Machine Learning Engineer Nanodegree

Capstone Project

Omar Magdy
December 7th, 2019

I. Definition

Project Overview

Recent advances in automation and robotics bring about new operating environments where the role of human participants is increasingly reduced to passive observation. While opening radically new venues for improvements in productivity and lifestyle, such operating environments also create new hazards related to the inability of human operators to maintain concentration on passive control tasks. One of the best approaches for monitoring the state of human individuals in such circumstances can be expected to be Brain-computer interfaces (BCIs). In this section, the use of BCIs in detection of the driver's mental attention is reviewed, and the motivation of this study is emphasized.

BCIs establish a direct communication channel between a brain and a computer or external device. BCIs offer a novel communication paradigm between human beings and computers that bypass conventional interaction channels such as keyboard input or speech. Non-invasive BCI, namely those that use electroencephalography (EEG), magnetoencephalography, or magnetic resonance imaging to observe brain activity, are of special interest for addressing the problem of observing the mental state of human individuals by directly monitoring their brain activity, thus avoiding the pitfalls of less direct methods. The use of EEG in this regard is of special interest given the established nature of EEG technology, the relative ease of use of modern EEG headsets, as well as the small size, cost, portability, and reliability of existing modern EEG solutions.

Several studies in the past made use of EEG for fatigue detection in the context of car driving. For instance, Hsieh, Liang, Ko, Lin, and Lin (2006) developed a system that estimates a vehicle's positioning the lane by using the driver's EEG signals in the context of fatigue detection of car drivers. Yeo, Li, Shen, and Wilder-Smith (2009) reported a connection between alerted and drowsy states of car drivers and the changes in the EEG beta and alpha rhythms. They subsequently proposed a system to automatically detect the onset of fatigue of car drivers as well as to test such a system in simulated and real driving conditions. Mardi, Ash- tiani, and Mikaili (2011) studied the association between drowsiness and certain chaotic features of EEG signals for a similar purpose of fatigue detection. Simon et al. (2011) studied the possibility of sleepiness detection of car drivers again by means of EEG alpha spindles. They demonstrated that changes in the parameters of alpha spindles in the EEG signals can be associated with the increase of the drivers' fatigue. Hashemi, Saba, and Resalat (2014) developed a Steady State Visually Evoked Potential EEG BCI for detecting the onset of drowsiness of car drivers.

In this project, I studied the problem of detecting mental state changes in human individuals who need to remain dormant or passive while also having to maintain a continuous significant level of concentration or attention. An example of such a scenario can be supervising automated processes or systems. Another example can be controlling robotic vehicles or drones or security monitoring. Yet another example can be long-term monitoring of aircraft pilots while under the control of the autopilot. In all these cases, non- interfering supervision of processes is desired, while also alertness and a quick reaction is required of the involved individuals.

With this study, I aimed to demonstrate that it is possible to identify pure mental states such as engaged and focused attention versus detached and unfocused monitoring from EEG data and tried to develop a machine learning-based system for solving such a task. Previous studies have used extra data (e.g., task engagement index or respiration data) to support EEG signals or limited the number of mental states detected to two to achieve high accuracy. I tried to detect the differentiation of engaged / focused, detached / unfocused, and drowsing mental states experimentally in a version of continuous performance tasks with 92% (best) and 88% (avg.) accuracies by using only the EEG data. The approach to the subjects' state detection used in this study is gener- ally accepted and therefore can easily be generalized in the future related to the development of subject state monitoring systems in different settings such as patients' state monitoring in hospitals, as well as helping improve safety mechanisms of modern automated and robotic systems.

Problem Statement

According to the World Health Organization (WHO), 1.25 million deaths result from road accidents every year. Consequently, lots of technologies have been developed in order to reduce the accidents to save people's lives. This project aims at developing a Brain-Computer Interface (BCI) that enables the prediction of the driver's behavior based on Electroencephalography (EEG) brain activity of car drivers. The project capitalizes on the extensive research that has been carried out on using spectral analysis of electroencephalogram (EEG) activity to estimate different physiological and psychological states. The developed system will relate the brain activity of the driver before starting to drive the car with the driving pattern. Spectral signatures that correspond to different mental states will be extracted from the recorded EEG. Such signatures will be used to train different machine learning algorithms that will be subsequently used to decode the driver's EEG activity to infer his/her driving behavior. Implemented algorithms will be examined on benchmark data of mental states available online. These algorithms will be then tailored to operate within the AUTOSAR framework. Based on the detected mental state, recommendations to the driver will be given. In addition, signals will be sent to different car components to limit the driver's capabilities in case of predicting a risky driving behavior.

The main objective of this project will be to use machine learning techniques to classify driver's mental states.

When given processed EEG brain activity signal form the headset worn by the person during driving, fed as a stream of electrical voltage readings sampled from the brain for a record of a few seconds duration. This record is then passed by some signal processing operations in frequency domain to extract the underlying features, which is one of the most widely used method to analyze EEG data, is to decompose the signal into functionally distinct frequency bands, such as delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-30 Hz), and gamma (30-100 Hz). We want to be able to detect the current mental state of the driver based on these frequency bands values. Using bands as the features to train the ML model to classify the brain concentration level, whether the person is: (focused, defocused or drowsy).

For our project, the **support vector machine** is chosen as the machine learning algorithm to solve this problem, which uses a discriminant hyperplane to identify classes. In addition, with **SVM**, the selected hyperplane is the one that maximizes the margins (i.e., the distance from the nearest training points), this technique learns the best decision boundary, which has been found to increase the generalization capabilities (Burges 1998; Bennett and Campbell 2000). This fits our problem.

Metrics

In this project, I used accuracy as a performance metric for grid search training, accuracy was chosen as a general base performance reference to compare models from the grid search during training, which does not need to be a complex metric just a simple one as accuracy would be enough to help the grid decide the best candidate model, then [F1-score]() is used as a final evaluation score on the testing set. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is: **F1 = 2 * (precision*recall) / (precision+recall)**. So **F1** was chosen to get a more accurate score not affected by data skewing, with balanced precision and recall contribution, this fits our data distribution, which does not favour precision on recall or vice versa. Since our problem is a **multiclass** detection, then micro averaging is used in the **F1-Score** to calculate metrics globally by counting the total true positives, false negatives and false positives.

For more detailed performance measure for each class of the dataset, confusion matrix is computed, which is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known, as shown in the figure below. Finally, a classification report is used, to get a full description of the model and a precise score for each class, with precision and recall percentages reported.

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig. 1. Confusion matrix

II. Analysis

Data Exploration

The data consists of 34 EEG brain signals records from 5 participants for approximately 30-50 minutes for each file, from 14 channels of the **Emotiv Epoc** headset, located on different places of the head, with sampling rate of 128 HZ.

