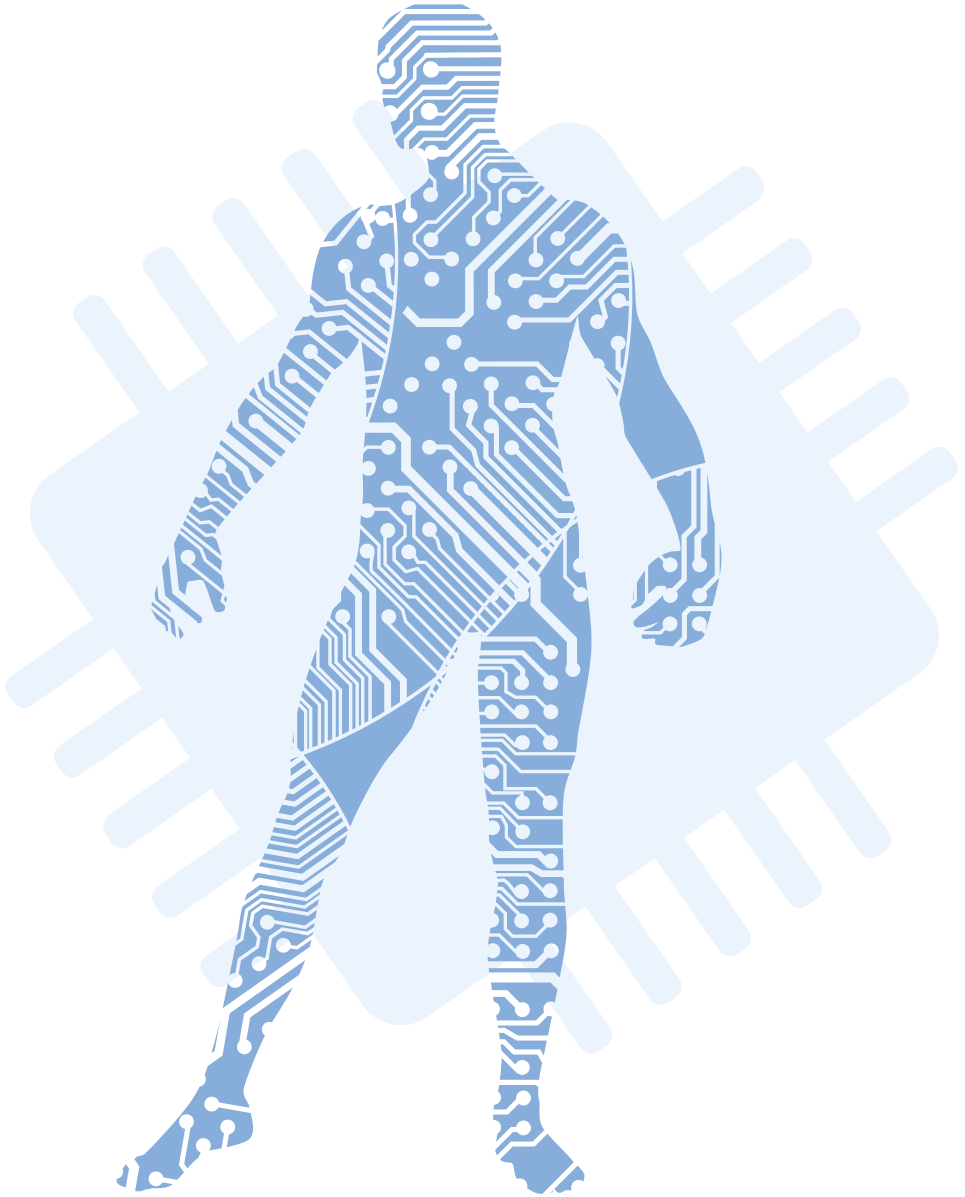
IFMBE Scientific Challenge

Contents



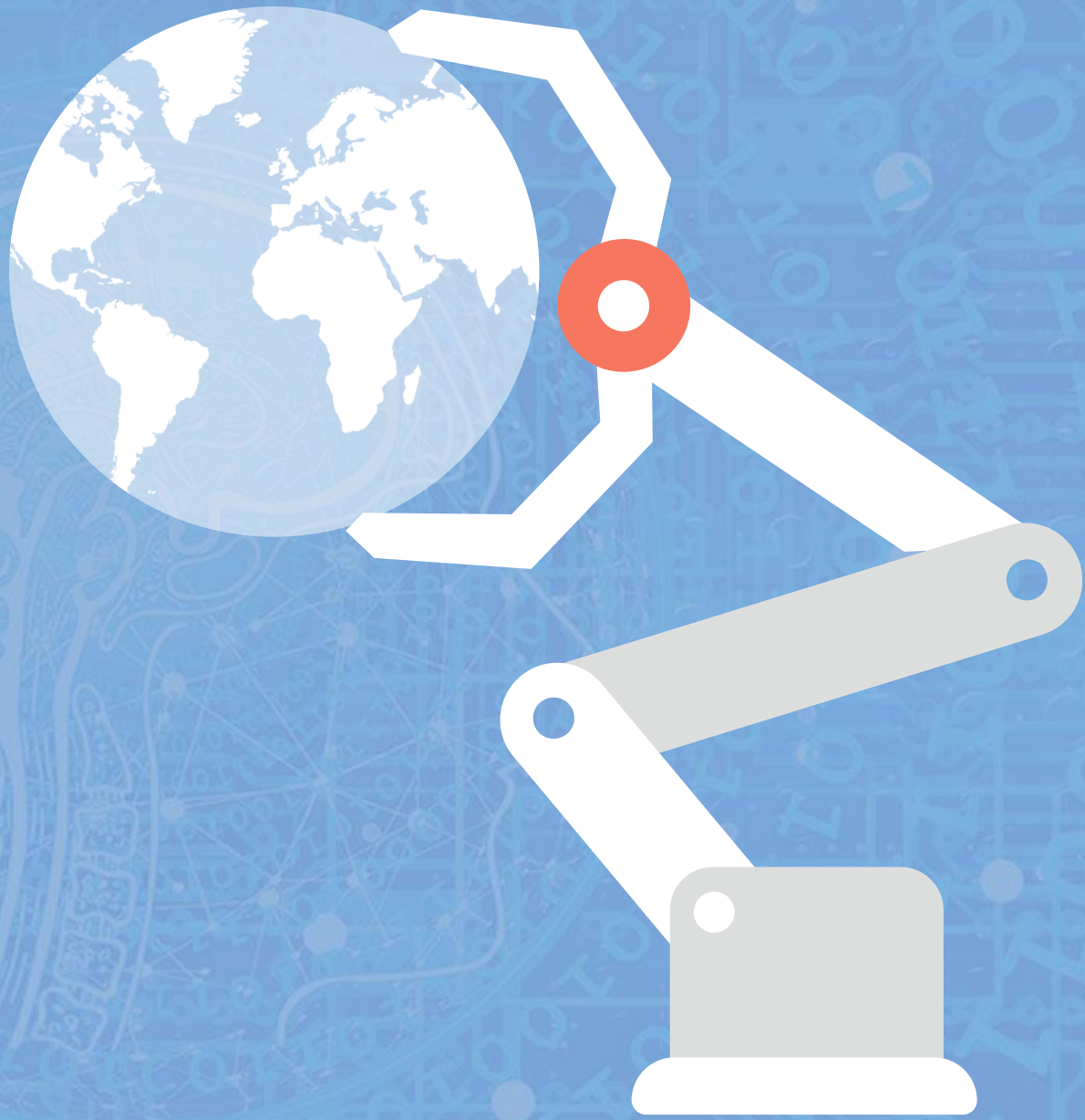
The Problem

P300 Signal Detection in the Brain

In short, our task was to train a model in order to detect P300 brain signal peaks that occur in the following scenario.

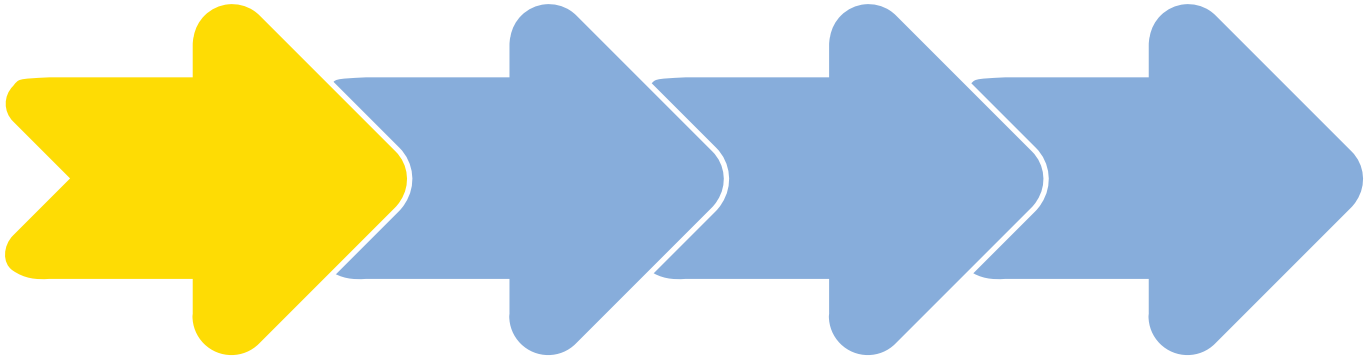
A test subject is exposed to different stimuli. He is told to pay attention to the appearance of one specific stimulus(which is rarer than the others). In theory, 250-500ms after the event occurring (reaction time depends on the subject), we observe a positive peak in the target EEG signal. Such peaks aren't observed for the other stimuli/events.

Or simply put, we want to make a model that recognizes when a subject has himself recognized some (rare, but expected and sought after) event.



Our Work

Phases



01 **Phase 1**
Research. Initial approach.

02 **Phase 2**
Data Preprocessing.

03 **Phase 3**
Training.

04 **Phase 4**
Testing.

Initial Approach

Brain/ EEG signals have 2 specific characteristics:

- High dimensionality
- Low signal to noise ratio (SNR)

Due to these attributes, we decided that it is essential to use a filter on the dataset to smooth out the signals.

Also, because results can heavily vary between subjects, we trained a separate model for each subject, in order not to “confuse” the model unnecessarily.

