INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 5 - V7I5-1265)
Available online at: https://www.ijariit.com

Optimal EEG Channels for emotion detection in humans

Samarth Kulkarni
kulkarni.samarth@yahoo.com
Indus International School, Bengaluru, Karnataka

ABSTRACT

A brain-computer interface (BCI) is a system that allows communication between a subject's brain and a computer, without requiring any physical movement. The high-level process for any BCI system is signal acquisition, signal processing and providing output through a physical device. Signal acquisition is done by sensors with multiple channels using invasive or non-invasive methods. The collected data usually contains more than 256 unique channels across the standard 32 sensors. In addition to this, the data is often sampled at a rate of 256Hz for long periods of time, resulting in massive datasets. The critical next step is to preprocess these signals to understand the patterns and changes in the various types of brain waves. However, the intrinsic neurophysiological changes in the brain pose a challenge to the processing of these signals. Identifying the key sensor-channels becomes vital not only for the above reason but also due to the large size of the data points captured. This paper is a study on identification of optimal channels for the detection of an emotion such as Fear, one of the primary emotions frequently observed during multiple studies. By detecting channels with highly prevalent features, this study aims to reduce the size of data. The data so optimized can be output to speed up complex classification/prediction algorithms.

Keywords*– BCI, Signal Processing, EEG channels, Emotion Detection, MNE-Python*

1. INTRODUCTION

Since the art of writing was invented, humans have strived to improve the way information is communicated. With the advent of computers and the internet, a new need arose. The need to be able to communicate with computers, and in turn, with fellow humans around the world. There have been many different forms of keyboards, from ergonomic, to wearable. The next logical step is to do away with these devices, which are often large and hard to carry around, and replace them with devices that we can control with nothing but our thoughts, i.e. computer systems interfacing with our brain signals.

BCI systems hold important applications for disabled individuals, who suffer from neurodegenerative diseases such as Amyotrophic Lateral Sclerosis (ALS) or Spinal Muscular Atrophy (SMA). These can cause the loss of control over muscles (including vocal muscles), making it difficult to communicate. A BCI model could be trained to recognize what kind of signals the brain emits when a particular word is being thought of. The ability to communicate in this fashion would allow these individuals to be less dependent on others. Another, more futuristic application would be using a trained BCI model to control a form of exoskeleton. This would allow individuals with severe motor neuron degradation to move around by themselves, in an almost humanlike manner.

The applications of BCI systems also extend to the entertainment sector, especially in Virtual and Augmented reality. BCI systems could make the use of such virtual environments and help disabled individuals simulate actions that they would perform in the real world. They can also be used to diagnose patients suffering from neurological irregularities and include workload and concentrationassessments.

Brain signal acquisition for these applications start with the use of the sensors either with an invasive or non-invasive technique. Invasive techniques require the sensors to be placed surgically onto the brain directly. This requires the presence of a medical professional and a subject who is willing to undergo the procedure. Two invasive modalities can be found in BCI research: electrocorticography, which places electrodes on the surface of the cortex, either above the dura mater (epidural electrocorticography) or under the dura mater (subdural electrocorticography).

These methods were introduced in an effort to improve the quality of brain signals monitored by BCIs. Though this technique yields high quality data, the signal from such a procedure deteriorates over time due to the buildup of scar tissue. Additionally, invasive methods are generally more expensive and require very experienced surgeons or advanced machinery, in most cases, both.

These procedures also involve significant health risks, which restricts their use to experimental settings. In non-invasive techniques, the sensors are placed on top of the scalp and require no surgical procedure. They are also much cheaper, and do not need medical intervention. Due to this, extensive research has been conducted and made available using data from non-invasive sensors. This approach has successfully been used by severely and partially paralyzed patients to reacquire basic forms of communication and to control neuro-prostheses and wheelchairs. These advantages made data from Non-Invasive procedures the natural pick for my research.

Amongst the main technologies used in the process of non-invasive BCIs are electroencephalogram (EEG), functional magnetic resonance imaging (fMRI), magneto-encephalography (MEG), near infrared spectroscopy (NIRS), and functional transcranial Doppler sonography (FTCD). EEG data is easy, fast, safe, and cheap to acquire, and is characterized by low spatial resolution (precision of measurement on how deep in the brain the signal originated), but high temporal resolution (precision of a measurement with respect to time), which makes it ideal for studies (Fig 1). EEG measures the electrical activity in our brain using electrodes (sensors) placed on our scalp. Each sensor consists of multiple channels, which capture brain signals of different wavelengths (Measuredin ranges labelled Gamma, Beta, Alpha, Theta and Delta).

