



UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND BIOMEDICAL INFORMATICS

Analysis and processing of Electroencephalography(EEG) Signals with deep learning techniques

Daskalou Areti

Supervisor

Delibasis Konstantinos

Associate Professor

Lamia,2024



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ

Επεξεργασία και Ανάλυση Σημάτων Ηλεκτροεγκεφαλογραφίας(ΗΕΓ) με την Χρήση Τεχνικών Μηχανικής Μάθησης

Δασκάλου Αρετή

Επιβλέπων

Δελήμπασης Κωνσταντίνος

Αναπληρωτής Καθηγητής

Λαμία, 2024

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι: 1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα. 2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία. 3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια 4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 19/06/2024

ΔΑΣΚΑΛΟΥ ΑΡΕΤΗ

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

Τριμελής Επιτροπή:

Δελήμπασης Κωνσταντίνος, Καθηγητής(επιβλέπων)

Ιακωβίδης Δημήτριος, Καθηγητής

Σαβελώνας Μιχάλης, Επίκουρος Καθηγητής

Contents

1	Introduction	7
1.1	Human Brain Anatomy	8
1.2	Electroencephalography -EEG	9
2	Literature Review	11
2.1	Music Perception	11
2.2	Music Processing in the Auditory System	13
2.3	The Primacy of Tonality in Music Perception	13
2.4	Linear models for the brain	14
2.5	The objectives of this work	14
3	Methodology	16
3.1	Temporal Response Function Estimation -TRF	17
3.1.1	Linear Forward Model	19
3.1.2	TRF calculation	20
3.1.3	Backward Models	22
3.1.4	Regularization	24
3.1.5	Time Lags	25
3.2	Multivariate Temporal Response Function Estimation -mTRF	26
3.3	Deep Learning	27
3.3.1	ED Architecture	28
3.3.2	Pretraining the Autoencoder	31
4	Results and Analysis	34
4.1	Dataset Acquisition	34
4.2	Electrode Preprocessing	35
4.3	Utilizing the mTRF	38
4.4	Deep Learning Predictive Model	42

4.4.1	Autoencoder part	46
4.4.2	Encoder-Decoder Part	50
4.5	Comparison of the 2 methods	54
5	Discussion and Conclusions	57
6	Bibliography	59

Abstract

Music, a universal language that transcends cultures and generations, has the profound ability to evoke emotions, trigger memories, and move us in profound ways. But how do we actually perceive music? What happens within our brains that allows us to decipher the complex interplay of pitch, rhythm, harmony, and timbre?

The interplay between music and the human brain has long captivated researchers, with music's profound impact on cognitive and emotional states being a focal point of numerous studies. Electroencephalography (EEG) offers a non-invasive window into the brain's electrical activity, making it an invaluable tool for investigating neural responses to musical stimuli. However, traditional EEG analysis methods often struggle to adequately handle the complex, high-dimensional nature of EEG data, especially in the context of music.

This thesis explores the application of advanced computational techniques, specifically encoder-decoder models and linear signal processing models based on the Multivariate Temporal Response Function (mTRF), to enhance the processing and analysis of EEG signals elicited by musical stimuli. We used EEG data from participants exposed to a musical piece, meticulously preprocessed to remove noise and artifacts.

Leveraging these advanced models and tools, we analyzed the temporal and spatial features of the EEG signals with respect to its input stimulus. Our findings demonstrate that encoder-decoder models, significantly outperform traditional methods such as mTRF method in predicting neural patterns associated with music perception, in terms of improved accuracy. This research highlights the potential of these advanced computational tools to revolutionize EEG analysis, offering robust methodologies for uncovering the neural dynamics of music perception.

1 Introduction

The study of the human brain's response to music has fascinated researchers for decades and that it because music triggers human brain in several ways. Electroencephalography (EEG) provides a non-invasive means to measure electrical activity in the brain, offering insights into the neural mechanisms underlying these effects. By analyzing EEG signals, we can gain a deeper understanding of how the brain processes musical stimuli and how these processes vary among individuals. Despite the advancements in EEG analysis, traditional methods often face challenges in handling the complexity and high dimensionality of EEG data. Conventional techniques may fall short in capturing intricate patterns and subtle variations in brain activity elicited by musical stimuli. This limitation has spurred interest in applying deep learning techniques to EEG analysis. Deep learning, with its ability to model complex, non-linear relationships, holds the potential to revolutionize our understanding of brain responses to music. The primary objective of this thesis is to explore and develop deep learning methodologies for processing and analyzing EEG signals obtained in response to music. By leveraging the capabilities of deep learning, we aim to:

1. Enhance the accuracy and efficiency of EEG prediction and classic signal processing.
2. Identify and characterize neural patterns associated with different neural (musical) stimuli.
3. Investigate the potential of deep learning models to differentiate between various cognitive and emotional states induced by music.

This research will involve the collection of EEG data from subjects exposed to a variety of musical pieces. The data will undergo preprocessing to remove noise and artifacts, followed by feature extraction using traditional and deep learning techniques. Encoder-Decoder models and regression techniques will be employed to analyze the temporal and spatial characteristics of the EEG signals. The results will be compared with those obtained from conventional methods to evaluate the effectiveness of deep learning in this context.

The findings of this study are expected to contribute significantly to the fields of neuroscience, psychology, and music therapy. Understanding how the brain responds to music can inform therapeutic practices for mental health conditions, enhance musical training programs, and provide insights into the cognitive processes involved in musical perception and appreciation.

Moreover, the application of deep learning to EEG analysis represents a cutting-edge approach that can pave the way for future research in neurotechnology and artificial intelligence.

1.1 Human Brain Anatomy

It is known that the human body consists of a series of systems that regulate functions and serve its needs. One of them, the nervous system is responsible for transmitting information from the external environment to the rest of the body, processing information and responding to various stimuli. The nervous system is divided into the Central Nervous System (CNS) which consists of the brain and spinal cord and the Peripheral Nervous System (PNS) which consists of the nerve branches of the rest of the body. Most of the brain's connections to the body are through the spinal cord, which descends from the spine. Thus, the CNS is enclosed in a bony sheath, the brain in the skull and the spinal cord in the spine. It is called Central because it is physically located at the centre of the nervous system, as well as because it is the basic structure that influences behaviour, perception, language, and memory. The nervous system is made up of one type of tissue: nervous tissue. In turn, nervous tissue is composed of nerve cells and glial cells. Nerve cells transmit nerve impulses (action potentials) which is the main role of nervous tissue, while glial cells assist this function by nourishing and protecting nerve cells and facilitating nerve impulses. Nerve cells or neurons, receive, modify and transmit the various stimuli while being responsible for regulating and performing both muscular movements and higher mental functions. The brain, being the most complex organ of our body, is divided into 2 cerebral hemispheres, almost symmetrically. Externally, the brain is surrounded by a folded layer of nerve tissue called the cerebral cortex. The cortex of each hemisphere includes clearly defined functional zones and is divided into 2 lobes, the frontal, temporal, occipital and parietal. The frontal lobe is located in the front of the brain and is associated with reasoning, movement and emotions, the parietal lobe is behind the frontal lobe and is associated with the perception of touch, pressure, temperature and pain while the temporal lobe behind from the lateral fissure (separates the frontal from the temporal and parietal lobes) and is associated with hearing and memory and finally the occipital lobe is located at the back of the brain, behind the parietal and temporal lobes and is associated with various aspects of vision. A detailed figure describing all

the regions is available in 1.

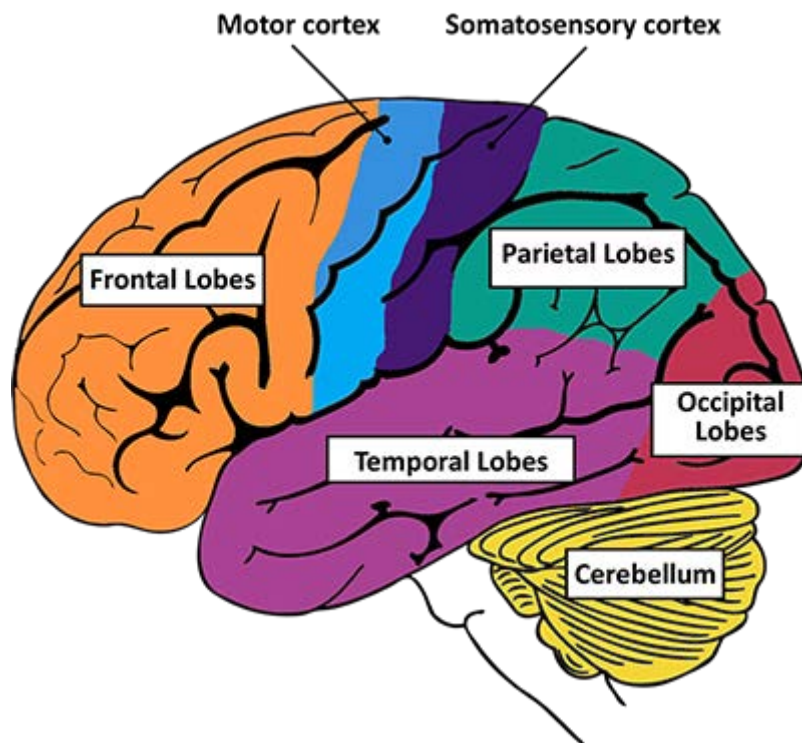


Figure 1: Brain regions [1]

1.2 Electroencephalography -EEG

Information is transmitted to the organs through electrical signals, which today it is possible to visualise in order to extract a graph of the electrical activity of the brain. This process, called electroencephalography (EEG), is a diagnostic tool and involves the recording of electrical signals from the cerebral cortex using electrodes placed on the surface of the skull. The brain reacts by changing its potential to stimulation of the sensory organs or sensory nerves resulting in the generation of an evoked potential which helps to detect the sensory centres of the brain. This potential is created by the movement of kations due to the unequal number and type of charges on both sides of the nerve cell membrane, which is activated and transitions from the resting potential to action potential. The electrodes record changes of electric potential in the extracellular space of the cortex. When performing the electroencephalogram, the electrodes are placed according to certain patterns which determine the organisation of the electrodes

according to the potential differences created at the positions where they are placed.

The electroencephalogram is performed by placing the electrodes in specific positions depending on the area of the brain where the electrode is placed and the potential difference that is recorded. In particular, the best-known method of placement to date is the one proposed by Jasper in 1958 and now called the International 10-20 System. At this point, we must point out some more specific regions, which are used as symbols to correspond to the electrodes located in them, more specifically:

- The prefrontal region (Fp), which essentially includes the frontal region.
- The frontal area (F), which is the area at the front of the brain, behind the forehead.
- The central area (C) directed along the central axis of the skull.
- The lateral or parietal area (P) defined as the area of the brain at the top of the head behind the frontal lobes.
- The occipital region (O) located at the back of the head.
- The temporal region (T), which includes the areas located at the side of the brain, behind the ears (A).

The odd numbered electrodes are placed on the left and the even numbered ones on the right. In figure 2 there is a montage of the EEG process along the brain regions.

The **electrocorticogram (ECoG)** is a brain signal recorded using a large electrode placed directly on the cerebral cortex. It's primarily used to locate the source of seizures in patients with epilepsy who don't respond to medication. Additionally, ECoG is gaining traction in research on brain function and brain-computer interfaces. Understanding the "spatial spread" of ECoG is crucial for accurately pinpointing the seizure zone and interpreting cognitive processes. This spread refers to the area of cortical tissue contributing to the activity picked up by the ECoG electrode. However, the exact extent of the ECoG spread remains unknown. Even the spread of a similar brain signal called the local field potential (LFP), recorded using smaller electrodes, is debated, with estimations ranging from a few hundred micrometres to several millimetres.

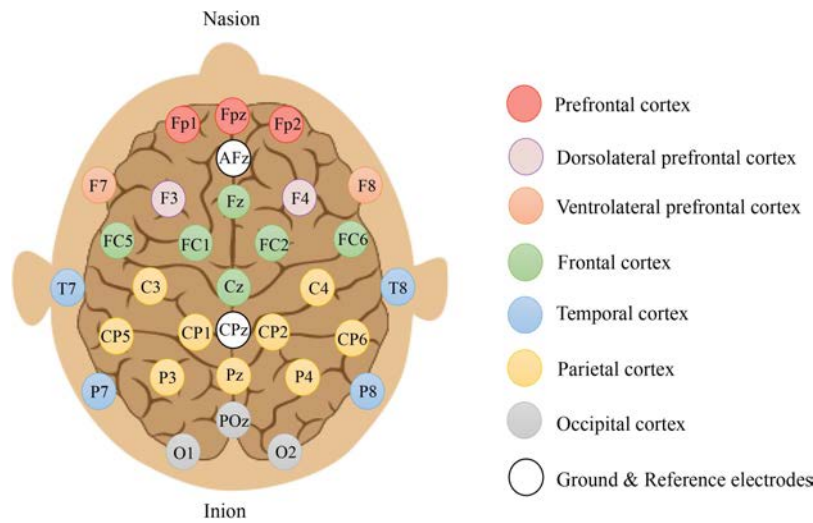


Figure 2: Electrodes for EEG acquisition. [2]

Researchers have attempted to estimate ECoG spread by measuring the "receptive field" (the area of the brain a stimulus activates) and multiplying it by a "cortical magnification factor." However, this method overestimates the spread because the size of the receptive field can be inflated by various factors. A more accurate approach might involve comparing the receptive fields of two different brain signals, such as LFP and multi-unit activity (MUA), using a specific model.

2 Literature Review

2.1 Music Perception

Music constitutes both a fundamental aspect of human expression, and one of the most complex and demanding cognitive challenges that the human brain can undertake. Comparatively with most other sensory–motor activities, music performance requires precise timing of several hierarchically organized actions, as well as precise and careful control over pitch interval production, tailored to the specific instrument being played. Exploring the neural correlates involved in or altered by the human engagement with music has become a growing focus in neuroscience, as it also represents one of the oldest and most basic socio-cognitive questions regarding human species[3] That, it is mostly studied from trying to determine the relationship between human

response and comprehension of different stimuli since the ancient years. This response, most of the times, affects main brain functions and every cognitive process involved: perception, action, cognition, social cognition, emotion, learning and memory. As a result, by recording brain activity during human engagement with music, we could understand and investigate more about of both the nature of different kind of responses and the underlying brain mechanisms. For this specific goal, one of the most prominent for auditory neuroscience, multiple studies have developed accurate and specific tests and experiments either behavioural (discrimination tasks [4], emotional reaction games [5]), neuroimaging (EEG [6], fMRI [7]) and also computational studies that include Artificial Neural Networks (ANNs [8]) to study brain activity. At the same time, musical perception is highly correlated with the personal characteristics of the listener such as age, cultural level, socioeconomic status, music interests, familiarity with the music genre, psychological state and music preference [9]. Moreover, the cognitive component in music understanding must be underlined due to what brain has learned from previous experiences on the tonal structure in our culture which is also part of our long-term memory, developed by the societal factors. Our ability to predict, anticipate and expect these patterns is connected to our perception of music and is occurring (most of the times) without us being aware of it. As we also referred before, it is crucial for each listener its exposure to different musical elements as it determines its emotional response. It has been stated that also, music preference is due to the completion of an expectation of a pattern and that, specifically infants tend to like familiar music structures, and their very early preferences can be altered by exposure to more than one culture specific structure [10]. Listeners familiar with certain styles while listening to them they can anticipate how the notes and melodies will unfold. Research by Zatorre shows that specific parts of the brain store musical patterns and interact with areas involved in memory and pattern recognition. This creates our expectations for what comes next in the music. While listening familiar musical styles, altering these expectations can evoke the feeling of surprise because unexpected chords can produce physical reactions, and the intensity of our emotional response is often tied to how strongly our expectations were violated. So, when listening to music we're used to, the feelings we get come from our brain's constant work of figuring out what happens next - expectation, prediction, and anticipation.

2.2 Music Processing in the Auditory System

Our journey into music perception begins with the auditory system. Sound waves, upon entering the outer ear, cause the eardrum to vibrate. These vibrations are then conveyed to the middle ear, where tiny bones amplify them before they reach the inner ear, or cochlea. Inside the cochlea, hair cells transform these vibrations into electrical signals transmitted through the auditory nerve to the auditory cortex in the brain. However, not only the auditory cortex but also other brain regions play major role in these signals' processing but also: *The limbic system*: This region, associated with emotions and memory, plays a vital role in connecting us to the emotional content of music.

The motor cortex: This area, involved in movement and coordination, can trigger physical responses like tapping our feet or nodding our heads in sync with the rhythm.

The prefrontal cortex: This higher-order area, responsible for complex cognitive functions like attention, memory, and decision-making, allows us to analyze and interpret the music we hear.

2.3 The Primacy of Tonality in Music Perception

In the realm of music perception, tonality plays a critical role. Present in Western music traditions since the medieval era, both in academic and popular contexts, tonality provides the foundational structure of harmony. This structure revolves around the relationships between musical intervals, which can be categorized as consonant (generally perceived as pleasant) or dissonant (generating a sense of tension or unease). Music perception as consonant and dissonant is also a research domain on modern neuroscience. Musical syntax often leverages these contrasting qualities; consonant chords tend to evoke relaxation while dissonant chords create tension. It's important to understand that music theory, encompassing the principles of harmony, is a vast and nuanced field transformed throughout history by various stylistic shifts and cultural influences. Our overview of intervallic relationships offers a simplified starting point. Another very important issue that has been extensively dealt is the predictability of the tonal system. Structurally, tonal music evolves temporarily within specific tonal/spectral ranges and with relatively low uncertainty (entropy) limits that make it reasonably predictable. Recent

theories along these lines, such as that of McDermot, analyze familiarization with the tonal system of society against theories of auditory neurobiology underlying the attraction factor to the tonal structure [11]. This aspect of neurobiology has been discussed by Bowling et al., who analyse this issue and conclude that without the exposure of the general population to the system or tonal structure, it is undeniable that there is underlying biological evidence that demonstrates that both concepts are closely linked [12]. Western academic music is also about defining rhythmic forms while listening, by recognizing patterns and forms as time goes by and for example being able to predict or expect a rhythm or a sequence.