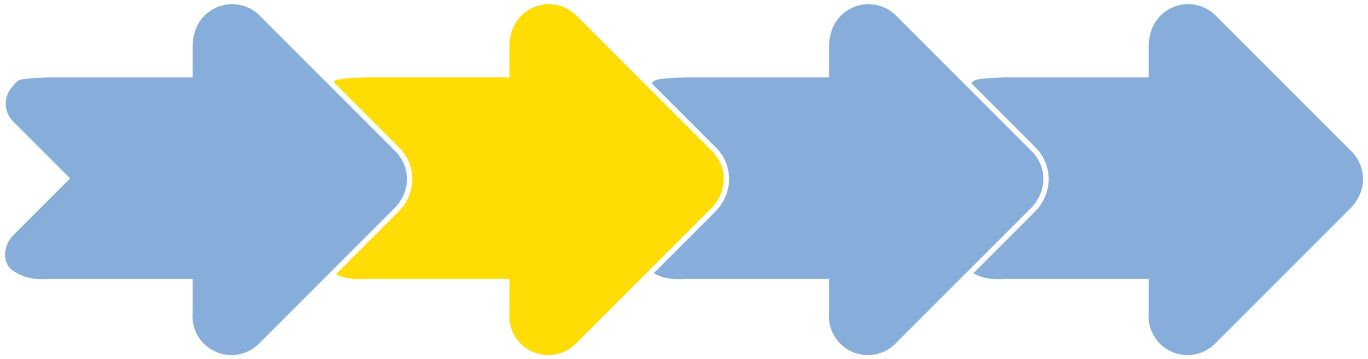
Initial Approach (2)

The goal here is for us to first clean up the data and prepare it for an input layer, and then run it through a base model.

The first model doesn’t have to be perfect, it just has to provide acceptable results for us to compare and then improve the model.

Certain parts of the model can be changed depending on the outcome, as the process of building the models goes on.

Phases



01 **Phase 1**
Research. Initial approach.

02 **Phase 2**
Data Preprocessing.

03 **Phase 3**
Training.

04 **Phase 4**
Testing.

Data aggregation and restructuring

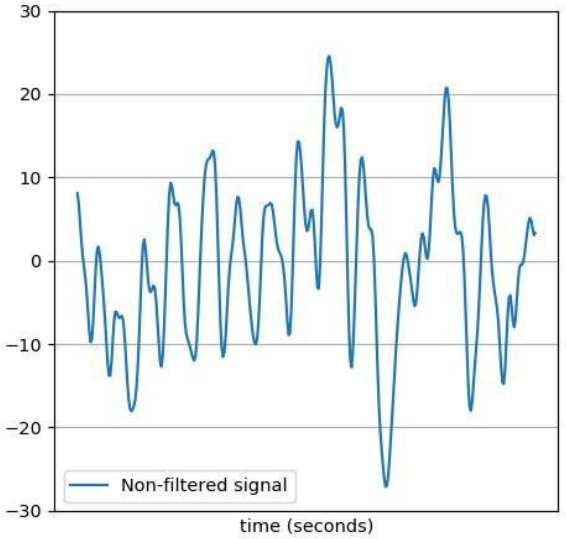
Considering the previous conclusion on subject specificity – the data was restructured such that we split it and then individually combined the complete training and test data for each subject.

This was done with a pipeline that groups and merges all calibration and online phases for each session, for each subject respectively.

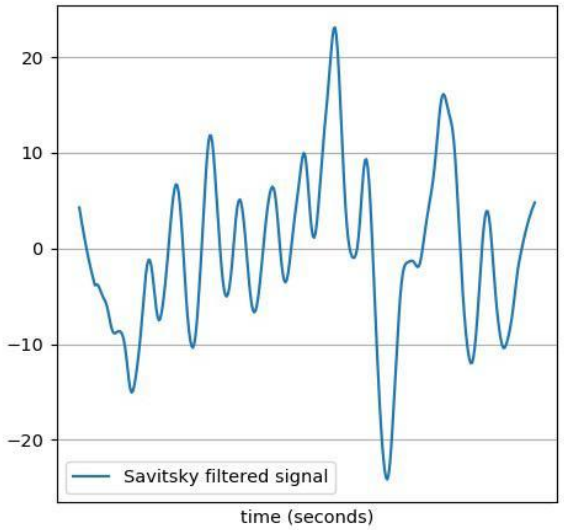
Data Preprocessing

As mentioned before, EEG signals are always accompanied with low SNR. Therefore, we used 2 smoothing filters on the data (separately). We ended up with data in 3 forms:

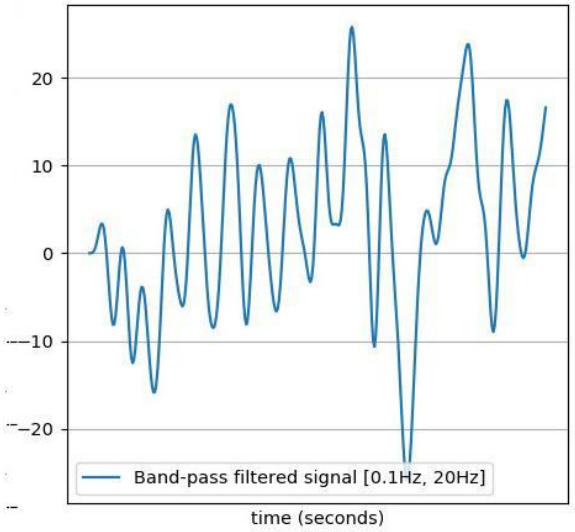
- Raw
- Bandpass filtered
- Savitzky-Golay filtered