After signal processing of the raw brain readings, the output data is loaded and split into features and target label. The data consists of 70 features and one target label the mental state. The feature vector for each data entry is about the five frequency bands values for each channel from the 14, concatenated in one vector representing the features. The state classes are represented in a numerical form, where: 1 is Focused, 2 is De-Focused and 3 is Drowsy.

The dataset contains 510 data entries collected from the different 34 EEG records of the 5 participants (i.e. about 7 records for each participant on average). Each successive 15 entry belong to one record, where each state class has 5 samples. We can take more samples to increase our dataset size, however 5 samples are enough to reach a satisfactory performance score.

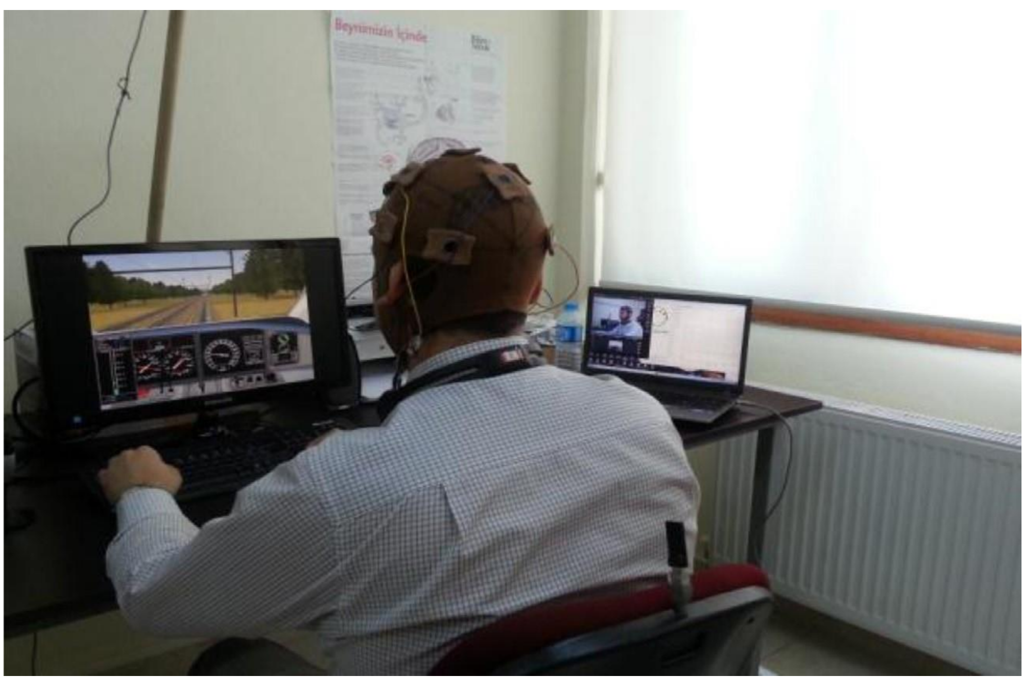


Fig. 2. Simulation and data recording process.

Table 1 lists some of the features of the mental state detector and its statistical properties, organized according to the **Intra class Correlation Coefficient (ICC)** of such features with the target detector states. **ICC** is a statistical measure of relatedness of a continuous predictor variable with a discrete outcome variable.

Table 1
Statistical properties of the 25 most significant features based on the ICC with the target mental state variable.

Rank	Channel	Frequency (Hz)	Mean	STD	Mean focused	Mean unfocused	Mean drowsy	ICC
1	C3	12	7.3701	6.6015	3.0069	2.8568	15.258	0.77489
2	F3	12	7.3351	7.1219	2.6434	2.4826	15.817	0.76973
3	C4	12	7.0274	6.9183	2.5108	2.2899	15.251	0.76694
4	Cz	12	7.1504	6.9895	2.5024	2.4753	15.436	0.76248
5	F4	12	7.7564	6.9367	3.2259	3.0626	15.954	0.75787
6	C3	11.5	7.2411	6.4331	2.6736	3.2756	14.824	0.75544
7	C4	11.5	6.8756	6.5842	2.2998	2.7417	14.616	0.75063
8	F3	11.5	7.1857	6.7928	2.3314	3.0661	15.161	0.74986
9	Pz	12	5.6787	5.9377	2.0338	1.5049	12.627	0.74438
10	Fz	12	6.24	7.2337	1.4592	1.4907	14.709	0.7438
11	F4	11.5	7.5599	6.7017	2.8145	3.5413	15.348	0.73483
12	Cz	11.5	7.0045	6.7105	2.3403	2.8885	14.807	0.73479
13	Fz	11.5	6.1487	7.0016	1.181	2.1192	14.144	0.71065
14	Pz	11.5	5.866	6.0477	1.9401	2.0206	12.772	0.70766
15	C4	12.5	5.9152	6.233	1.9	1.9717	12.988	0.6987
16	F3	12.5	6.3208	6.3337	2.2655	2.2949	13.502	0.69763
17	C3	12.5	6.2436	5.8986	2.4349	2.5351	12.924	0.69603
				…				
43	F4	3	10.72	3.4544	14.252	9.0921	9.0286	0.50204
44	F4	3.5	9.5554	3.2217	12.813	8.2334	7.8354	0.49144
45	F4	2.5	11.889	3.4799	15.402	10.138	10.323	0.49129
46	F4	4	8.86	3.049	11.921	7.5095	7.3397	0.48387
47	F4	4	8.1726	2.8992	11.054	6.8883	6.7534	0.47408
48	F4	2.5	11.351	3.2627	14.548	9.691	9.9839	0.46457
				…				

$$\textbf{ICC} = \frac{\text{variance of group means}}{\text{full variance}}$$

Table 1 shows two types of differences in the EEG signals associated with different mental attention states – an increase in **alpha** band power at frequencies **8-13 Hz**, especially over the **parietal lobe** (electrodes C3, C4, Cz and Pz), and a decrease at **lower delta** and **theta** band frequencies **1-4 Hz**, especially over the **frontal lobe** (electrodes F3, F4 and Fz).

Exploratory Visualization

The EEG data was acquired using a modified Epoc EEG head- set and its classic wet electrodes. Whereas the original Epoc headset only allowed electrode coverage over frontal and occipital areas within a rigid plastic spider-web cap format. Thus, the positions of the electrodes used in this work were F3-Fz-F4-C3-Cz- C4-T3-T4-T5-T6-Pz in the standard 10–20 system. In the EEG device, 4 leads (here identified by the locations T3, T4, T5, and T6) were used to supply current and establish the EEG reference and could not be used for data collection. The acquired data from the 7 leads were identified by F3, F4, Fz, C3, C4, Cz, and Pz.