2.4 Linear models for the brain

It is well known that linear models are the first method of choice for signal processing. Thus they have been broadly used in neuroscience in order to investigate multiple brain signals. Specific applications include, processing of different kinds of stimuli in speech, such as [13], and also the investigation of transition between brain states [14]. Furthermore, others are trying to inspect the characteristics of commonly used filters as Gabor filters to decompose the information of Spectro-Temporal Receptive Fields (STRFs) in more Gabor filters as it is analyzed in [15]. Current ridge regression methods for performing reverse correlation obtain STRF estimates that are robust under variations of second-order correlations in the stimulus ensemble, including to some extent natural stimuli ([16], [17]). A major concern is also the size of the dataset in order to fit the linear models because in most cases it is unclear the size of data needed to achieve legit results as it is investigated in [18]. The complex problem of neural tracking is also researched in [19] where mTRFs are also used in order to compute linear bands of frequencies in the Electroencephalography Signals and their variations.

2.5 The objectives of this work

The main purpose of this work is to investigate possible methods to predict the neural response of specific EEG electrodes, given the dynamic spectrogram of a sound stimulus, in this case a music piece. The mathematical techniques used are both linear and non-linear methods and to gain meaningful insights from their performance and evaluation. First traditional signal analysis

techniques are used, specifically the mTRF which is a generalization into multiple frequencies of the STRF. Subsequently, deep learning techniques are also tested in the scope of this problem. We are proposing the architecture of auto-encoder (AE), which is a special case of the encoder-decoder (ED) models to solve the problem of feature extraction of our stimulus data, learn the patterns of stimulus and compress them into a much smaller latent space. Then another ED utilises the training of the AE, in order to generate meaningful estimations of the brain responses (EEG signal) in specific electrodes. We also attempt to analyze the correlation of mTRF or other underlying brain mechanisms with neighboring brain regions.

3 Methodology

In this study, we investigate the prediction one's EEG response to a stimulus of a song. More specifically, a song by Pink Floyd was played to different patients in a long-term study that is trying to achieve sound reconstruction of this complex stimulus using classic signal analysis techniques [20]. We tried to find different non-linear techniques that could predict the patient's response effectively and we will present them below. Our approach can be summarized in the following steps:

Step 1: First we used mTRF method with input the song spectrogram and the EEG recording of one electrode in order to get the mTRF of this specific channel

Step 2: Subsequently, we used the predicted mTRF in combination with the input in order to get a prediction for the output EEG response which we compared with our ground truth responses

Step 3: Moreover, we used a Deep Learning approach where first we pretrained an autoencoder with the spectrogram inputs

Step 4: Subsequently, we used the pretrained encoder part in order to train a Encoder-Decoder model which predicted the EEG responses

Step 5: We compared both techniques on their ability to predict the output EEG response.

Fig.3 provides an overview of all the methods with the main steps used.

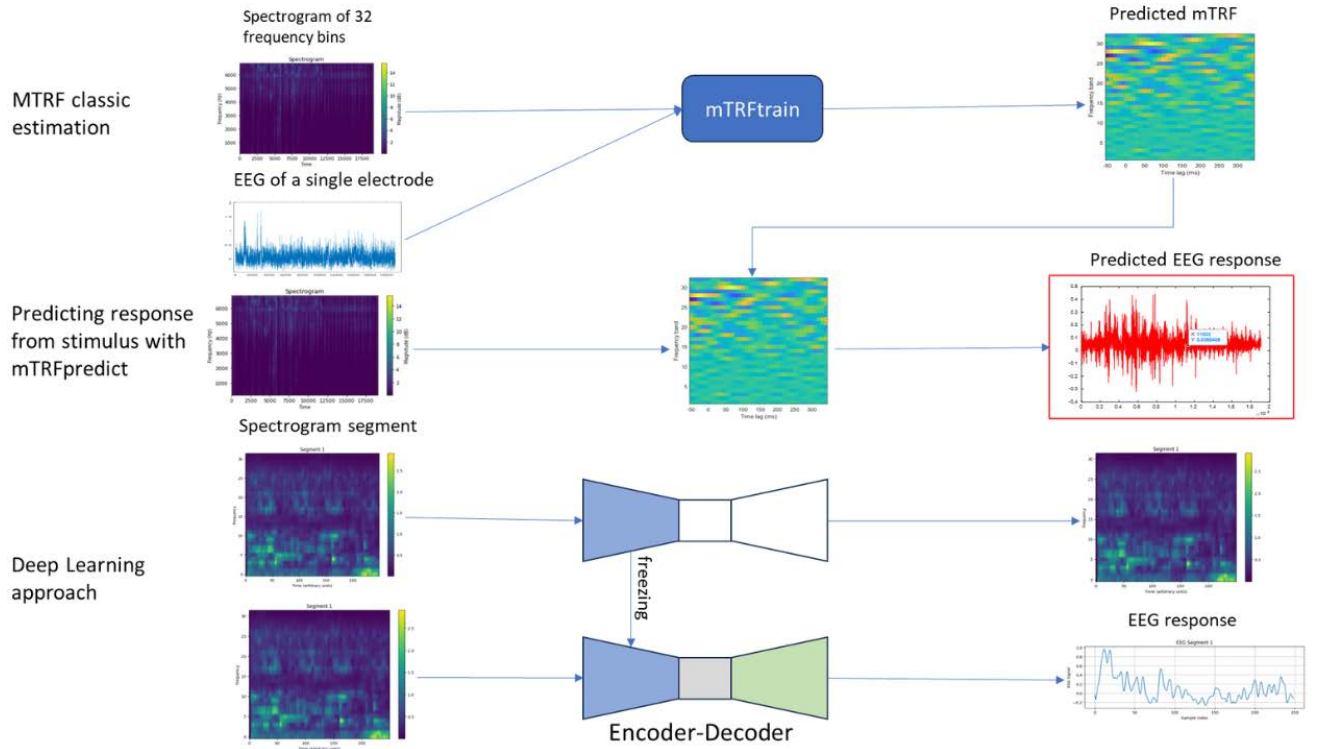


Figure 3: Overview of the proposed methods

3.1 Temporal Response Function Estimation -TRF

As the auditory system encodes music information, the sensory neurons involved, are usually modeled using linear models and more specifically linear Finite Impulse Response (FIR) filters. FIR filters are linear filters therefore their output is a linear function of their input and they pass a frequency lower than a selected cutoff frequency, while they attenuate signals higher than the cutoff frequency. This function can be expressed by the convolution operator that acts on the input signal. We define convolution of two functions, $f(x)$ and $g(x)$ to be the integral:

$$(f * g)(x) = \int f(t)g(x - t) dt \quad (1)$$

In some sense, one is looking at a sum of the overlaps of one of the functions and all the shifted versions of the other function. In this case, the two components of the convolution are the input signal and the impulse response of the filter [17]. Consequently, it can be used to understand the connections between sound features and the neuron's activity. In the auditory system, the

linear Spectro -Temporal Receptive Field (STRF) is implemented like a FIR filter representing the neural activation of sensory neurons. The Spectro -Temporal Receptive Field (STRF) of an auditory neuron acts like a fingerprint by capturing the neuron's sensitivity in both frequency and time. On the frequency side, the STRF highlights which specific sound frequencies the neuron excites more or the ones it suppresses, providing information of the tuning frequencies of each neuron. At the same time, on the temporal side, the STRF indicates how the neuron response unfolds over time and if it is continuous or not. Therefore, we are able to discover both the spectral and the temporal abilities of neurons of interest and to explore their firing patterns along frequency and time.

In this scope, mTRF method was developed, in order to map the stimulus representation of a signal to its Spectral and Temporal Response by predicting the STRF of the input signal[21]. This mapping is done with a System Identification (SI) technique called regularized linear regression , responsible for calculating Temporal Response Functions for the input signal. The Temporal Response Function is practically the impulse response function of the sensory neuron convolved with an impulse stimulus input as we described before. the implemented functionality is publicly available as a Matlab Toolbox [22]. We used this toolbox for the results in this work.

There are many ways of estimating the TRF but in this case we will describe the SI technique that includes mathematically modelling an equation that predicts how neurons in a sensory system (like vision or hearing) respond to a particular aspect of the stimulus (like light intensity or sound frequency). The most discussed method is the forward approach meaning the mapping from stimulus to response while it is also interesting to discover the opposite direction of this system to investigate how different properties of a stimulus are reflected in the firing patterns of neurons. This process is called stimulus reconstruction while we are trying to predict inputs' meaningful characteristics from the patients response. It provides a lot of advantages when it comes to EEG, because the multi-channel system could enhance accuracy of prediction and at the same time it is inclusive while it automatically downplays the influence of irrelevant channels by assigning them low weights. This also allows the model to capture additional variations in the data using channels that might have been excluded by feature selection techniques.

3.1.1 Linear Forward Model

The models that have been described before, try to predict the patient's EEG recording using the corresponding stimulus, thus they are called forward models. Often they are also mentioned as encoding models because they are encoding input's information according to [23]. We will describe them mathematically as following below, referring to them as Temporal Response Function or TRF. The weights of a TRF describe how the EEG signal on a given recording channel in response to an input impulse. The temporal dynamics of a TRF typically correspond to those of an ERP(Event-Related Potential) according to [24] , [25]. However, while ERP components describe how the EEG signal on a given recording channel changes in response to the entire stimulus event, TRF allows for predicting the neural response to a given stimulus subsequence, which allows better generalization to new data [23] , [26]). TRF in combination with the input sequence can provide a useful representation of the output neural response. This simultaneous neural response of N recording channels can be represented as $r(t, n)$ sampled times $t = 1...T$ at channel n , comprises of a convolution of the stimulus instance $s(t)$ with an unknown channel-specific TRF, $w(\tau, n)$. For discrete time system , we can describe the response model as follows:

$$r(t, n) = s * w + \varepsilon(t, n) = \sum_{\tau=-\tau_{min}}^{\tau_{max}} w(\tau, n)s(t - \tau) + \varepsilon(t, n) = \hat{r}(t, n) + \varepsilon(t, n), \quad (2)$$

where $*$ is the convolution operator and $\varepsilon(t, n)$, is the residual response from each channel n not included in our linear model. The TRF $w(\tau, n)$ describes this transformation for a specific time lag range τ , which is relative to the simultaneous occurrence of the stimulus feature $s(t)$. The values of τ_{min} and τ_{max} represent the absolute minimum and maximum time lags measured in discrete samples. These lags represent the temporal range over which the influence of the stimulus signal s on the neural response r is investigated as it was before mentioned. In our model, typically, the time range of τ must be relative to the simultaneous stimulus input $s(t)$.

3.1.2 TRF calculation

The convolution of Eq.(2) can be implemented as a multiplication with the Toeplitz matrix S of the input signal s , where S is given by:

$$S = \begin{pmatrix} s(1 - \tau_{\min}) & s(-\tau_{\min}) & s(-1 - \tau_{\min}) & s(1) & 0 & \dots & \dots & 0 \\ s(2 - \tau_{\min}) & s(1 - \tau_{\min}) & s(-\tau_{\min}) & \vdots & s(1) & \dots & \dots & \dots \\ s(3 - \tau_{\min}) & s(2 - \tau_{\min}) & s(1 - \tau_{\min}) & \vdots & \vdots & \dots & 0 & \dots \\ \vdots & \vdots & \dots & \dots & \vdots & \vdots & \dots & s(1) \\ s(T) & \vdots & \dots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & s(T) & \dots & \dots & \vdots & \vdots & \dots & \vdots \\ \vdots & 0 & s(T) & \dots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & 0 & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & s(T) & s(T-1) & \dots & s(T - \tau_{\max}) \end{pmatrix}$$

The TRF $w(\tau, n)$, is estimated by minimizing the mean-squared error (MSE), ε between the actual neural response $r(t, n)$ and the predicted value which is calculated by the convolution in the equation (2). Matrix S is constructed by arranging each time lag τ (excluding zero) as a separate column. To ensure causality, non-zero lags are padded with zeros in S . Causality implies that the neural response cannot precede the stimulus[13]. If we denote by $\hat{r}$ the predicted response and by n the index of the recorded channel (electrode), then:

$$\min \varepsilon(t, n) = \sum_t [r(t, n) - \hat{r}(t, n)]^2 \quad (3)$$

By using Newton's pseudo-inverse matrix [27] we can solve this linear system of equations as follows:

$$w = (S^T S)^{-1} S^T r, \quad (4)$$

The window size is defined as $\tau_w = \tau_{\max} - \tau_{\min}$. This window size determines the temporal extent of the influence of $s(t, n)$ on $r(t, n)$. Consequently, the dimensions of matrix S are

$T \times \tau_w$, where T represents the total number of samples in the stimulus signal. To incorporate the constant term (y-intercept) in the regression model, a column vector containing only ones is concatenated to the left side of matrix S . This column accounts for the baseline neural response independent of the stimulus that must be included. In Equation (4), variable r represents a matrix containing all the recorded neural response data. This data is arranged with channels as columns, resulting in a matrix of dimensions $T \times N$. Here, N signifies the total number of recorded neural channels. The estimated transfer function (TRF), denoted by w , is a matrix of dimensions $\tau_w \times N$. Each column of w represents the time-varying, univariate mapping from the stimulus s to the neural response in a specific channel. In other words, each column of w captures the temporal profile of how the stimulus influences the activity in a particular neural channel.

A crucial aspect of this analysis is its explicit consideration of the auto covariance structure inherent in the stimulus signal. Unlike white noise (uncorrelated at all time lags), natural stimuli, such as speech, exhibit time-varying intensity. This characteristic leads to non-zero lag correlations within the stimulus itself. Consequently, a straightforward cross-correlation between the speech envelope and the neural response would result in a "smearing" effect on the estimated impulse response function. This smearing arises because the cross-correlation captures not only the true relationship between the stimulus and the neural response but also the inherent correlations within the stimulus at different time lags. Equation (3) addresses this issue by incorporating the auto covariance structure of the stimulus into the model. By dividing out this structure, the model effectively removes the spurious correlations between different time lags within the stimulus, leading to a more accurate estimate of the TRF. The TRF approach, therefore, offers a significant advantage over a simple cross-correlation by mitigating the temporal smearing effect. The next section will demonstrate this benefit through a worked example. We are visualizing here an example of a speech TRF with figure 4.

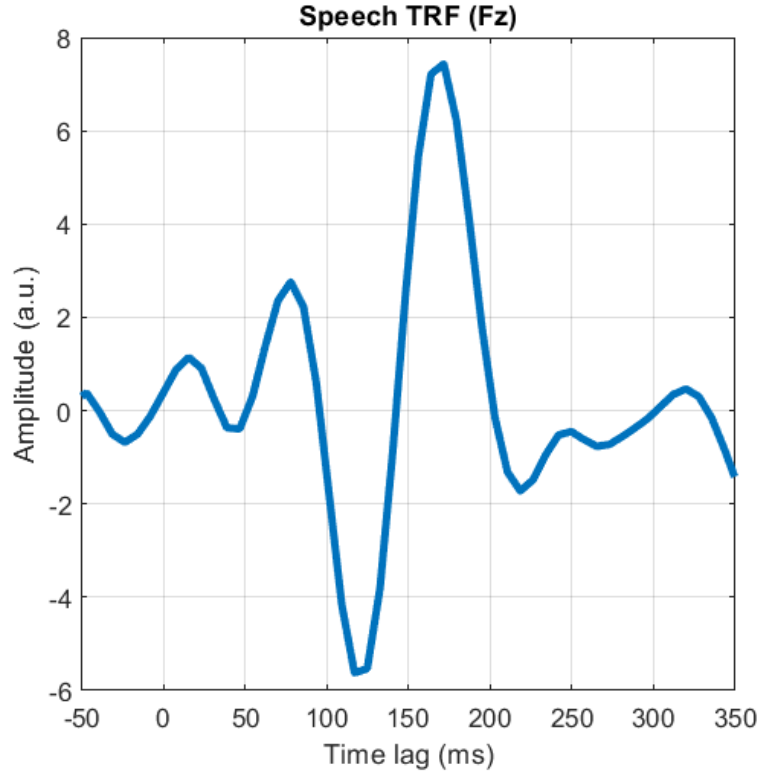


Figure 4: TRF estimation of a speech signal digitized at 512Hz in a time window of (-50 - 350)ms

3.1.3 Backward Models

In order to achieve the inverse of the encoding models, we introduce here the decoding or backward models. Those linear methods, consider each neural response channel as a separate univariate characteristic and try to map it in order to unravel the stimulus sequence. Backward model uses all multivariate neural data in order to map the and reconstruct the input signal. Therefore, we are able to quantify how well specific stimulus features could be decoded given the neural activity under a variety of experimental conditions or in different patients from the same subject group. As they are trying to decode the input , they are often mentioned as decoders and they can be modeled and represented as the TRF in the previous section. Let's declare decoder $g(\tau, n)$ the linear transformation of the output $r(t, n)$ to the input $s(t)$.

$$\hat{s}(t) = \sum_n \sum_{\tau} r(t + \tau, n) g(\tau, n) \quad (5)$$

with $\hat{s}$ to be the decoded stimulus instance in the time range of the time lags τ . Those time lags are supposed to match the stimulus neural dynamics, so they have to be accurate and precise to get the range correctly. While forward TRF looks at how past stimuli affect current neural activity, the decoder looks at current neural activity to infer past stimuli forming this temporal inversion, key to decoding. It is necessary, to reverse the time lags used before, ($\tau_{min} = -100$ ms, $\tau_{max}=400$ ms). Also, the decoder $g(\tau, n)$ is estimated from the MSE difference of $s(t)$ and $\hat{s}(t)$ for that frequency channel:

$$\min \varepsilon(t) = \sum_t [s_f(t) - \hat{s}_f(t)]^2 \quad (6)$$

Using again reverse correlation the equation is solved and the decoder can be represented as follows:

$$g = (R^T R + \lambda I)^{-1} R^T s \quad (7)$$

with R to be the lagged time series of the output matrix r , where also R can be initialized for a single channel:

$$R = \begin{pmatrix} r(1 - \tau_{min}, 1) & r(-\tau_{min}, 1) & \dots & r(1, 1) & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & r(1, 1) & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \dots & \dots & \vdots & \vdots & r(1, 1) \\ r(T, 1) & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ 0 & r(T, 1) & \dots & \vdots & \vdots & \dots & \vdots \\ \vdots & 0 & \dots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & r(T, 1) & r(T - 1, 1) & \dots & r(T - \tau_{max}, 1) \end{pmatrix}$$

The above solution can be applied for the case of N channels by substituting every column of matrix R with N columns for the N channels, transforming the dimension of R with

$N\tau$ window. The matrix R is padded with zero columns on the left in order to ensure causality. [13]. Channel weights in this case include the amount of information of each channel that is used in reconstruction meaning that the magnitude of very informative channels is higher while the ones providing less information have almost zero weights.

