

Βιοπληροφορική και Νευροπληροφορική

Πρόγραμμα Σπουδών

Διπλωματική Εργασία

Διαφοροποίηση σημάτων ΗΕΓ για οπτικά ερεθίσματα: Προκλήσεις
στην ταξινόμηση ψηφίων

Αύγουστος Τσαμουργκέλης

Επιβλέπων καθηγητής: Καθηγητής Αδάμ Αδαμόπουλος,

Τμήμα Ιατρικής, Δημοκρίτειο Πανεπιστήμιο Θράκης

Πάτρα, Ιούνιος 2024

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του/της φοιτητή/φοιτήτριας («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Διαφοροποίηση σημάτων ΗΕΓ για οπτικά ερεθίσματα: Προκλήσεις
στην ταξινόμηση ψηφίων

Αύγουστος Τσαμουργκέλης

Επιτροπή Επίβλεψης Πτυχιακής / Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:

Καθηγητής Αδάμ Αδαμόπουλος

Τμήμα Ιατρικής

Δημοκρίτειο Πανεπιστήμιο Θράκης

Συν-Επιβλέπων Καθηγητής:

Νευροχειρουργός Αθανάσιος Αλκίνοος,

Μέλος ΣΕΠ

Ελληνικό Ανοικτό Πανεπιστήμιο

Πάτρα, Ιούνιος 2024

Περίληψη

Η ηλεκτροεγκεφαλογραφία (ΗΕΓ) υπάρχει από τις αρχές του 20ου αιώνα. Έχει αποδειχθεί ζωτικής σημασίας εργαλείο για ηλεκτροφυσιολογικές μελέτες καταστάσεων όπως η επιληψία. Πρόσφατα αναζωογονήθηκε, καθώς αναπτύχθηκε ο τομέας της μηχανικής μάθησης, διευρύνοντας τη χρησιμότητά της μεταξύ μιας πληθώρας νευρολογικών παθήσεων καθώς και σε εφαρμογές διεπαφής εγκεφάλου-υπολογιστή (BCI). Η παρούσα μελέτη εμβαθύνει στην περίπλοκη διαδικασία ταξινόμησης των σημάτων ΗΕΓ που προκαλούνται από οπτικά ερεθίσματα του υποκειμένου που βλέπει τα ψηφία 0 και 1 και μια κενή οθόνη. Επικεντρωνόμαστε στην ανάπτυξη μιας ολοκληρωμένης ροής εργασίας για την προεπεξεργασία του EEG, καθώς και την εξαγωγή χαρακτηριστικών και την ταξινόμηση του σήματος. Επιτυγχάνουμε ισχυρές δυνατότητες διαφοροποίησης μεταξύ ψηφίων και μη ψηφίων με όλους τους αλγορίθμους ταξινόμησης. Ωστόσο, η μελέτη μας αναδεικνύει επίσης τις σημαντικές νευρολογικές προκλήσεις που αντιμετωπίζονται κατά τη διάκριση μεταξύ των ψηφίων, καθώς το μοντέλο μας, εμπνευσμένο από τη σχετική βιβλιογραφία, δεν ήταν σε θέση να διαφοροποιήσει τις τιμές 0 και 1. Τα ευρήματα αυτά υπογραμμίζουν την πολυπλοκότητα της αριθμητικής επεξεργασίας στον εγκέφαλο, αποκαλύπτοντας κρίσιμες πληροφορίες σχετικά με τους περιορισμούς και τις δυνατότητες της ταξινόμησης ψηφίων με βάση το EEG και αποκαλύπτει επίσης την ανάγκη για διαύγεια στην κοινότητα της βιοπληροφορικής.

Λέξεις – Κλειδιά

Ηλεκτροεγκεφαλογράφημα, Βιοπληροφορική, Εξαγωγή Χαρακτηριστικών, Μηχανική Μάθηση, Ταξινόμηση.

Differentiation of EEG Signals for Visual Stimuli: Challenges in Digit Classification

Augoustos Tsamourgelis

Abstract

Electroencephalography (EEG) has been around since the early 20th century. It has proven a vital tool for electrophysiological studies of conditions like epilepsy. Recently it has been revitalized, as the field of machine learning has been developing, widening its usefulness among a plethora of neurological conditions as well as in brain-computer interface (BCI) applications. This study delves into the intricate process of classifying EEG signals elicited by visual stimuli of subject viewing digits 0 and 1, and a blank screen. We focus on developing a comprehensive workflow for EEG preprocessing, as well as feature extraction, and signal classification. We achieve strong differentiation capabilities between digit and non-digit values in all classification algorithms. However, our study also highlights the profound neurological challenges encountered in distinguishing between the digit values, as our model, inspired from related bibliography, was unable to differentiate between digit values 0 and 1. These findings underscore the complexity of numerical processing in the brain, revealing critical insights into the limitations and potential of EEG-based digits classification and also reveals the need for clarity in the bioinformatics community.

Keywords

Electroencephalography, Bioinformatics, Feature Extraction, Machine Learning, Classification.

Περιεχόμενα

Περίληψη.....	iv
Abstract	v
Περιεχόμενα	vi
Κατάλογος Εικόνων / Σχημάτων	vii
Κατάλογος Πινάκων	viii
1. Introduction	1
2. Related work on EEG-based digit classification.....	2
3. Materials and Methods	5
3.1. Dataset description	6
3.2. Data preprocessing	7
3.2.1. Resampling.....	7
3.2.2. Butterworth filter.....	8
3.2.3. Artifact removal	9
3.2.4. Z-score normalization	11
3.3. Feature extraction	12
3.3.1. Coherence.....	12
3.3.2. Granger Causality.....	14
3.3.3. Directed Transfer Function	16
3.3.4. Partial Directed Coherence	18
3.3.5. Transfer Entropy	20
3.3.6. Empirical Wavelet Transform.....	22
3.3.7. Wavelet Scattering Transform	25
3.4. Classification	27
3.4.1. Support Vector Machine	28
3.4.2. Extreme Gradient Boosting.....	29
3.4.3. Naïve Bayes	30
3.4.4. Artificial Neural Networks.....	31
3.4.5. Convolutional Neural Networks	32
3.4.6. Long Short-Term Memory networks	33
4. Results	35
5. Discussion	38
5.1. Comparison to previous works	39
5.2. Neurophysiological perspective	40
6. Conclusions	43

Κατάλογος Σχημάτων

Figure 3.1 Placement of EEG electrodes from EPOC (in blue) and MU (in red).....	6
Figure 3.2 Histogram plots for EEG signal lengths before and after resampling	8
Figure 3.3 Signal sample before and after Butterworth filtering	9
Figure 3.4 ERP signal before and after ATAR	11
Figure 3. 5 ERP signal before and after normalization	12
Figure 3. 6 Coherence matrices for a single trial of digit 0 for the 14 channels.	14
Figure 3. 7 Granger causality score for the 150 trials for digit 0 for the 14 channels.....	16
Figure 3. 8 DTF matrices for a single trial of digit 0 for the 14 channels.....	18
Figure 3. 9 PDC matrices for a single trial of digit 0 for the 14 channels.	20
Figure 3. 10 TE matrices for a single trial of digit 0 for the 14 channels.	22
Figure 3. 11 Plotting of the original sample signal and its 4 different frequency segments.	25
Figure 3. 12 Plotting of the original sample signal and its 0 th , 1 st and 2 nd scattering components.	27
Figure 3.13 Workflow diagram for the LSTM unit.	35

Κατάλογος Πινάκων

Table 4.1 Validation accuracies from the EPOC and MU datasets.....	37
--	----

Abbreviations

Electroencephalography (EEG),
Brain-Computer Interface (BCI),
Granger Causality (GC),
Directed Transfer Function (DTF),
Partial Directed Coherence (PDC),
Transfer Entropy (TE),
Empirical Wavelet Transform (EWT),
Wavelet Scattering Transform (WST),
Support Vector Machine (SVM),
Naïve Bayes (NB),
eXtreme Gradient Boosting (XGBoost),
Artificial Neural Networks (ANN),
Convolutional Neural Networks (CNN),
Long Short-Term Memory (LSTM)

1. Introduction

Electroencephalography (EEG) is a non-invasive method for recording the electrical activity of the brain (Gasser & Molinari, 1996). Utilizing scalp electrodes EEG measures the cumulative electrical potentials generated by neural activity, offering excellent time resolution in the order of milliseconds at the cost of low spatial resolution. Algorithmic techniques like surface Laplacian, combined with a high number of channel scalp electrodes are able to increase the temporal resolution, yet not at the level of techniques like Positron Emission Topography or functional Magnetic Resonance Imaging (Burle et al., 2015). Neuronal activity results in oscillations across a frequency range mainly between 0.5Hz and 50Hz. This range is empirically divided into different frequency bands that include delta (0.5-4Hz), theta (4-7 Hz), alpha (8-12 Hz), beta (13-30 Hz), and gamma (greater than 30Hz) (Nayak & Anilkumar, 2024).

Resting EEG serves as a diagnostic tool for a plethora of conditions like epilepsy, dementia and different psychiatric disorders (Noachtar & Rémi, 2009; Micanovic & Pal, 2014; Park et al., 2021). Also, the EEG is recorded while stimuli are presented in order to capture event-related potentials (ERP), with this protocol requiring repeated trial stimulations in order for the ERPs to be represented with a high signal to noise ratio. Stimuli lead to EEG activity and averaging among many trials provide the characteristic waveform components for each stimulus. Positive peaks are represented with P and a number representing the time in milliseconds of the peak of the wave (i.e. P300) and negative peaks are represented with an N (Sharp et al., 2011).

Transient external visual stimulation, such as checkerboard patterns or flashes of light, may be recorded with trial EEG acquisition in the form of visual evoked potentials (VEPs), which are the expression of neuronal activity of the visual pathways from the optic nerve to the calcarine occipital cortex (Di Russo et al., 2002). The time locked pattern of VEPs, provide functional information for the optical pathways. In the case of demyelinating lesions localized in the optic nerve in multiple sclerosis, prolonged latencies of the VEPs, mainly the P100, were clearly recorded highlighting the importance of VEPs in medical diagnosis among neurodegenerative diseases affecting the visual pathways (Calugaru et al., 2016).

Steady state visual evoked potentials (SSVEPs) are elicited by periodic, continuous visual stimuli. Increased attention and focus to sensory activation leads to increased SSVEP

amplitude partially attributable to rising synchronous neuronal activity (Joon Kim et al., 2007). SSVEPs have proven important in the field of brain-computer interface (BCI). BCI utilizes technologies that allow for the non-physical connection of human subjects and prostheses or assistive devices for patients with physical disabilities and motor impairment (Pattern, 2022). The basis of BCI is the automatic algorithmic classification of EEG recordings based on the features of neuronal activity. SSVEPs based BCI experiments indicate it many potential uses such as wheelchair movement control in physically impaired patients and for online communication patients with locked-in syndrome (complete paralysis with unaffected cognitive functions) (Siribunyaphat & Punsawad, 2023)(Lesenfants et al., 2014).

Recent advancements in EEG analysis have shown its utility in decoding different perceptual and attentional states. List et al. (2017) were able to decode the perception of face versus non-face stimuli from trial-based EEG analysis. They were also able to distinct states between locally-focused versus globally-focused attention using text based images. Magnetoencephalography (MEG), sharing many common principals with EEG, measuring magnetic fields generated by neuronal electrical activity, thus providing better spatial resolution, has successfully been used for word and phrase decoding of audio stimuli (Défossez et al., 2023). These results are suggestive of the distinct feature patterns of lexical and contextual semantic representations taking place in neuronal activity during listening. Also, EEG data has proven to contain information that allow for the classification of different images viewed by the subject that the data is being collected from (Spampinato et al., 2017).

In this work we focus on the novel field of digit classification from EEG data. This study aims to investigate the efficacy of various EEG-based binary classification methods in distinguishing between simple visual stimuli of digits between 0 and 1, and between digits and blank screen. First we are going to discuss current related work focusing on areas of improvement. After, we are going to mention the EEG data bank of MindBigData. Last, we are going to present the different methods used for preprocessing, feature extraction and machine learning classification. Python was used for all computations and the source code is provided for review separately. We are going to close with some neuroscientific interpretations of our results.

2. Related work on EEG-based digit classification

Machine learning tools paired with EEG analysis may be used for classification of different visual stimuli. There have been many attempts for digit classification of visual stimulation EEG recordings collected while viewing digits on a monitor. It is important to note that the high complexity of the different steps of EEG preprocessing, feature extraction and deep-learning classification pose risks regarding the integrity and reproducibility of the results. It is of importance to note that all the following papers and their results are based on the same data bank of MindBigData MNIST of BRAIN DIGITS¹.

Simple methods of convolutional neural networks (CNN) and recurrent neural networks (RNN) have been used claiming accuracies north of 70% for the 10 digit classification problem. Mishra et al. (2021) used a simple CNN with 2 single dimension convolution layers followed by a fully connected dense layer on the raw data (after filtering, noise reduction and channel selection). Using this architecture they were able to achieve 70% classification accuracy. Also, in another study using combinatory architecture of CNN and generative adversarial network (GAN) on the raw data (after normalization) an accuracy for digit classification of 95.4% was achieved (Khaleghi et al., 2023). Using different feature extraction algorithms instead of feeding preprocessed signal data in the CNN propose classification accuracies north of 90%. Kumari et al. (2021) used short time Fourier transform as a feature extraction achieving 91.29% accuracy, while Tiwari et al. (2023) used energy-frequency-time spectrum obtained using the Hilbert-Huang Transform for feature extraction achieving accuracy of 98.97%.

However, high accuracies for difficult classification problems using somewhat simple algorithms should always be reevaluated and reproduced. A common pitfall of extensive preprocessing, feature extraction and deep-learning classification is incorrect matrix dimension handling and transposition. It is of great importance to avoid data leakage. Providing information to the deep-learning algorithm from different trials while it is supposed to only receive input from single trial data will allow the algorithm to cheat and lead to high training and also testing accuracies. This is shown by the work of Perez (2023) for digit classification in which by incorrect dimension handling the Recurrent

¹ <https://mindbigdata.com/opendb/>

Convolutional Neural Network (R-CNN) provides the label not from the digits but from the channels. This effectively performs channel classification thus the 88% accuracy achieved is to be expected. In the work of Chown et al. (2024) and their published code² using a similar combinational RNN-CNN they were unable to surpass randomness with accuracy of 10%.