Table 2
Sample data.

Cnt.	Intp.	Channel F3	Channel FZ	Channel T3	Channel F4	Channel C3	Channel CZ	Channel C4	Channel T4	Channel PZ	X	Y
80	0	4020.51	4915.90	4036.92	4328.72	4285.13	4035.90	4250.77	4304.62	4102.56	1570	1719
81	0	4021.03	4913.85	4039.49	4329.23	4283.59	4035.90	4250.26	4304.62	4107.18	1569	1720
82	0	4016.41	4909.23	4038.97	4327.69	4277.95	4035.38	4250.26	4304.62	4106.67	1568	1721
83	0	4009.23	4907.69	4036.41	4325.64	4270.77	4028.21	4246.15	4304.62	4105.64	1569	1719
84	0	4003.08	4905.64	4036.41	4326.15	4264.10	4017.44	4245.13	4304.62	4107.18	1568	1722
85	0	3994.36	4902.56	4036.41	4323.59	4255.38	4010.26	4243.59	4304.62	4106.67	1569	1721
86	0	3988.21	4904.10	4034.87	4322.05	4248.72	4006.15	4236.41	4304.62	4106.15	1567	1721
87	0	3988.72	4903.08	4035.38	4324.62	4252.31	4003.08	4234.36	4304.62	4107.69	1568	1722
88	0	3991.28	4902.05	4034.36	4327.18	4258.46	4002.56	4238.46	4304.62	4106.15	1567	1721
89	0	3991.79	4903.08	4034.36	4326.67	4261.03	40 0 0.0 0	4236.92	4304.62	4101.03	1566	1721
90	0	3991.28	4898.97	4034.36	4325.64	4260.00	3994.87	4232.31	4304.62	4097.44	1566	1720
91	0	3990.77	4893.33	4034.87	4325.64	4255.90	3991.79	4234.36	4304.62	4093.85	1565	1719
92	0	3988.72	4893.85	4035.38	4324.10	4254.87	3991.79	4233.85	4304.62	4090.26	1564	1717
93	0	3988.21	4895.38	4034.36	4322.05	4257.44	3992.31	4230.77	4304.62	4090.77	1563	1717
94	0	3994.36	4894.36	4034.36	4325.64	4261.03	3994.36	4234.87	4304.62	4092.82	1562	1715

For illustration of brain signals reading versus time, please find below a 30-seconds extract of real slow-wave sleep from one young individual. The sampling frequency is 100 Hz and the channel is F3.

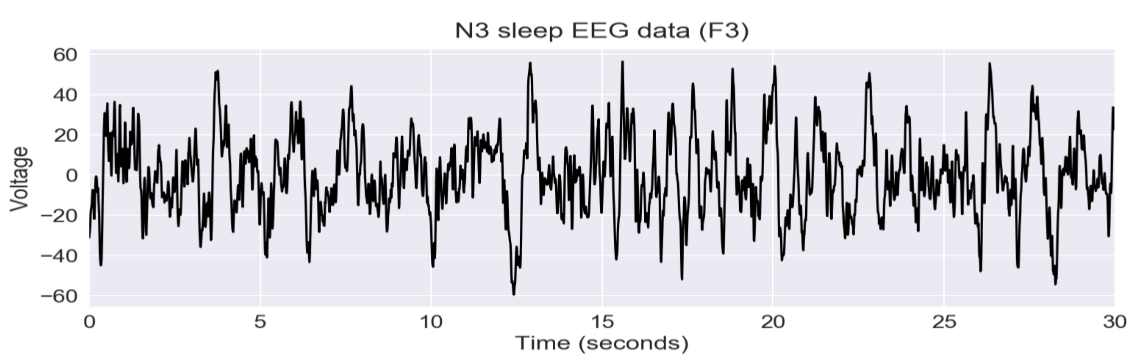


Fig. 3. EEG signal time graph.

A plot showing a typical example of the weight vectors W for a participant over the three most distinctive EEG channels of F3, F4 and Fz is given in Fig. 4. As can be seen there, the examination of SVM weight vectors indicates that elevated EEG power in the frequency range of **1–5 Hz** is treated as a positive evidence of the presence of **"focused"** state by the detector, while the elevated power in the frequency range of **10–15 Hz** is taken as an indication of the **"drowsing"** state. These observations are in agreement with the findings concerning the EEG signature of drowsy and alert mental states in the literature. The indicator of the **"unfocused"** or disengaged mental state is the **reduction** in the EEG power at both **1–5 Hz and 10–15 Hz frequencies** (Fig. 4 B). This corresponds to the learning of the EEG spectrum in the disengaged mental state.