3.1.4 Regularization

A crucial aspect of calculating the Transfer Function (TRF) is *regularization*. This technique addresses two main challenges: ill-posedness and overfitting. The estimation of the TRF often involves inverting the auto covariance matrix of the stimulus, denoted as $S^T S$. Matrix inversion can be numerically unstable, especially when dealing with finite-precision computer calculations. Even minor changes in $S^T S$ (e.g., rounding errors) can lead to significant variations in the estimated TRF (w) if the matrix is ill-conditioned (has a low determinant). In simpler terms, the estimate of w becomes highly sensitive to small variations in the stimulus data. This instability is less of a concern for stochastic stimuli (e.g., white noise) because their auto covariance matrices are typically full-rank [24]. However, natural stimuli like speech have non-white characteristics leading to potentially singular auto covariance matrices (determinant of zero). Regularization techniques aim to reduce the variance of the TRF estimate, even if it introduces a slight bias. The overall estimation error arises from both bias (difference between the estimate and the true value) and variance. Regularization strategically increases the bias in a controlled manner to significantly reduce the variance, leading to a lower overall estimation error. Furthermore, regularization addresses the issue of overfitting. The reverse correlation analysis used for TRF estimation doesn't consider the underlying biological nature of the neural response data. Consequently, an unregulated TRF might be optimized for the specific dataset used for its calculation (e.g., minimizes least squares error), potentially capturing noise or dataset-specific features. This can lead to biologically implausible characteristics like high-frequency fluctuations. Using such an overfit TRF for prediction on unseen data will likely perform poorly because the high-frequency features won't generalize well to the new data's noise characteristics. Regularization helps prevent overfitting to these dataset-specific noise components, particularly in low-variance dimensions [28]. One common approach penalizes large differences between

neighboring TRF values, enforcing smoothness in the estimated TRF. This smoothness promotes better generalization to unseen data by reducing the TRF's dependence on the specific training data. The problem is that $S^T S$ has some eigenvalues of zero or nearly zero, so inverting it is not a good practice. That is why, we add a *ridge* value on the diagonal: $(S^T + \lambda I)$ where all eigenvalues are increased by λ as follows:

$$w = (S^T S + \lambda I)^{-1} S^T r, \lambda \in [0, \inf) \quad (8)$$

when $\mathbf{I}$ is the identity matrix known from mathematics and λ is the ridge parameter as it occurs in Ridge Regularization and can be altered from using cross-correlation in order to maximize the correlation between predicted $\hat{r}(t, n)$ and actual response $r(t, n)$. Ridge regularization is often used in statistics in order to correct the problem of overfitting while λ value is subjective and depends of the problem and amounts to a bias-variance trade-off. According to [24] there is the quadratic approach which is quadratically minimizing the absolute values between each neighboring term of $\mathbf{w}$:

$$w = (S^T S + \lambda \mathbf{M})^{-1} S^T r, \text{ where } \mathbf{M} = \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ \vdots & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix} \quad (9)$$

In 9 the signal's amplitude is preserved and that's why it is an improved estimate of the system's output when in Ridge Regression smoothing is not sustained.

3.1.5 Time Lags

When building your model, it's crucial to consider the time window of brain responses you want to capture. This is often referred to as the time lag. We know from brain wave studies (ERP) that sensory information takes time to travel through the brain, with different delays reflecting distinct processing stages. These delays can be in the range of milliseconds. Including

time lags in your model allows it to represent the dynamic relationship between stimuli (inputs) and brain responses (outputs). This lets the model use information from specific past or future time points to make better predictions. Both forward models and backward models use time lags for predictions. The optimal time window (range of time lags) for each model depends on the expected relationship between the stimulus and the brain response. You can determine this window either by referencing existing research by looking at studies on Time-Resolved Functional Connectivity (TRF) by experimenting with your model by trying different time windows and check the results. Examples of Time Windows are cortical responses that they typically occur within hundreds of milliseconds (e.g., 0 to 300 milliseconds) and auditory brainstem responses, that are much faster, typically happening within tens of milliseconds (e.g., 0 to 10 milliseconds). If you have a more specific research question, you might choose to focus on a narrower time window. For instance, you could analyze: Early vs. Late Processing in order to compare lags within 0 to 150 milliseconds (early) and 150 to 300 milliseconds (late). Also with time-Dependent Effects, examine how model performance changes across different time lags using a moving window or individual lags [29], [30] Including time lags before and after the response window (e.g., -100 to 400 milliseconds) can be helpful for visualization purposes. This allows you to see the baseline brain activity or estimate the background noise level. While using a wider range of time lags can improve prediction accuracy, it also increases the number of parameters in your model. This can lead to overfitting, where the model performs well on training data but poorly on unseen data. Fortunately, techniques like regularization (discussed later in the Model Training and Testing section) can help prevent the model from overfit. So, in our case we will use the typical time window, sufficient to cover the cortical response components of an Event Related Potential (ERP), which is (-100 - 400)ms. The TRF value at -50ms would indicate the change in the amplitude of the speech envelope of the EEG signal, 50ms later.

3.2 Multivariate Temporal Response Function Estimation -mTRF

The previous section explained how a single, simple feature of a sound (like its volume over time) can be related to brain activity measured by electrodes. However, real-world sounds such as music and song stimulus are usually more complex for the brain to process. For instance,

when we hear speech, our inner ear (cochlea) transforms the sound into a spectrogram through all the auditory pathway. This spectrogram shows how much sound energy is present at different frequencies over time. Our brain then uses this spectrogram information to understand the speech. A model has been developed called the "Multivariate Transfer Function (mTRF)" to understand how the brain processes spectrograms. This model assumes that different frequency bands in the spectrogram are processed separately by the brain. To calculate the mTRF, researchers use a special matrix that represents the spectrogram $s(t, f)$ over time. In this matrix, each column represents a specific frequency band from the set $f = 1 \dots F$. The mTRF is reached by an extension of the stimulus lag matrix S , that we described in equation 4, where each column is replaced with F columns as the different frequency bands and the resulting matrix has dimension $T \times F\tau_{window}$. By analyzing this matrix and brain activity recordings, the mTRF reveals how each frequency band in the spectrogram contributes to the activity measured by each electrode and it is a combination of all components: $w(f, \tau, n) = (F\tau_{window} \times N)$. Afterwards, it is a choice of the user if regularization will be applied. The standard regularization method used for simple sound features can cause problems with spectrograms. It can blur the information across different frequency bands, leading to an inaccurate picture of how the brain processes each frequency. Therefore, when analyzing spectrograms with mTRF, scientists often choose a different type of regularization that doesn't blur information across frequencies. However, in some specific cases, the standard method might still be appropriate.

3.3 Deep Learning

We tested an approach that utilises a Neural Network that tries to predict the patient's response, being fed with the sound dynamic spectrogram. An Encoder-Decoder (ED) is neural architecture has been selected, a specific type of deep neural network used to encode and decode information with the profound ability to process information much faster than traditional methods. In order to facilitate the training of the ED, we utilise an Auto-Encoder (AE) that is trained to reproduce the dynamic spectrogram that is received as input. AEs have been introduced [31],[32] as an unsupervised technique that could be trained in order to reproduce the input.

3.3.1 ED Architecture

We first designed a neural model with the structure of a classic Encoder-Decoder model. Such models are commonly used in task involving Machine Translation and Natural Language Processing(NLP) and also for other tasks involving prediction of a specific sequence such as in video classification[33]. To our purpose, we utilized such a model in order to predict a meaningful representation of the patient's response and to compare the prediction ability of the ED with another method. The ED consists of the encoder part and the decoder part. The encoder part is used to encode feature characteristics and patterns in order to extract them and create a representative instance of the extracted features. The input is being processed step by step and summarized the information in the context vector, which aims to capture the essential information of the input. Then the decoder takes the context vector produced by the encoder and generated an output sequence of whatever size. With the same technique as the encoder, the decoder inserts the hidden layer state to the context vector and produces step by step the output sequence. It is important that, the decoder is conditioned on the context vector and also with previous outputs allowing to generate an output based on learnt sequences of the network. By jointly training the encoder and decoder, the model learns to effectively encode and decode the input-output relationship. In our approach the encoder-decoder is a CNN network which converts an input spectrogram of 250(samples)x32(frequency bins) size to a predicted response of size 250(samples)x1(EEG signal). Our encoder-decoder model belongs to CNN networks as it is using convolutional layer both for the encoding and the decoding part. It is common knowledge in computer vision that, filters are used to convolve images and extract features from them. For example, different kinds of filters are used in order to extract and highlight different pieces of information such as Laplacian Filter which enhances edges by highlighting regions of rapid intensity changes.

In essence, a Convolutional Neural Network (CNN) utilizes a range of parameter-trainable filters of varying scales to convolve the input image and intermediate results, thereby extracting multi-scale features from the image. If we denote the output of layer i of the CNN as $z_i \in \mathbb{R}^{m_i \times n_i \times c_i}$, where m_i , n_i , and c_i represent the dimensions and the number of channels, respectively, the relationship between the output z_i^j of a single channel j (referred to as a feature

map) and the output of the preceding layer can be expressed as:

$$z_i^j = \sigma \left(\sum_{k=1}^{c_{i-1}} h_{ijk} * z_{i-1}^k + b_i^j \right) \quad (10)$$

Here, $h * z$ denotes a 2D convolution operation of an image z with a filter h , b is a bias term, and σ represents a nonlinear activation function, such as the rectified linear unit (ReLU) [34].

With the advancements in CNN architectures, the typical components now include convolutional layers (CONV), nonlinear activation functions (ReLU), batch normalization (BN) [35], and pooling layers (POOL). The output of each layer serves as the input to the subsequent layer, where it is convolved with new filters. This iterative process can result in a highly deep network structure, known as a deep convolutional neural network (DCNN). The hyperparameters, such as the filters h and biases b , are optimized through supervised learning and back-propagation algorithms [36]. Utilizing optimization techniques like stochastic gradient descent (SGD), the network can progressively minimize the error between its outputs and the labeled data, effectively learning the mapping from the training set to the target set.

Here, we treat EEG prediction as a semantic segmentation task from which each piece of the spectrogram is encoded and then the model tries to illustrate the encoding with a footprint of the EEG response.

Utilizing this method allows for the generation of segmented images of any size and offers a significant speed advantage over previous techniques. Consequently, almost all advanced semantic segmentation methods have adopted the CNN model, leading to the widespread popularity of this state-of-the-art technology. In recent years, numerous CNN models have been proposed for semantic segmentation tasks, including U-Net [37], SegNet [38], and DeconvNet [39]. These models, which demonstrate strong performance in semantic segmentation, are primarily based on encoder–decoder CNN architectures.

In this section, we describe the architecture of the ED network used for training and testing on y_{train} data. The network consists of both an encoder and a decoder, designed to learn an efficient representation of the input data and reconstruct the output with minimal error. Notably, the encoder part of the network utilizes a pre-trained autoencoder model, leveraging its pre-learned features.

- **Input Layer:** The input to the network is defined with a shape of (250, 1, 32), indicating 250 samples, 1 channel, and a depth of 32.
- **Encoder Layers:**
 - **Layer 1 (Conv2D):** This layer applies 32 convolution filters of size 3x3 with strides (2,1), using 'same' padding and ReLU activation.
 - **Layer 2 (Conv2D):** This layer applies 64 convolution filters of size 3x3 with strides (2,1), using 'same' padding and ReLU activation.
 - **Layer 3 (Conv2D):** This layer applies 128 convolution filters of size 3x3 with strides (2,1), using 'same' padding and ReLU activation.
- **Freezing Encoder Layers:** The weights of the encoder layers (Layer 1, Layer 2, and Layer 3) are optionally frozen to retain the pre-learned features, which can help in transfer learning scenarios.
- **Decoder Layers:**
 - **Layer 4 (Conv2DTranspose):** This layer applies 128 convolution transpose filters of size 2x1 with strides (2,1) and ReLU activation to increase the spatial dimensions of the feature maps.
 - **Layer 5 (Conv2DTranspose):** This layer applies 64 convolution transpose filters of size 2x1 with strides (2,1) and ReLU activation.
 - **Cropping Layer (Cropping2D):** This layer crops the feature maps, removing 3 rows from the bottom.
 - **Layer 6 (Conv2DTranspose):** This layer applies 32 convolution transpose filters of size 2x1 with strides (2,1) and ReLU activation.
- **Final Layer (Conv2D):** This layer applies a single convolution filter of size 3x3 with 'same' padding and Leaky ReLU activation to generate the final output.

The encoder part of this network is pre-trained using another autoencoder model, which helps in leveraging pre-learned features and improving the performance of the network. The weights for the encoder are loaded from the file `encoder_weights.h5`.

The ED model is compiled using the Adam optimizer with a learning rate of 0.001. The loss function used is mean squared error, which measures the average squared difference between the predicted and actual values. This setup ensures that the network is trained to minimize the reconstruction error effectively. Overall, this ED network is designed to efficiently encode and decode the input data, leveraging pre-trained weights in the encoder for better performance. The detailed layer architecture and functions ensure that the network can handle complex data representations and provide accurate reconstructions. The whole methodology is summarized in the figure [/refAutoencoder](#).

3.3.2 Pretraining the Autoencoder

The opted to use an autoencoder (AE) in order to encode the input into a compressed form and then decode it back such as the reconstructed input is similar to the original. AEs have been introduced [32] as an unsupervised technique that could be trained in order to reproduce the input in an attempt to compress the information of the input into a significantly smaller latent space.

Consequently, the latent captures significant features of the original signal, facilitating unsupervised feature extraction. The core structure of an AE is divided into these three parts:

1. *Encoder*: Feed forward neural network which compresses the input into lower dimensional latent space.
2. *Latent Space*: It is the intermediate level between encoder-decoder and contains the compressed image which will pass afterwards in the decoder part
3. *Decoder*: Decodes the encoded image from latent space and achieves an input reconstruction.

We used this architecture of unsupervised learning in order to train and test our stimulus spectrogram data. We used this model to first train it with input the segments of the song's spectrogram and generate the same compressed segments in order of our model to learn how to produce perfectly the input segments.

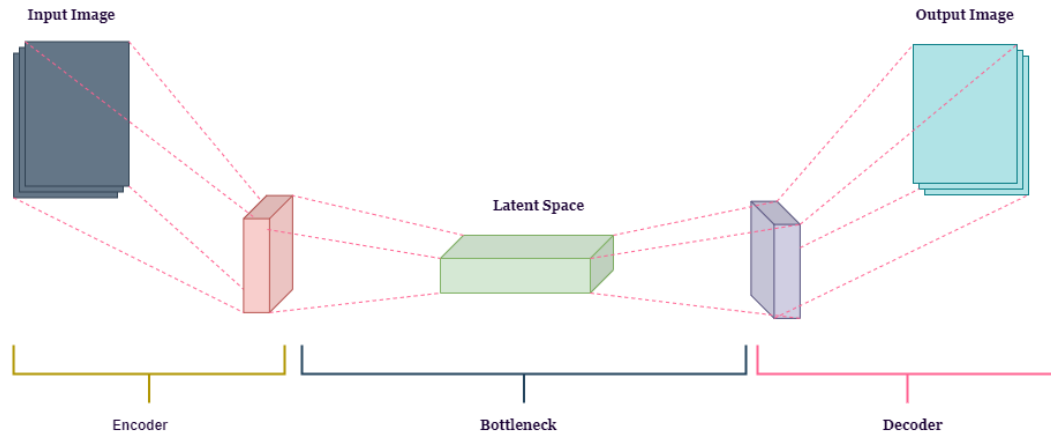


Figure 5: Autoencoder Architecture

A general description of the structure is given in Fig. 5. This section describes the architecture of the autoencoder model used for training and testing on y_{train} data. The model aims to reconstruct the input data while learning compressed representations in the latent space. The structure of the AE we used to train and test our data is as follows.

- **Input Layer:**

- The model takes an input image with a shape of $(250, 1, 32)$. This implies 250 samples per segment, 1 segment, and 32 frequency bins, representing the segments of the dynamic spectrum of the sound data (stimulus).