This discrepancy between similar machine learning architectures performed on the same data bank rise suspicions for the validity of the claimed results and highlights the importance of source code publication following any bioinformatics programming research. Chen et al. (2024) in a great undertaking performed benchmarks among 15 different deep-learning classification algorithms with most barely surpassing randomness for digit classification with their most successful novel EEGNet8,2 model (combination of 2 dimensional convolution and depthwise 2-dimensional convolution) providing the highest classification accuracy of 16.69%. It is immensely valuable to compare different classification problems side by side, particularly those that were able to surpass randomness with statistically significant results. This is somewhat reassuring for the validity of their code structure, yet source code publication would be appreciated. Using the similar but improved algorithm of EEGNeX Hang (2023) suggest in their PhD dissertation submission, a digit classification accuracy of 22.42%.

In the work of Muglikar (2021) they compared different classification algorithms (Long-short term memory network (LSTM), Fully connected network (FCN), Transformer, Random Forest, K-nearest neighbors (KNN)) with the greatest being the Transformer algorithm (which is an attention based model that relies in encoder and decoder architecture) providing digit classification accuracy of 19.42% for the MindBigData MNIST EPOC dataset and of 17.72% for the MindBigData MNIST MU dataset. Interestingly, similarly with our results as later discussed, they were able to achieve increased accuracies using the Transform algorithm north of 98% for binary calcification between digits and non-digits (All algorithmic methods provided accuracies north of 90%). Similarly in their work, Kalita (2023) using a simple CNN, they performed binary classification between digit and non-digits with accuracy of 95%. However, in their source code that they generously provided, they seem to perform EEG averaging across all channel values which is a technique not usually performed in EEG analysis due to

² <https://github.com/zchown/MindMnist-8BClassification>

temporal and spatial information distortion. Yet this method does not seem to result in any data leakage, thus the validity of their results is not disputed.

In an extensive study comparing different classification algorithms between different classification problems, digit classification of the MindBigData dataset yielded an overall accuracy of 31.35% using a novel classification algorithm (Bird et al., 2019). The strongest algorithm used was a Deep evolutionary classifier which is an approach that combines principles from deep learning and evolutionary algorithms inspired by natural selection. Finally they optimized the algorithm with adaptive boosting which is a method often performed to weak classifiers performing slightly better than random guessing to increased performance. It is important to state that in this study, using the same algorithm to two different classification problems yielded mental state classification accuracy of 79.7% and emotional state classification accuracy of 96.23% highlighting the overall complexity of digit classification.

3. Materials and Methods

3.1. Dataset description

In our study we use the MindBigData, the “MNIST” of Brain Digits data set, as is the case for the previously mentioned studies. The data collected are from a single Test Subject (named David Vivancos) capturing a 2 second visual stimulus while viewing digits on a screen from 0 to 9. Also, a portion of recordings correspond without the stimulus of seeing the digits indexed as -1 in the files. It is important to note that only commercial grade EEG capture devices were used and in our case we focused on the data captures from the Emotiv EPOC (EPOC) and the Interaxon Muse (MU) device. The EPOC device captured EEG signals from 14 channels (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) with a sampling rate of 128Hz, while the MU device provides data from 4 channels (TP9, FP1, FP2, TP10) with a 220Hz sampling rate. Electrode placement is depicted in Figure 3.1.

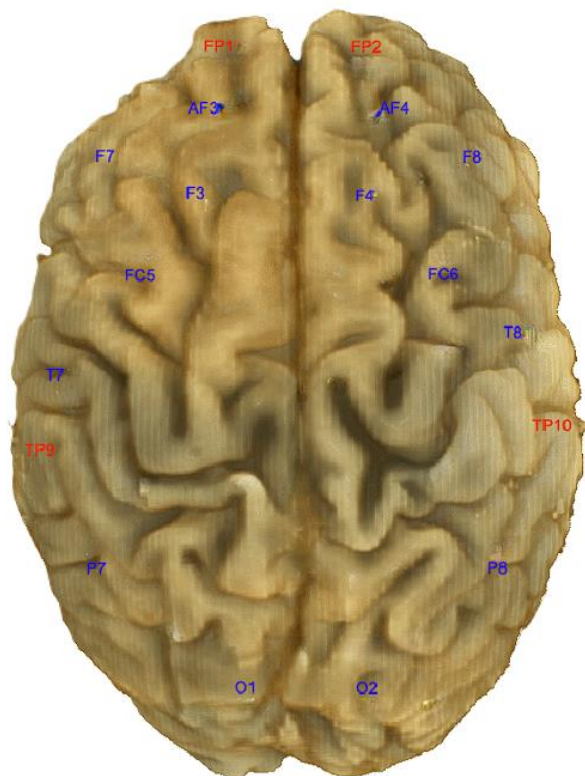


Figure Error! No text of specified style in document.1 Placement of EEG electrodes from EPOC (in blue) and MU (in red)

The digits appeared in random order on screen for a total of 2 seconds with a black screen between each image. For the EPOC set 910,476 total EEG recordings were captured while for the MU set the total number was 163,932. The construction of the dataset was an immense undertaking demonstrating the dedication of the researcher. The EPOC set contains around 90,000 recordings for each digit and 2,226 blank ones and the MU set contains around 12,000 recordings for each digit and 44,412 blank ones. Due to computational restraints we are going to be working with the binary classification problem with 150 trial recordings for digits 0, 150 for 1 and 150 for blank recordings (-1).

3.2.Data preprocessing

3.2.1. Resampling

Since there exist discrepancies with fluctuating sampling rates in the data which results in data for the 2 seconds recording period, we perform resampling using linear interpolation to achieve uniform sampling rates (128Hz for the EPOC dataset and 220Hz for the MU). This involves interpolating between existing data points to generate data at the desired sampling rate.

First we create out new time points: $t_i = \frac{i}{fs}$

Where:

- i is the original timepoints
- fs is the desired sampling frequency

For each new time point t_i we find the two original datapoints (x_k, y_k) and (x_{k+1}, y_{k+1}) such that

$$x_k \leq t_i \leq x_{k+1}$$

Then we interpolate using the formula:

$$y_i = y_k + (y_{k+1} - y_k) \frac{t_i - x_k}{x_{k+1} - x_k}$$

Focusing on the EPOC dataset, we can see the fluctuating sampling rate with a histogram depicting the time points of each signal (Figure 3.2). Given that the sampling rate is 128Hz for a 2 second recording we identify a significantly skewed mean value of around 260 time points. We can simply define a function using the `interp1d` function from the `scipy.interpolate` module in order to resample at the desired value of 256 time points per recording (i.e. 128Hz sampling rate).

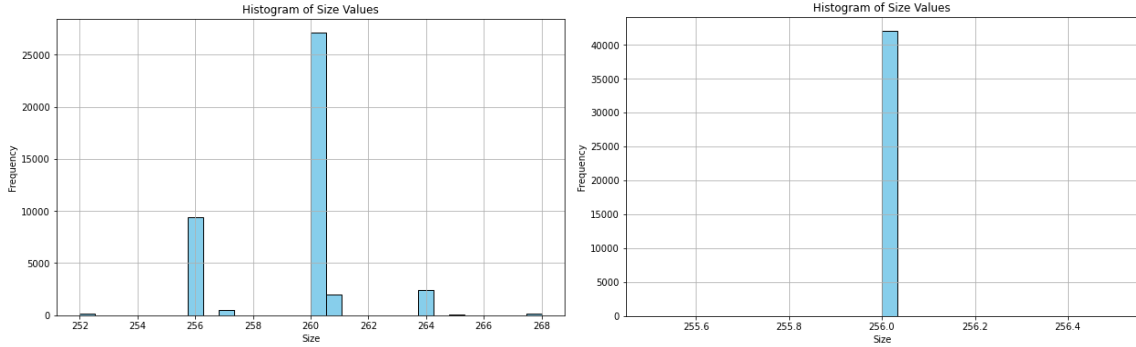


Figure 3.2 Histogram plots for EEG signal lengths before and after resampling

3.2.2. Butterworth filter

The researcher states that almost all EEG recordings were manually selected to avoid blinking and movement artifacts. However, they state no further processing was performed. First of all, we applied Butterworth filtering. This filter can be used as a high, low pass and band pass filter. The first step is normalizing the cutoff frequencies which in our case we used 0.5Hz for the lowest frequency and 30Hz for the highest) by the Nyquist frequency (which is half of the sampling frequency).

$$low = \frac{0.5}{nyq}$$

$$high = \frac{30}{nyq}$$

For the n th order Butterworth filter we use the following transfer function:

$$H(s) = \frac{1}{\sqrt{1 + \left(\frac{s}{w_c}\right)^{2n}}}$$

Where:

- s is the complex frequency
- w_c is the cutoff frequency
- n is the order of the filter

The band pass filter is basically the combination of high and low pass filters:

$$H_{BP}(s) = H_{LP}(s) \times H_{HP}(s)$$

This can be represented as:

$$H_{BP}(s) = \frac{1}{\sqrt{1 + \left(\frac{s}{w_1}\right)^{2n}}} \times \frac{1}{\sqrt{1 + \left(\frac{w_2}{s}\right)^{2n}}}$$

Where w_1 and w_2 are the low and high cutoff frequencies.

We need to perform bilinear transform to implement the filter in our digital data in order to map the analog filter's transfer function to the digital domain:

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

Where:

- T is the sample period defined as $T = 1/fs$
- s is the complex frequency in the analog domain
- z is the complex frequency in the digital domain

The transformed digital filter can then be applied to the discrete EEG data. Using a high cutoff frequency of 30Hz removes from concern line frequency noise at the 50Hz or 60Hz (depending on the geographical location). We perform this in our data using the butter and filtfilt functions from the scipy.signal module. Here we show an example from applying the filter with order $n = 4$ and low cutoff frequency of 0.5Hz and high cutoff of 30Hz in the Figure 3.3.

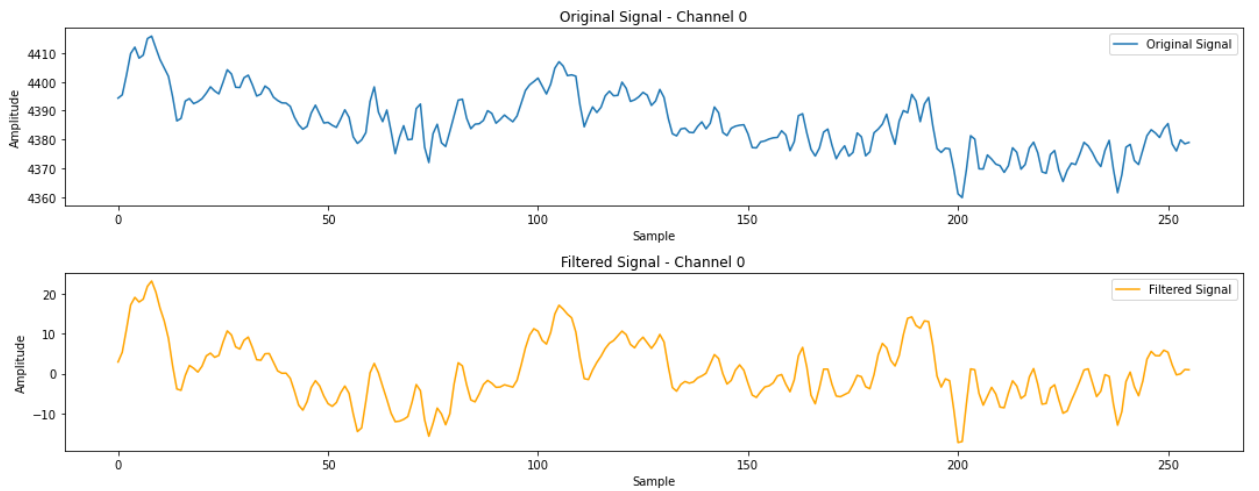


Figure 3.3 Signal sample before and after Butterworth filtering

3.2.3. Artifact removal

We perform the novel approach of Automatic and Tunable Algorithm (ATAR) for EEG artifact removal using wavelet decomposition (Bajaj et al., 2020). The main principle of this method is to interpret the probability of wavelet components containing non-artefactual information based on their amplitude superseding predefined thresholds. For our experiments we used the soft-thresholding mode defined as follows:

$$\lambda_s(w) = \begin{cases} w & \text{if } |w| \leq \theta_\gamma \\ \frac{1 - e^{-aw}}{1 + e^{-aw}} \theta_\alpha & \text{otherwise} \end{cases}$$

Where:

- $a = -\frac{1}{\theta_\gamma} \log \frac{\theta_\alpha - \theta_\gamma}{\theta_\alpha + \theta_\gamma}$, with $\theta_\alpha > \theta_\gamma$
- w is the wavelet coefficient
- θ_α is the given threshold for maximum attenuation
- θ_γ is the given threshold that attenuation begins and smoothly limit them to θ_α with hyperbolic tangent function

The main benefit of the ATAR method compared to other methods like Independent Component Analysis is that it is left to the user to tune the aggressiveness of artifact removal. Also, performing artifact removal based on wavelet components allows for more focused attenuation of artifacts, leaving localized patterns basically unharmed from specific frequency band noise. Very important to also note is that the nature of the algorithm allows for specific channel by channel and trial by trial filtering eliminating any possibilities for data leakage. To achieve artifact removal using ATAR we call the `sp` function from the `spkit` module as shown in Figure 3.4.