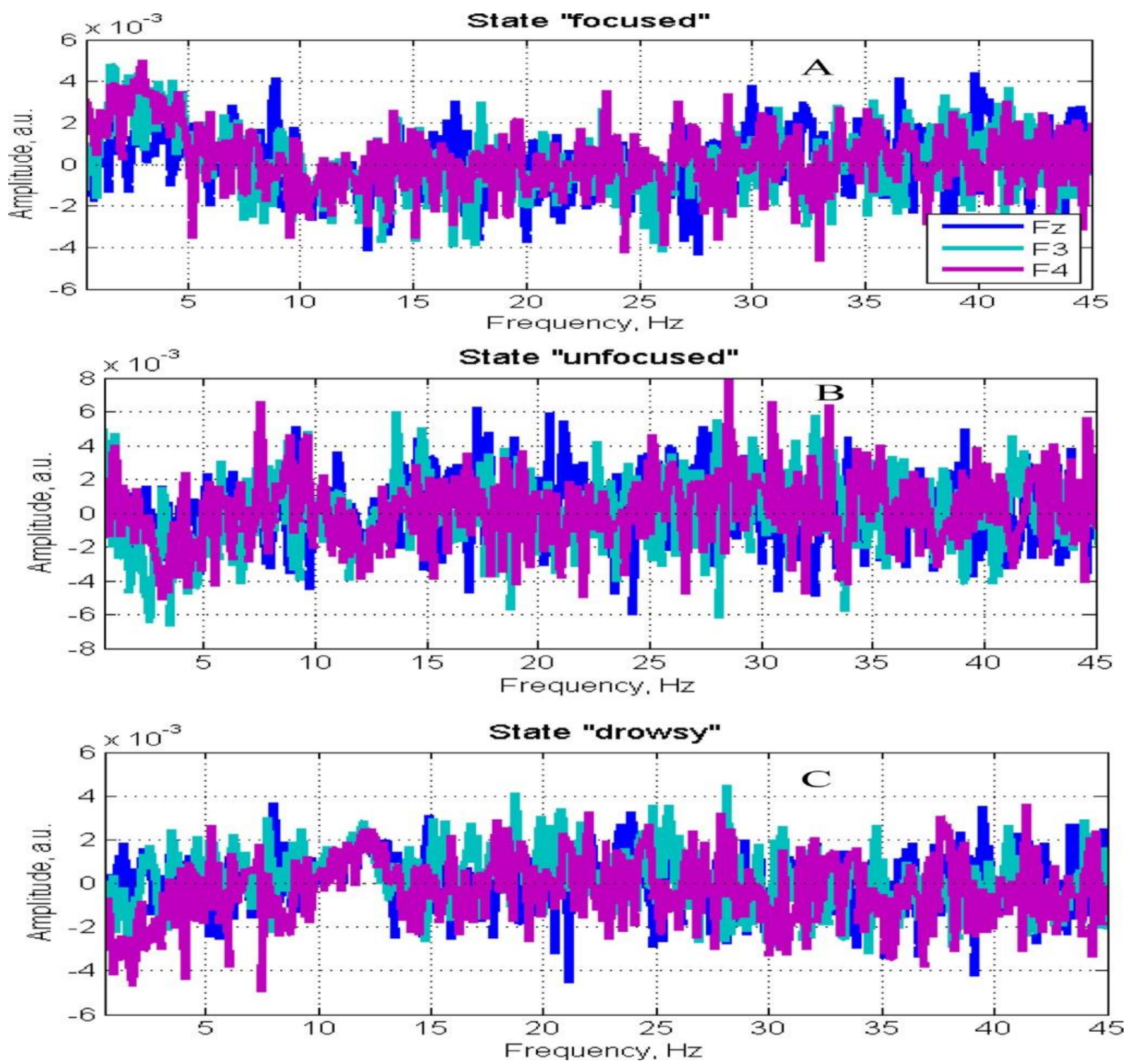


Fig. 4. An example of SVM weight vectors for focused (A), unfocused (B) and drowsing (C) mental states.

Fig. 5 shows a sample EEG data before the SVM-based classifier training part. The differences in the EEG signals associated with the different attention states can be seen in the spectrograms. Red dashed lines indicate the different attention state periods corresponding to the time-intervals of **0–10 min (focused)**, **10–20 min (unfocused)**, and **20–35 min (drowsy)**.

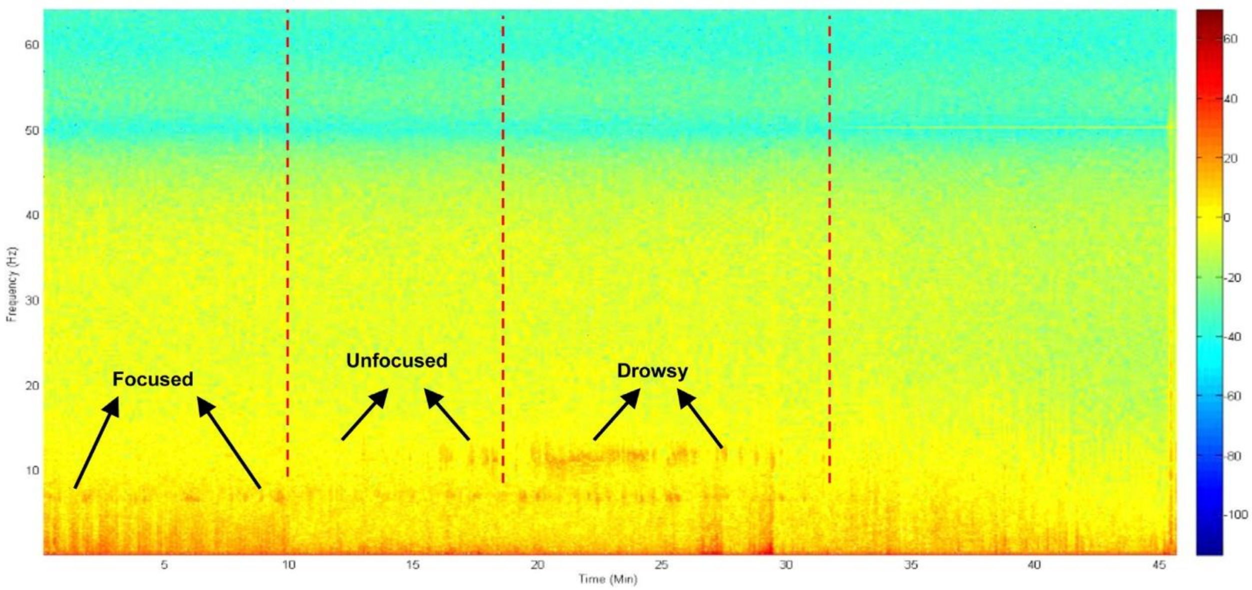


Fig. 5. A sample EEG data before the SVM-based classifier training part.

Algorithms and Techniques

In BCI design, EEG signal processing aims at translating raw EEG signals into the class of these signals, i.e., into the estimated mental state of the user. This translation is usually achieved using a pattern recognition approach, whose two main steps are the following:

- **Feature Extraction:** The first signal-processing step is known as "feature extraction" and aims at describing the EEG signals by (ideally) a few relevant values called "features" (Bashashati et al. 2007). Such features should capture the information embedded in EEG signals that is relevant to describe the mental states to identify, while rejecting the noise and other non-relevant information. All features extracted are usually arranged into a vector, known as a feature vector.
- **Classification:** The second step, denoted as "classification," assigns a class to a set of features (the feature vector) extracted from the signals (Lotte et al. 2007). This class corresponds to the kind of mental state identified. This step can also be denoted as "feature translation" (Mason and Birch 2003). Classification algorithms are known as "classifiers".