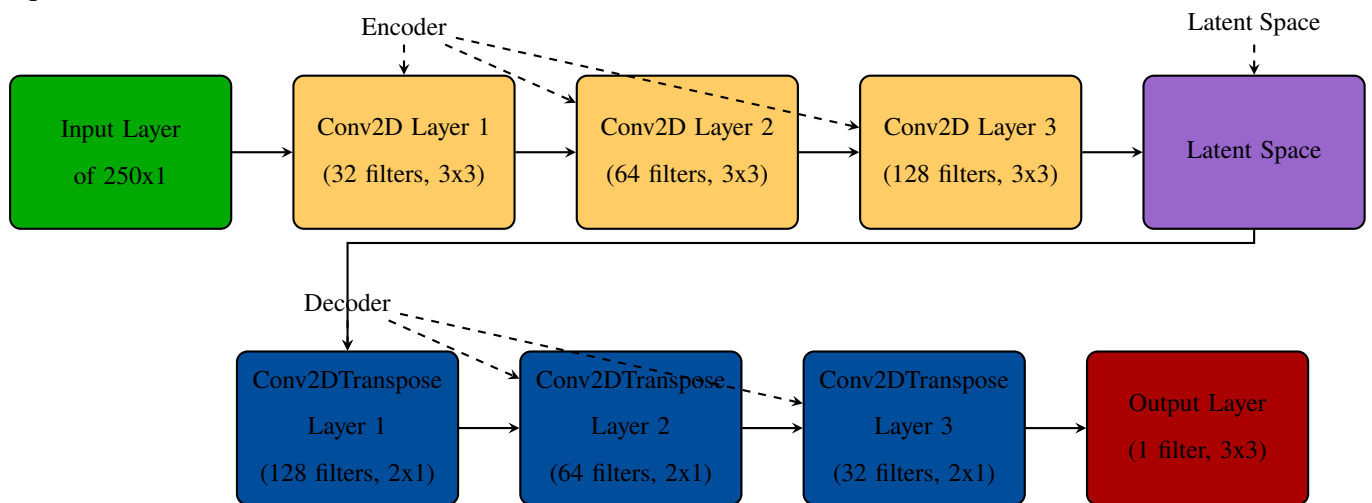
- **Encoder Layers:**

- *Layer 1:* 32 filters of size 3×3 , stride $(2, 1)$, and same padding. This layer downsamples the temporal dimension by half while preserving the spatial dimension.
- *Layer 2:* 64 filters of size 3×3 , stride $(2, 1)$, and same padding. This layer further downsamples the temporal dimension and increases the number of feature maps.
- *Layer 3:* 128 filters of size 3×3 , stride $(2, 1)$, and same padding. This layer performs the final downsampling in the temporal dimension, resulting in the latent representation.

- **Decoder Layers:**

- *Layer 4*: 128 filters of size (2, 1), stride (2, 1), and upsamples the temporal dimension while increasing the number of feature maps.
- *Layer 5*: 64 filters of size (2, 1), stride (2, 1). This layer further upsamples the temporal dimension.
- **Cropping Layer**: This layer removes the last three elements from the temporal dimension due to the difference in input and output shapes after upsampling.
- *Layer 6*: 32 filters of size (2, 1), stride (2, 1). The final upsampling layer restores the original temporal dimension.

The model is compiled with Adam optimizer with a learning rate of 0.001 and the mean squared error (MSE) as loss function.



4 Results and Analysis

In this section we are going to apply the methodology steps and techniques that we described before in an open dataset provided in [40].

4.1 Dataset Acquisition

With this study, we got to investigate how is it possible to predict one's response to a song stimulus that is played to different patients in a long-term study that is also trying to achieve sound reconstruction of this complex stimulus. The patients passively listened to the song *Another Brick in the Wall, Part 1*, by Pink Floyd (released on the album *The Wall*, Harvest Records/Columbia Records, 1979) and they were instructed to listen attentively to the music, without focusing on any special detail. The duration of the song was 190.72 seconds and in the data collection its given in a form of a spectrogram of 32 and 128 frequency bins non-linearly distributed. The corresponding frequency range is (188-6745) Hz that is partitioned into frequency bins. The partitioned is non-linear and it is depicted in Fig. 6.

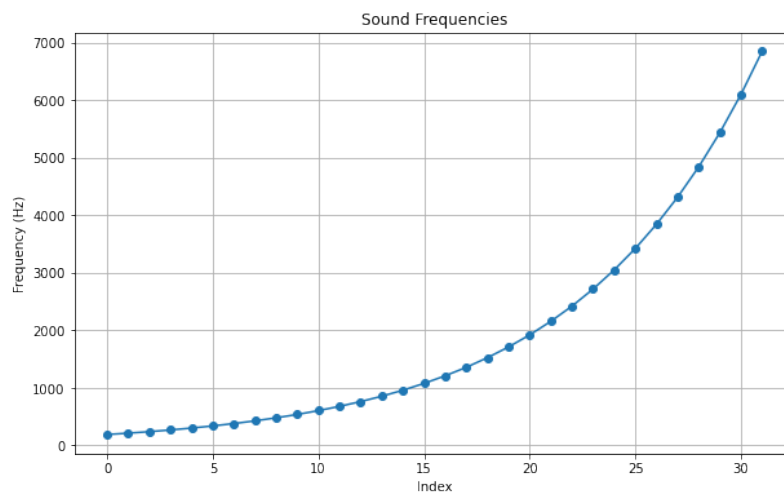


Figure 6: The range of the 32 frequencies

The dynamic spectrogram (the 32 aforementioned frequency bins) of the whole stimulus (19,072 samples) is shown in Fig. 7.

The dataset includes apart from the spectrograms of the raw stimuli, the post processed



Figure 7: Spectrogram with 32 Frequency bins of the song stimulus

ECoG responses of 29 patients with pharmaco-resistant epilepsy and the reconstructed signal that was achieved with linear and non-linear methods[20].

4.2 Electrode Preprocessing

The first step of *preprocessing* is the removal of the epileptic electrodes and the ones containing an artifact from the raw ECoG representations. Those were included in the dataset information. The epileptic electrodes were marked with their indices corresponding to the same indices in the ECoG matrix, while the artifact electrodes were in a separate .mat file, indicated by 1. As a result, from 2.668 electrodes, we first excluded patient 8 from our study, because all its electrodes were marked as artifacts, getting 2.592 electrodes. Then, we detected and removed for each patient its epileptic electrodes, ending up with 2.048 artifact-free electrodes.

We decided to focus our study to the bilateral Superior Temporal Gyrus as it is crucial music processing in the brain cortex level, [41],[42]. We must also note that it has been proven

that tonotopic maps located in the STG increased in size or volume in trained musicians, a characteristic that helps them significantly in perceiving better music components. Tonotopic maps refer to topographical organisation of neurons as a function of their response to different frequencies and are present in the auditory system and cortex, responsible for processing these responses.

Considering the importance of STG, we kept 411 electrodes with the anatomic label characterising them as “superior temporal” either in the left or the right hemisphere, across all patients. We were able to detect these electrodes over the patients scalp and to differ their positions as we were given their anatomic label provided in the dataset. The positions of the electrodes for all patients were provided with respect to the MNI template. This template, created by the Montreal Neurological Institute and Hospital is similar to the Talairach coordinates. It is a 3D model of the human brain, average of 305 T1-weighted MRI scans from healthy individuals[43]. This template is publicly available and can be downloaded in [44].

In figure 8 we are plotting all electrodes of patient 6, using different colors for each anatomical area, as shown in the legend. The triangulated model was generated using an isosurface at an appropriate value to segment the MRI soft tissue.

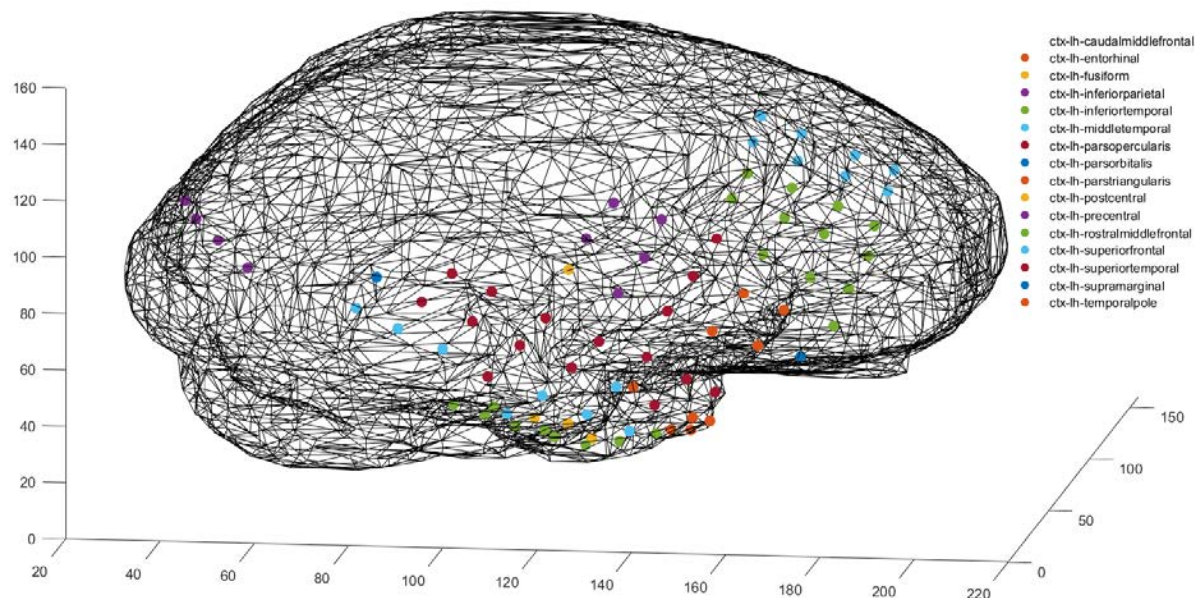


Figure 8: All electrodes from patient 6

Figure 9 also plots the STG (Superior Temporal Gyrus) electrodes of patients 6 and 7 with different color, which will be used in this work.

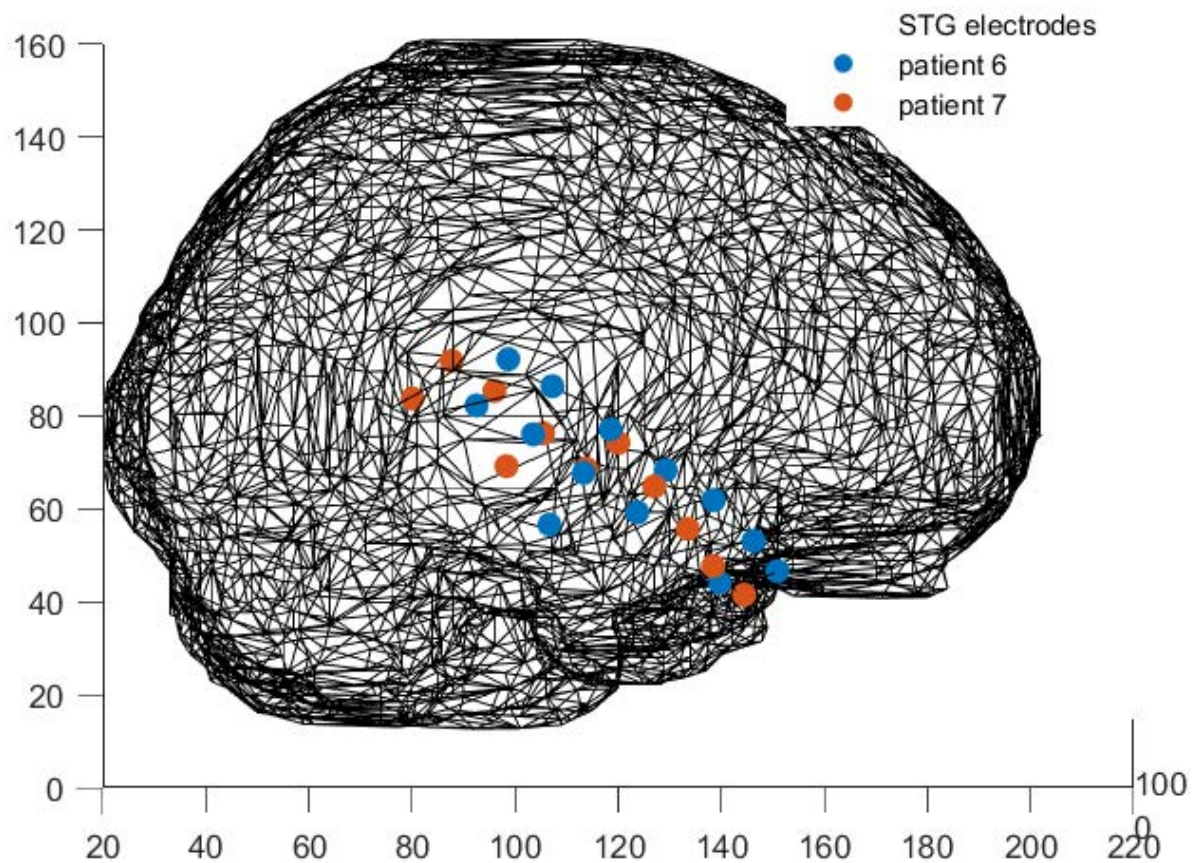


Figure 9: All STG electrodes from patients 6,7

4.3 Utilizing the mTRF

The classic method of multivariate Temporal Response Function determines the mTRF from given stimulus and response from several electrodes. Having calculated mTRF for each electrode, it is possible to predict the EEG response given the patient's stimulus.

This functionality has been implemented in a Matlab toolbox [21]. One of its major functions allows computing multivariate and univariate linear regression based on the equation 8. So, for a given trial, we use `mTRFtrain` function which is initialized as follows:

$model = mTRFtrain(stim, resp, fs, Dir, tmin, tmax, lambda);$

The λ parameter in Eq.(7) is the ridge parameter and should be declared empirically beforehand using a cross-validation method. The stimulus and response data should have the same sampling frequency fs (measured in Hz). The window of the time lags ($tmin, tmax$) must be inserted in milliseconds (ms). Sometimes, more time lags need to be inserted in order to visualize the responses better considering the possible artifacts.

We used for input stimulus the dynamic sound spectrogram and for response the response of one electrode from a patient.

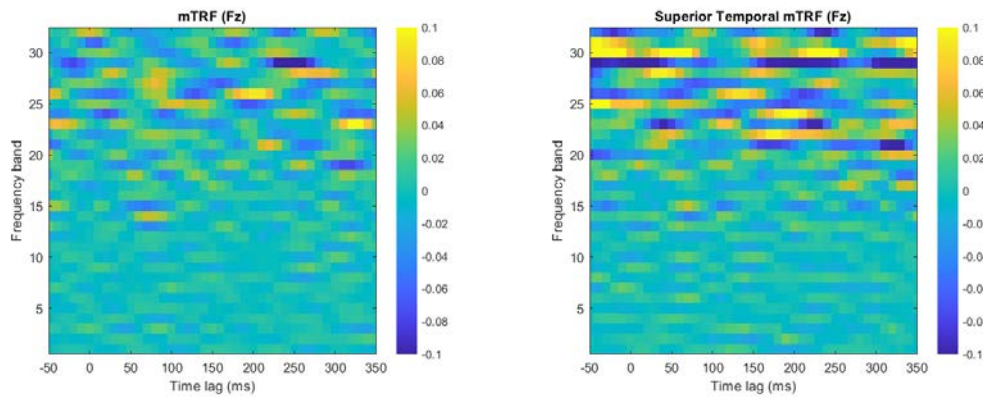


Figure 10: mTRF of electrode 2 and electrode 15 patient 1

In figure 10 we plot the mTRF for 2 electrodes of patient 1, electrode 2 and electrode 15. We have to mention that mTRF of electrode 15 on the Superior Temporal Gyrus (STG), shows a increased activation of higher frequencies than the mTRF of electrode 2, which lies at the supramarginal region, a brain area mostly associated with learning and memory.

Therefore, we selected the 18 electrodes located in the STG region from patient 1 and we calculated the Euclidean distance map of the locations of these specific electrodes. Figure 11 depicts the calculated distance map.

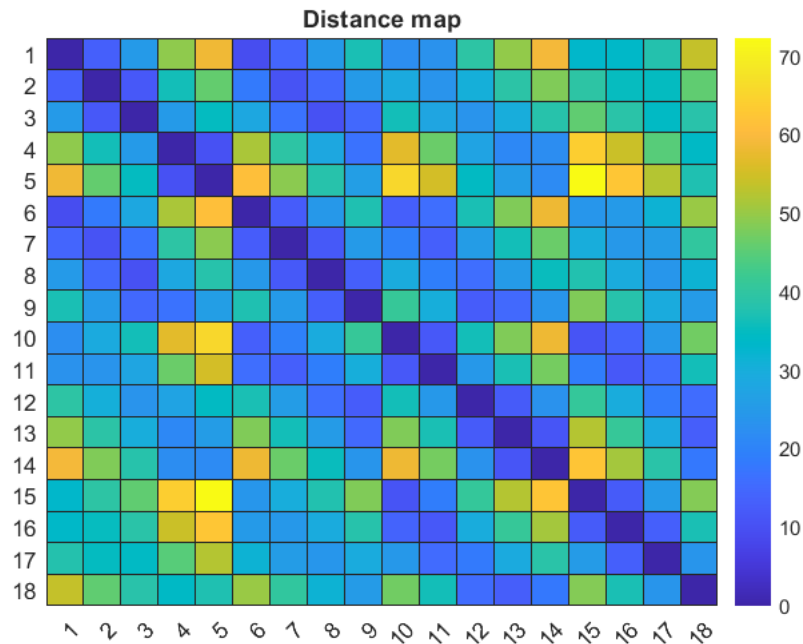


Figure 11: Distances between electrodes for the 18 STG electrodes of patient 1.

For each of these specific electrodes, we used mTRFtrain in order to obtain the mTRFs and then we created the correlation map between them, in order to assess if the distance affects the mTRF prediction. Fig. 12 visualizes the result. We observed that most values indicating correlation between mTRFs were below 0.5 which is not an indicator of high correlation.

The relationship between the mTRF correlation and electrode distance is investigated in Fig. 13, where we plot the mTRF correlations versus electrode distances for the 18 electrodes of the STG for patient 1. As it can be observed, no clear trend is visible.

Afterwards, we produced different mTRFs for all these 18 electrodes of the first patient and we depict them in figure 14 for electrodes in the area of interest another areas as well.