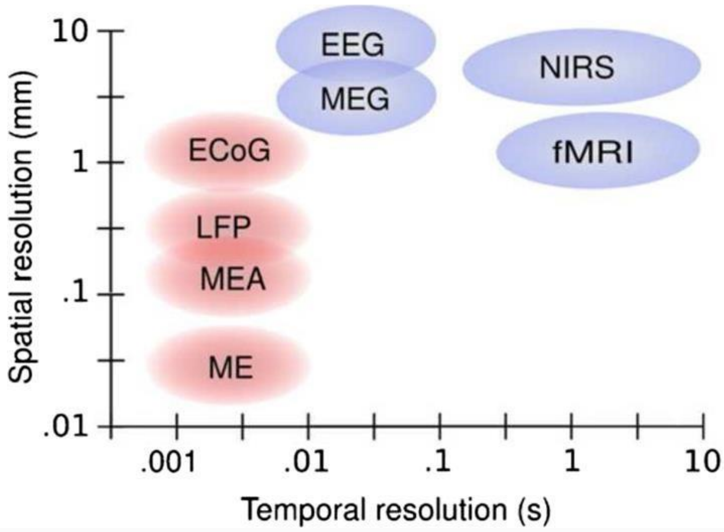


Fig 1 - Comparison of spatial and temporal resolution (Hill et al. 2006b)

A prerequisite to building an effective BCI system is being able to identify channels with the most prevalent features. This paper discusses the process to identify these EEG sensor-channels from EEG datasets to limit the amount of data to be analyzed, speeding up the process of classification. Fig 2 below shows the high-level Process flow diagram of a BCI System

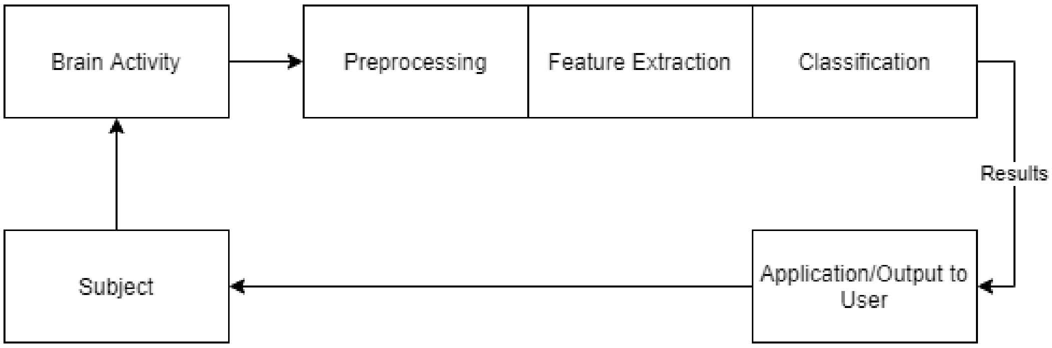


Fig 2 - High Level Process flow diagram of a BCI System

2. RELATED WORK

A study conducted by Turky Alotaiby, Fathi E Abd El-Samie, Saleh A Alshebeili and Ishtiaq Ahmad (found at SpringerOpen), concluded that “[the] performance of the classification/detection tasks, to make use of a small set of EEG channels ranging from 10 to 30 % of the available channels”, remains largely unchanged when compared to using all channels available.

Another study conducted by Jian Kui Feng, Jing Jin, Ian Daly, Jiale Zhou, Yugang Niu, Xingyu Wang, and Andrzej Cichocki (found at Hindawi) proposed an advanced method of choosing relevant channels “based on common spatial pattern (CSP) rank channel selection”, which further improve upon the accuracy of classifiers.

3. PROPOSED METHODOLOGY

My process differed from the above cited studies. With a particular focus on the detection of fear, I identified the major feature to be a large spike in the center of the dataset. This peak was then used as reference in order to detect similar features in other channels. The diagram below (Fig 3) shows the process that I followed over the course of my study.