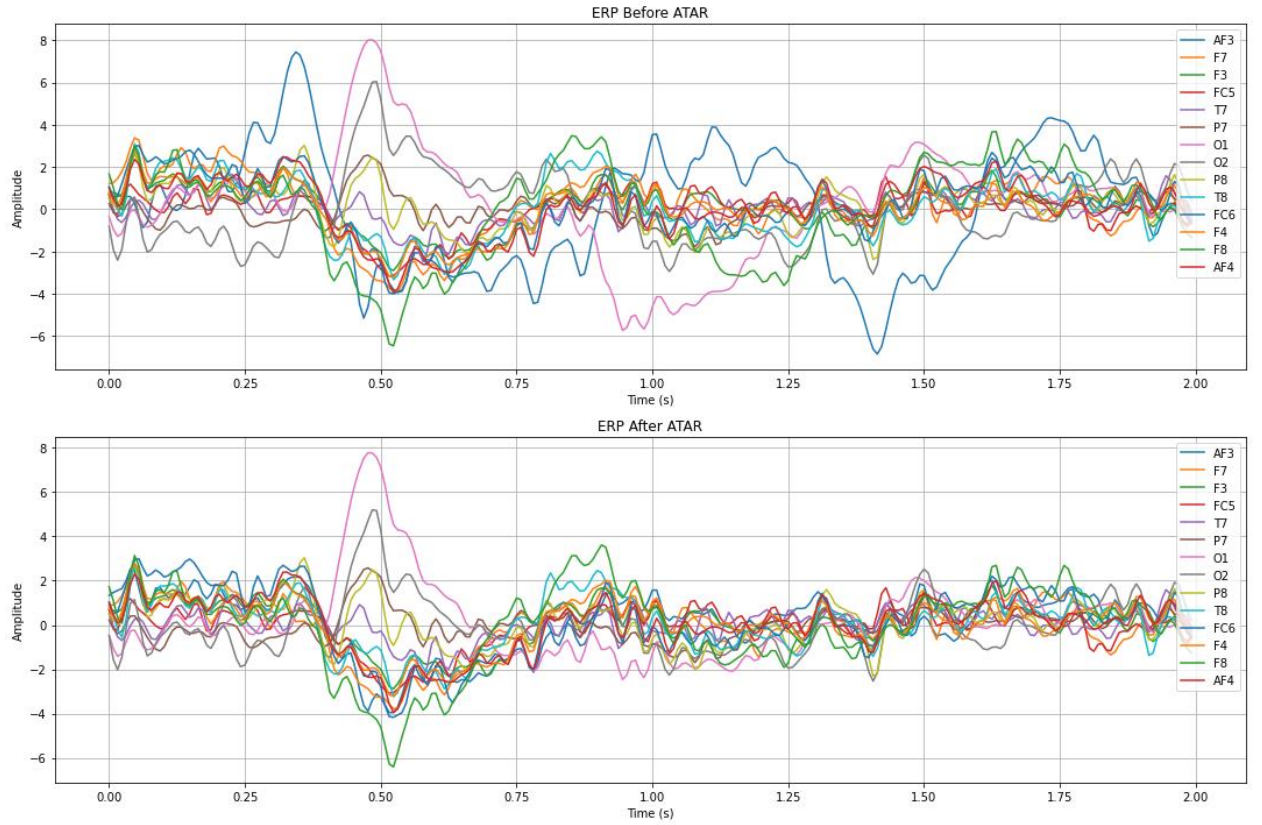


Figure 3.4 ERP signal before and after ATAR

3.2.4. Z-score normalization

We perform a simple normalization method using Z-scoring. We first combine all digit trials and then apply the algorithm to avoid any data leakage. From the combined digit trial data, we calculate the mean and the standard deviation of all trials and then perform z-score normalization for each time step value of each trial:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Where:

- x_i are the signal values for each timepoint i
- μ is the mean value from all trials
- σ is the standard deviation of the values based on all trials

We then separate again the data based on digit values. We use here the zscore function from the scipy.stats module, shown in Figure 3.5. Note that the shape of the event related potentials do not change in shape but only in magnitude.

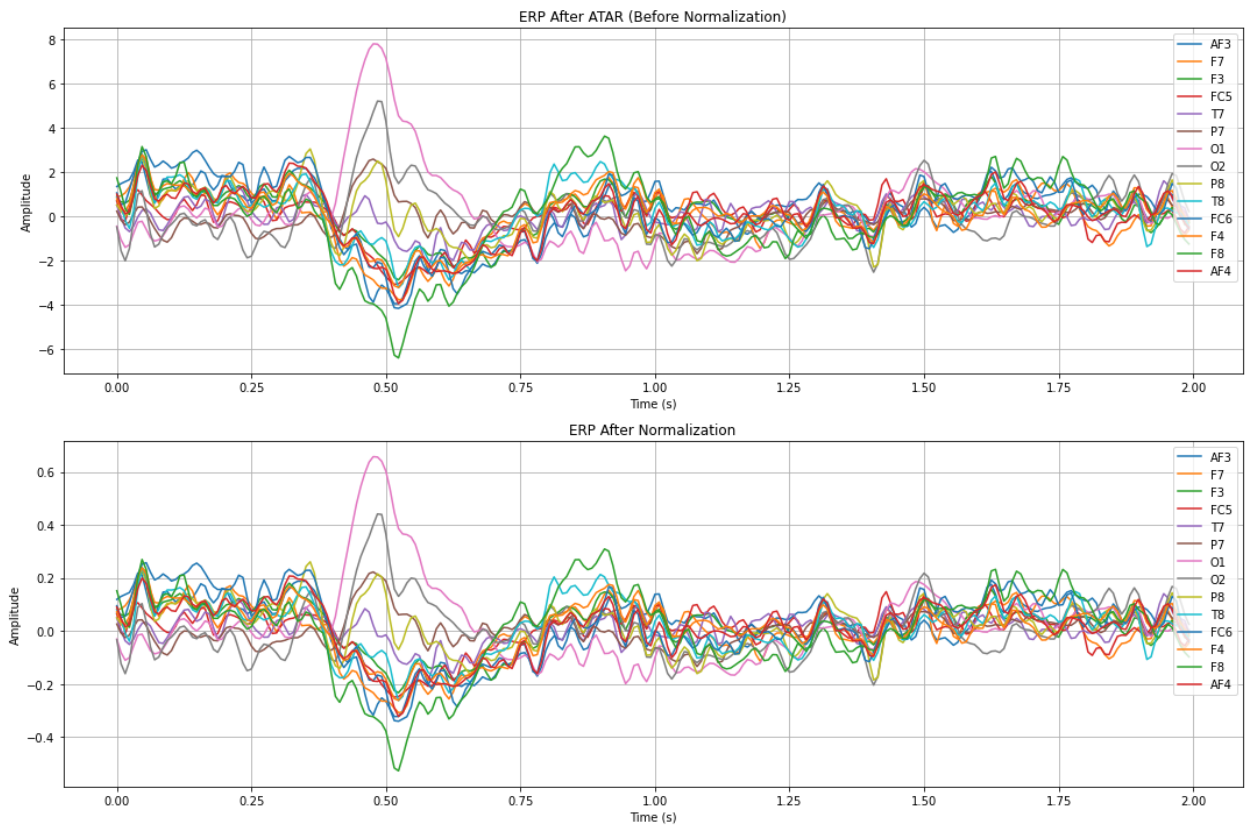


Figure 3Error! No text of specified style in document.5 ERP signal before and after normalization

3.3.Feature extraction

In addition to applying different classification algorithms to the EEG preprocessed data, we also perform a plethora of different feature extraction techniques. This is an attempt to reduce the dimensionality of the raw data, or amplify different features in the spectral and temporal domain. The methodology was inspired by a paper on the identification of speech from subjects imagining of speaking (Chengaiyan et al., 2020). We calculate the EEG Coherence, Granger Causality (GC), Directed Transfer Function (DTF), the Partially Directed Coherence (PDC) and the Transfer Entropy (TE). We also perform Empirical Wavelet Transform (EWT) and Wavelet Scattering Transform (WST), both recently proven as very strong feature extraction methods (Anuragi & Sisodia, 2020)(Buriro et al., 2021).

3.3.1. Coherence

The coherence metric attempts to identify functional connectivity between different electrodes based on the synchronization of the recordings (Nunez et al., 1997). Coherence ignores directionality of connectivity. It quantifies the frequency domain relationship between two signals given by the formula:

$$C_{xy}(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f)P_{yy}(f)}$$

Where:

- $C_{xy}(f)$ is the coherence between channels x and y at frequency $C_{xy}(f)$ taking values from 0 to 1.
- $P_{xy}(f)$ is the cross power spectral density between x and y which is basically the multiplication of one signal's Fourier transform with the complex conjugate of the other signal.
- $P_{xx}(f)$ is the auto power spectral density for x
- $P_{yy}(f)$ is the auto power spectral density for y

In our source code we use the coherence and get_window functions from the scipy.signal module. Coherence matrices are shown in Figure 3.6. Keep in mind that due to the nature of coherence calculation, the diagram is symmetrical across the diagonal.

Coherence Matrix For A Single Trial Of Digit 0



Figure 3.6 Coherence matrices for a single trial of digit 0 for the 14 channels.

3.3.2. Granger Causality

Granger causality (GC) measures the degree to which past values of a signal may contain information able to predict future values of another signal (Granger, 2008). First, the Autoregressive (AR) Model describes each time series as a function of its own past values. For the signal Y :

$$Y_t = \sum_{i=1}^p a_i Y_{t-i} + \varepsilon_t$$

Where:

- Y_t is the value of signal Y at time t
- a_i are the coefficients of the AR model.
- p is the number of past timepoints (lag).
- ε_t is the residual error for the part of Y_t that is not predicted by past values of Y .

Then we use the Multivariate Autoregressive models (MVAR) to extend the AR model as to include past values of a different signal X :

$$Y_t = \sum_{i=1}^p a_i Y_{t-i} + \sum_{i=1}^p b_i X_{t-i} + \varepsilon'_t$$

Where:

- b_i are the coefficients that measure the influence of past values of X on Y .
- ε'_t is the new residual error.

GC can also be performed in the frequency domain, as later discussed. Also, the lag points accounted from each signal does not have to be the same, yet for simplicity we chose the same time points in our method. In the end we calculate the sum of squared residual errors for the AR (SSR_{AR}) and the MVAR model (SSR_{MVAR}) and compute the F-statistic:

$$F = \frac{(SSR_{AR} - SSR_{MVAR})/p}{SSR_{MVAR}/(n - 2p - 1)}$$

Where:

- n is the number of observations

The degrees of freedom ($n - 2p - 1$) are changed to ($n - k$) when different number of timelags are considered from the 2 signals where $k = p + q$ (q is the number of lags from signal X) (Sims, 1972). If the F exceed the critical values set based on the required significance level, then we state that X causes Y . Keep in mind that due to the nature of GC calculation the results are binary i.e. channel X either causes channel Y or it does not.

We use the `grangercausalitytests` function from the `statsmodels.tsa.stattools` module to perform our GC calculations. Due to the binary nature of the results we opt to create a heat map using the `sns` function from the `seaborn` module plotting the count of GC outputs across the trials (150 trials for the digit 0 on the plotted example in Figure 3.7).

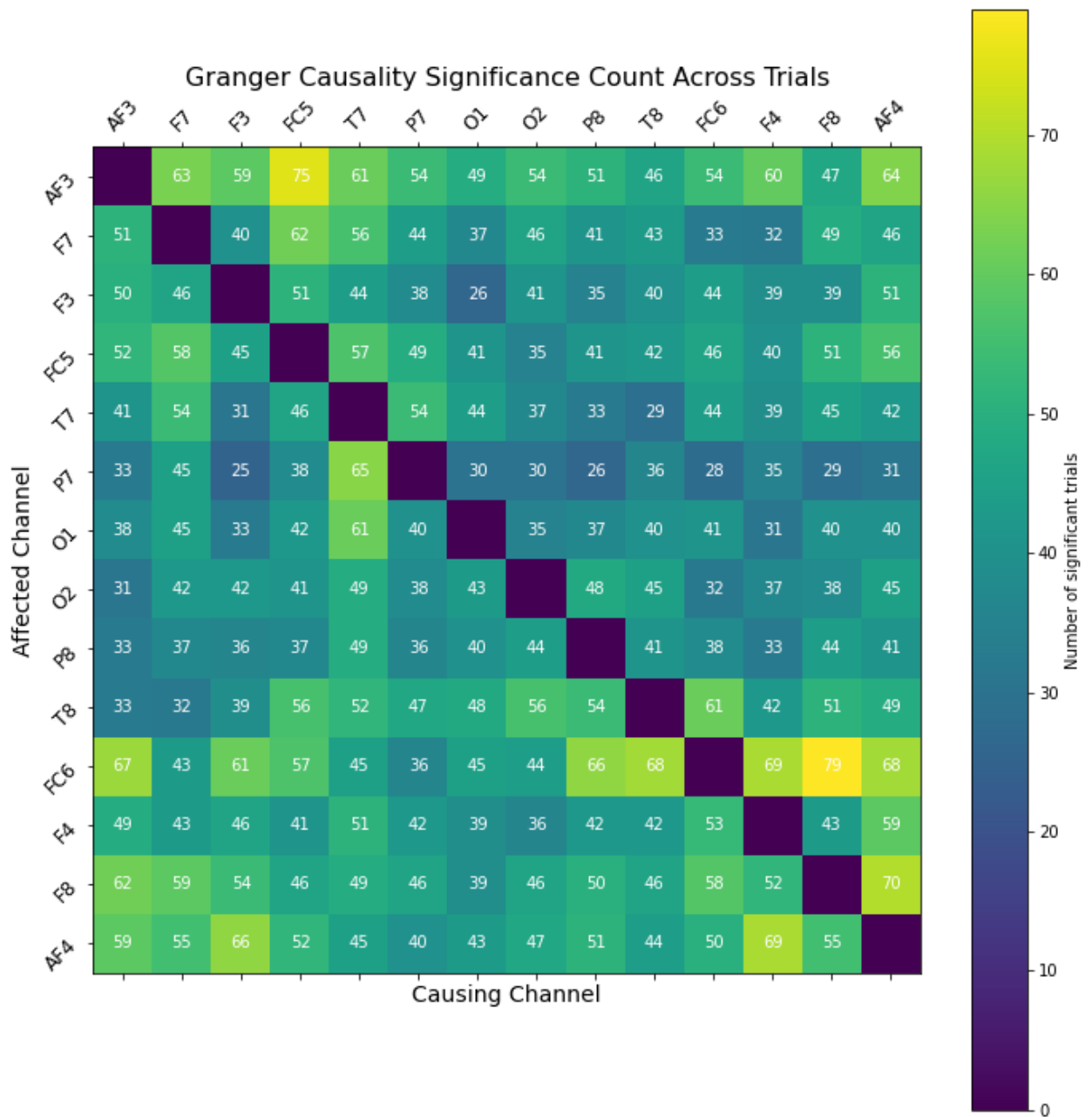


Figure 3.7 Granger causality score for the 150 trials for digit 0 for the 14 channels.

3.3.3. Directed Transfer Function

DTF is used to describe the directional influence of a channel signal on that of another, maintaining information of directionality, based on the frequency domain (Kamiński et al., 2001). It considers the overall influence, including both direct and indirect pathways though other time series. We begin by modeling each channel signal as already performed for the MVAR model for GC but we attribute the different coefficients of the AR model as matrices:

$$V_t = \sum_{k=1}^p A_k Y_{t-k} + E_t$$

Where:

- V_t here is depicted as a vector of EEG signals at time t
- A_k are the matrices of coefficients for each lag k
- E_t is a vector of noises

Then we convert the time domain to the frequency domain using Fourier transform with the function resulting in the following transfer matrix:

$$H_f = \left(I - \sum_{k=1}^p A_k e^{-i2\pi f k} \right)^{-1}$$

Where:

- I here is the identity matrix

The frequency domain representations of V_t are provided by:

$$V_f = H_f E_f$$

Where:

- E_f is the Fourier transform of the noise E_t

Last we get the DTF (ranging from 0 to 1) from channel j to channel i at frequency f from:

$$DTF_{j \rightarrow i_f} = \frac{|H_{ij_f}|}{\sqrt{\sum_{k=1}^p |H_{ik_f}|^2}}$$

Where:

- H_{ij_f} is the element from the transfer matrix H_f corresponding to the influence of channel j to channel i at frequency f .