The objective now is to use the produced mTRF and in combination with the input signal to predict the response of the corresponding electrode. This is done with a function (mTRFpredict) of the MATLAB toolbox [21] that allows prediction of the output of a forward encoding model

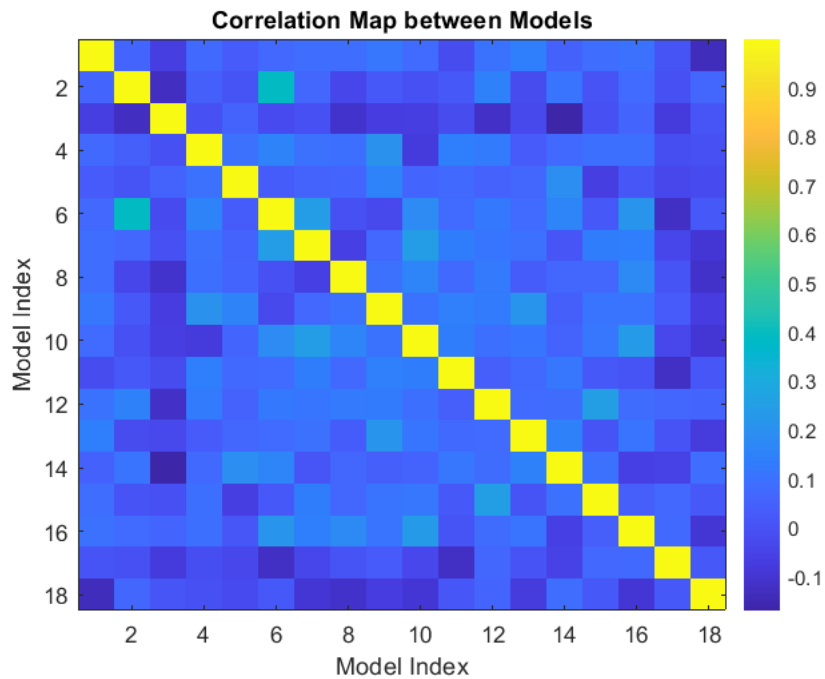


Figure 12: Correlation between mTRFs for the 18 STG electrodes for patient 1.

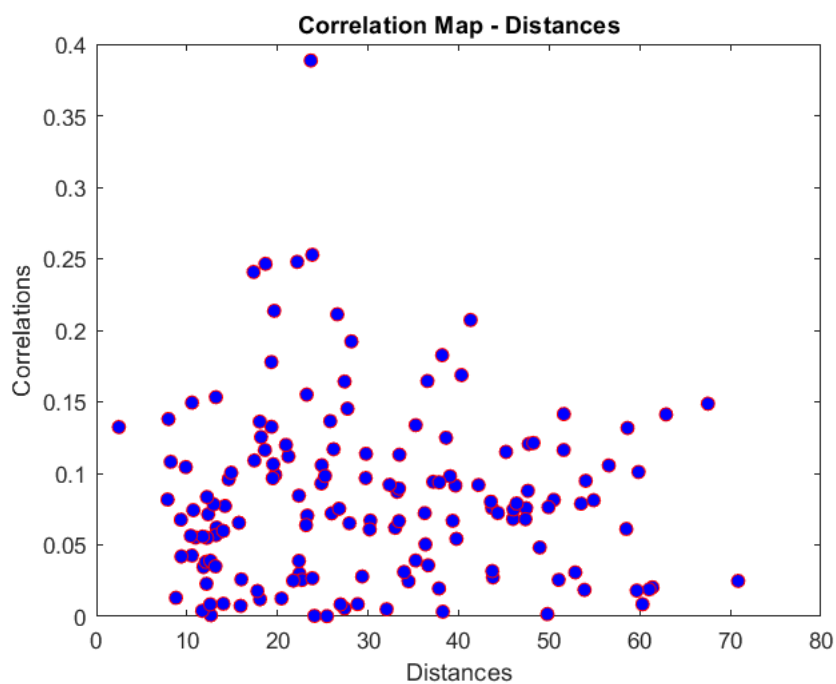


Figure 13: Distances-Correlations for the 18 STG electrodes

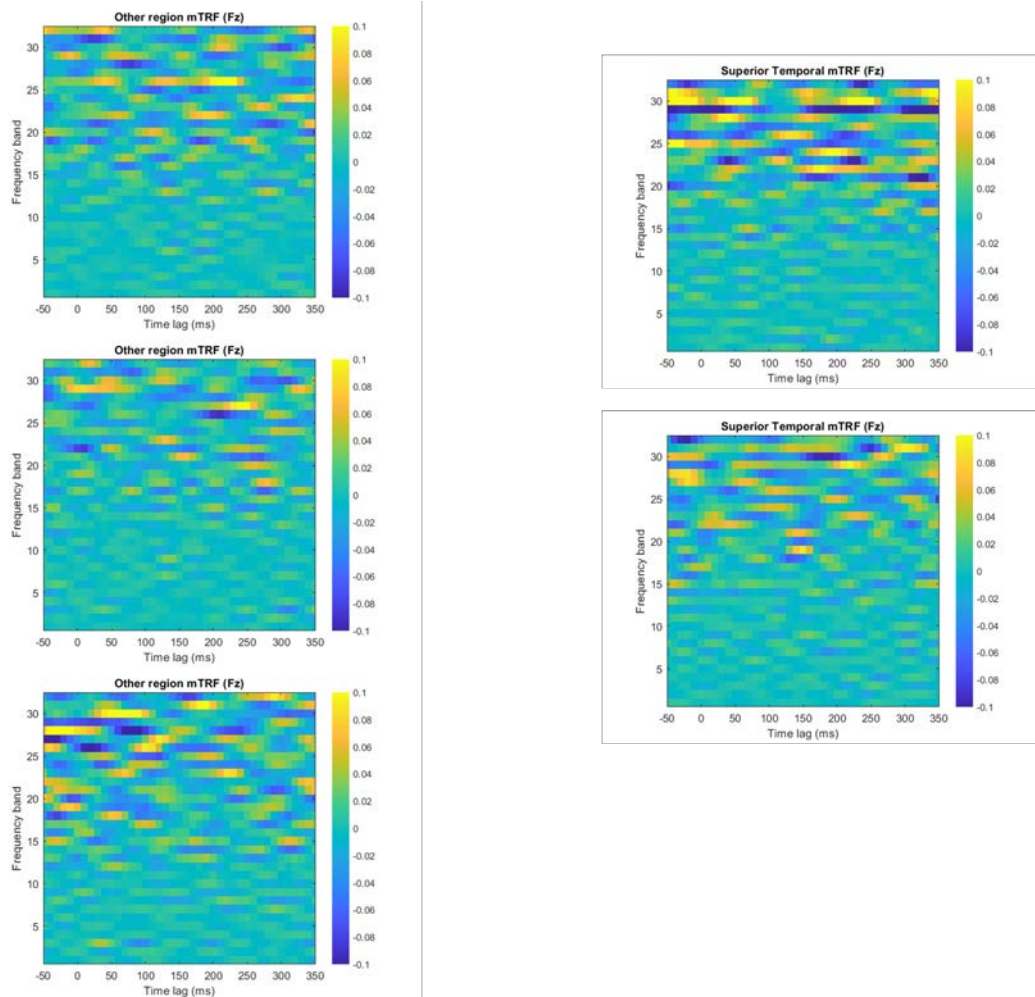


Figure 14: mTRFs for 5 electrodes of patient 1.

(stimulus to neural response) using time-lagged input features. We used this function to predict the responses for the first 5 electrodes. The predicted and true response of each electrode are shown in Fig. 15 with red and blue color respectively.

Prediction models are typically evaluated using error metrics such as correlation between the actual and predicted values. This is done with function (mTRFevaluate) In mTRF-Toolbox. The function returns two evaluation metrics: r is the Pearson correlation coefficient, and err is the mean squared error (MSE). Alternatively, we can specify a Spearman correlation using the 'corr' parameter and the mean absolute error (MAE) using the 'error' parameter.

We will provide here a matrix with the output MSE error for the first 5 electrodes of our dataset on Table 1 .

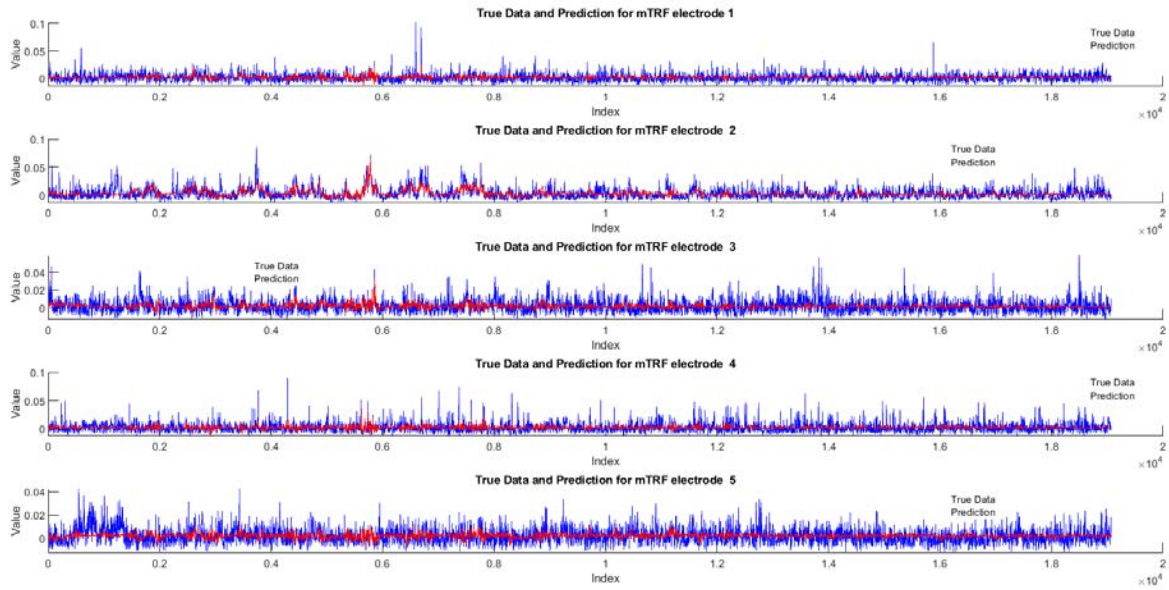


Figure 15: Actual output (blue curves) and predictions (red curves) for 5 STG electrodes, using the corresponding mTRFs in Fig. 14

Electrode	Average MSE
1	4.222e-05
2	6.848e-05
3	5.149e-05
4	5.260e-05
5	5.015e-05

Table 1: MSE errors from mTRF method for 5 electrodes

4.4 Deep Learning Predictive Model

Using the architecture that we previously described in 3.3.1 and the post-processed data we will present here the results and the aforementioned methodology.

Data Segmentation With the purpose of achieving a proper analysis for our dataset we segmented both the spectrogram and ECoG data to apply our prediction methods. The original shape of our spectrogram as we previously mentioned is 19072 samples x 32 frequency bins so, we had to split our samples into meaningful segments. We should consider that, meter is the temporal framework according to which rhythm is perceived as defined in [45]. At the most basic level, the perception of meter involves a sense of pulse which is a pattern of beats at equally

spaced intervals [46], ([47]). It is well known that the 4 beats per measurement or 4/4 time signature meter (four beats per measure) is the most common meter across genres and we will use this meter for music perception as it is suggested in [20]. In [20], 8 bars of 4/4 measures are included in the duration of 20 seconds, so we consider each 4/4 measure as 2.5 seconds which corresponds to the segment duration that we selected for our segments in order to be well perceived.

Firstly, we did split our dataset for the whole procedure according to the rule of the 2.5 seconds (equivalent to 250 samples because of 100Hz sampling rate) that we must follow as we noted when we were segmented our data. Following that, we applied also an overlap useful to enhance feature extraction, to smooth the transitions between segments. By sliding overlapping windows over the input data, the autoencoder can capture finer details and dependencies within the data, enhancing its ability to reconstruct the input accurately. This approach is particularly useful in scenarios where the input data has spatial or temporal dependencies, such as in image or time series data, allowing the autoencoder to learn more robust representations. The overlap window technique helps the autoencoder to achieve better reconstruction performance and can lead to more effective feature extraction and compression. Without overlap, there might be abrupt changes at the boundaries of windows, which can introduce artifacts or discontinuities in the analysis. Hence, we tried different duration for our overlap window, those of 50 seconds, 100 seconds and 150 seconds that we are depicting below.

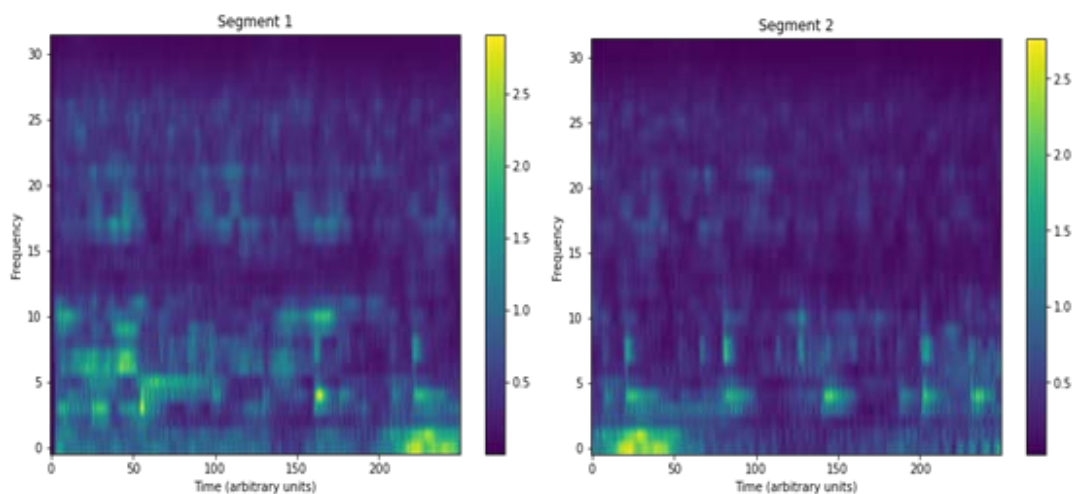


Figure 16: Spectrogram with 50 samples overlap window

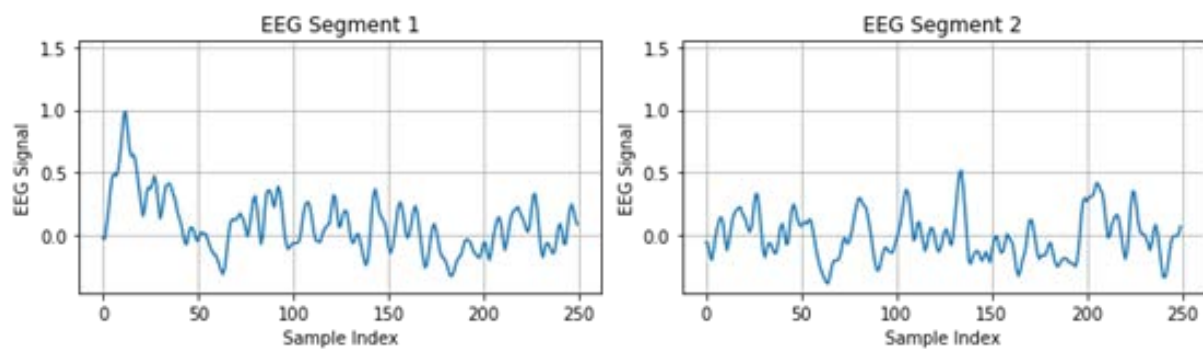


Figure 17: EEG with 50 samples overlap window

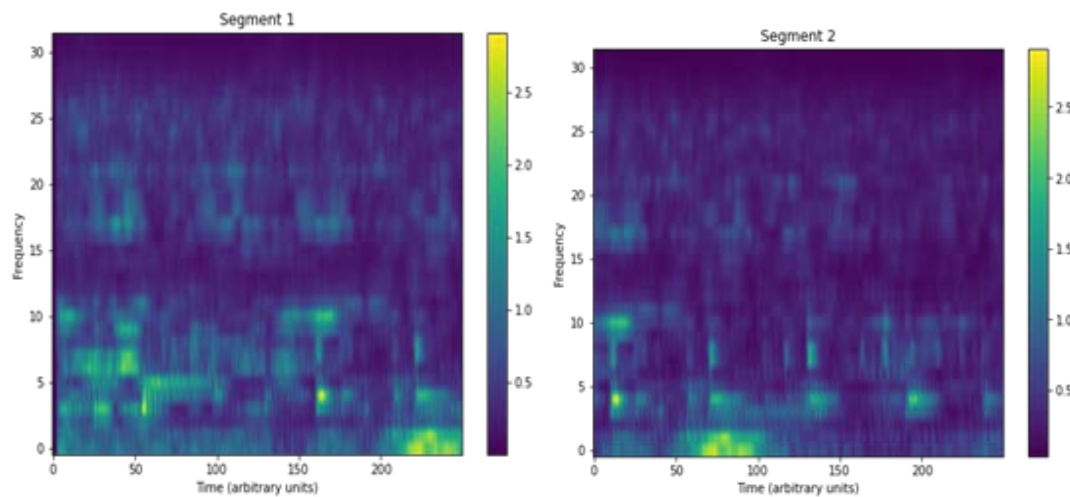


Figure 18: Spectrograms of two consecutive stimulus segments with 100 samples overlap

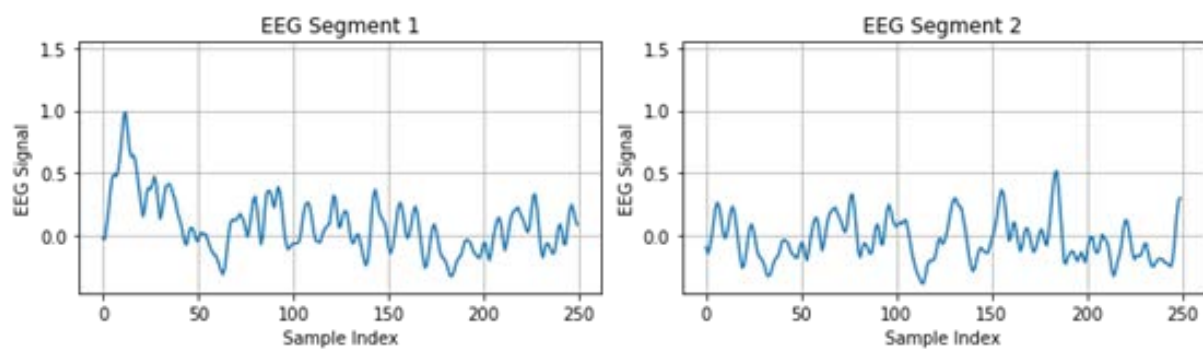


Figure 19: EEG with 100 samples overlap.