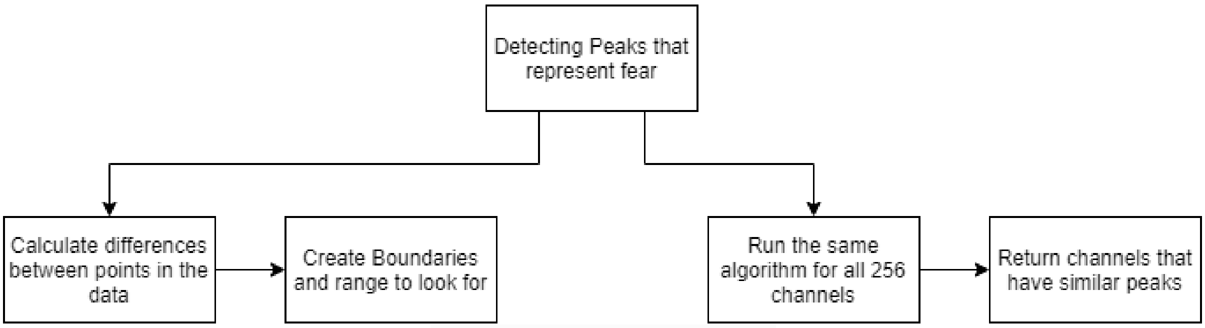


Fig 3 – Proposed Methodology

3.1 EEG Data and Data Collection

Most current BCIs obtain the relevant information from the brain activity through electroencephalography (EEG). Electroencephalography is by far the most widely used neuroimaging modality, owing to its high temporal resolution, relative low cost, high portability, and few risks to the users. BCIs based on electroencephalography consist of a set of sensors that acquire EEG signals from different brain areas. However, the quality of EEG signals is affected by scalp, skull, and many other layers as well as background noise. Noise removal is key to electroencephalography and to other neuroimaging methods, insofar as it reduces the Signal to Noise Ratio (SNR) and therefore the ability to extract meaningful information from the recorded signals.
The data from the dataset (link at the bottom of this paper), is from a study carried out at UCSD, where participants were played a track that prompted a certain emotion. Fear was one of these emotions being tested for.

There were a total of 31 unique individuals tested. The EEG data I was working with was stored in the form of a .BDF file and had a sampling rate of 256Hz. The data could be extracted into a four (4) dimensional array using python's *MNE* package. We can also extract events from this same file, giving us the exact timestamp when a certain clip was triggered, and when the track finished.

3.2 Signal Processing

The graph below (Fig 4) shows the signal in the section that refers to fear. This signal is represented by 38423 unique points, for a total duration of 150.1 seconds (out of a total of 4452 seconds recorded for this subject). This is just a mere 3% of the entire signal. Additionally, there are 255 more channels with the same length of data. If all this data is run through a classification algorithm at once, it takes a long time to accurately process them and would take even longer on machines with limited processing power.

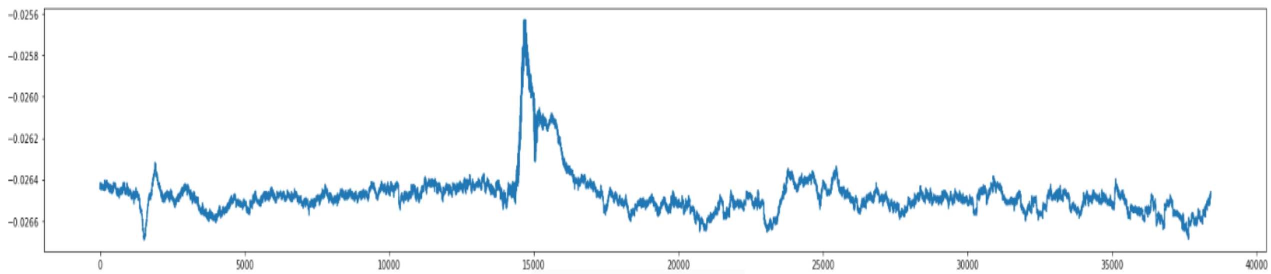


Fig 4 – Fear Signal

Fear is a sudden emotion, which shows up as the largest peak in the system. As shown in the graph (Fig 4 and Fig 5), this occurs just before the 15000 mark. The objective was to now detect the magnitude of the peak and create bounds that would isolate channels that had similar peaks duringthe same timeframe.