Raw Data



S-G Smoothing Filter

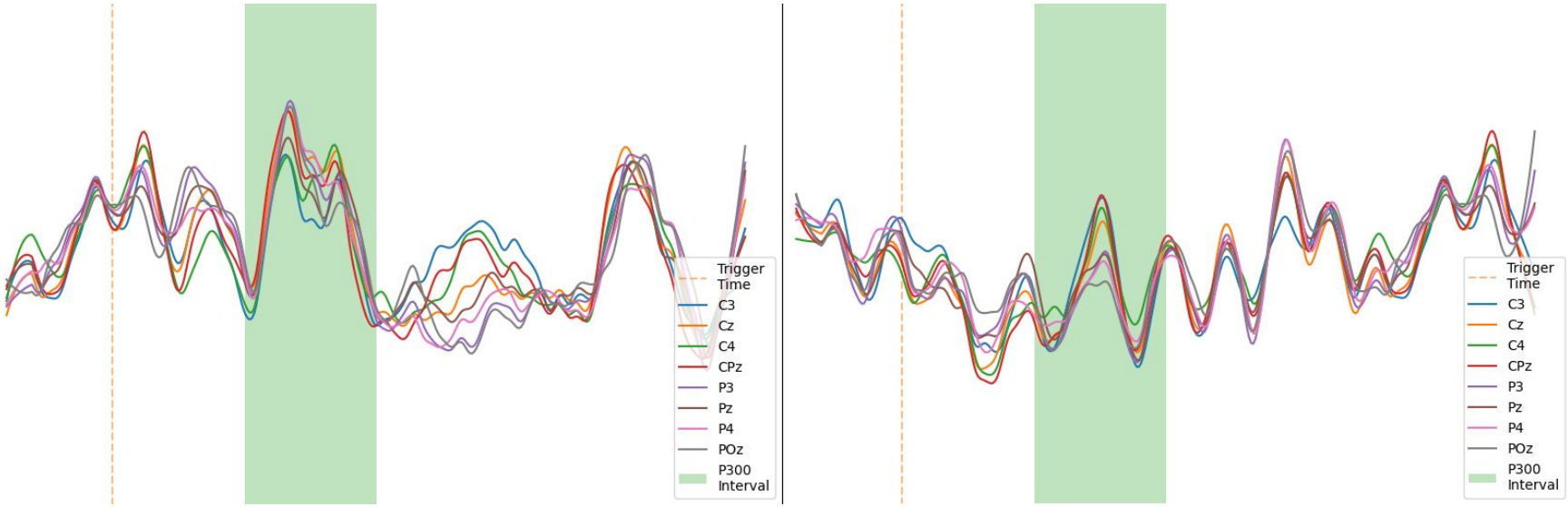


Bandpass Filter

Data Visualization

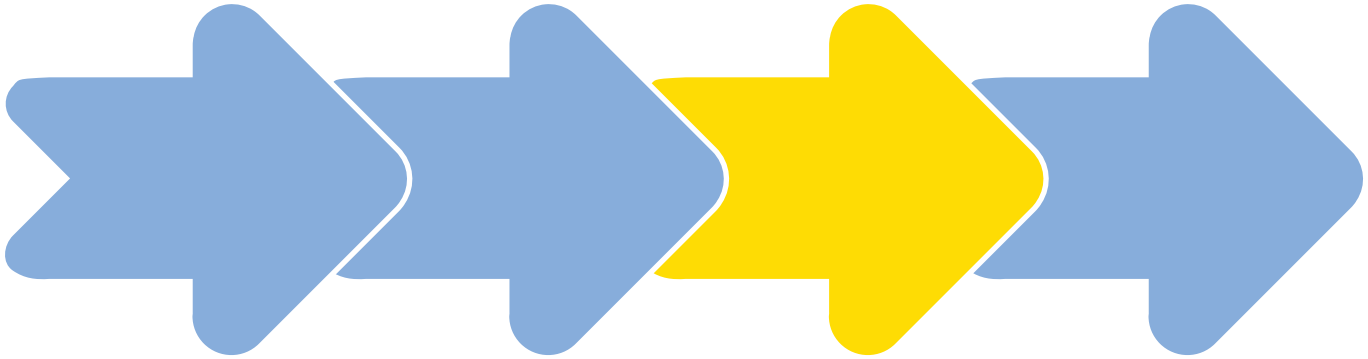
This was done so we could get a better understanding of what we want the model to learn, thus possibly helping us in the task. We plotted the various plots of the brain signals:

- Target events vs. non-target events
- Mean values of all sensor data for target and nontarget events
- Effects of filters on the signal.



Presence(left) vs. Absence(right) of a P300 signal

Phases



01 **Phase 1**
Research. Initial approach.

02 **Phase 2**
Data Preprocessing.

03 **Phase 3**
Training.

04 **Phase 4**
Testing.