For the calculation of DTF we manually define the function in Python code based on the above mathematical functions above, utilizing the VAR function from the statsmodels.tsa.vector_ar.var_model module. Example results are shown in Figure 3.7.

DTF Matrix For A Single Trial Of Digit 0

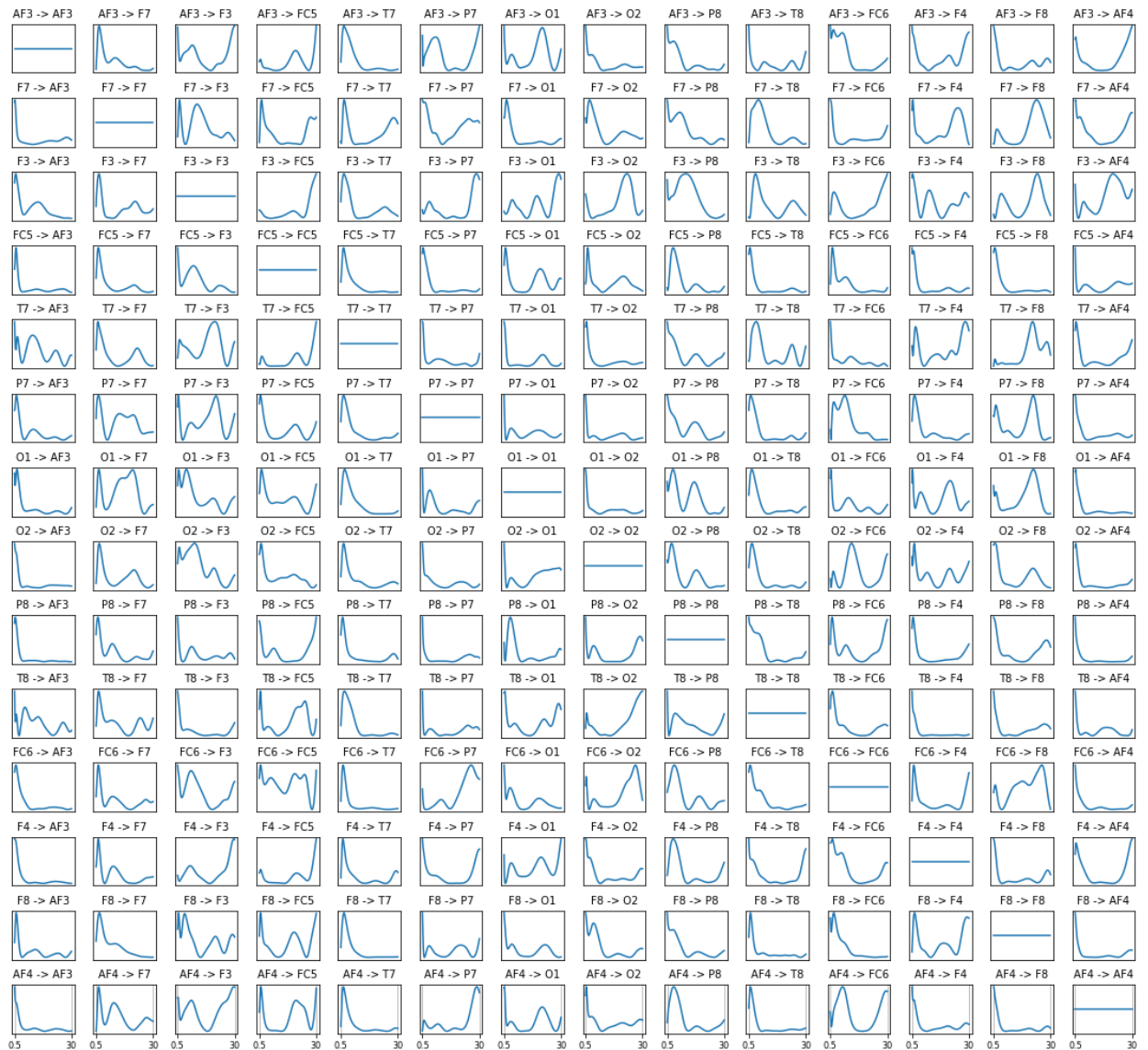


Figure 3. 8 DTF matrices for a single trial of digit 0 for the 14 channels.

3.3.4. Partial Directed Coherence

Partial Directed Coherence (PDC) is another technique used to determine the directional influence between different channels, similar to DTF, but it takes into account the influence of all other channel signals of the trial (Sameshima & Baccalá, 1999). PDC attempts to identify direct influence between time series, filtering out indirect influences. We use the same MVAR model as in DTF:

$$V_t = \sum_{k=1}^p A_k Y_{t-k} + E_t$$

We also use the same frequency domain representation:

$$V_f = H_f E_f$$

The key difference is that in PDC we use the MVAR model coefficients in the frequency domain. PDC (ranging from 0 to 1) from channel j to channel i at frequency f results from the function:

$$PDC_{j \rightarrow i_f} = \frac{A_{ij_f}}{\sqrt{\sum_{k=1}^p |A_{ik_f}|^2}}$$

Where:

- $A_f = (I - H_f)$ is the matrix of the Fourier transformed MVAR coefficients.
- A_{ij_f} is the element of A_f corresponding to the influence of channel j on the channel i at frequency f

For the calculation of PDC we, again, manually define the function in Python code based on the above mathematical functions above, again utilizing the VAR function. Sample results are shown in Figure 3.9. Feel free to observe the similarities in the two plots between DTF and PDC.

PDC Matrix For A Single Trial Of Digit 0

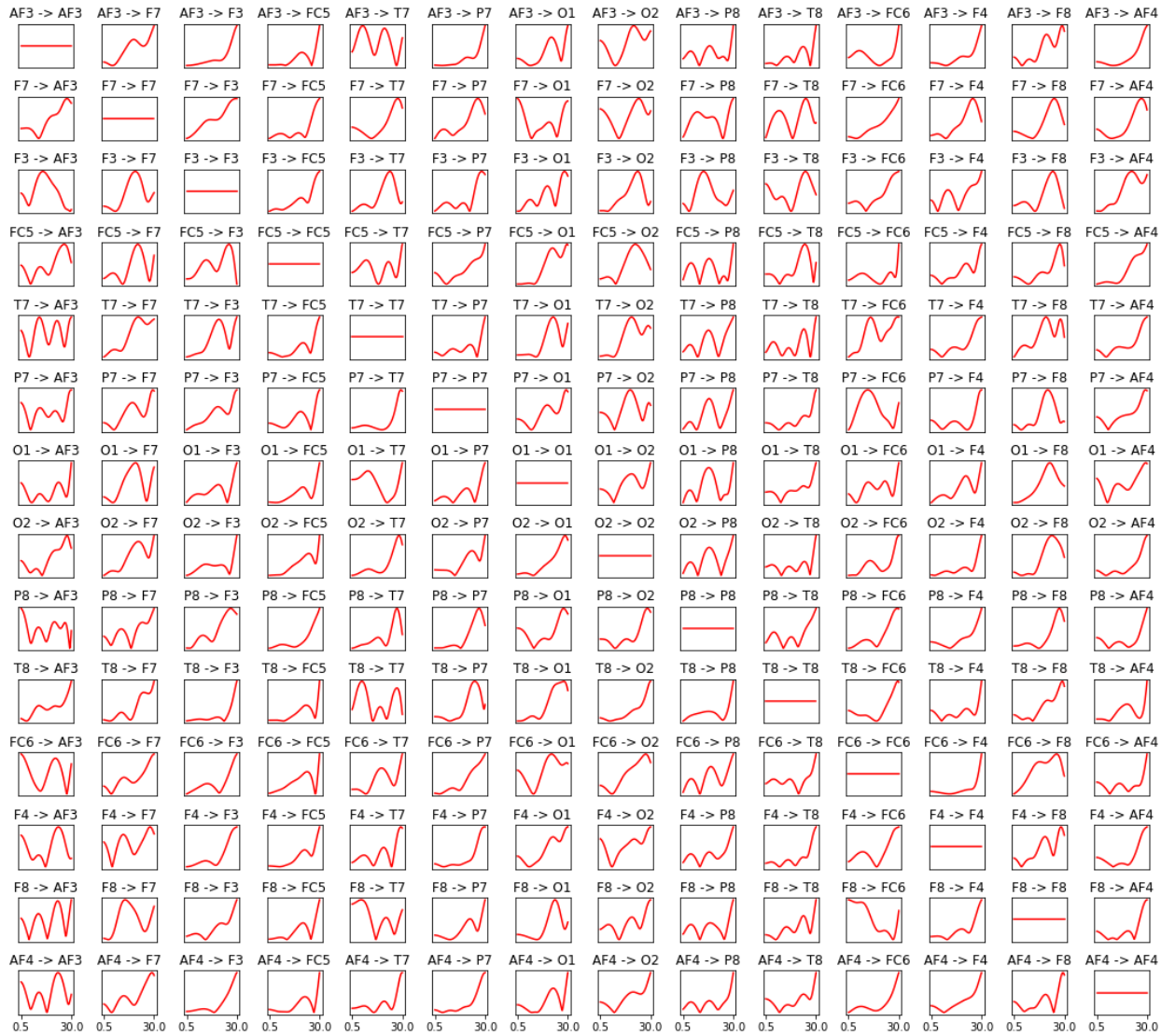


Figure 3.9 PDC matrices for a single trial of digit 0 for the 14 channels.

3.3.5. Transfer Entropy

Transfer Entropy (TE) shares properties from the Mutual Information theory, while taking into account the dynamics of non-linear information transfer among time series (Schreiber, 2000). Similarly with DTF and PDC, it attempts to derive future states of a channel signal based on past states of different channel time series. The formula providing transfer entropy is model-free. The non-linearity aspect is useful in capturing complex channel interactions that linear models like DTF and PDC might miss.

First we calculate some different probabilities that are going to be used in the TE function. Let's calculate the joint probability i.e. the likelihood of observing a specific combination of “future” states y_{t+1} and “past” states of y_t^L and x_t^L up to lag L :

$$p(y_{t+1}, y_t^L, x_t^L) = \frac{\text{count}(y_{t+1}, y_t^L, x_t^L)}{N}$$

Where:

- $\text{count}(y_{t+1}, y_t^L, x_t^L)$ is the number of occurrences of the triplet (y_{t+1}, y_t^L, x_t^L) in the time series
- N is the number of time points in the time series

After, let's calculate the conditional probability i.e. the likelihood of observing “future” states y_{t+1} given the “past” states y_t^L and x_t^L :

$$p(y_{t+1}|y_t^L, x_t^L) = \frac{p(y_{t+1}, y_t^L, x_t^L)}{p(y_t^L, x_t^L)}$$

Where:

- $p(y_{t+1}, y_t^L, x_t^L) = \frac{\text{count}(y_{t+1}, y_t^L, x_t^L)}{N}$
- $p(y_t^L, x_t^L) = \frac{\text{count}(y_t^L, x_t^L)}{N}$

Finally, let's calculate the conditional probability i.e. the likelihood of observing the “future” state y_{t+1} given the “past” state y_t^L :

$$p(y_{t+1}|y_t^L) = \frac{p(y_{t+1}, y_t^L)}{p(y_t^L)}$$

Where:

- $p(y_{t+1}, y_t^L) = \frac{\text{count}(y_{t+1}, y_t^L)}{N}$
- $p(y_t^L) = \frac{\text{count}(y_t^L)}{N}$

These probabilities can be used to calculate TE for channels X and Y in the following equation:

$$TE_{X \rightarrow Y} = \sum_{y_{t+1}, y_t^L, x_t^L} p(y_{t+1}, y_t^L, x_t^L) \log \frac{p(y_{t+1}|y_t^L, x_t^L)}{p(y_{t+1}|y_t^L)}$$

We perform these calculations manually in our python code and then we plot a single trial results. Here we suggest a novel methodology for TE feature extraction by calculating TE

over sliding windows of a fixed stepping size, attempting to capture temporal information. Unlike tradition TE that produces a single value, representing the average information transfer over the whole signal, our implementation attempts to maintain temporal resolution allowing a more dynamic analysis. A sample of the working algorithm is depicted in Figure 3.10. Further investigation into adaptive TE in brain connectivity studies should implement adaptive windowing, data smoothing and multiscale analysis across multiple temporal resolutions to enhance this technique.



Figure 3.10 TE matrices for a single trial of digit 0 for the 14 channels.

3.3.6. Empirical Wavelet Transform

Empirical Wavelet Transform (EWT) attempts to provide an adaptive and dynamic methodology for signal decomposition. Basically, it splits the signal to different frequency bands based on the intrinsic characteristics of the signal, itself. It is a very strong tool for capturing non-stationary characteristics of EEG data (Gilles, 2013). Also, it allows for user hyperparameter tuning, aiding its application in different types of signals. First, we compute the Fourier transform of the signal X_t . Then we divide the Fourier spectrum \hat{X}_f into N segments (defined by the user). The method of segmentation is left to the user providing flexibility in the EEG analysis. For our analysis we use the default local maxima method, detecting the maxima that correspond to the dominant frequencies and splitting the spectrum between 2 local maxima points.

Calculations begin by defining the empirical scaling function to capture low-frequency components and to ensure smooth transition and cut-off in the frequency domain:

$$\hat{\phi}_N(\omega) = \begin{cases} 1 & \text{if } |\omega| \leq \omega_N - \tau_N \\ \cos \left[\frac{\pi}{2} \beta \left(\frac{1}{2\tau_N} (|\omega| - \omega_N + \tau_N) \right) \right] & \text{if } \omega_N - \tau_N \leq |\omega| \leq \omega_N + \tau_N \\ 0 & \text{otherwise} \end{cases}$$

Where:

- ω_i are the boundaries of the segments
- τ_N is the parameter defining the transition width
- $\beta(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ x^4(35 - 84x + 70x^2 - 20x^3) & \text{if } 0 < x < 1 \\ 0 & \text{if } x \leq 0 \end{cases}$

Arbitrary function satisfying $\beta(x) + \beta(1 - x) = 0$ suggested in the literature.