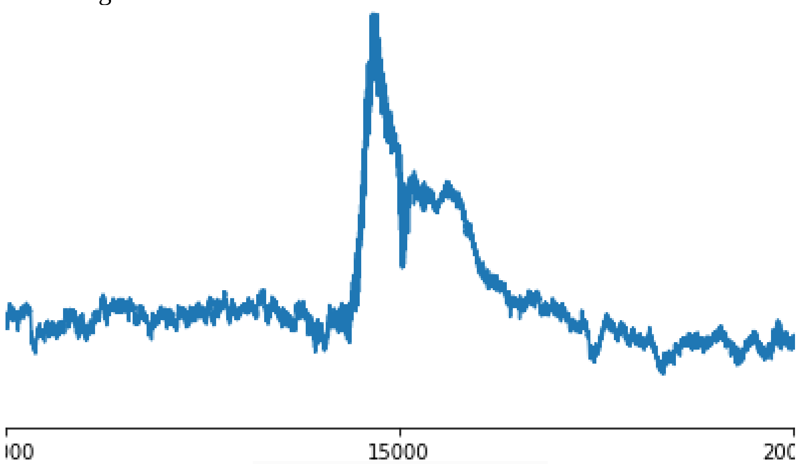


Fig 5 – Fear Signal (Peak)

There was also the task of removing any non-relevant peaks. The problem lay in the number of samples the signals consist of. While plotted at the above size, the internal peaks look like a lack of antialiasing, however, zooming in on the peak results in a signal like the one below (Fig 6). There are a lot of tiny peaks, most of which would interfere with our data if no lag was introduced.

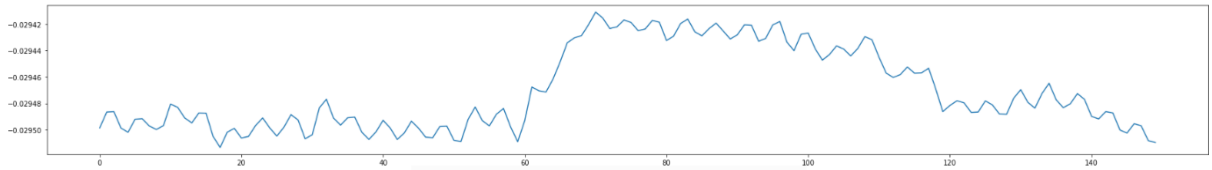


Fig 6 – Fear Signal (zoom on peak)

Even the larger peak that I was targeting (which is the large jump in the center of the graph) was made up of two smaller peaks, which makes it difficult to measure the total difference in height. This can be solved in two ways: Down-sampling to a point where the peaks are merged or introducing a parameter in the algorithm that allows it to ignore the previous few readings when observing a peak. I tried both these methods.

The Algorithm

My objective was to detect the channels in the signal that were best suited for the detection of fear. The first task to be completed was establishing the range between which a peak could be categorized as "fear". The diagram below (Fig 7) shows the process I followed to develop this part ofthe algorithm.

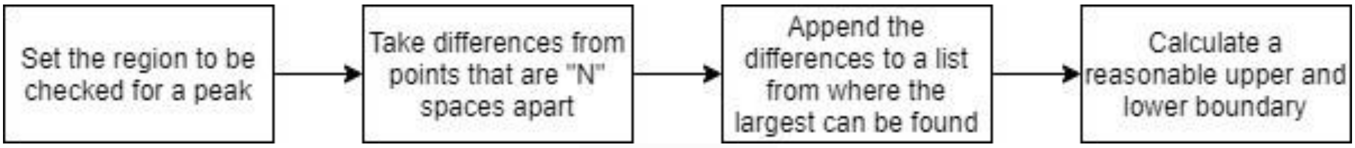
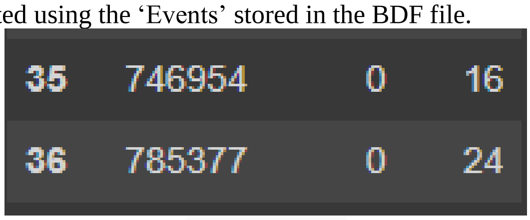


Fig 7 - Algorithm

The region to be checked could be selected using the 'Events' stored in the BDF file.



35	746954	0	16
36	785377	0	24

Fig 8 – Events

These are the timestamps for one of the files I was referencing. The code "16" (Fig 8) refers to the start of the track depicting fear. The code "24" (Fig 8) represents a 'recall' or a return to normal. Looking within this range will give us the peak that attributed to fear.

Step 1: Detecting Peaks

To detect the peak, I used the difference between points. However, using only the point behind the current one led to different peaks being detected.

Introducing a lag of 500 timesteps, allowed the algorithm to ignore the smaller peaks and vastly reduced the selection of erroneous channels. The graph below (Fig 9) shows the result, with a reddot placed at the highest peak.