Model framework

Our models constitute of 2 parts:

- P300 signal detection
- Per block probabilities aggregation

Two different models were created:

- Simple CNN model
- Deep CNN utilizing spatial and depthwise convolutions

Note: The models differ only in the P300 signal detection (first part), while the aggregation of probabilities for each object stays the same in both of them.

First Model

Input

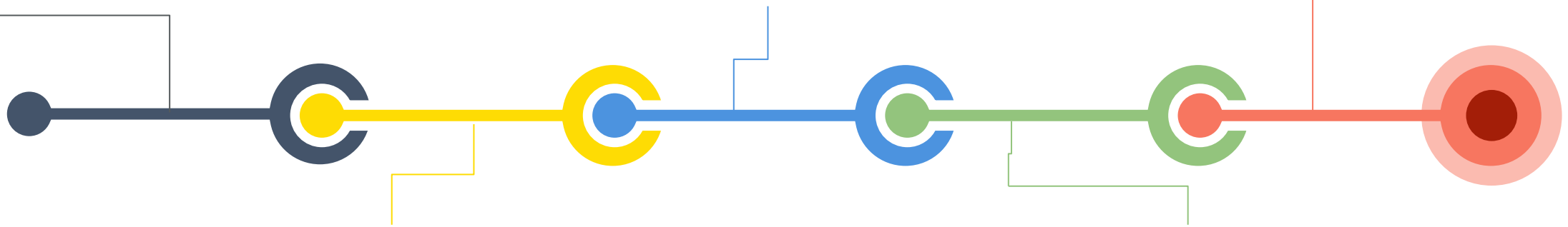
The input for the first model is an event which has two dimensions, representing the number of channels and the temporal dimension - (8,350)

Dropout layer

Regularization and preventing overfitting

Output

The output of the model is a value [0,1] which represents the probability that the event was a target event



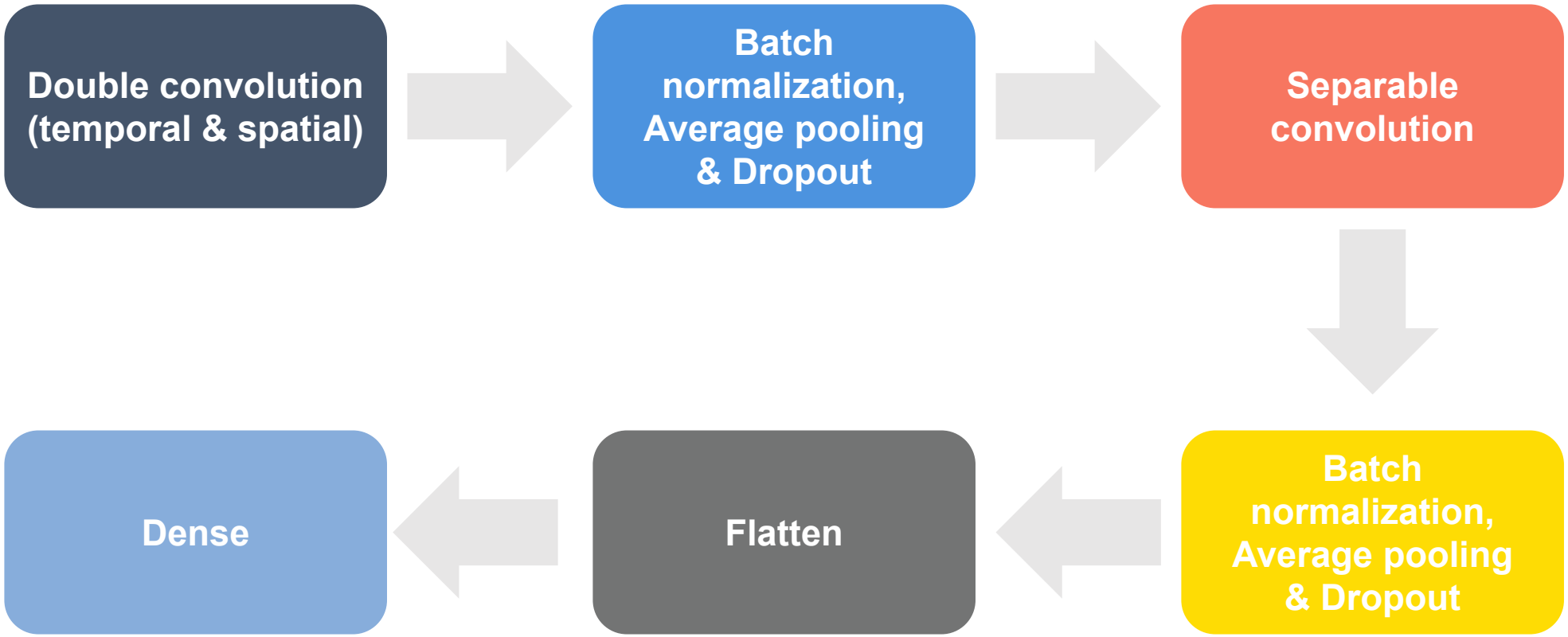
Convolution layer

Generating spatial and temporal features

Flattening layer

Reducing the output of the convolution layer to a single dimension

Second Model



Hyperparameters and model training

When setting the hyperparameters we looked into previous research on the topics and combined that information with the results we got from our initial tests on the data.