Then the empirical wavelets are defined by the equation:

$$\hat{\psi}_N(\omega) = \begin{cases} 1 & \text{if } \omega_N + \tau_N \leq |\omega| \leq \omega_{N+1} - \tau_{N+1} \\ \cos \left[\frac{\pi}{2} \beta \left(\frac{1}{2\tau_{N+1}} (|\omega| - \omega_{N+1} + \tau_{N+1}) \right) \right] & \text{if } \omega_{N+1} - \tau_{N+1} \leq |\omega| \leq \omega_{N+1} + \tau_{N+1} \\ \sin \left[\frac{\pi}{2} \beta \left(\frac{1}{2\tau_N} (|\omega| - \omega_N + \tau_N) \right) \right] & \text{if } \omega_N - \tau_N \leq |\omega| \leq \omega_N + \tau_N \\ 0 & \text{otherwise} \end{cases}$$

Finally we get the reconstruction of the signal from:

$$f(t) = W_f^\varepsilon(0, t) * \varphi_1(t) + \sum W_f^\varepsilon(N, t) * \psi_N(t)$$

$$= \left(\widehat{W}_f^\varepsilon(0, \omega) * \widehat{\varphi}_1(\omega) + \sum \widehat{W}_f^\varepsilon(N, \omega) * \widehat{\psi}_N(\omega) \right)^\vee$$

Where:

- $W_f^\varepsilon(n, t) = \langle f, \psi_N \rangle$ i.e. the detail coefficients of the EWT are given by the inner products of the signal with the empirical wavelets
- $W_f^\varepsilon(0, t) = \langle f, \varphi_1 \rangle$ i.e. the approximation coefficients are given by the inner products of the signal with the scaling function
- \wedge denotes the Fourier transform
- \vee denotes the inverse Fourier transform

In our source code we divide the Fourier spectrum into 4 segments and we perform EWT calculations using the ewtpy Python module, as shown in Figure 3.11. Interesting, yet also expected, is the tendency of the EWT boundaries to naturally almost match the known EEG frequency bands of Delta (0.5-4Hz), Theta (4-7Hz), Alpha (8-12Hz), Beta (13-30Hz). Yet, the division happens dynamically for each EEG signal.

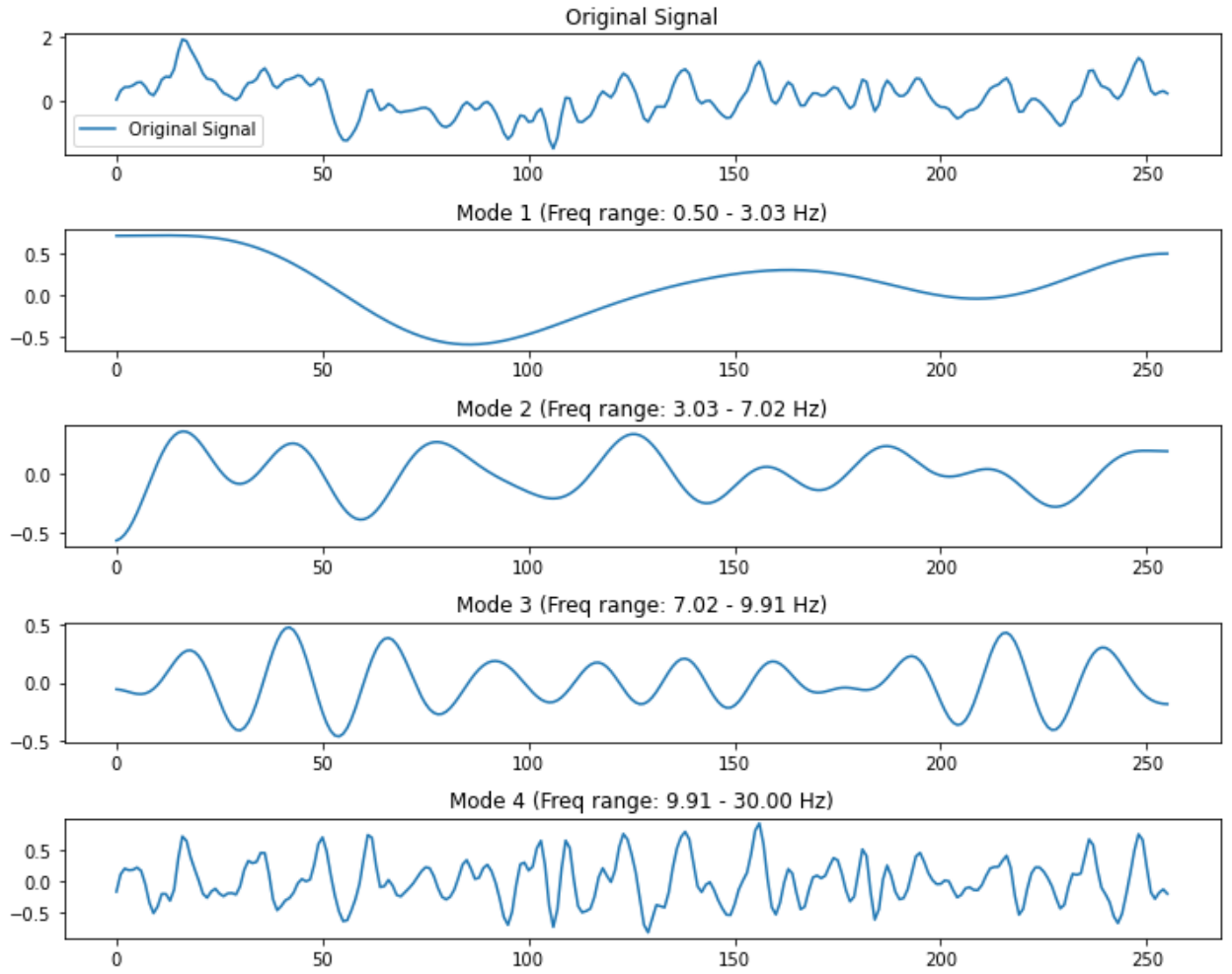


Figure 3 Error! No text of specified style in document. **11 Plotting of the original sample signal and its 4 different frequency segments.**

3.3.7. Wavelet Scattering Transform

Wavelet Scattering Transform (WST) extends the traditional wavelet transform by including convolution operations of non-linear modulus and averaging processes, allowing the capture of higher-order interactions in the EEG signal (Mallat, 2012)(Ghezaiel et al., 2020). It provides robustness against deformations and noise. The transform decomposes the signal using wavelet convolution and modulus operations, capturing information at different scales.

The zero-order scattering coefficient is simply a low-pass filter of the signal:

$$S_0x(t) = x * \varphi(t)$$

Where:

- x is the signal

- φ is the low-pass filter, typically a Gaussian filter.

The first-first order scattering coefficients capture information about the local variations of the signal by convolving the inputs signal $x * \psi_{\lambda_1}$, where λ_1 denotes the center frequency of the first-order ψ_{λ_1} wavelets, defining the scale of the operation. Then a simply modulus operator is applied $|x * \psi_{\lambda_1}|$, to implement non-linearity. Finally, we perform averaging via a low-pass filter φ . The formula of the first-order operation is:

$$S_1x(t, \lambda_1) = |x * \psi_{\lambda_1}| * \varphi(t)$$

For the second-order scattering coefficient we simply repeat the steps. First we apply wavelet transform on the first-order coefficients using the ψ_{λ_2} wavelets like so: $|x * \psi_{\lambda_1}| * \psi_{\lambda_2}$. Then we apply the modulus $||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|$ and finally we apply the low-pass filter φ , getting:

$$S_2x(t, \lambda_1, \lambda_2) = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \varphi(t)$$

We can dive deeper into higher-order scattering, capturing features of the signal at different scales and orientations. For the Nth order the scattering coefficient are calculated:

$$S_Nx(t, \lambda_1, \lambda_2, \dots, \lambda_N) = ||\dots |x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \dots * \psi_{\lambda_N}| * \varphi(t)$$

Note that the low-pass filter is always applied outside of the scattering operations in each step of the WST and low-pass filtering in lower layers does not affect scattering coefficients of higher layers.

We perform WST using the Scattering1D function of the kymatio.numpy module. We calculate up to the second-order scattering coefficients and, later, we use the deepest coefficients for classification. Here we plot a sample from the original signal, along with the zero, first and second-order scattering, shown in Figure 3.12.

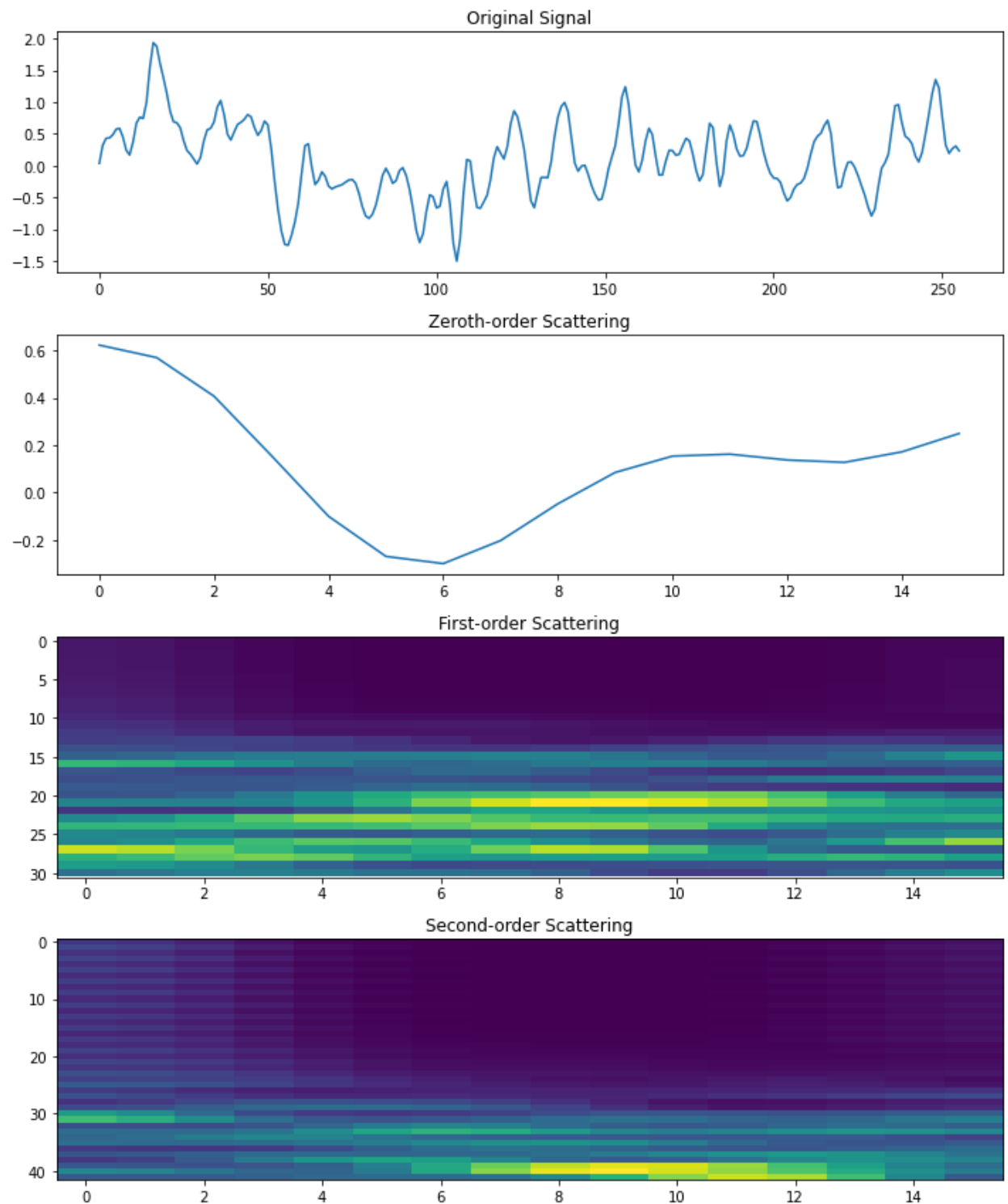


Figure 3 Error! No text of specified style in document.12 Plotting of the original sample signal and its 0th, 1st and 2nd scattering components.

3.4. Classification

We experimented with the raw data (after preprocessing) as well as with the outputs from each feature extraction method, with different classification algorithms. We use Naïve Bayes, Support Vector Machine with the Radial Basis Function as its kernel, eXtreme Gradient Boosting which is an optimized Gradient boosting algorithm, as well as Artificial Neural Networks, Convolutional Neural Networks and Long short-term memory Recurrent Neural Networks. Due to limited computing power, this is not a comparative study for the maximal efficiency of each classification algorithm, yet a simple introduction in each method and a proof of concept regarding their use for EEG signal classification. The complexity of each algorithm architecture and hyperparameter tuning is kept to a rudimentary level. Also, note that all the classification algorithms are used in a supervised learning manner. We train all the models and then we calculate the validation accuracy to measure performance for each model.

3.4.1. Support Vector Machine

Support Vector Machines (SVM) are widely used in bioinformatics due to their ability to handle high-dimensional, non-linear data with excellent classification capabilities (Ben-hur & Weston, 2010). The SVM algorithm projects data points in a high dimensional space and attempts to draw a hyperplane that best classifies the origin of these data points, with the largest margin decision boundary. The term support vectors describe the data points closest to the hyperplane that influences its position and orientation in the high dimensional space.

We focus on the Radial Basis Function (RBF) which basically, handles data that are not linearly separable, by projecting this data in higher dimensions that allow for linear separation(Cho et al., 2008). The function for the RBF is:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

Where:

- x_i and x_j are the input vectors
- $\|x_i - x_j\|$ is the Euclidean distance of the input vectors
- γ defines the spread of the kernel, typically Gaussian, that defines the influence of a single training example.

Using the training data to fit the SVM model we attempt to optimize the problem using:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

Subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, i \in [1, n]$$

Where:

- y_i is the binary class label for the data point i
- x_i is the data point i
- w is the weight vector defining the orientation of the hyperplane
- b is the bias term
- ξ_i are the slack variable that allow for misclassification for non-perfectly separable cases
- C is a parameter controlling the trade-off between low error on the training and achieving a large margin
- n is the number of data points

Minimizing the term $\frac{1}{2} \|w\|^2$ maximizes the margin (Goar & Yadav, 2024). $C \sum_{i=1}^n \xi_i$ allows for some misclassified data points to rely inside the margin without them regarded as support vectors. Classification among multiple classes can be handled as a series of binary classifications.