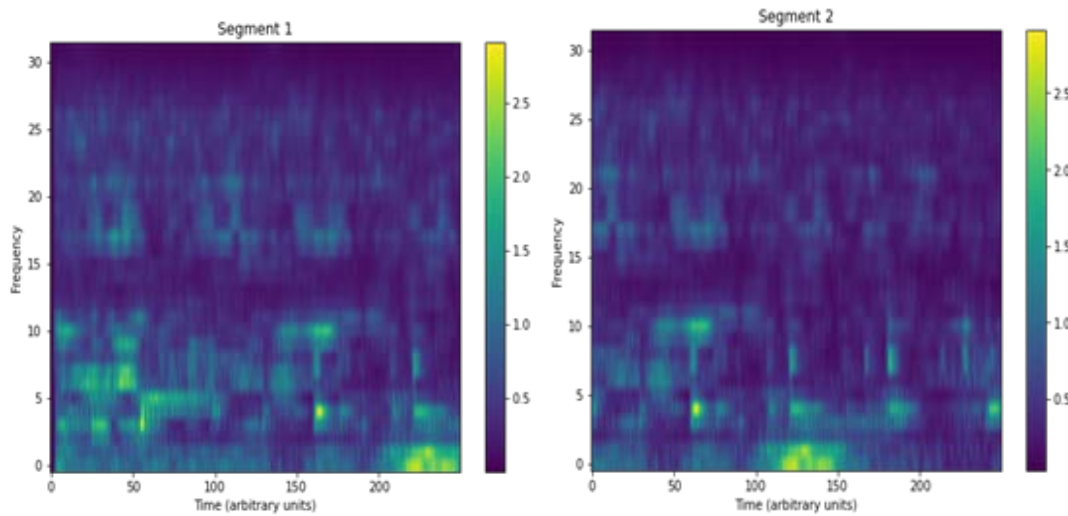


Figure 20: Spectrogram with 150 samples overlap.

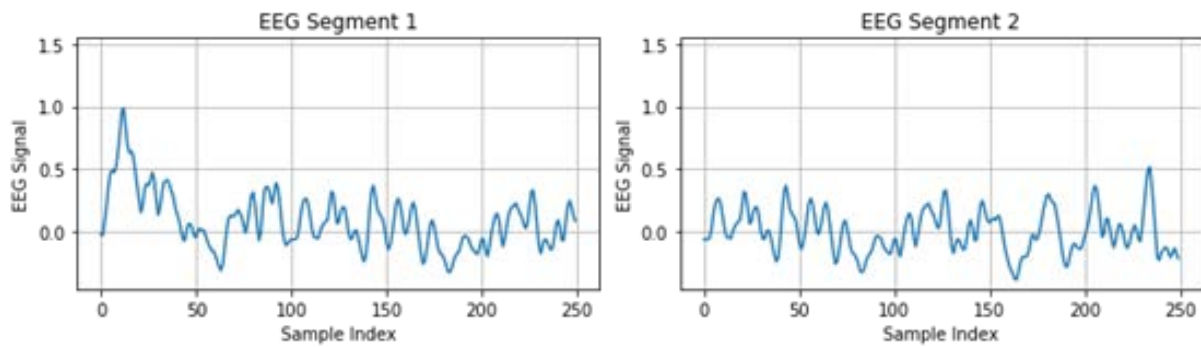


Figure 21: EEG with 150 seconds overlap window

We selected an overlap of 150 samples and in this way we increased the size of the available dataset, while preserving essential information. We applied the `train_test_split` method to partition the dataset into 80% training and 20% testing subsets. Input data consists of spectrogram segments (X) and corresponding EEG responses(y). Moreover, we opted not to shuffle the data (`shuffle=False`), maintaining the original order of instances, which might be essential for preserving temporal or sequential patterns in the dataset. This splitting strategy enables robust evaluation of model performance and generalization to unseen data. The result of the split method:

X_{TRAIN} : 151 segments $\times$ 250 samples $\times$ 32 frequencies

X_{TEST} : 38 segments $\times$ 250 samples $\times$ 32 frequencies

y_{TRAIN} : 151 segments $\times$ 250 samples

y_{TEST} : 38 segments $\times$ 250 samples

4.4.1 Autoencoder part

Starting with the autoencoder model that we used, we first fed to our network the spectrogram segments both in the input and the output. We used this model:

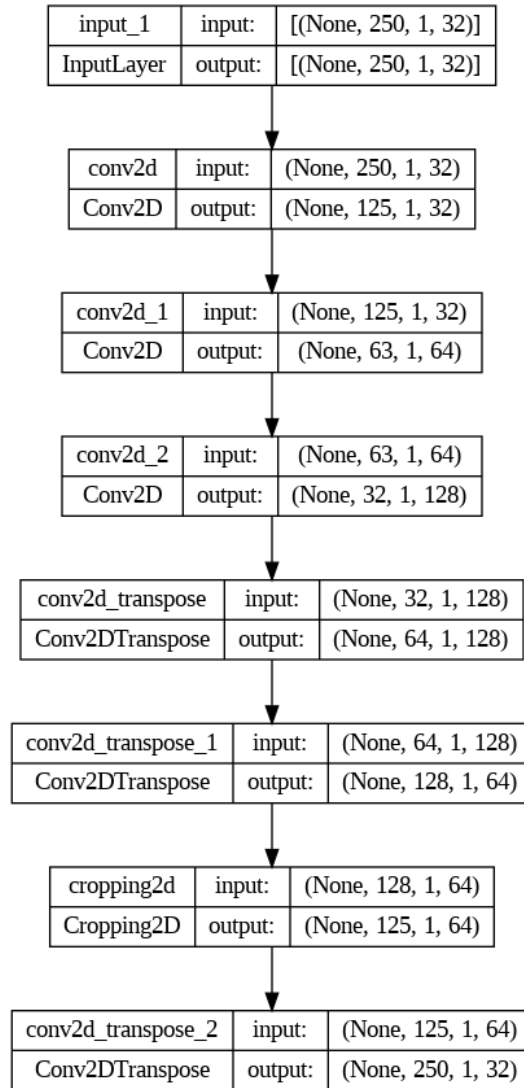


Figure 22: Autoencoder Applied

Then, we trained our model, using the training data for a specified number of epochs (50 in this case). The autoencoder is trained to reconstruct its input, hence spectrogram train data

is passed as both the input and the target output. This configuration aims to minimize the reconstruction error, thereby learning a compressed representation of the input data.

During the training process, the fit method updates the model's parameters based on the optimization algorithm (e.g., stochastic gradient descent) and the loss function in our case, mean squared error specified in the model configuration. The objective is to minimize the discrepancy between the reconstructed output and the original input, thereby learning a representation of the data that captures important features while minimizing reconstruction error. Specifically, in the context of autoencoder models, the fit method is utilized to iteratively update the model's parameters (weights and biases) based on the provided training data.

Additionally, the EarlyStopping callback is employed to monitor the validation loss during training. The training process will stop early if the validation loss does not improve for a certain number of epochs, thus preventing overfitting and potentially saving computational resources. We achieve this by trying to check the best value for patience parameter which indicates the number of epochs with no improvement for the loss after which training will be stopped. We must note that the parameters of epochs and the patience parameter can be changed and are going to represent our hyperparameter values for the following experiments along with MSE error. The results from using fit function are the following:

$$\begin{aligned} & \text{autoencoder22.fit}(X_train, X_train, \text{epochs} = 50, \\ & \text{callbacks} = [\text{EarlyStopping}(\text{monitor} = 'val_loss', \text{patience} = 5)]) \end{aligned} \quad (11)$$

Using the spectrogram segments X_train as we referred, we get the results are illustrated in the figure 13:

After, we tested the model with X_test data and we got:

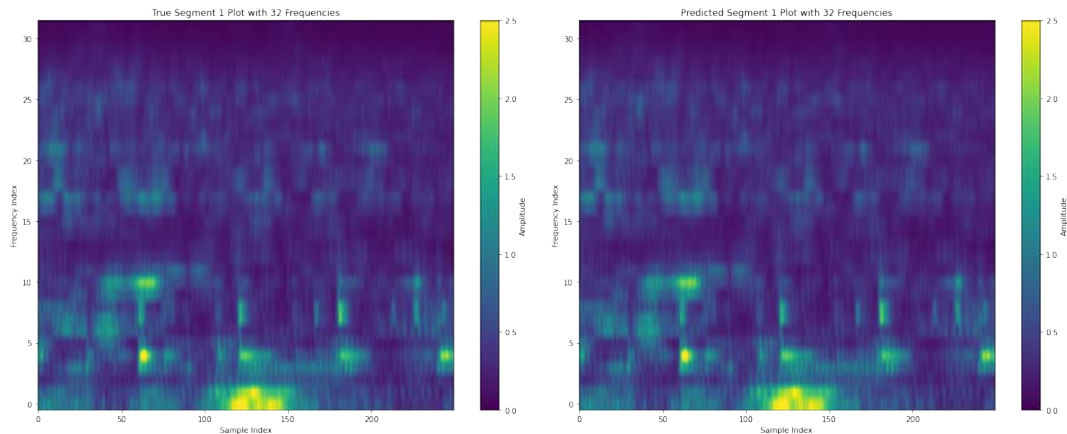


Figure 23: Ground truth and predicted train data

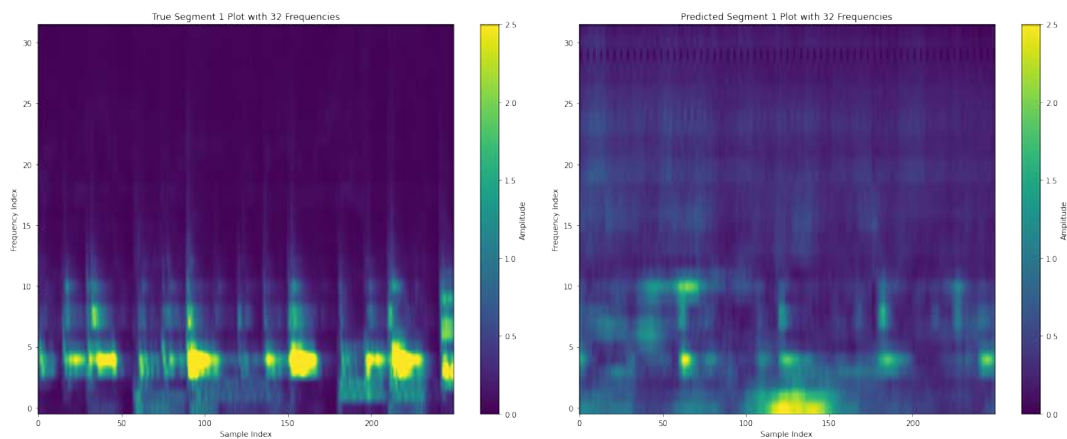


Figure 24: True and predicted test data

Hyperparameter Tuning We will focus here to a strategy for hyperparameter tuning focused on epochs, early stopping, and average Mean Squared Error (MSE) commonly used for machine learning models, particularly those involving neural networks.

Epochs: Epochs represent a complete iteration through the entire training dataset. During each epoch, the model updates its internal parameters to minimize the loss function. Tuning the number of epochs is crucial. Too few epochs may lead to underfitting, where the model fails to capture the underlying relationships in the data. Conversely, too many epochs can lead to overfitting, where the model memorizes the training data and performs poorly on unseen data. *Early Stopping:* Early stopping is a regularization technique that prevents overfitting. It monitors the validation loss (loss on a held-out set) during training. If the validation loss fails

to improve for a predefined number of consecutive epochs (patience), training is stopped. This helps to identify the point where the model generalizes best to unseen data. *Average MSE Error:* MSE is a common metric used to evaluate the performance of regression models. As we have mentioned, it measures the average squared difference between the predicted values and the actual values. Here, we'll use the average MSE error across multiple training runs with different hyperparameter settings. This helps to account for potential randomness in the training process. *Learning Rate:* Learning Rate is one of the most common metrics for neural networks as it is a tuning parameter that determines the step size at each iteration while moving toward a minimum of a loss function. Consequently, a high learning rate value can lead to faster convergence but may also cause the optimization algorithm to overshoot or oscillate around the optimal solution, while a lower learning rate resulted in very slow convergence. So, the decision of this tuning parameter must be taken very carefully. In our case we tried to make predictions with altering the learning rate but our network was very sensitive to the data. Therefore, the value of 10^{-2} was chosen as a balanced value to ensure stable and efficient training. We are also attaching here a table with the hyperparameter tuning regarding this model.

Epochs	Patience	Average Testing MSE error	Average MSE over epochs
20	5	0.1137	
20	10	0.0326	0.0598
20	15	0.0331	
50	5	0.1368	
50	10	0.0310	0.0657
50	15	0.0295	
80	5	0.0569	
80	10	0.0285	0.0372
80	15	0.0262	
100	5	0.0340	
100	10	0.0841	0.0547
100	15	0.0461	

Table 2: Autoencoder Hyperparameter Tuning

Also, regarding the evaluation method that we used which was MSE error we would like to highlight here the MSE error per segment for all my train set:

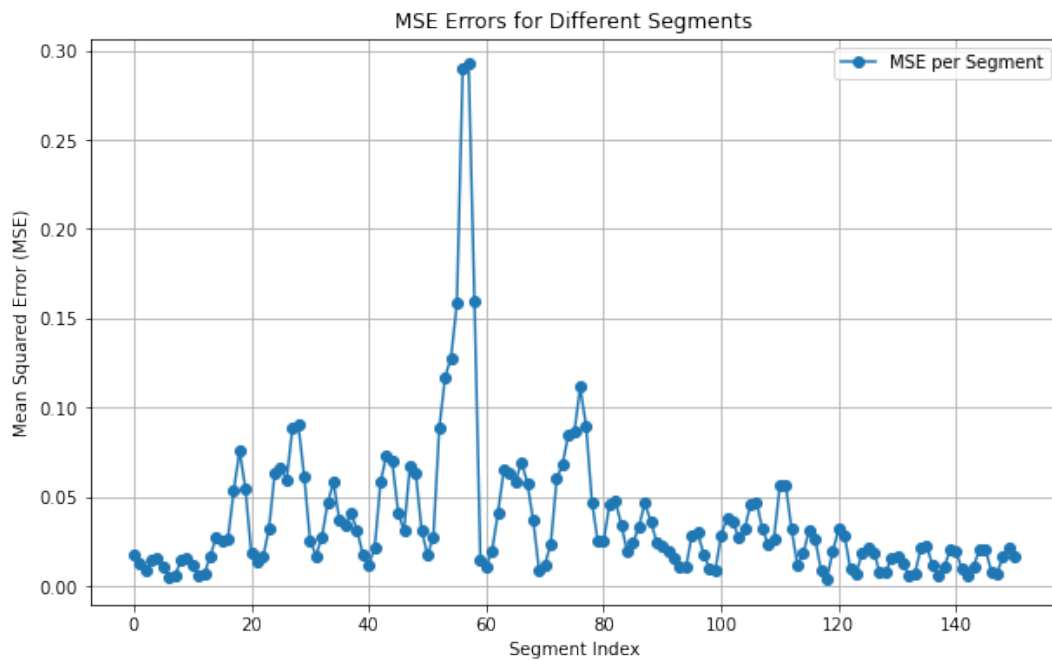


Figure 25: MSE error per segment for 100 epochs and patience = 5

4.4.2 Encoder-Decoder Part

Continuously, we used the weights of the AE (that was used before) in order to train the ED that will now predict the EEG responses which was initially our purpose. The architecture of such a model could be summarized in the following plot:

Now, after training and testing our AE, we are ready to load the weights of the encoder to our Encoder-Decoder System in order to output the EEG predictions. For this purpose, we will use the train and test sets from the first electrode that we are using and next we are going to test the models to new electrode data and evaluate the results.

Using the spectrogram data train and response data train for the output of the ED model, we produced a prediction matrix with dimension 151x250 for the train part of my segments and all the duration of my spectrogram:

After we tested our model with response test data from the same electrode and we got:

The training part with the train data was conducted with the same function that we mentioned in 11 and we performed the same hyperparameter tuning process.