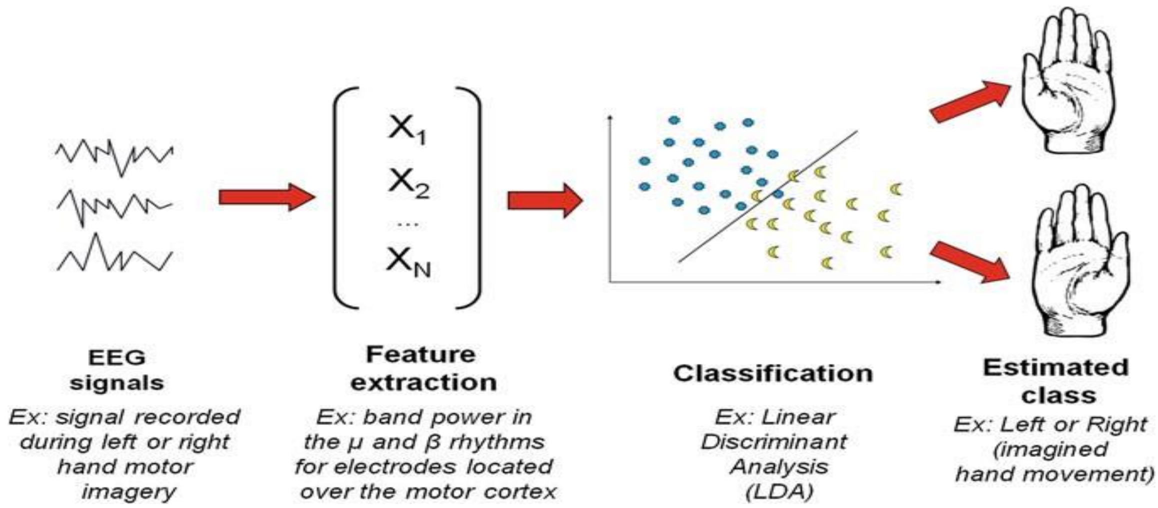


Fig. 6. A classical EEG signal-processing pipeline for BCI.

In this project, I used **Spectral Analysis** source to extract features from EEG signals:

- **Spectral (frequential) information:** Such features would describe how the power in some relevant frequency bands varies. In practice, this means that the features will use the power in some specific frequency bands.

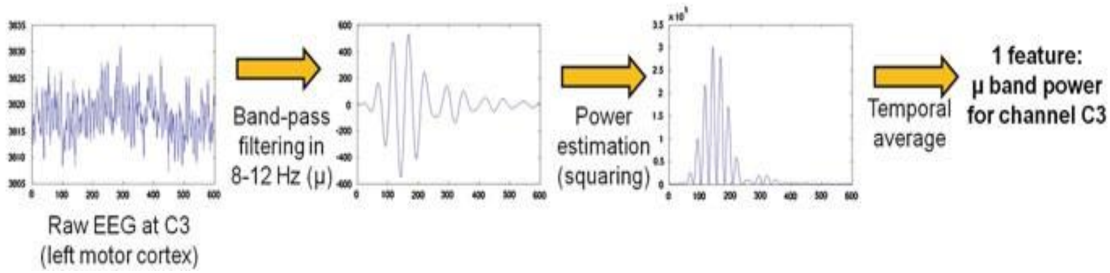


Fig. 7. Signal-processing steps to extract band-power features from raw EEG signals.

For classification, very popular classifiers for BCI are linear discriminant analysis (LDA) and support vector machine (SVM) (Bennett and Campbell 2000). **For this project SVM is used**, which also uses a discriminant hyperplane to identify classes (Burges 1998). However, with SVM, the selected hyperplane is the one that maximizes the margins, i.e., the distance from the nearest training points, which has been found to increase the generalization capabilities (Burges 1998; Bennett and Campbell 2000).

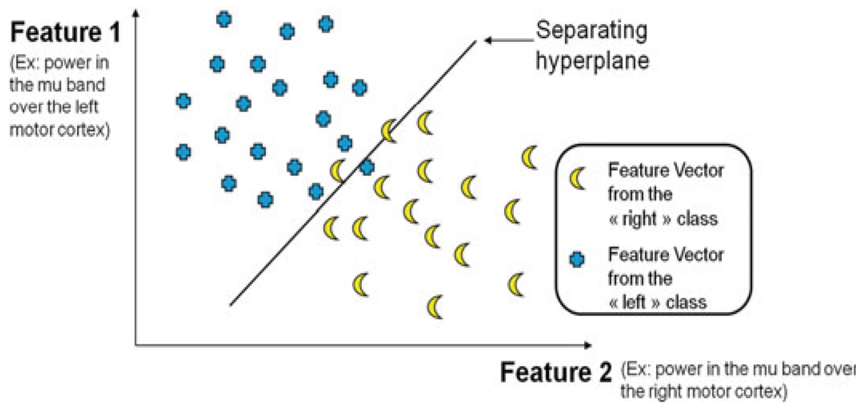


Fig. 8. Discriminating two types of motor imagery with a linear hyperplane using a linear discriminant analysis (LDA) classifier.

Generally, regarding classification algorithms, it seems that very good recognition performances can be obtained using appropriate off-the-shelf classifiers such as LDA or SVM (Lotte et al. 2007).

Benchmark

For benchmark model, we will use the algorithms outlined in this paper "Distinguishing mental attention states of humans via an EEG-based passive BCI using machine learning methods" results as threshold for comparing across performances obtained by my solution.

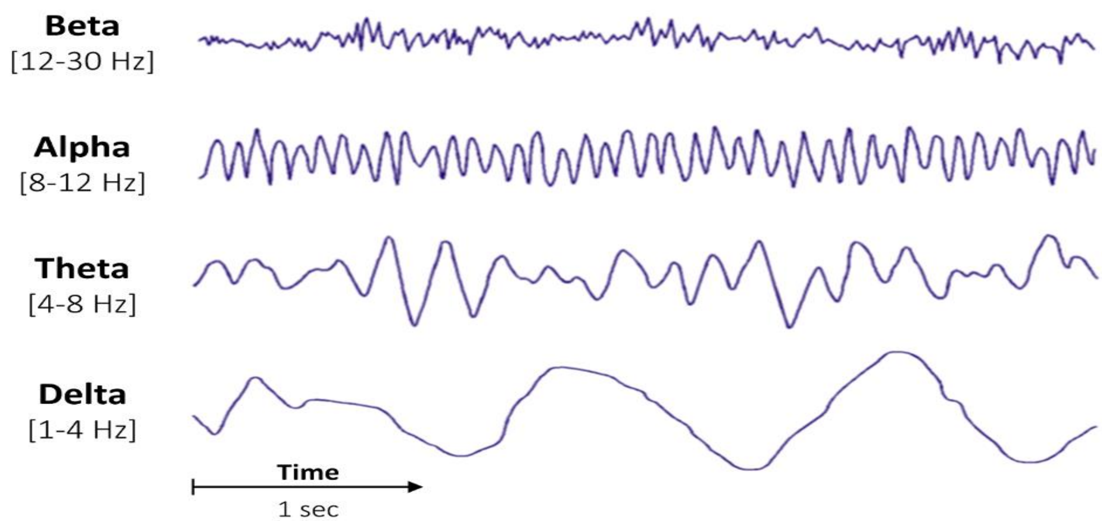
Abstract: Recent advances in technology bring about novel operating environments where the role of human participants is reduced to passive observation. While opening new frontiers in productivity and lifestyle, such environments also create hazards related to the inability of human individuals to maintain focus and concentration during passive control tasks. A passive brain-computer interface for monitoring mental attention states of human individuals (focused, unfocused, and drowsy) by using electroencephalographic (EEG) brain activity imaging and machine learning data analysis methods is developed in this work. An EEG data processing pipeline and a machine learning mental state detection algorithm using the Support Vector Machine (SVM) method were designed and compared with k-Nearest Neighbor and Adaptive Neuro-Fuzzy System methods. To collect 25 h of EEG data from 5 participants, a classic EEG headset was modified. We found that the changes in EEG activity in frontal and parietal lobes occurring at 1 – 5 Hz and 10 –15 Hz frequency bands were associated with the changes in individuals' attention state. We demonstrated the ability to use such changes to identify individuals' attention state with 96.70% (best) and 91.72% (avg.) accuracy in experimental settings using a version of continuous performance task with SVM-based mental state detector. The findings help guide the design of future systems for monitoring the state of human individuals by means of EEG brain activity data.