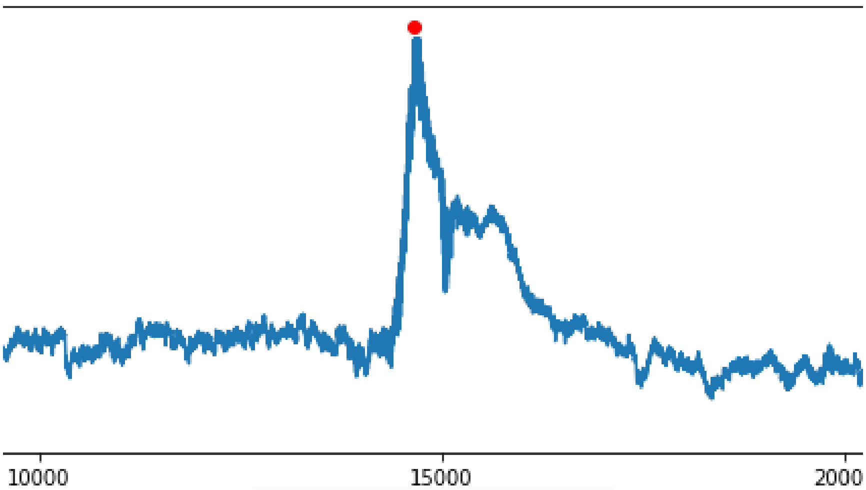


Fig 9 – Result (Detecting Peaks)

The characteristics of this peak are also stored, including its position, value and the difference between the two points taken. Another approach is to down-sample the dataset before performing this algorithm. This can help drastically improve processing times, but reduces the resolution of the data, making it harder to run prediction algorithms on it. The lag introduced in the previous method can also be removed as the number of samples being taken has been drastically reduced, and there are no longer any sub-peaks under a major peak.

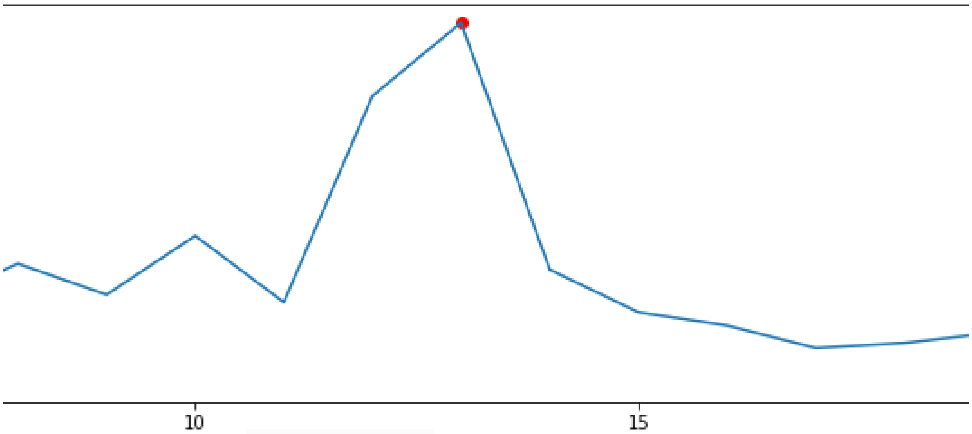


Fig 10 – Graph (Down sample the dataset)

Applying these changes results in a graph like the one above (Fig 10). Again, the red dot represents the highest peak found by the algorithm. The reason I chose to work with the first method was that the higher resolution of data allowed for more precise measurements of the difference between the points that made up a peak. It also made it easier to apply to an entire channel when searching for fear.

Step 2: Establishing the upper and lower boundaries

This step is relatively simple. I decided a fixed percentage above and below the peak just detected (from the first channel). This was initially set to 20% above and below the range. However, this resulted in a lot of false positives.

To narrow this down, I reduced the boundaries, both manually and programmatically, till it started losing good channels. Reducing the boundaries from 20% on both sides to 10% on both sides drastically narrowed down and improved the results. A 20% boundary returned 48 unique channelsas having the features I was looking for.

Reducing this to 10% allowed the algorithm to return just 9 unique channels, each with explicit features. There is still room for this property to be tuned, but this is within the accuracy we can workwith.

Step 3 & 4: Finding the channels

These steps were used to both identify significant channels and perfect the boundaries found in step 2. The algorithm used here was similar to the algorithm in step 1. The primary difference was that instead of exiting after storing the maximum difference and index in an array, the value was compared to the boundaries I established in step 2.