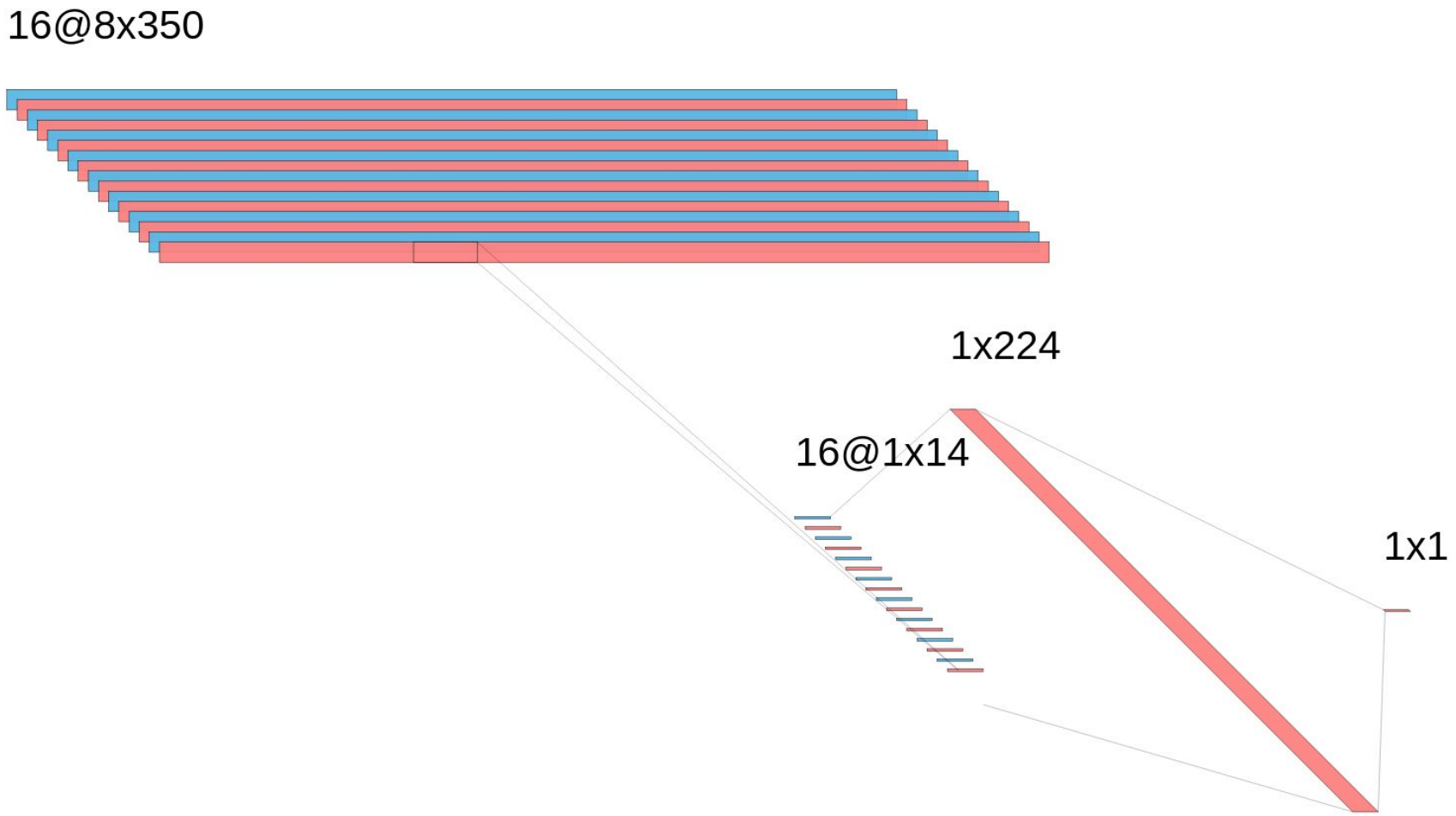
After some testing, we set 30 as the number of epochs spent training while using 20% of the data for validation purposes.

The positive class weight was set to 8, with the negative set to 1 - in accordance with the data.

Hyperparameters - 1st model

- Convolutional layer:
  - 16 convolutional filters
  - 8x25 kernel size
  - 1x25 stride (no overlap, slides only in temporal dimension)
- Dropout rate of 0.25
- Flatten the 16x14 output of the convolution to a single dimension of 224 neurons
- Fully connected to output layer (1 output neuron)