III. Methodology

Data Preprocessing

Signal Processing

One of the most widely used method to analyze EEG data is to decompose the signal into functionally distinct frequency bands, such as delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz), and gamma (30–100 Hz).



This implies the decomposition of the EEG signal into frequency components, which is commonly achieved through Fourier transforms. The almost invariably used algorithm to compute the Fourier transform (and arguably the most important signal processing algorithm) is the Fast Fourier Transform (FFT), which returns, for each frequency bin, a complex number from which one can then easily extract the amplitude and phase of the signal at that specific frequency. In spectral analysis, it is then common to take the magnitude-squared of the FFT to obtain an estimate of the power spectral density (or power spectrum, or **periodogram**), expressed in (micro)-Volts2 per Hertz in the case of EEG data.

Although a myriad of analyses can be performed from the power spectral density, I am going to focus here on a very simple one: the **average band power**, which consists in computing a single number that summarizes the contribution of the given frequency band to the overall power of the signal. This particularly useful in our machine-learning approach, when we will want to extract some keys features from the data and have a single number that could summarize a particular aspect of the data (i.e. Frequency bands average power).

- **Computing the power spectral density:**

  In order to compute the average bandpower in the delta band, we first need to compute an estimate of the power spectral density. The most widely-used method to do that is the Welch's periodogram, which consists in averaging consecutive Fourier transform of small windows of the signal, with or without overlapping.

  The Welch's method improves the accuracy of the classic periodogram. The reason is simple: EEG data are always time-varying, meaning that if you look at a 30 seconds of EEG data, it is very (very) unlikely that the signal will looks like a perfect sum of pure sines. Rather, the spectral content of the EEG changes over time, constantly modified by the neuronal activity at play under the scalp. Problem is, to return a true spectral estimate, a classic periodogram requires the spectral content of the signal to be stationary (i.e. time-unvarying) over the time period considered. Because it is never the case, the periodogram is generally biased and contains way too much variance (see the end of this tutorial). By averaging the periodograms obtained over short segments of the windows, the Welch's method allows to drastically reduce this variance. This comes at the cost, however, of a lower frequency resolution. Indeed, the frequency resolution is defined by:

$$F_{res} = \frac{F_s}{N} = \frac{F_s}{F_s t} = \frac{1}{t}$$

  where F_S is the sampling frequency of the signal, N the total number of samples and t the duration, in seconds, of the signal. In other words, if we were to use the full length of our data (30 seconds), our final frequency resolution would be 1/30=0.033 Hz, which is 30 frequency bins per Hertz. By using a 4-second sliding window, we reduce this frequency resolution to 4 frequency bins per Hertz, i.e. each step represents 0.25 Hz.

  How do we define the optimal window duration then? A commonly used approach is to take a window sufficiently long to encompasses at least two full cycles of the lowest frequency of interest. In our case, our lowest frequency of interest is 0.5 Hz so we will choose a window of 2/0.5=4 seconds.

```
from scipy import signal

# Define window length (4 seconds)
win = 4 * sf
freqs, psd = signal.welch(data, sf, nperseg=win)
```

ML Processing

Fortunately, for the dataset, there are no invalid or missing entries we must deal with, however, there are some qualities about certain features that must be adjusted. This pre-processing can help tremendously with the outcome and predictive power of nearly all learning algorithms.

- **Transforming Skewed Continuous Features:**
  A dataset may sometimes contain at least one feature whose values tend to lie near a single number but will also have a non-trivial number of vastly larger or smaller values than that single number. Algorithms can be sensitive to such distributions of values and can underperform if the range is not properly normalized. With our dataset for example, feature such as: **delta** values from **F7** channel **(2nd channel)**, fit this description.

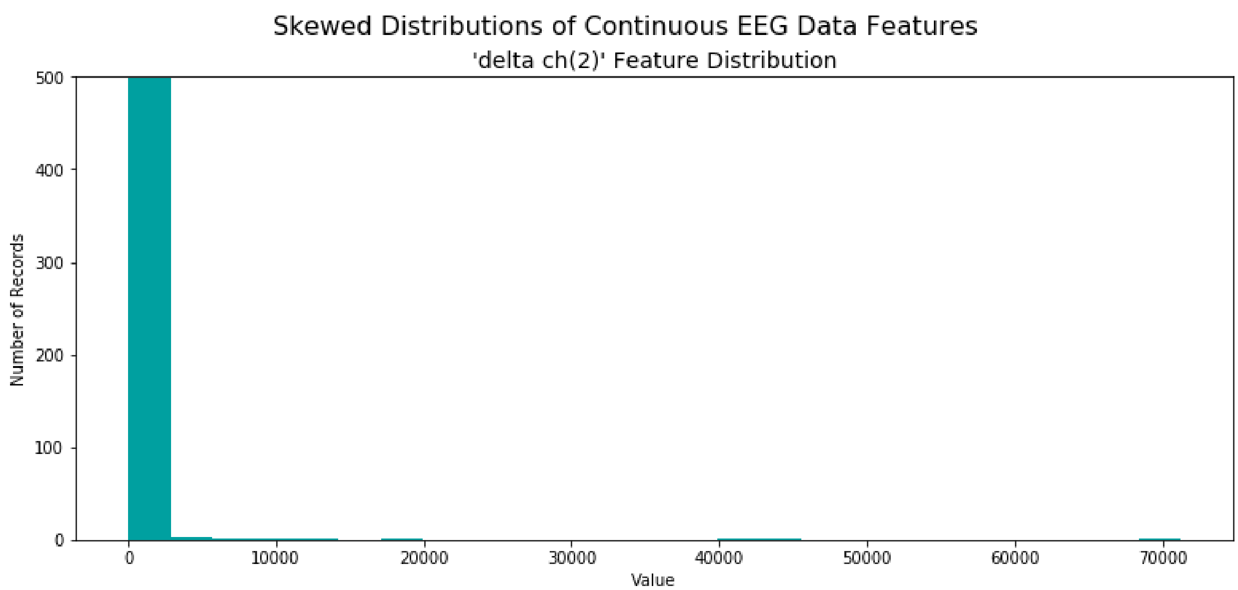


Fig. 9. Skewed distribution before log transformation.