3.4.2. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoosting) is a powerful sparsity-aware algorithm approximating tree learning for classification and regression (T. Chen & Guestrin, 2016). It intergrades regularization techniques, thus minimizing over fitting and also, it perfectly handles missing values. It is an ensemble of additive model composed of multiple base learners (trees). The primary goal is to minimize the objective function:

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{t-1}) + f_t(x_i) + \Omega(f_t)$$

Where:

- $l(y_i, \hat{y}_i^{t-1})$ is the loss function comparing the real values y_i with the previous predictions \hat{y}_i^{t-1} . Different loss functions are used for different classification and regression problems.

- $f_t(x_i)$ is the new prediction
- $\Omega(f_t)$ is the regularization term that $\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$ where T is the total number of tree leaves, γ is the penalty coefficient for the T , w_j are the weights of the leaves and λ is the regularization parameter of w_j (Tarwidi et al., 2023).

The loss function measures how far are the model's predictions from the actual values while the regularization term penalizes the complexity of the model to avoid over fitting. Using Taylor approximations we get:

$$L^{(t)} \approx \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

Where:

- g_i and h_i are the first and second order derivatives of the loss function named Gradient and Hessian accordingly.

A new tree is built using the g_i and h_i , by calculating the weights of the leaves in the new tree, aiming to correct the errors of the previous prediction. The new tree's predictions are scaled and added to the existing predictions to improve accuracy and then the process is repeated. The final model is a combination of all trees contributing to a more accurate prediction.

3.4.3. Naïve Bayes

Naïve Bayes (NB) is a simple yet powerful probabilistic classifier based on Baye's theorem. It assumes a strong (naïve) conditional independence between the features (Vanneschi & Silva, 2023). However, the assumption that EEG vectors are independent from one another is a main disadvantage considering the correlated nature of the signal. Nevertheless, NB for EEG analysis is a real consideration with strong classification capabilities (Hosseini et al., 2021). First we calculate the prior probabilities:

$$P(C_k) = \frac{\text{Number of instances in class } C_k}{\text{Total number of instances}}$$

Where:

- $P(C_k)$ is the prior probability of each class C_k

Then we calculate the Likelihoods:

$$P(X_i|C_k) = \frac{\text{Number of instances where } X_i \text{ and } C_k \text{ co-occur}}{\text{Number of instances in class } C_k}$$

Where:

- $P(X_i|C_k)$ is the probability of feature X_i given class C_k

After, we apply Baye's Theorem:

$$P(C_k|X) = \frac{P(X|C_k) \cdot P(C_k)}{P(X)}$$

Where:

- $P(C_k|X)$ is the posterior probability of class C_k given the feature vector X
- $P(X)$ is the probability of observing the vector X

For multiple features, this becomes:

$$P(C_k|X) \propto \prod_{i=1}^n P(X_i|C_k) \cdot P(C_k)$$

3.4.4. Artificial Neural Networks

Artificial Neural Networks (ANN) functionally mimic the neural architecture seen in human brain. It constitutes of an input layer, the hidden layers, and last there is the output layer. For our example the input layer receives the EEG data either after preprocessing or after feature extraction. The hidden layers are mainly responsible for the analysis. Each neuron in the hidden layer applies a weighted sum of its inputs and produces a result through a non-linear activation function; most commonly Rectified Linear Unit (ReLU). This function outputs a zero for any negative input. The output layer produces the final classification, commonly using the sigmoid function for binary classification or softmax for multiple classes.

After setting the model architecture, we train the model by adjusting its weights based on the desired output. This is done by using backpropagation. We begin with randomized weight values and compute the output of the network for a given input. Then we calculate the loss function for our predicted output and the real output. A very common and simple method for calculating the losses for classification is the multi-class Cross-Entropy Loss (Alzubaidi et al., 2021):

$$\text{Cross - Entropy Loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c})$$

Then we perform the backward pass updating the weights of the network to minimize the loss. The weights are calculated using the gradient descent algorithm indicating the direction and magnitude of weight adjustment:

$$w_{new} = w_{old} - a \frac{\partial \text{Loss}}{\partial w}$$

Where:

- a is the learning rate, adjustable by the user
- w are the weights in the network

3.4.5. Convolutional Neural Networks

In Convolutional Neural Networks (CNN) we again rely on techniques like backpropagation and gradient descent. The theory of CNNs can be simplified as feeding the output of convolution operations in a normal ANN (Alzubaidi et al., 2021). They have excellent capabilities for extracting spatial and temporal features. Let's focus on how the convolution operations are performed. The input layer remains the same as for ANN. Convolutional layer apply convolution operations in order to detect patterns in the input layer. A mathematical operation where a small matrix called kernel, slides over the input data, performing element-wise multiplication and summing the results to produce a feature map as an output:

$$(I * K)(i, j) = \sum_m \sum_n (i + m, j + n) \cdot K(m, n)$$

Where:

- I is the input matrix
- K is the kernel matrix
- i, j indicate the position in the input matrix
- m, n indicated the position in the kernel matrix

As in ANNs each neuron receives its value for an activation function, also commonly ReLU. Finally, in CNN we commonly perform a pooling function to retain the most

important features. For the case of max pooling, a pooling window with preset small dimensions, slides over the feature map and outputs only the maximum value within the window.

The output of the convolution layers is then passed through one or more fully connected (dense) layers with architecture similar to that of ANNs, and then classification is performed in the output layer. After the forward pass we compute the loss function, again cross-entropy commonly used, and we update the weights using gradient descent.

3.4.6. Long Short-Term Memory networks

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) created for appropriately dealing with the vanishing gradient problem found in traditional RNNs. As stated above gradient descent is computed based on the partial derivative of the loss function. With increased iterations in RNNs the gradient magnitude decreases, slowing the training process (Hochreiter & Schmidhuber, 1997). LSTM thrive on capturing temporal patterns, crucial for EEG analysis, and they are frequently used for time-series analysis. LSTM complexity comes from their architecture and not so much from their mathematical background. Let's try to provide a simple explanation of their structure using a simple LSTM network as an example.

We begin with the input layer as with all neural networks, in our case with the preprocessed EEG data or with the output of feature extraction. Then we proceed with the LSTM layer. Each layer consists of LSTM units that have a memory cell and three gates (input, forget and output gate) (Ian Goodfellow, Yoshua Bengio, 2017). The forget gate decides what information to discard from the cell state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Where:

- f_t is the forget gate activation vector
- σ is the sigmoid function
- W_f is the weight matrix for the forget gate
- h_{t-1} is the previous hidden state
- x_t is the current input
- b_f is the bias vector

The input gate is calculated by:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Where:

- W_i is the weight matrix for the input gate
- b_i is the bias for the input gate

The candidate cell state is the new candidate values to be added to the actual cell state, i.e. the “memory” of the data:

$$\tilde{C}_t = \tan(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Where:

- \tilde{C}_t is the candidate cell state
- W_c is the weight matrix for the candidate cell state
- b_c is the bias for the candidate cell state

The current cell state is updated:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The output gate controls the information flow from the cell state to the hidden state determining the output of the LSTM unit:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Where:

- W_o is the weight matrix for the output gate
- b_o is the bias for the output gate

Then the hyperbolic tangent function is applied to the output gate to form values between -1 and 1 for the new hidden state:

$$h_t = o_t * \tan(C_t)$$

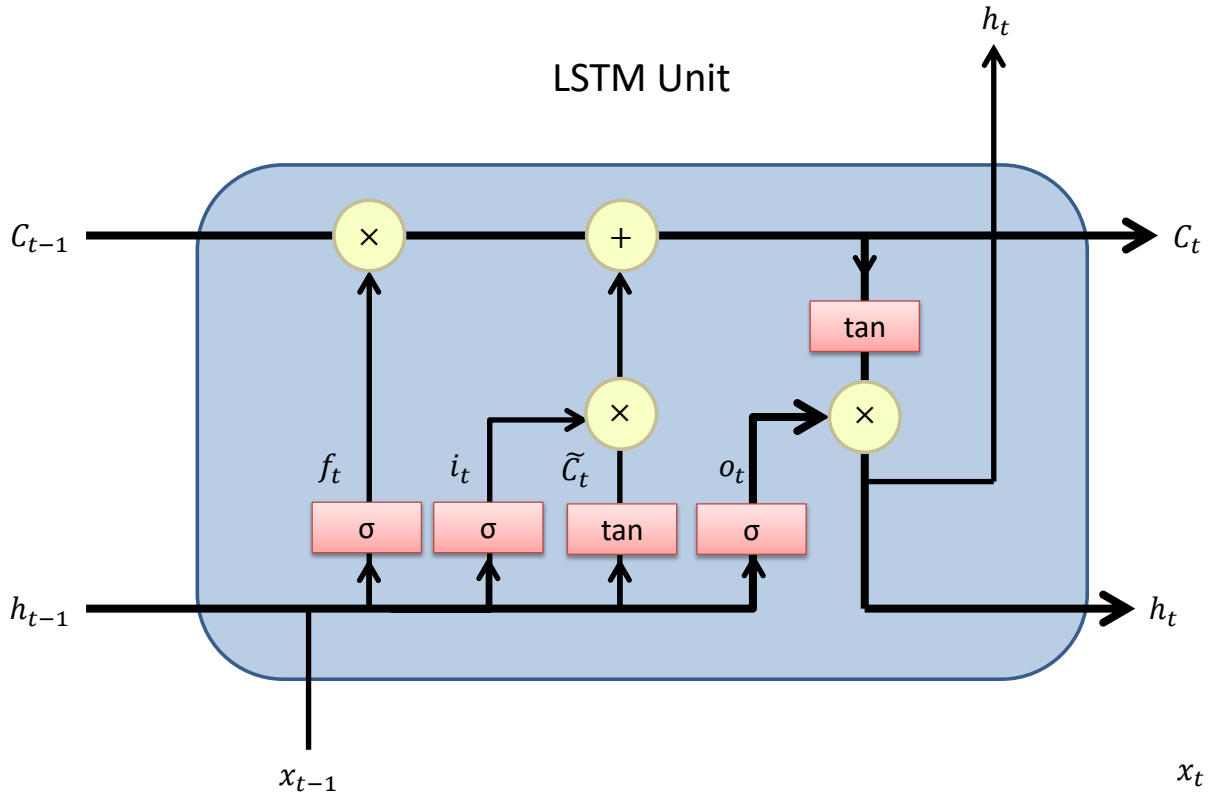


Figure 3.13 Workflow diagram for the LSTM unit.

Let's attempt an intuitive explanation of the LSTM unit function, shown in Figure 3.13. We can visualize the structure by imagining 3 inputs (previous hidden state, previous hidden state, current input) and 2 outputs (cell state, hidden state). The previous cell state C_{t-1} and the current input x_t , along with the previous hidden state h_{t-1} are fed into the LSTM unit. The forget gate f_t calculates which parts of the previous cell state C_{t-1} to forget. The input gate i_t and the candidate cell state \tilde{C}_t determine which new information to store in the cell state C_t . The cell state C_t combines the forget output f_t with the previous cell state C_{t-1} and the input gate output i_t with the candidate cell state \tilde{C}_t . In the end the new hidden state h_t is calculated from the output gate o_t .

After the LSTM layers, again we connect their output with a fully connected (dense) layers and we proceed in a similar fashion as for the CNN. Training consists again by loss computation and updating the weights using gradient descent.

4. Results

As stated before due to limited computational resources, only 150 trials were used for each of the classes for EEG recording for digits 0, 1 and non-digit, blank recordings. The hyperparameters for all classification algorithms were exactly the same between the 0-1 classification and 0-blank but different for the EPOC and MU datasets.

For the SVM model we used the RBF kernel using the `sklearn.svm` module, with default parameters $C = 1$ and $\gamma = 'scale'$. For the NB and XGBoost we used the default setting from their `sklearn.naive_bayes` and `xgboost` python modules. For XGBoost the default hyperparameters were a learning rate of 0.3, maximum depth of 6 and 100 estimators (basic trees). For the ANN we chose a single hidden layer with 8 neurons with a ReLU activation function for the hidden layers, a dropout layer and a sigmoid activation function for the output layer. For the CNN we used a single 8 neuron convolutional layer followed by a fully connected dense layer with 16 neurons. After the convolutional layer we also applied MaxPooling and BatchNormalization. For the preprocessed data and for Granger causality we used two-dimensional convolution while for the rest we implemented three-dimensional convolution. For the LSTM network, again due to computational limitations, we connected a single LSTM layer with 16 neurons with a dropout layer directly to the output layer for binary classification. For all deep learning algorithms we used the `keras` library. We didn't perform each classification algorithm to each feature extraction method. For the preprocessed data we perform CNN, ANN, LSTM, SVM, XGBoost and NB. For coherence we perform CNN, SVM, XGBoost, NB. We focused only on CNN classification for Granger causality. For Directed Transfer Function, Partially Directed Coherence and Transfer Entropy we used CNN, SVM, XGBoost and NB. Last for Empirical Wavelet Transform and for Wavelet Scattering Transform we used all mentioned classification algorithms (CNN, ANN, LSTM, SVM, XGBoost, NB). We used only validation accuracy for evaluation in order to avoid clutter and in order for our results to be directly comparable with other related work. For binary classification accuracy is calculated using the formula:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

We were unable to prove any classification capabilities stronger than random for digits 0 and 1. Similar were the results when comparing different digits, but we don't present these findings as they were similar in nature. These results directly oppose many of the related works discussed above. The results are summarized in Table 4.1. On the other hand, we were able to provide excellent classification for almost all combinations of feature extraction and classification algorithms for differentiating digit 0 from non-digit blank recordings as demonstrated in the following tables. Some techniques were able to reach validation accuracies north of 95% for 0 vs blank for classification. Similar were the results for the comparison of all other individual digits when compared with non-digit EEG recordings. These results are not presented here to avoid clattering, yet we invite the reader to perform these experiments using our provided source code. We are unable to perform a legitimate comparison for the different classification methods, as these methods, especially the ones using deep-learning are not in their optimal configuration due to computational limitations.

Table 4.2 Validation accuracies from the EPOC and MU datasets.