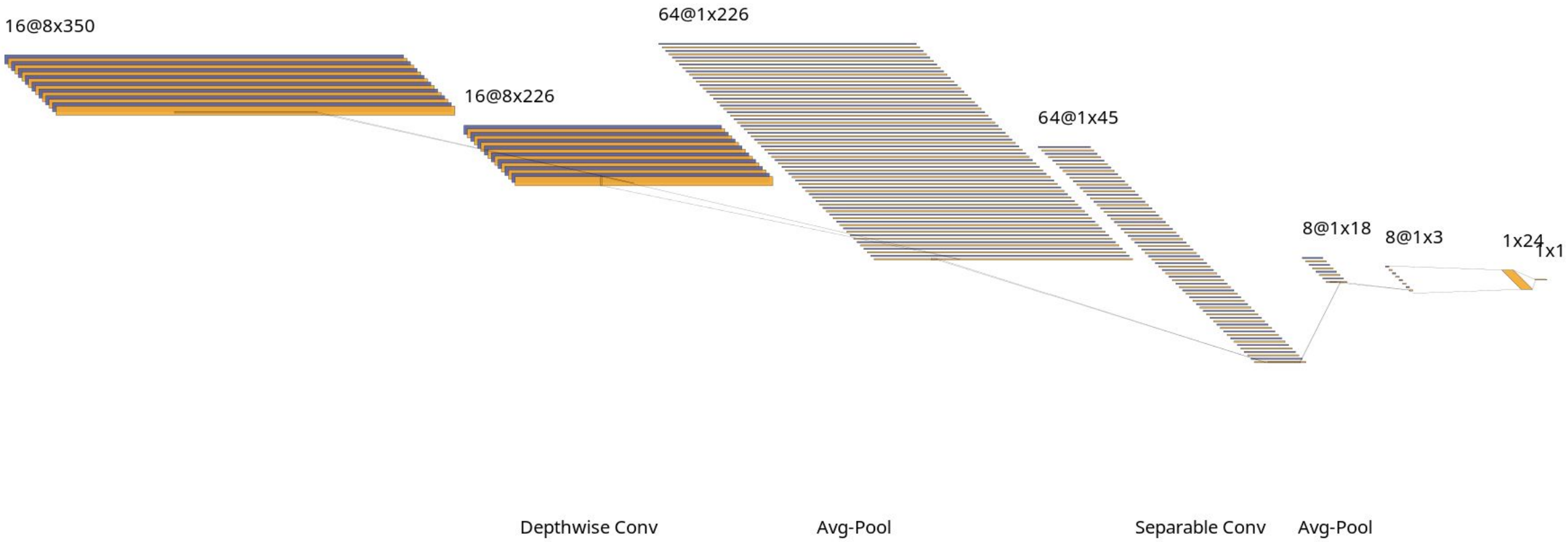
First model visualized



Hyperparameters - 2nd model

- Convolution layer (temporal):
  - 16 convolutional filters
  - 1x125 kernel size
  - 1x1 stride
- Depthwise convolution layer:
  - 4 depth
  - 1x28 kernel size
- Average pooling layers had 1x5 filter size
- Dropout rates of 0.5
- All convolutions used a linear activation function, and were passed through a ReLU layer after batch normalization layers.

Second model visualized



Hyperparameters - preprocessing filters

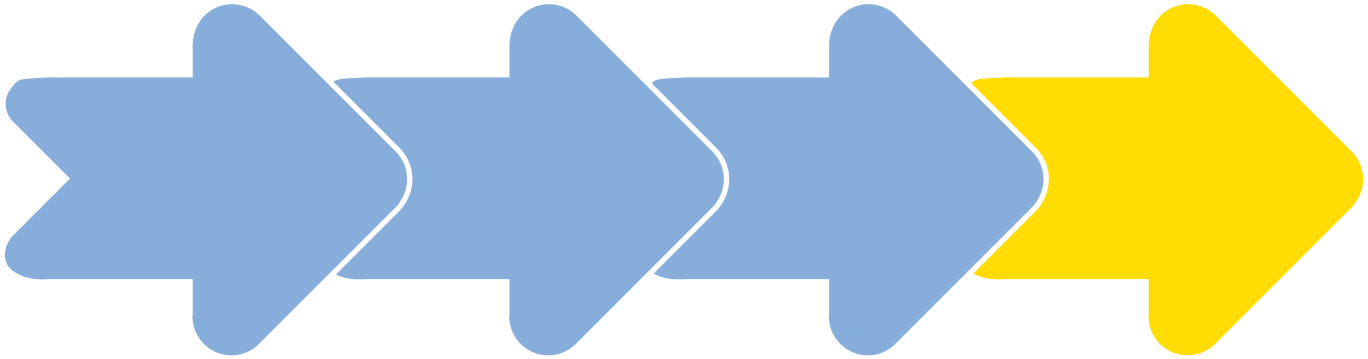
Bandpass filter:

- Low pass of 0.1Hz
- High pass of 20Hz

Savitzky-Golay filter:

- Polynomial order was set to 2
- Frame size was set to 25

Phases



01 **Phase 1**
Research.

02 **Phase 2**
Data Preprocessing.

03 **Phase 3**
Training.

04 **Phase 4**
Testing.

Testing - P300 detection DNNs

The outputs from the P300 detectors are probabilities of the event containing a P300 signal.

	Model 1			Model 2		
	Band-pass	Savitsky-Golay	Raw Data	Band-pass	Savitsky-Golay	Raw Data
Model accuracy	0.8707	0.899	0.895	0.875	0.9174	**0.9178**
Precision score	0.044	0.653	0.609	0.0	**0.784**	0.768
Recall score	0.0016	0.383	0.389	0.0	0.455	**0.477**
F1 score	0.0032	0.476	0.470	0.0	0.563	**0.580**

TABLE I: Results for both P300 detection DNNs on all types of data, averaged for all subjects

The Bandpass filter clearly overfits - we won't use it any further.