For highly-skewed feature distributions such as **'delta ch(2)'**, it is common practice to apply a logarithmic transformation on the data so that the very large and very small values do not negatively affect the performance of a learning algorithm. Using a logarithmic transformation significantly reduces the range of values caused by outliers. Care must be taken when applying this transformation however: The logarithm of 0 is undefined, so we must translate the values by a small amount above 0 to apply the the logarithm successfully.

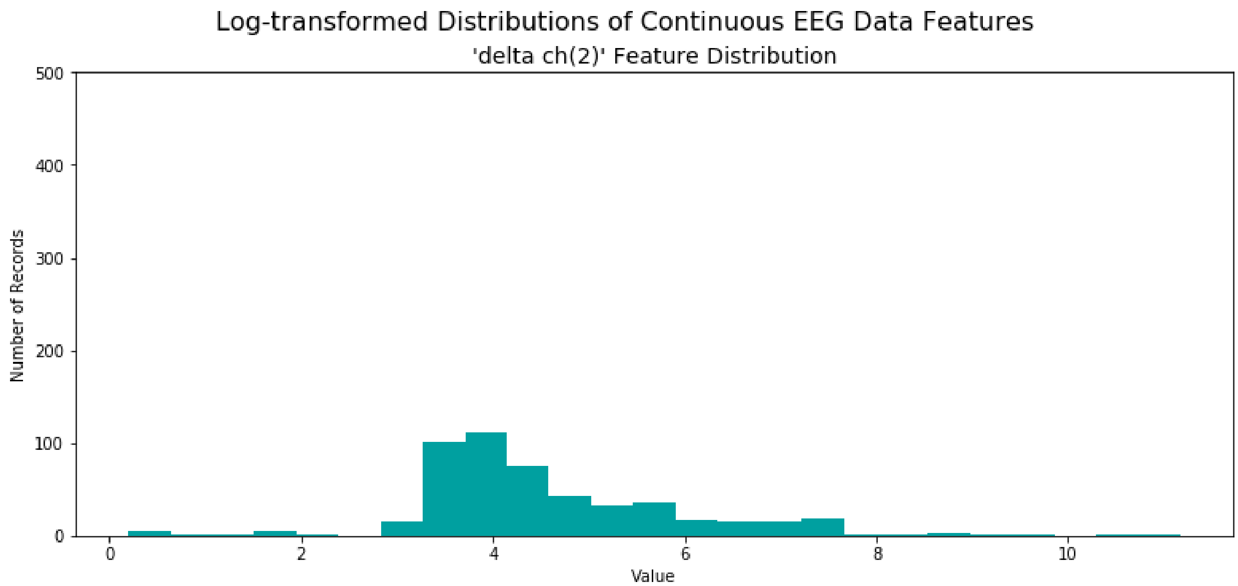


Fig. 10. Log transformed EEG data distribution.

- **Outlier Detection:**

  Detecting outliers in the data is extremely important in the data pre-processing step of any analysis. The presence of outliers can often skew results which take into consideration these data points. There are many "rules of thumb" for what constitutes an outlier in a dataset. **For our dataset**, we will use Tukey's Method for identifying outliers: An outlier step is calculated as 1.5 times the interquartile range (IQR). A data point with a feature that is beyond an outlier step outside of the IQR for that feature is considered abnormal.

- **Normalizing Numerical Features:**

  In addition to performing transformations on features that are highly skewed, it is often good practice to perform some type of scaling on numerical features. Applying a scaling to the data does not change the shape of each feature's distribution (such as **'delta ch(2)'** above); however, normalization ensures that each feature is treated equally when applying supervised learners.

Implementation

In this section of the project, we will develop the tools and techniques necessary for a model to make a prediction. Being able to make accurate evaluations of each model's performance through the use of these tools and techniques helps to greatly reinforce the confidence in the model's predictions.

Initial Model

I used a **Linear Discriminant Analysis (LDA)** classifier in my machine learning algorithm design as my initial model, without outliers' removal and dimensionality reduction. The model attained a score of about **55%**, which indicates a poor performance, but helped me in designing a general frame for the whole process

Final Model

The final achieved a great optimization and enhancements compared to the initial one. In this model I used another very popular classifier for BCI, which is the **support vector machine (SVM)**. In addition, I used the **IQR** score for outliers' detection and removal, also performed some dimensionality reduction using **PCA.** The model's implementation can be outlined in the following three main steps:

- **Shuffle and Split Data:**
  After scaling the data to a more normal distribution and has had any necessary outliers removed. As always, we will split the data (both features and their labels) after shuffling into training and test sets. 80% of the data will be used for training and 20% for testing.

- **PCA Transformation:**
  We then apply PCA to the data for dimensionality reduction, to fit the relatively small size of the data compared to its features number, also to discover which dimensions about the data best maximize the variance of features involved. In addition to finding these dimensions, PCA will also report the explained variance ratio of each dimension — how much variance within the data is explained by that dimension alone. Note that a component (dimension) from PCA can be considered a new "feature" of the space, however it is a composition of the original features present in the data.

- **Fitting The Model:**
  Our final implementation step requires that we bring everything together and train a model using the **support vector machine** algorithm to figure out the decision boundaries to be able to classify the states.

Some of the complications I faced was to decide which hyperparameters have the greatest effect on the classifier to tune. Also choosing the number of principal components for **PCA** was one of the difficulties until used the cumulative **Explained Variance** summation graph, when plotting this data it was such an easy task to select the number of components which preserves around **97%** or **98%** of the total variance of the data.

Refinement

Some optimizations performed and improvements are made upon the algorithms and techniques used in the implementation. This helped the model attain better accuracies.

- **Hyperparameters Tuning:**

  To ensure producing an optimized model, the model is trained using the **grid search technique** to optimize the **'gamma'** and **'C'** parameters for the **SVM** classifier:

  - **gamma:**
    - Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
    - Search values: [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1]
  - **C:**
    - Regularization parameter. The strength of the regularization is inversely proportional to C.
    - Search values: [1e3, 5e3, 1e4, 5e4, 1e5]

  Parameter **'gamma'** is **0.1** for the optimal model.
  Parameter **'C'** is **1000** for the optimal model.