Validation accuracies	0 vs blank EPOC	0 vs 1 EPOC	0 vs blank MU	0 vs1 MU
Preprocessed data-CNN	91.7%	52.5%	92.5%	59.2%
Preprocessed data-ANN	67.5%	45.0%	84.2%	52.5%
Preprocessed data-LSTM	82.5%	54.2%	79.2%	51.7%
Preprocessed data-SVM	66.0%	57.0%	80.0%	53.0%
Preprocessed data-XGBoost	79.0%	53.0%	88.0%	62.0%
Preprocessed data-NB	90.0%	54.0%	86.0%	53.0%
Coherence-CNN	94.1%	53.0%	85.8%	52.5%
Coherence-SVM	96.0%	60.0%	92.0%	53.0%
Coherence-XGBoost	95.0%	53.3%	87.0%	49.0%
Coherence-NB	90.0%	49.0%	78.0%	57.0%

GC-CNN	86.7%	54.4%	72.2%	52.0%
DTF-CNN	87.5%	48.3%	83.3%	51.67%
DTF-SVM	92.0%	59.0%	82.0%	58.0%
DTF-XGBoost	83.0%	50.0%	83.0%	61.0%
DTF-NB	82.0%	59.9%	76.0%	53.0%
PDC-CNN	91.1%	60.0%	82.2%	52.2%
PDC-SVM	96.0%	58.0%	87.0%	57.0%
PDC-XGBoost	86.0%	46.0%	81.0%	57.0%
PDC-NB	88.0%	52.0%	71.0%	53.0%
TE-CNN	85.6%	52.2%	73.3%	47.8%
TE-SVM	82.0%	42.0%	74.0%	41.0%
TE-XGBoost	81.0%	53.0%	68.0%	50.0%
TE-NB	80.0%	50.0%	78.0%	41.0%
EWT-CNN	92.2%	45.6%	88.9%	56.7%
EWT-ANN	72.2%	57.8%	76.7%	53.3%
EWT-LSTM	84.4%	47.8%	80.0%	46.7%
EWT-SVM	86.0%	47.0%	88.0%	52.0%
EWT-XGBoost	82.0%	52.0%	90.0%	52.0%
EWT-NB	88.0%	54.0%	84.0%	52.0%
WST-CNN	87.5%	47.5%	67.8%	45.6%
WST-ANN	88.3%	46.7%	92.2%	61.1%
WST-LSTM	90.8%	49.2%	94.4%	60.0%
WST-SVM	89.0%	49.0%	86.0%	47.0%
WST-XGBoost	94.0%	42.0%	94.0%	52.0%
WST-NB	88.0%	47.0%	72.0%	46.0%

5. Discussion

5.1. Comparison to previous works

Our findings regarding 0 vs 1 classification directly opposes many of the related works discussed above. We do not state that our result are able to disprove any of these publications, mainly because of the simplicity of our classification algorithms and due to the use of only a small percentage of the data from the EPOC and MU sets. It is important to note that any attempt to increase architecture complexity only increased over fitting resulting in extremely high training accuracy and low validation accuracies (not exceeding randomness). There are enough publications on the MINST digit dataset that a meta-analysis on digit classification algorithms would prove extremely illuminating on EEG analysis, yet this requires coordinated scientific transparency.

It is very important to perform very careful dimension manipulation during all the steps of EEG analysis (i.e. loading the data, preprocessing, feature extraction and classification) as the classification algorithms, especially neural networks, have excellent capabilities to detect any leaked data as class features. From experimentation with forced leaked data comparing EEG recording from digits 0 and 1, we were able to receive false validation accuracies north of 90%. Common practices that introduce data leakage are feeding the wrong dimension into the classification algorithm like channels instead of trials, normalizing each signal separately for each class, data scaling based on both the training and the test set and non-proper class randomization for the input layer. This is why it is of extreme importance for any novel bioinformatics publication to also release the source code in order for avoid computation and programming errors and also to aid future researchers to implement the developed tools in their own work.

It is interesting that digit vs non-digit classification yielded excellent results, especially considering the deep-learning models' simplicity. This pattern is also present in some of the related work performed by Muglikar (2021) and from the works of Kalita (2023). This proves that the inability for our code to differentiate between digits 0 and 1 is not due to error. The disclosure of this debate is possible via the publication of the source code from the authors of all related publications. We simply can state that digit classification imposes many difficulties as we will discuss later from a neurophysiological point of view.

We suggest further research in order to identify classification capabilities for EEG digit recordings. Genetic algorithms (GA) and particle swarm optimization (PSO) can be used to not only optimize hyperparameters but also to explore and optimize the architecture and complexity of different classification algorithms (Hewahi & Hamra, 2017). These optimization techniques are particularly well-suited for tasks that involve searching through large, complex, and poorly understood search spaces, which is often the case with neural network architectures. Also, not further discussed in this paper, combinational architecture as well as semi-supervised implementations of Deep Belief Networks (which are a type of restricted Boltzmann machines) may deserve further experimentation for EEG analysis and classification (Wulsin et al., 2012).

5.2. Neurophysiological perspective

There is no lack of prior knowledge as to why we achieved excellent differentiation between EEG recordings from viewing a digit number and from viewing a blank screen. During viewing a blank image there is minimal processing taking place. The algorithms are therefore differentiating between visual event potentials (VEPs) elicited by the visual stimuli and the resting state EEG recordings. Machine learning is an excellent tool for differentiating between functional differences of EEG signals. However, the difficulty for differentiating between viewings of different digits is a more complicated question deserving further neurophysiological explanation.

Fares et al. (2019) successfully used a stacked bi-directional LSTM model for object image classification from EEG recordings of a dataset including 40 categories of images, each containing 50 images, as a Rapid Serial Visual Presentation (RSVP), with accuracy of 97.3%. In the same study they also attempt emotion classification while associating the image class “gun” with negative emotion and the class “phone” with neutral emotion. They found increased signal amplitude in the “Fz” location (prefrontal cortex, midline) for the EEG data from the stimuli from the negative category “gun” when compared with “phone” especially in the Gamma frequency range. They also were able to detect differences with statistical significance between the classes “gun”, “phone” and “panda”. They state the “panda” class to be correlated with positive emotions based on societal and cultural interpretations. Again here the category “gun” showed the highest gamma response at the prefrontal sites. Also, in another study on visual classification, the

researchers used a Generative Adversarial Network (GAN) to generate images from EEG recordings of image viewings (Shimizu & Srinivasan, 2022). They present a sample of generated images with their classification testing accuracies. It is very interesting that the generated image with the highest testing accuracy of 0.75% is one closely resembling a panda.

These findings could be used as evidence that image classification relies to some extent on the neural activity features of emotional response. It is known that the state of fear correlates with relatively heightened amygdala activation (Shin & Liberzon, 2010). Also, the hypothalamus is involved with autonomic and endocrine aspects of the fear response (i.e. release of adrenaline and cortisol, increased heart rate, sweating and pupil dilation (Herman et al., 2005). Neuronal activity from these areas when shown images associated with fearful emotions may present an identifying pattern of EEG activity.

On the other hand the “happiness” state has a different neurophysiologic profile. While amygdala also plays a role in processing positive emotions, areas like the Nucleus Accumbens and the Ventral Tegmental Area (VTA) are more directly associated with pleasure and happiness. The Nucleus Accumbens is a part of the basal ganglia that regulates the process of pleasurable stimuli and reinforces rewarding behaviors (Kringelbach & Berridge, 2009). Its activation is linked with dopamine release, which is critical for the conscious aspects of pleasure and happiness. The VTA also regulates the release of dopamine and also projects to the Nucleus Accumbens as well as other brain regions (Wise, 2002).

Machine learning algorithms are able to differentiate between novel and old stimuli. Chang et al. (2020) proposed an experiment where they used pictures of faces coupled with auditory stimuli of the name of the person depicted from familiar and unfamiliar people. They were able to classify the familiar from the unfamiliar person with the best accuracy of 90.58% in delta frequency band using SVM. This suggests that memory could be a neurophysiological feature providing EEG classification capabilities for visual stimuli. When novel stimuli is presented there is increased activity in the anterior hippocampus handling encoding of new information, while the posterior hippocampus is related with retrieval when there is familiarity (Strange et al., 2014).

Also interesting is that when a memory is recalled there is activation of the exact pathway that formed during its encoding. Memory is encoded in different parts of the brain depending on the origin of the stimuli. Traumatic events leading to Post Traumatic Stress

Disorder (PTSD) implement regions like the hippocampus and the amygdala suggesting that these areas are involved in the consolidation of the memory of traumatic events (Bremner, 2006). While using trauma reminders in PTSD patients, there was a significant increase in neuron activation in ventral hypothalamus and basolateral amygdala (Ritov et al., 2014). Also, the visual cortex in the occipital lobe has both the roles of visual processing as well as consolidation of topographic visual memory (van de Ven et al., 2012). As for auditory stimuli there is evidence that auditory fearful memories are consolidated with the help of the auditory cortex after auditory signal processing (Cambiaghi et al., 2016).

It can be theorized that visual classification depends on some extend to memory recollection. For example an image of a boat could activate visual pathways during retrieval while an image of a guitar might activate auditory pathways. Also, some objects are associated with emotional, visual and auditory pathways. For example an image of clown could activate the amygdala if the person viewing the images has coulrophobia (phobia of clowns). For another person who was introduced to clowns via their distinct bright color patterns, the visual cortex might be more involved while viewing the image. If a person associates clowns with carnival music, the auditory cortex might be more involved. All these statements of course remain theoretical and remain to be tested. However this might be an interpretation of why visual stimuli of images are able to be classified from EEG data.

For the average human subject there is no known emotional response while viewing digits. Numbers lack the complexity of associations that images of objects have. They retain no auditory or complex visual aspects. Numerical processing is partly performed in the intraparietal sulcus (IPS) which is part of the parietal lobe (Schwizer Ashkenazi et al., 2024). It's contribution is to the analysis of single object features like shape, orientation, outline and shape topology (Xu, 2007). Electrophysiological studies suggest that there is increased specific neuronal activity in the intraparietal cortex when presented with many objects. Specific areas in the IPS are differently activated while subjects are shown a collection of green and blue rectangles depending on their ratio (Butterworth & Walsh, 2011). Also, there exist specific brain areas that are activated with different activation profiles, in the cerebral cortex, for reading digits vs retrieving simple arithmetic facts vs actual arithmetical computations (Zago et al., 2001). In its essence numerical cognition, including digit recognition and arithmetic processing relates to complicated network

processing within the brain rather than a single topographically organized area. This complexity is not yet fully understood, leading to either poor or conflicting results from the classification of EEG recordings of digit image viewings. Advanced classification algorithms able to capture and interpret the neural connectome of mathematical processing could lead to more robust results.

6. Conclusions

In our study we dive into different methods for EEG feature extractions. We describe the mathematical background and the code implementation for the techniques of Coherence, Granger Causality, Directed Transfer Function, Partial Directed Coherence, Empirical Wavelet Transform and Wavelet Scattering Transform. We also demonstrate different classification methodologies. The basic mathematical functions taking place are discussed for Support Vector Machines, naïve Bayes, XGBoosting, Artificial Neural Networks, Convolutional Neural Networks and Long Short-Term Memory networks. We remind the reader that the implementation of the different classification algorithm mainly aims as a template and not as a direct comparison of their relative effectiveness due to the small sample size used and because of their simplified architecture. This is due to the lack of the computational power that is required to perform classification evaluation.

Our results indicate that our source code and the models used work as intended, and they are able to provide excellent differentiation between digit and non-digit recordings, despite the rudimentary structure. However we were unable to provide any differentiation capabilities for digit classification between digit values. We intend for our code to aid in the further exploration of the EEG analysis of the MindBigData MNIST digits databank. Due to the existence of conflicting results we invite all authors of the related work that already exist to make their source code public, aiming for the further development of robust classification techniques. There exist some papers that implement high complexity next generation deep learning classification methods that support our findings of poor classification capabilities for the different digit values.

From a neurophysiological point of view, there is strong evidence suggesting that EEG analysis is a perfectly valid method for visual stimuli classification. We suggest that this is true due to the complex network interactions taking place while viewing different object images. However, numerical operations and mathematical symbols rely in coordinated neuronal activity of different brain regions not yet well understood. The development of new deep learning classification algorithms performed in EEGs from recordings under strict, well documented experimental parameters, could lead to further understanding of the brain functions related to math. As a closing note we suggest a focus on quality over quantity for the creation of EEG databanks in order to not misrepresent the performance of classification algorithms.

References

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing.
<https://doi.org/10.1186/s40537-021-00444-8>
- Anuragi, A., & Sisodia, D. S. S. (2020). Empirical wavelet transform based automated alcoholism detecting using EEG signal features. *Biomedical Signal Processing and Control*, 57, 101777. <https://doi.org/10.1016/j.bspc.2019.101777>
- Bajaj, N., Requena Carrión, J., Bellotti, F., Berta, R., & De Gloria, A. (2020). Automatic and tunable algorithm for EEG artifact removal using wavelet decomposition with applications in predictive modeling during auditory tasks. *Biomedical Signal Processing and Control*, 55, 101624. <https://doi.org/10.1016/j.bspc.2019.101624>
- Ben-hur, A., & Weston, J. (2010). Chapter 13: A User's Guide to Support Vector Machines. *Data Mining Techniques for the Life Sciences*, 223–239.
<https://doi.org/10.1007/978-1-60327-241-4>
- Bird, J. J., Faria, D. R., Manso, L. J., Ekárt, A., & Buckingham, C. D. (2019). A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction. *Complexity*, 2019. <https://doi.org/10.1155/2019/4316548>
- Bremner, J. D. (2006). Traumatic stress: Effects on the brain. *Dialogues in Clinical Neuroscience*, 8(4), 445–461. <https://doi.org/10.31887/dcns.2006.8.4/jbremner>
- Buriro, A. B., Ahmed, B., Baloch, G., Ahmed, J., Shoorangiz, R., Weddell, S. J., & Jones, R. D. (2021). Classification of alcoholic EEG signals using wavelet scattering transform-based features. *Computers in Biology and Medicine*, 139, 104969. <https://doi.org/10.1016/j.combiomed.2021.104969>
- Burle, B., Spieser, L., Roger, C., Casini, L., Hasbroucq, T., & Vidal, F. (2015). Spatial and temporal resolutions of EEG: Is it really black and white? A scalp current density view. *International Journal of Psychophysiology*, 97(3), 210–220.
<https://doi.org/10.1016/j.ijpsycho.2015.05.004>