Afterwards , we performed training of our network with different electrodes. For example,

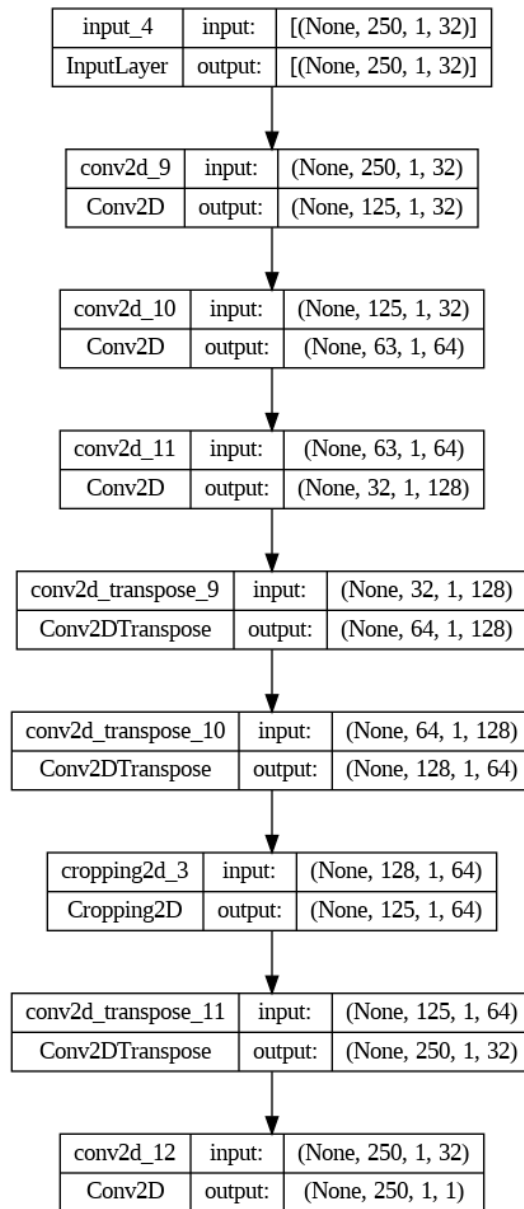


Figure 26: Encoder-Decoder instance

we trained with electrode 18 which is the closer electrode to electrode 1 in order to check how the model is predicting the result and we got:

Also, we trained our network with the most distant electrode from electrode 1 in order to check how distance between electrodes affects prediction. The results that we got for test data were:

We must underline that in this specific electrode the algorithm doesn't seem to predict really

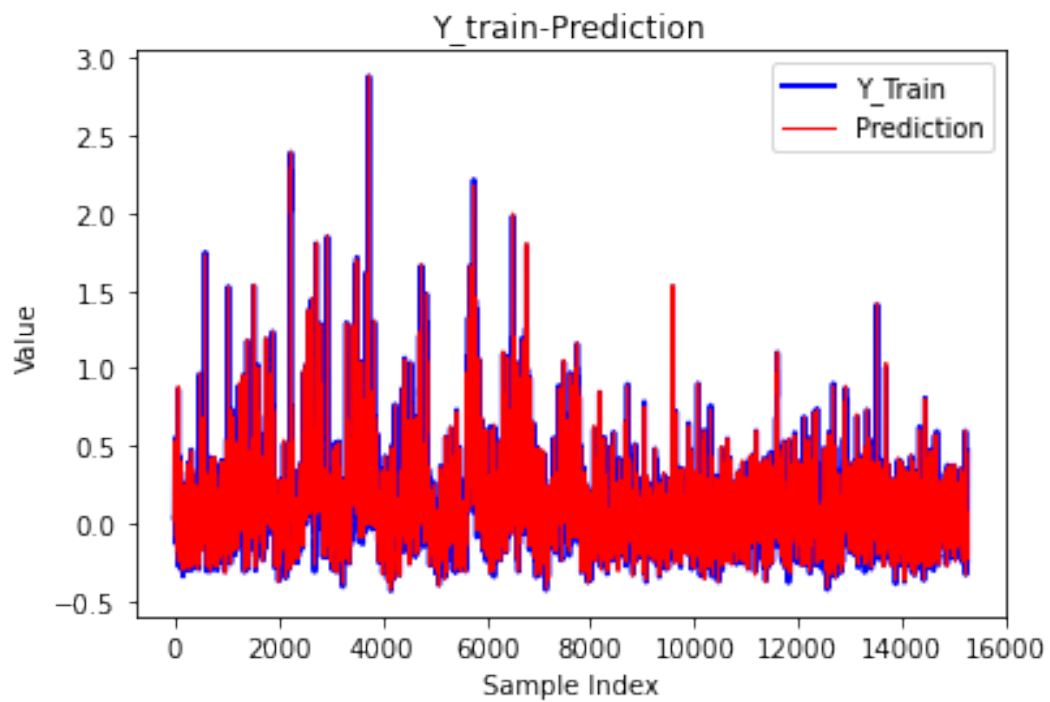


Figure 27: Predictions of EEG response on Patient 1 Electrode 1 on train data

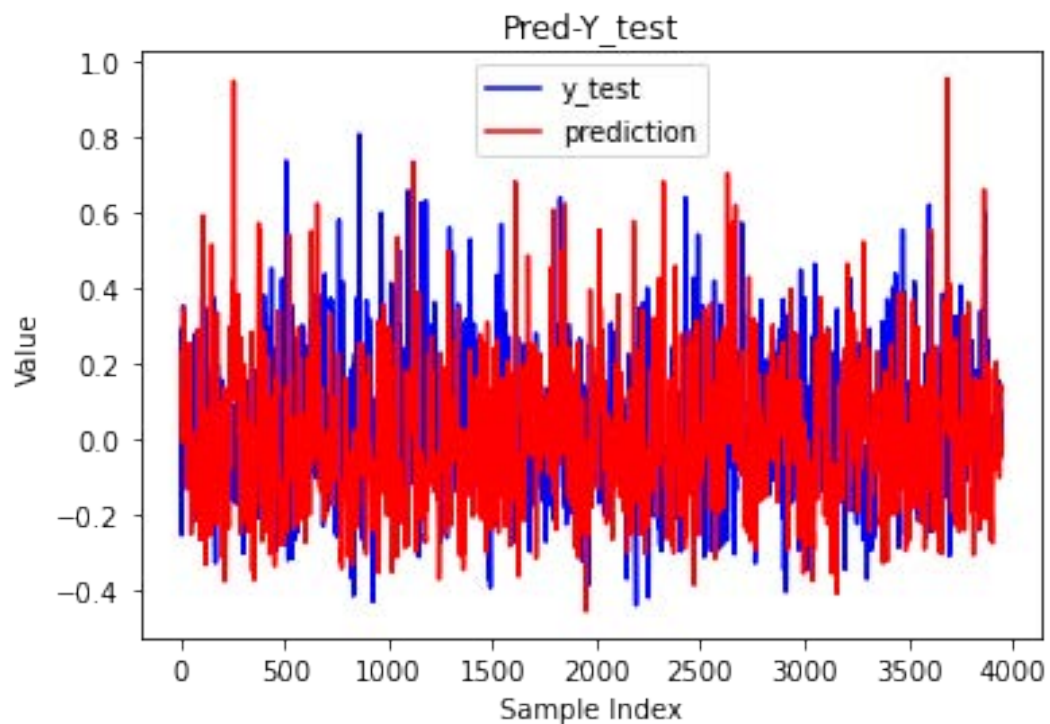


Figure 28: Predictions of EEG response on Patient 1 Electrode 1 on test data

Epochs	Patience	Average Testing MSE error	Average MSE over epochs
20	5	0.0625	
20	10	0.0478	0.0480
20	15	0.0338	
50	5	0.0169	
50	10	0.0108	0.0311
50	15	0.0051	
80	5	0.0022	
80	10	0.0013	0.0083
80	15	0.0016	
100	5	0.0018	
100	10	0.0005	0.0008
100	15	0.0003	

Table 3: Encoder Decoder Hyperparameter Tuning

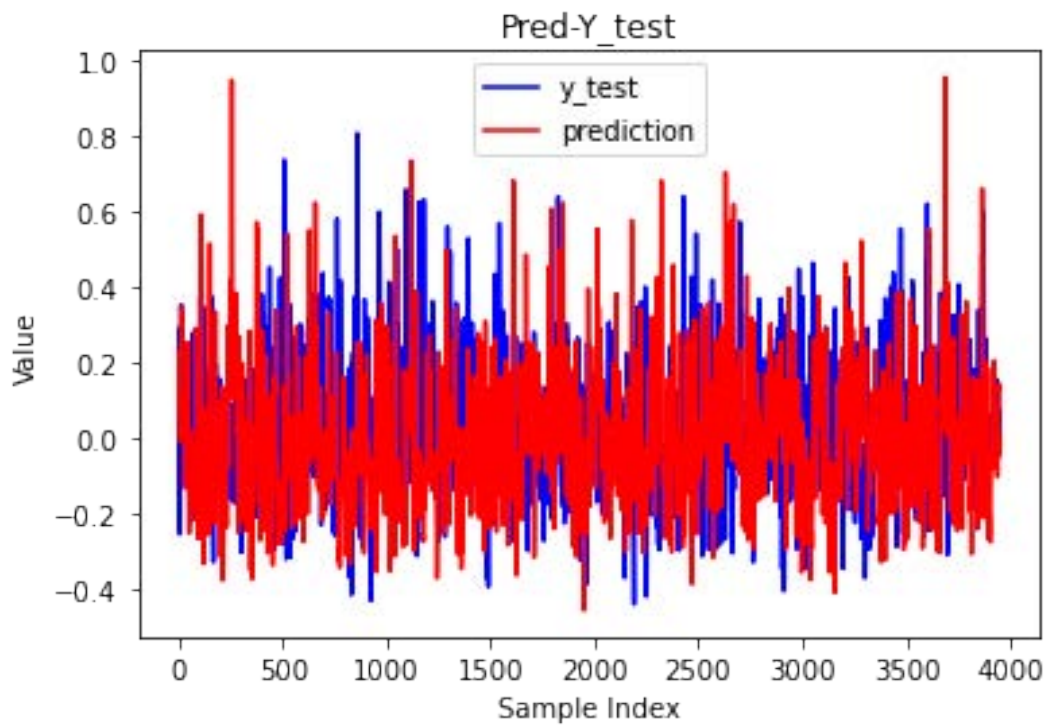


Figure 29: Predictions of EEG response on Patient 1 Electrode 18 on test data

well the output contributing to the fact that it is about the most distant electrode from electrode one so the prediction is not that precise.

Secondly, we also used another method of partitioning our dataset and we tried to split the dataset as 50% of the data to be the training set and the other half to be the test set. So, the

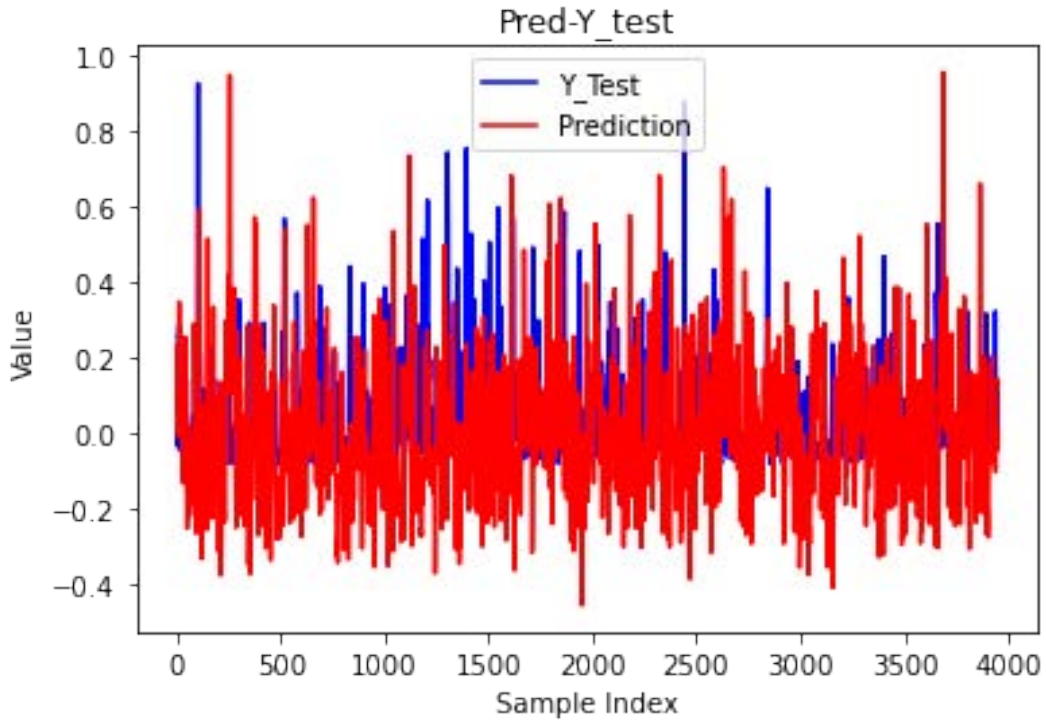


Figure 30: Predictions of EEG response on Patient 1 Electrode 6 on test data

dimensions of our data were:

X_{TRAIN} : 95 segments $\times$ 250 samples $\times$ 32 frequencies

X_{TEST} : 94 segments $\times$ 250 samples $\times$ 32 frequencies

y_{TRAIN} : 95 segments $\times$ 250 samples

y_{TEST} : 94 segments $\times$ 250 samples

We also applied the same methodology as before with the spectrogram data to pretrain the model and we got the results visualized in figure 31. While also we trained and tested with the ED model we figured out that the

4.5 Comparison of the 2 methods

Here, we will provide a comparison section for the 2 techniques that we used in order to better evaluate their performances comparatively. They were trained to produce the same result and the were fed with same segments of input.

At first, we will attach here a plot of the MSE error per segment for each method that we

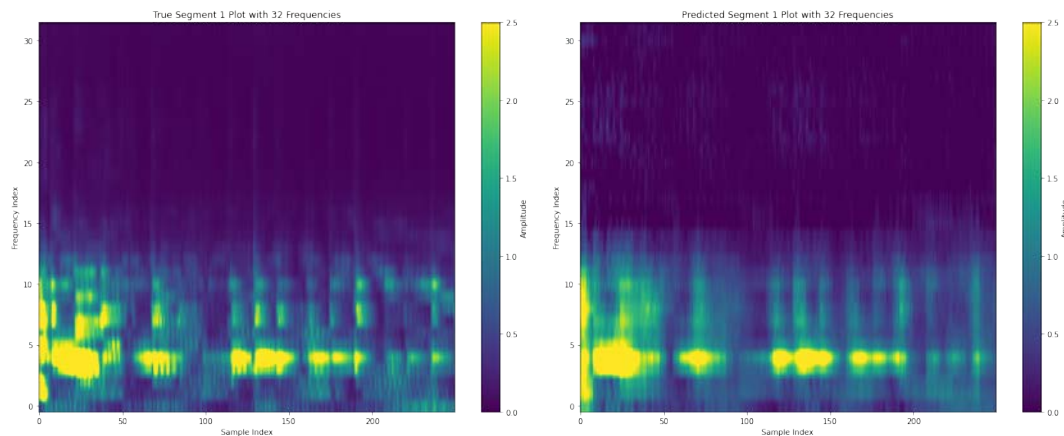


Figure 31: Predictions on spectrogram test data

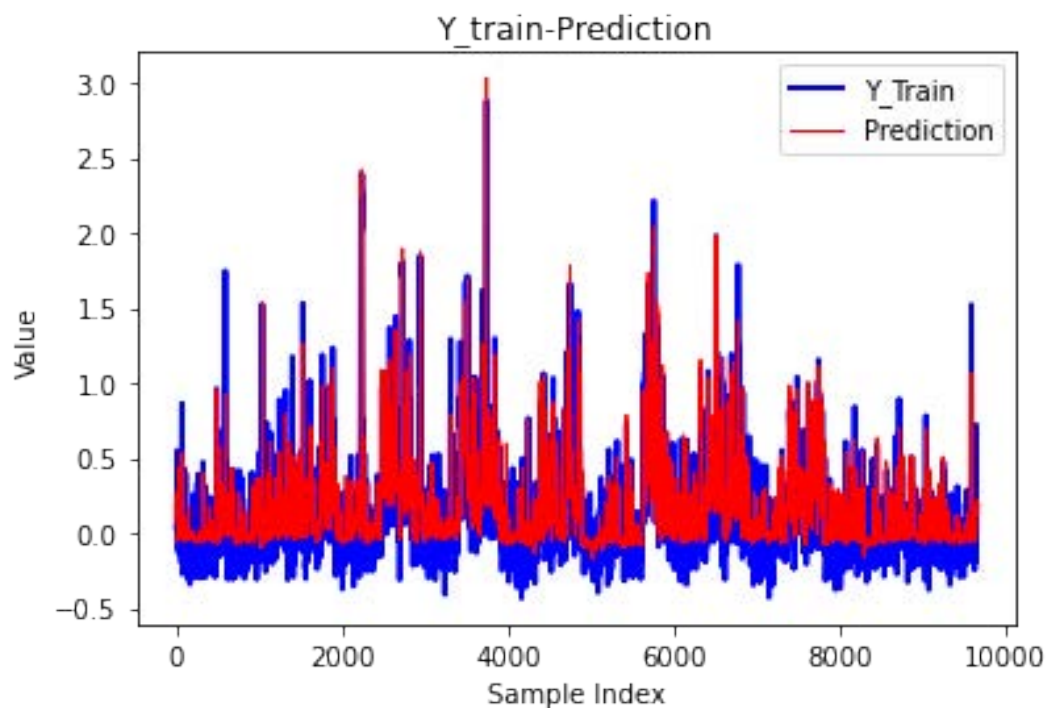


Figure 32: Predictions of EEG response on train data

used. The MSE error is our assessment method based on which the deep learning method seems to overpass the mTRF method in average.

In figure 36 we observe that the deep learning technique follows the actual response More accurately than the mTRF, both in the train and test part of the data.