- **Cross Validation:**

  In addition, the implementation is using **ShuffleSplit()** for an alternative form of cross-validation. While it is not the K-Fold cross-validation technique, this type of cross-validation technique is just as useful! The **ShuffleSplit()** implementation will create 10 **('n_splits')** shuffled sets, and for each shuffle, **20% ('test_size')** of the data will be used as the *validation set*.

IV. Results

Model Evaluation and Validation

Once the model has been trained on the given set of data, it can now be used to make predictions on new sets of input data. In the case of a **support vector machine**, the model should have learned **how the frequency bands values relate to the mental state of the driver** and can respond with a detection of the current **mental state**. We can use these predictions to gain information about data where the value of the target variable is unknown — such as data the model was not trained on (i.e. testing data or future readings of the driver's brain signals).

The model's average score was increasing with every pre-processing procedure added to the data. The number of principal components for **PCA** was decided from the cumulative sum of the **"Explained Variance"** graph, to select a number that could preserve something around 97% or 98% of the total variance of the data. This also improved the performance. In addition to refinement steps through **grid search** and **cross validation**, to optimize the model as more as possible. We reached our final model, with the best estimated parameters.

- **Evaluation:**

  The model attained a final average score of about **88%**, with a confusion matrix showing minimal mistakes for all classes' predictions. From the classification report the scores appeared to be uniformly distributed across the three mental state classes.

- **Validation:**

  For sensitivity analysis, the model was tested with: random shuffles before the train-test split of the data for many times, changing the size of data by varying the number of samples and manipulating the interquartile range value to control the percentage of outliers removal. The model achieved a **best** score of around **92%**, a **minimum** of **78%**, with **88%** on **average**.

Justification

Compared to the benchmark model, which achieved a 96.70% (best) and 91.72% (avg.) accuracy in experimental settings using a version of continuous performance task with SVM-based mental state detector. My model scores are close enough.

The model attained a testing accuracy of around **88%**, uniformly distributed on the three state classes, which indicates a good performance of the model that we can rely on for **mental state** detection.

V. Conclusion

Free-Form Visualization

Following is a plot of the **Confusion Matrix** for the tuned final **SVM** Model. This describes the model's performance in general, with detailed predictions for each **Mental State** class.

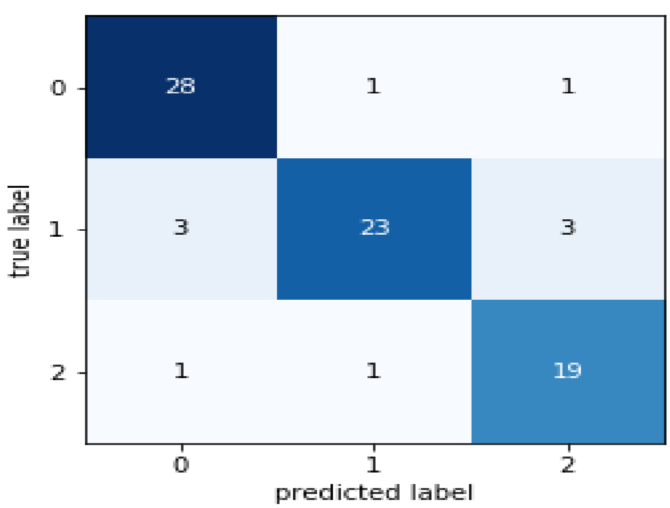


Fig. 11. Final model's Confusion Matrix.

From the plot, the model managed to detect correctly **28 “Focused”** states out of **30** for the 1st class, with only misclassifying the remaining two: one as **“De-Focused”** and the other as **“Drowsy”**. For the 2nd class the classifier managed to detect **23 “De-Focused”** states out of **29** correctly, misclassifying **3** as **“Focused”** and another **3** as **“Drowsy”**. Finally, for the 3rd class, it managed to correctly detect **19 “Drowsy”** states out of **21**, misclassifying one as **“Focused”** and the other as **“De-Focused”**.

This indicates a really good performance, with highest prediction score for the 1st class **“Focused”**, in the second place the score of the 3rd class **“Drowsy”** and for the last place the score of the 2nd class **“De-Focused”**. The variations between classes’ scores are small, with approximately uniform score distribution upon the three states, which is emphasized by the **Classification Report** bellow.

	precision	recall	f1-score	support
1	0.88	0.93	0.90	30
2	0.92	0.79	0.85	29
3	0.83	0.90	0.86	21
micro avg	0.88	0.88	0.88	80
macro avg	0.87	0.88	0.87	80
weighted avg	0.88	0.88	0.87	80

Fig. 12. Model’s Classification Report.

Reflection

The process used for this project can be summarized with the following steps:

1. The initial problem was defined and relevant public dataset was located.
2. The data was explored and analysed.
3. Data was pre-processed and features were extracted.
4. An initial model was trained and evaluated.
5. A further model was trained and refined.
6. The final model was evaluated.

From the initial exploration of the data in step 2, I envisaged that the pre-processing work in step 3 would be incredibly time consuming. However, this was actually relatively easy thanks to the Python tool scikit-learn. I also thought that the dimensionality reduction would be a lot trickier but again scikit-learn, shortened the effort required immensely.

Feature vector we extracted in signal processing step perform much better than I had expected. Welch's periodogram saved us a lot of detailed fourier transformation and electrical signal power averaging.

Overall, the model performed better than planned and reached close results to that achieved in the benchmark model.

Improvement

If we were to continue with this project there is a number of additional areas that could be explored:

- Test the model's performance with Real-time EEG signals.
- Train the model for real world data. This would likely involve augmenting the training data in various ways such as:
  - Adding a variety of different driving situations.
  - Adjusting the rest periods between tasks switching.
  - Changing the starting position of the recording sample.
- Experiment to see if per-class accuracy is affected by using training data of different record durations.
- Experiment with other techniques for feature extraction such as multitaper method.

References

[1] Fabien Lotte. A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces. Eduardo Reck Miranda; Julien Castet. Guide to Brain-Computer Music Interfacing, Springer, 2014. hal01055103: https://hal.inria.fr/hal-01055103/document

[2] Distinguishing mental attention states of humans via an EEG-based passive BCI using machine learning methods: https://www.sciencedirect.com/science/article/pii/S0957417419303926

[3] Compute the average bandpower of an EEG signal: https://raphaelvallat.com/bandpower.html?fbclid=IwAR2XL6sA1qYLU4J4bk0a5GjJ14JmoDV_yKPnOYJiEkRs-40orDA-GyH2oso