better with more data. This should be considered if the costs associated with further data collection, and deploying a product with algorithm training on the user-end are in line with the overall business goals.