If the value was within the range, the channel number and time-step of the peak were stored. Going back and forth between these steps and step 2 allowed me to tweak the boundaries until it reachedacceptable levels of accuracy.

RESULTS

The result of these steps of processing enabled me to identify the channels with the most prevalent features related to Fear. These channels can be further used to train Machine Learning or Artificial Intelligence algorithms to detect Fear. Using only these channels drastically reduces the total data that a prediction/classification algorithm needs to work with and can improve training duration when run in real time. In a standard 10-20 sensor layout, the channels returned by the algorithm are:

0, 7, 73, 106, 204, 143, 151, 249, 250

These channels have been manually verified to contain prevalent features that we require for thedetection of fear as well.

CONCLUSION AND FUTURE SCOPE

The most relevant 9 channels out of a total of 254 have been identified for detection of the emotion ‘Fear’. This process can actively help in reducing the amount of data that a BCI system would need to process to take the final action while maintaining the accuracy of the system. The algorithms and methods involved in each step of this project were chosen after careful consideration and trials. I have implemented these methods to achieve a balance between speed and accuracy. The analysis methodology can be extended to the detection of other emotions such as Happiness, Sadness, etc. It can be utilized by any BCI system that would like to reduce the time taken to understand a user’s emotions such as sentiment analysis while a user is on social media, in gaming, psychology, medicine and education.

Dataset

Main Link: https://headit.ucsd.edu/studies/3316f70e-35ff-11e3-a2a9-0050563f2612

Data-file Used for this paper: https://headit.ucsd.edu/recording_sessions/0fd04a38-3600-11e3-a2a9-0050563f2612

Notebook

Link: https://colab.research.google.com/drive/1c6SN3nDcbJbafd0ZdW4r74vciQ8QB7pO?usp=sharing

REFERENCES

[1] Anatomy of the Brain. (2018, April 1). Retrieved from Mayfield Brain & Spine:https://mayfieldclinic.com/pe-anatbrain.htm
[2] Barjinder Kaur, D. S. (2018). EEG Based Emotion Classification Mechanism in BCI.
[3] David A. Moses, M. K. (2019, July 30). Real-time decoding of question-and-answer speech dialogue using human cortical activity. Retrieved from Nature Communications: https://www.nature.com/articles/s41467-019-10994-4
[4] David Feess, M. M. (2013, July 2). Comparison of Sensor Selection Mechanisms for an ERP-Based Brain-Computer Interface. Retrieved from PLOS One: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0067543
[5] Edgar P. Torres, E. A.-Á. (2020, September 7). EEG-Based BCI Emotion Recognition: A Survey.
[6] Retrieved from National Institutes of Health: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7570756/
[7] Haiyun Huang, Q. X. (2019). An EEG-Based Brain Computer Interface for Emotion Recognition and Its Application in Patients with Disorder of Consciousness.
[8] How to Reshape Input Data for Long Short-Term Memory Networks in Keras. (2017, August 30).
[9] Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/reshape- input-data-long-short-term-memory-networks-keras/
[10] How to Update LSTM Networks During Training for Time Series Forecasting. (2017, April 14).
[11] Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/update- lstm-networks-training-time-series-forecasting/
[12] Luis Fernanto Nicolas-Alonso, J. G.-G. (2011, December 29). Brain Computer Interfaces, a Review.
[13] Retrieved from MDPI: https://www.mdpi.com/1424-8220/12/2/1211/htm
[14] Meazzini, L. (2020, January 21). Everything you need to know about Time Series. Retrieved from Towards Data Science: https://towardsdatascience.com/everything-you-need-to-know-about-time-series-5fa1834d5b18
[15] Step-by-Step Guide — Building a Prediction Model in Python. (2020, October 15). Retrieved from Towards Data Science: https://towardsdatascience.com/step-by-step-guide-building-a- prediction-model-in-python-ac441e8b9e8b
[16] Taciana Saad Rached, A. P. (2012, June 27). Emotion Recognition Based on Brain-Computer Interface Systems. Retrieved from Intechopen: https://www.intechopen.com/books/brain-computer- interface-systems-recent-progress-and-future-prospects/emotion-recognition-based-on- brain-computer-interface-systems
[17] Ulrich Hoffmann, J.-M. V. (2007, November 1). Recent Advances in Brain-Computer Interfaces.
[18] Retrieved from Researchgate.net: https://www.researchgate.net/publication/37452383_Recent_Advances_in_Brain-Computer_Interfaces