Testing - full models - methodologies

Considering how the data is structured, using the *train_test_split* function from scikit-learn is inappropriate. We instead devised **3 different ways of testing the data:**

1. A custom made 5-fold CV function that shuffles the events inside every block, isolated from the events of other blocks.

2. Combining the last 4 blocks (online phase) from each session into a validation set and training on the rest

3. Training and validating on completely separate sessions. Specifically we used the first two sessions for training and the third for validation.

Testing - full models - results

Validation split	Random split (5 fold CV)		4 Last blocks		Third session	
Data type	Savitsky-Golay	Raw	Savitsky-Golay	Raw	Savitsky-Golay	Raw
Model 1	0.9433	0.9411	0.8833	0.8778	0.83	0.8667
Model 2	**0.9856**	0.9756	**0.95**	0.9389	**0.9333**	0.9267

TABLE II: Results (accuracy) of both model architectures on filtered and raw data

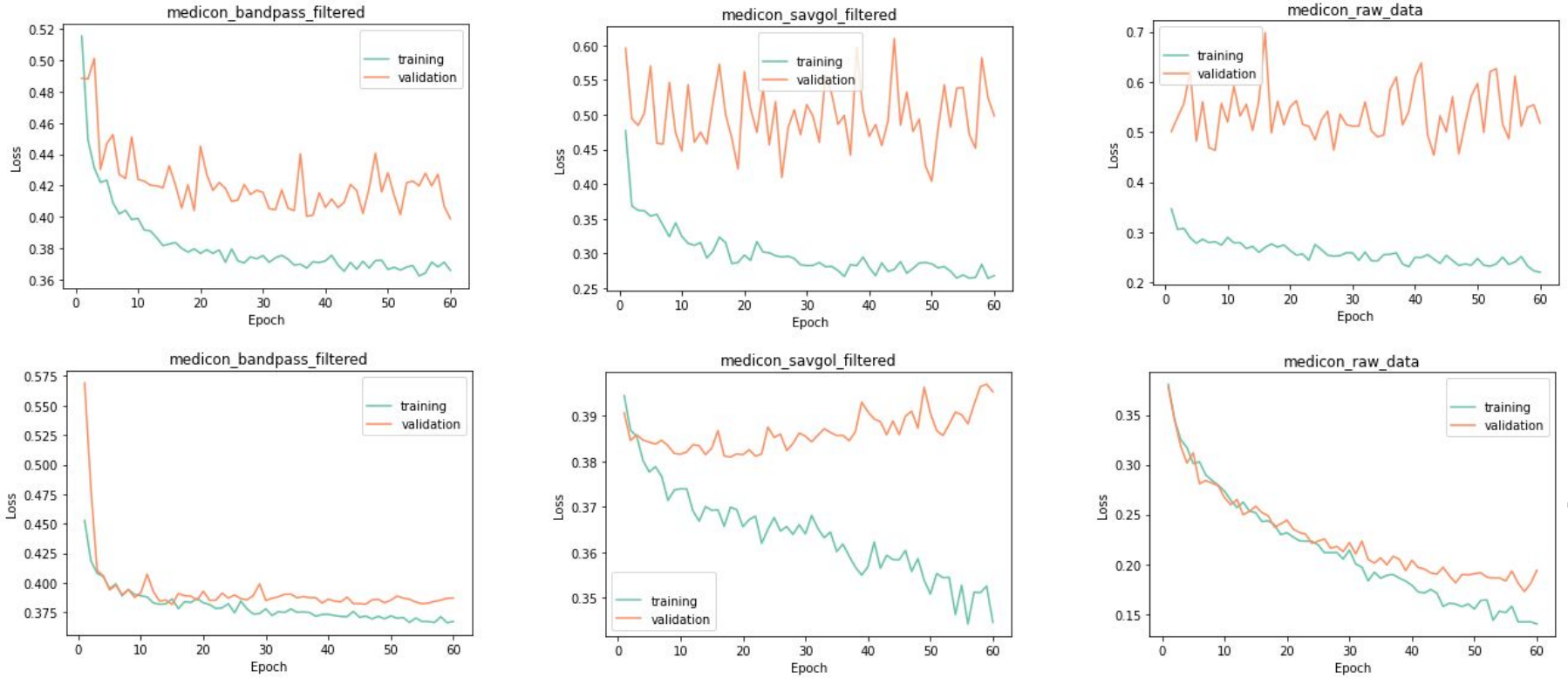
As previously discussed, band-pass filtered data results are not shown.

The improvements in the more complex model are clearly visible, it outperforms the simpler model in all of our tests.

Although the S-G processed data gave better results, in fear of potential overfitting on the yet unknown real test data, we opted to use the raw data as a safer, but well performing, alternative.

Training vs validation loss

Our decision to use the second model with raw data was also supported by the training vs validation loss graphs for the P300 detectors.



As can be seen in this graph, the combination of Raw Data and the Model2 P300 detector DNN gave the best training and validation loss results.

From left to right - band-pass filtered, s-g filtered, raw data.
Row 1 == Model 1 and Row 2 == Model 2's P300 DNN detector.

Results and Final Thoughts

Test data results

On the test data our model achieved **93.91%** accuracy.

We can conclude that this was a successful take on the IFMBE Scientific Challenge on the 15th Mediterranean Conference on Medical and Biological Engineering and Computing.