- Butterworth, B., & Walsh, V. (2011). Neural basis of mathematical cognition. *Current Biology*, 21(16), R618–R621.
<https://doi.org/https://doi.org/10.1016/j.cub.2011.07.005>
- Calugaru, L., Calugaru, G. T., & Calugaru, O. M. (2016). Evoked Potentials in Multiple Sclerosis Diagnosis and Management. *Current Health Sciences Journal*, 42(4), 385–389. <https://doi.org/10.12865/CHSJ.42.04.08>
- Cambiaghi, M., Grosso, A., Renna, A., & Sacchetti, B. (2016). Differential Recruitment of Auditory Cortices in the Consolidation of Recent Auditory Fearful Memories. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, 36(33), 8586–8597. <https://doi.org/10.1523/JNEUROSCI.0561-16.2016>
- Chang, W., Wang, H., Yan, G., & Liu, C. (2020). An EEG based familiar and unfamiliar person identification and classification system using feature extraction and directed functional brain network. *Expert Systems with Applications*, 158, 113448.
<https://doi.org/10.1016/j.eswa.2020.113448>
- Chown, Z., & Hilty, C. (2024, May 6). *Classifying MNIST Digits from Brain Waves using Machine Learning in Python*. Medium. <https://medium.com/@caelenhilty/mindnist-29f6fdda948d>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-Aug*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Chen, X., Teng, X., Chen, H., Pan, Y., & Geyer, P. (2024). Toward reliable signals decoding for electroencephalogram: A benchmark study to EEGNeX. *Biomedical Signal Processing and Control*, 87. <https://doi.org/10.1016/j.bspc.2023.105475>
- Chengaiyan, S., Retnapandian, A. S., & Anandan, K. (2020). Identification of vowels in consonant–vowel–consonant words from speech imagery based EEG signals. *Cognitive Neurodynamics*, 14(1), 1–19. <https://doi.org/10.1007/s11571-019-09558-5>
- Cho, B. H., Yu, H., Lee, J., Chee, Y. J., Kim, I. Y., & Kim, S. I. (2008). Nonlinear support vector machine visualization for risk factor analysis using nomograms and localized radial basis function kernels. *IEEE Transactions on Information Technology in*

Biomedicine, 12(2), 247–256. <https://doi.org/10.1109/TITB.2007.902300>

- Défossez, A., Caucheteux, C., Rapin, J., Kabeli, O., & King, J. R. (2023). Decoding speech perception from non-invasive brain recordings. *Nature Machine Intelligence*, 5(10), 1097–1107. <https://doi.org/10.1038/s42256-023-00714-5>
- Di Russo, F., Martínez, A., Sereno, M. I., Pitzalis, S., & Hillyard, S. A. (2002). Cortical sources of the early components of the visual evoked potential. *Human Brain Mapping*, 15(2), 95–111. <https://doi.org/10.1002/hbm.10010>
- DU, H. (2023). *MATHEMATICAL AND STATISTICAL ANALYSIS OF NON-STATIONARY TIME SERIES DATA*. 9.
- Fares, A., Zhong, S. H., & Jiang, J. (2019). EEG-based image classification via a region-level stacked bi-directional deep learning framework. *BMC Medical Informatics and Decision Making*, 19(Suppl 6), 1–11. <https://doi.org/10.1186/s12911-019-0967-9>
- Gasser, T., & Molinari, L. (1996). The analysis of the EEG. *Statistical Methods in Medical Research*, 5(1), 67–99. <https://doi.org/10.1177/096228029600500105>
- Ghezaiel, W., Brun, L., & Lezoray, O. (2020). Wavelet Scattering Transform and CNN for Closed Set Speaker Identification. *IEEE 22nd International Workshop on Multimedia Signal Processing, MMSP 2020*. <https://doi.org/10.1109/MMSP48831.2020.9287061>
- Gilles, J. (2013). Empirical wavelet transform. *IEEE Transactions on Signal Processing*, 61(16), 3999–4010. <https://doi.org/10.1109/TSP.2013.2265222>
- Goar, V., & Yadav, N. S. (2024). Foundations of machine learning. In *Intelligent Optimization Techniques for Business Analytics*. <https://doi.org/10.4018/979-8-3693-1598-9.ch002>
- Granger, C. W. J. (2008). Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Essays in Econometrics Vol II: Collected Papers of Clive W. J. Granger*, 37(3), 31–47. <https://doi.org/10.1017/ccol052179207x.002>
- Herman, J. P., Ostrander, M. M., Mueller, N. K., & Figueiredo, H. (2005). Limbic system mechanisms of stress regulation: hypothalamo-pituitary-adrenocortical axis. *Progress in Neuro-Psychopharmacology & Biological Psychiatry*, 29(8), 1201–1213.

<https://doi.org/10.1016/j.pnpbp.2005.08.006>

- Hewahi, N. M., & Hamra, E. A. (2017). A Hybrid Approach Based on Genetic Algorithm and Particle Swarm Optimization to Improve Neural Network Classification. *J. Inf. Technol. Res.*, 10, 48–68. <https://api.semanticscholar.org/CorpusID:36538668>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hosseini, M. P., Hosseini, A., & Ahi, K. (2021). A Review on Machine Learning for EEG Signal Processing in Bioengineering. *IEEE Reviews in Biomedical Engineering*, 14(c), 204–218. <https://doi.org/10.1109/RBME.2020.2969915>
- Ian Goodfellow, Yoshua Bengio, A. C. (2017). Deep Learning. *MIT Press*, 521(7553), 785. <https://doi.org/10.1016/B978-0-12-391420-0.09987-X>
- Joon Kim, Y., Grabowecy, M., Paller, K. A., Muthu, K., & Suzuki, S. (2007). Attention induces synchronization-based response gain in steady-state visual evoked potentials. *Nature Neuroscience*, 10(1), 117–125. <https://doi.org/10.1038/nn1821>
- Kalita, D. (2023, August 9). *Decoding thoughts with deep learning: EEG-based digit detection using CNNs*. Medium. <https://dxganta.medium.com/decoding-thoughts-with-deep-learning-eeeg-based-digit-detection-using-cnns-cdf7eee20722>
- Kamiński, M., Ding, M., Truccolo, W. A., & Bressler, S. L. (2001). Evaluating causal relations in neural systems: granger causality, directed transfer function and statistical assessment of significance. *Biological Cybernetics*, 85(2), 145–157. <https://doi.org/10.1007/s004220000235>
- Khaleghi, N., Hashemi, S., Ardabili, S. Z., Sheykhivand, S., & Danishvar, S. (2023). Salient Arithmetic Data Extraction from Brain Activity via an Improved Deep Network. *Sensors*, 23(23). <https://doi.org/10.3390/s23239351>
- Kringelbach, M. L., & Berridge, K. C. (2009). Towards a functional neuroanatomy of pleasure and happiness. *Trends in Cognitive Sciences*, 13(11), 479–487. <https://doi.org/10.1016/j.tics.2009.08.006>
- Kumari, N., Anwar, S., & Bhattacharjee, V. (2021). *Convolutional Neural Network-Based*

Visually Evoked EEG Classification Model on MindBigData BT - Proceedings of Research and Applications in Artificial Intelligence (I. Pan, A. Mukherjee, & V. Piuri (eds.); pp. 233–241). Springer Singapore.

- Lesenfants, D., Habbal, D., Lugo, Z., Lebeau, M., Horki, P., Amico, E., Pokorný, C., Gómez, F., Soddu, A., Müller-Putz, G., Laureys, S., & Noirhomme, Q. (2014). An independent SSVEP-based brain-computer interface in locked-in syndrome. *Journal of Neural Engineering*, 11(3). <https://doi.org/10.1088/1741-2560/11/3/035002>
- List, A., Rosenberg, M. D., Sherman, A., & Esterman, M. (2017). Pattern classification of EEG signals reveals perceptual and attentional states. *PLoS ONE*, 12(4), 1–23. <https://doi.org/10.1371/journal.pone.0176349>
- Mallat, S. (2012). Group Invariant Scattering. *Communications on Pure and Applied Mathematics*, 65(10), 1331–1398. <https://doi.org/10.1002/cpa.21413>
- Micanovic, C., & Pal, S. (2014). The diagnostic utility of EEG in early-onset dementia: A systematic review of the literature with narrative analysis. *Journal of Neural Transmission*, 121(1), 59–69. <https://doi.org/10.1007/s00702-013-1070-5>
- Mishra, R., Sharma, K., & Bhavsar, A. (2021). Visual Brain Decoding for Short Duration EEG Signals. *European Signal Processing Conference, 2021-Augus*, 1226–1230. <https://doi.org/10.23919/EUSIPCO54536.2021.9616192>
- Muglikar, O. D. (2021). *Application of Deep Learning Techniques for EEG Signal Classification. December*.
- Nayak, C. S., & Anilkumar, A. C. (2024). *EEG Normal Waveforms*.
- Noachtar, S., & Rémi, J. (2009). The role of EEG in epilepsy: A critical review. *Epilepsy and Behavior*, 15(1), 22–33. <https://doi.org/10.1016/j.yebeh.2009.02.035>
- Nunez, P. L., Srinivasan, R., Westdorp, A. F., Wijesinghe, R. S., Tucker, D. M., Silberstein, R. B., & Cadusch, P. J. (1997). EEG coherency I: Statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and Clinical Neurophysiology*, 103(5), 499–515. [https://doi.org/10.1016/S0013-4694\(97\)00066-7](https://doi.org/10.1016/S0013-4694(97)00066-7)

- Park, S. M., Jeong, B., Oh, D. Y., Choi, C. H., Jung, H. Y., Lee, J. Y., Lee, D., & Choi, J. S. (2021). Identification of Major Psychiatric Disorders From Resting-State Electroencephalography Using a Machine Learning Approach. *Frontiers in Psychiatry*, 12(August), 1–12. <https://doi.org/10.3389/fpsy.2021.707581>
- Pattern, Q. R. C. (2022). *Steady-State Visual Evoked Potential-Based Brain – Computer*.
- Perez, A. (2023). *Neural decoding of InterAxon Muse data using a recurrent convolutional neural network*. Google colab. https://colab.research.google.com/drive/1q1mvIC1xgEuPNNvgASrAG7zie3qRjr-8#scrollTo=fBUn_aFMDk3h
- Ritov, G., Ardi, Z., & Richter-Levin, G. (2014). Differential activation of amygdala, dorsal and ventral hippocampus following an exposure to a reminder of underwater trauma. *Frontiers in Behavioral Neuroscience*, 8, 18. <https://doi.org/10.3389/fnbeh.2014.00018>
- Sameshima, K., & Baccalá, L. A. (1999). Using partial directed coherence to describe neuronal ensemble interactions. *Journal of Neuroscience Methods*, 94(1), 93–103. [https://doi.org/10.1016/s0165-0270\(99\)00128-4](https://doi.org/10.1016/s0165-0270(99)00128-4)
- Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2), 461–464. <https://doi.org/10.1103/PhysRevLett.85.461>
- Schwizer Ashkenazi, S., Roell, M., McCaskey, U., Cachia, A., Borst, G., O’Gorman Tuura, R., & Kucian, K. (2024). Are numerical abilities determined at early age? A brain morphology study in children and adolescents with and without developmental dyscalculia. *Developmental Cognitive Neuroscience*, 67, 101369. <https://doi.org/https://doi.org/10.1016/j.dcn.2024.101369>
- Sharp, R., Swerdlow, N. R., & Braff, D. L. (2011). EEG and ERPs. In *NIH Public Access* (Issue 619). <https://doi.org/10.1002/0471142301.ns0625s52.Electroencephalography>
- Shimizu, H., & Srinivasan, R. (2022). Improving classification and reconstruction of imagined images from EEG signals. *PLoS ONE*, 17(9 September), 1–16. <https://doi.org/10.1371/journal.pone.0274847>
- Shin, L. M., & Liberzon, I. (2010). The neurocircuitry of fear, stress, and anxiety

- disorders. *Neuropsychopharmacology*, 35(1), 169–191.
<https://doi.org/10.1038/npp.2009.83>
- Sims, C. A. (1972). Money, Income, and Causality. *The American Economic Review*, 62(4), 540–552. <http://www.jstor.org/stable/1806097>
- Siribunyaphat, N., & Punsawad, Y. (2023). Brain–Computer Interface Based on Steady-State Visual Evoked Potential Using Quick-Response Code Pattern for Wheelchair Control. *Sensors*, 23(4). <https://doi.org/10.3390/s23042069>
- Spampinato, C., Palazzo, S., Kavasidis, I., Giordano, D., Souly, N., & Shah, M. (2017). Deep learning human mind for automated visual classification. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 4503–4511. <https://doi.org/10.1109/CVPR.2017.479>
- Strange, B. A., Witter, M. P., Lein, E. S., & Moser, E. I. (2014). Functional organization of the hippocampal longitudinal axis. *Nature Reviews Neuroscience*, 15(10), 655–669. <https://doi.org/10.1038/nrn3785>
- Tarwidi, D., Pudjaprasetya, S. R., Adytia, D., & Apri, M. (2023). An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach. *MethodsX*, 10(December 2022), 102119. <https://doi.org/10.1016/j.mex.2023.102119>
- Tiwari, S., Goel, S., & Bhardwaj, A. (2023). EEG Signals to Digit Classification Using Deep Learning-Based One-Dimensional Convolutional Neural Network. *Arabian Journal for Science and Engineering*, 48(8), 9675–9691.
<https://doi.org/10.1007/s13369-022-07313-3>
- van de Ven, V., Jacobs, C., & Sack, A. T. (2012). Topographic contribution of early visual cortex to short-term memory consolidation: a transcranial magnetic stimulation study. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, 32(1), 4–11. <https://doi.org/10.1523/JNEUROSCI.3261-11.2012>
- Vanneschi, L., & Silva, S. (2023). Bayesian Learning. In *Natural Computing Series*.
https://doi.org/10.1007/978-3-031-17922-8_9
- Wise, R. A. (2002). Brain reward circuitry: Insights from unsensed incentives. *Neuron*,

36(2), 229–240. [https://doi.org/10.1016/S0896-6273\(02\)00965-0](https://doi.org/10.1016/S0896-6273(02)00965-0)

Wulsin, D., Gupta, J., & Mani, R. (2012). Modeling EEG waveforms with semi-supervised deep belief nets. *Journal of Neural Engineering*, 8(3), 1–28.

<https://doi.org/10.1088/1741-2560/8/3/036015>.Modeling

Xu, Y. (2007). The role of the superior intraparietal sulcus in supporting visual short-term memory for multifeature objects. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, 27(43), 11676–11686.

<https://doi.org/10.1523/JNEUROSCI.3545-07.2007>

Zago, L., Pesenti, M., Mellet, E., Crivello, F., Mazoyer, B., & Tzourio-Mazoyer, N. (2001). Neural correlates of simple and complex mental calculation. *NeuroImage*,

13(2), 314–327. <https://doi.org/10.1006/nimg.2000.0697>

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.