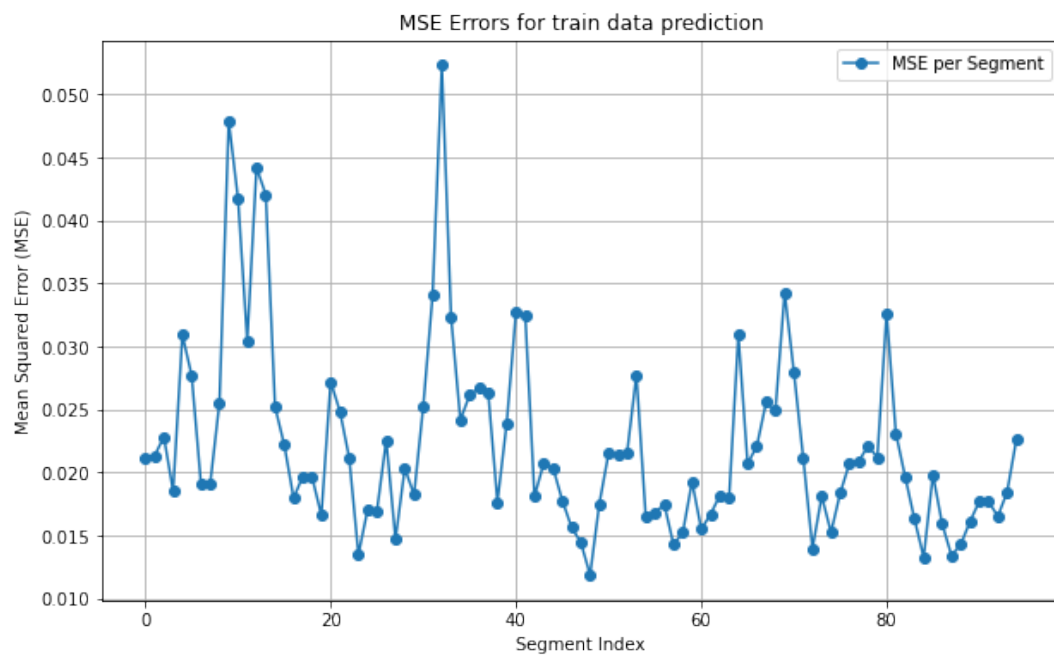


Figure 33: Deep Learning MSE for each test segment.

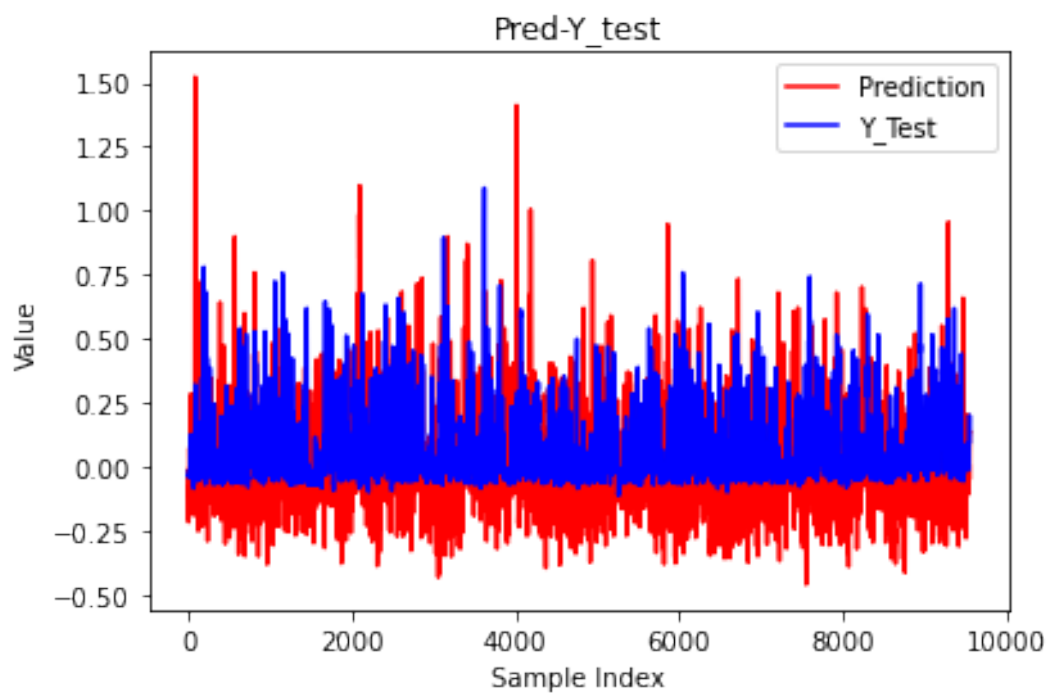


Figure 34: Predictions of EEG response on test data, using the proposed deep learning approach

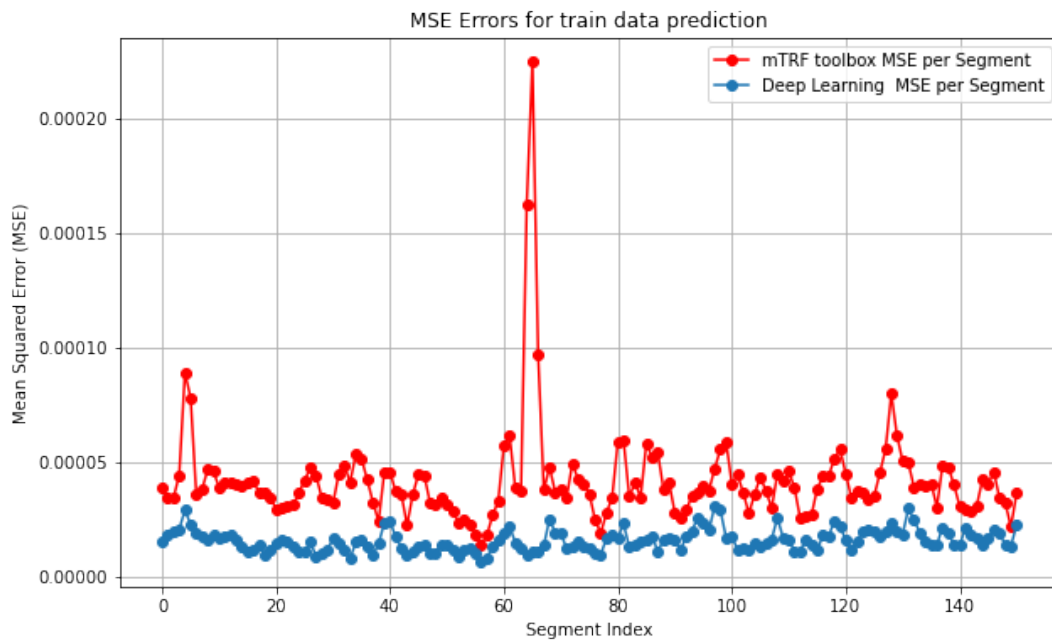


Figure 35: MSE Errors per Segment for each method

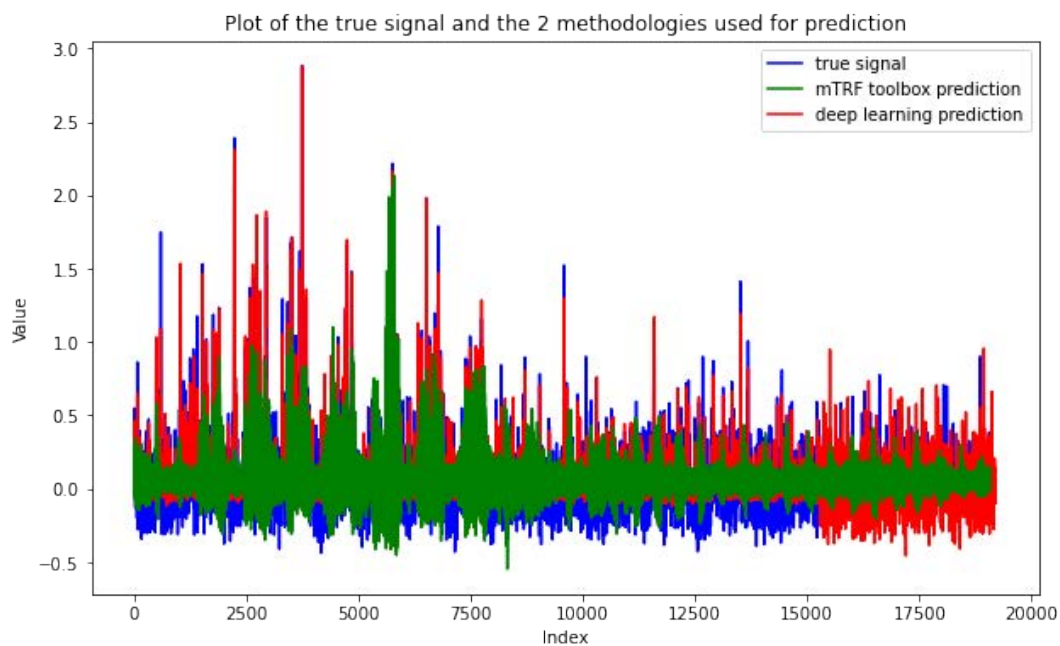


Figure 36: Comparison of both prediction methods and the ground truth data (blue curve).

5 Discussion and Conclusions

In conclusion, we applied our linear and non-linear methods in order to solve the problem of the prediction and we summarized that Deep Learning techniques are very effective and

provide a very promising technique. This study has shed light on the comparative effectiveness of deep learning over the mTRF method in EEG prediction tasks. Our findings strongly support the notion that deep learning techniques outperform mTRF in terms of predictive accuracy and robustness. With such complex neural network architectures, deep learning not only accommodates nonlinear relationships within EEG data but also enhances the overall predictive performance compared to the linear models employed by mTRF.

Additionally, our investigation into the impact of distance on EEG prediction yielded intriguing results. Contrary to initial hypotheses, distance between electrodes did not significantly influence the predictive capabilities of our method. This suggests that EEG prediction models, particularly those employing deep learning, may generalize well across different spatial configurations of electrodes.

Our analysis of mTRF performance across different brain regions underscored notable variations in predictive accuracy. This highlights the importance of considering anatomical differences and functional specialization within the brain when applying mTRF methods for EEG analysis. These insights contribute to a deeper understanding of how methodological choices can influence EEG prediction outcomes.

In essence, while deep learning emerges as a superior approach to EEG prediction compared to mTRF, the nuanced findings regarding electrode distance and regional variations in mTRF efficacy underscore the complexity of neuroscientific inquiry. Future research may benefit from integrating these insights to further refine EEG prediction models and deepen our understanding of brain dynamics. Also, it would be nice to examine the same patterns over a classification on the natural stimuli, a well-known technique in order to check if the models are adapting on the differences that might contain altered sequences, vocal or non-vocal series and different kind of meters.

6 Bibliography

Whole bibliography

- [1] N. I. of Neurological Disorders and S. (NINDS). *Brain Basics: Know Your Brain*. <https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain>. Accessed: 2024-06-17. 2024.
- [2] M. T. Chai et al. “Exploring EEG effective connectivity network in estimating influence of color on emotion and memory”. In: *Frontiers in neuroinformatics* 13 (2019), p. 66.
- [3] S. Koelsch and W. A. Siebel. “Towards a neural basis of music perception”. In: *Trends in Cognitive Sciences* 9.12 (2005), pp. 578–584. issn: 1364-6613. doi: <https://doi.org/10.1016/j.tics.2005.10.001>. url: <https://www.sciencedirect.com/science/article/pii/S1364661305002901>.
- [4] C. MacGregor and D. Müllensiefen. “The Musical Emotion Discrimination Task: A New Measure for Assessing the Ability to Discriminate Emotions in Music”. In: *Frontiers in Psychology* 10 (2019). issn: 1664-1078. doi: 10.3389/fpsyg.2019.01955. url: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2019.01955>.
- [5] D. Quintana. “Understanding the Influence of Agency in Music on the Players’ Experiences of Games and their Emotional Responses”. Dissertation. Unknown, 2023.
- [6] G. Almudena, S. Manuel, and G. J. Jesús. “EEG Analysis during Music Perception”. In: *Electroencephalography*. Ed. by H. Nakano. Rijeka: IntechOpen, 2020. Chap. 2. doi: 10.5772/intechopen.94574. url: <https://doi.org/10.5772/intechopen.94574>.
- [7] D. J. Levitin and S. T. Grafton. “Measuring the representational space of music with fMRI: a case study with Sting”. In: *Neurocase* 22.6 (Dec. 2016). Erratum in: *Neurocase*. 2016 Dec;22(6):i. PMID: 27687156. Epub 2016 Aug 12, pp. 548–557. doi: 10.1080/13554794.2016.1216572.

- [8] N. McNeal et al. “Relating Human Perception of Musicality to Prediction in a Predictive Coding Model”. In: *Unknown* (2022). Unpublished.
- [9] C. S. Pereira et al. “Music and Emotions in the Brain: Familiarity Matters”. In: *PLOS ONE* 6.11 (2011), e27241. doi: 10.1371/journal.pone.0027241.
- [10] G. Soley and E. E. Hannon. “Infants prefer the musical meter of their own culture: A cross-cultural comparison”. In: *Developmental Psychology* 46.1 (2010), pp. 286–292. doi: 10.1037/a0017555.
- [11] J. H. McDermott et al. “Indifference to dissonance in native Amazonians reveals cultural variation in music perception”. In: *Nature* 535 (2016), pp. 547–550. doi: 10.1038/nature18635.
- [12] D. L. Bowling et al. “The Nature and Nurture of Musical Consonance”. In: *Music Perception* 35.1 (2017), pp. 118–121. doi: 10.1525/mp.2017.35.1.118.
- [13] N. Mesgarani et al. “Influence of context and behavior on stimulus reconstruction from neural activity in primary auditory cortex”. In: *Journal of Neurophysiology* 102.6 (2009), pp. 3329–3339. doi: 10.1152/jn.91128.2008.
- [14] C. Li et al. “One-hot Generalized Linear Model for Switching Brain State Discovery”. In: *arXiv preprint arXiv:2310.15263* (2023).
- [15] J.-H. Bach, B. Kollmeier, and J. Anemüller. “Matching Pursuit Analysis of Auditory Receptive Fields’ Spectro-Temporal Properties”. In: *Frontiers in Systems Neuroscience* 11 (2017), p. 4.
- [16] G. B. Christianson, M. Sahani, and J. F. Linden. “The consequences of response nonlinearities for interpretation of spectrotemporal receptive fields”. In: *Journal of neuroscience* 28.2 (2008), pp. 446–455.
- [17] F. E. Theunissen, K. Sen, and A. J. Doupe. “Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds”. In: *Journal of neuroscience* 20.6 (2000), pp. 2315–2331.

- [18] M. Desai, A. M. Field, and L. S. Hamilton. “Dataset size considerations for robust acoustic and phonetic speech encoding models in EEG”. In: *Frontiers in Human Neuroscience* 16 (2023), p. 1001171.
- [19] A. Attaheri et al. “Cortical tracking of sung speech in adults vs infants: A developmental analysis”. In: *Frontiers in neuroscience* 16 (2022), p. 842447.
- [20] L. Bellier et al. “Music can be reconstructed from human auditory cortex activity using nonlinear decoding models”. In: *PLOS Biology* 21.8 (2023), e3002176. DOI: 10.1371/journal.pbio.3002176. URL: <https://doi.org/10.1371/journal.pbio.3002176>.
- [21] M. Crosse et al. “The Multivariate Temporal Response Function (mTRF) Toolbox: A MATLAB Toolbox for Relating Neural Signals to Continuous Stimuli”. In: *Frontiers in Human Neuroscience* 10 (2016), p. 604. DOI: 10.3389/fnhum.2016.00604.
- [22] MathWorks. *mTRF Toolbox*. <https://www.mathworks.com/matlabcentral/fileexchange/74260-mtrf-toolbox>. Accessed: 2024-06-17. 2024.
- [23] Haufe, Stefan and Meinecke, Frank C. and Görgen, Kai and Dähne, Sven and Haynes, John-Dylan and Blankertz, Benjamin and et al. “On the interpretation of weight vectors of linear models in multivariate neuroimaging”. In: *NeuroImage* 87 (2014), pp. 96–110. DOI: 10.1016/j.neuroimage.2013.10.067.
- [24] E. C. Lalor et al. “The VESPA: a method for the rapid estimation of a visual evoked potential”. In: *Neuroimage* 32 (2006), pp. 1549–1561. DOI: 10.1016/j.neuroimage.2006.05.054.
- [25] E. C. Lalor et al. “Resolving precise temporal processing properties of the auditory system using continuous stimuli”. In: *Journal of Neurophysiology* 102.1 (2009), pp. 349–359.
- [26] N. Kriegeskorte and P. K. Douglas. “Interpreting encoding and decoding models”. In: *Current opinion in neurobiology* 55 (2019), pp. 167–179.
- [27] R. de Boer and P. Kuyper. “Triggered correlation”. In: *IEEE Transactions on Biomedical Engineering* 15.3 (July 1968), pp. 169–179. DOI: 10.1109/tbme.1968.4502561.

- [28] N. Mesgarani et al. “Phoneme representation and classification in primary auditory cortex”. In: *J. Acoust. Soc. Am.* 123 (2008), pp. 899–909. DOI: 10.1121/1.2816572.
- [29] M. J. Crosse, F. Di Russo, and M. Gwizda la. “MEG phase connectivity reflects the timing and content of visual word recognition”. In: *Journal of Neuroscience* 36.14 (2016), pp. 3867–3880.
- [30] B. A. O’Sullivan et al. “Encoding task-related information in human auditory cortex using time-resolved MEG”. In: *Journal of Neuroscience* 35.8 (2015), pp. 3363–3380.
- [31] G. E. Hinton and R. S. Zemel. “Autoencoders, Minimum Description Length and Helmholtz Free Energy”. In: *Proceedings of the International Conference on Neural Information Processing Systems*. 1993, pp. 3–10.
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. Ed. by D. E. Rumelhart and J. L. McClelland. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. URL: <http://dl.acm.org/citation.cfm?id=104279.104293>.
- [33] G. Nousias et al. “Video-based eye blink identification and classification”. In: *IEEE Journal of Biomedical and Health Informatics* 26.7 (2022), pp. 3284–3293.
- [34] V. Nair and G. E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.
- [35] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. 2015, pp. 448–456.
- [36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1985), pp. 533–536.
- [37] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

- [38] V. Badrinarayanan, A. Kendall, and R. Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *arXiv preprint arXiv:1511.00561* (2015).
- [39] H. Noh, S. Hong, and B. Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.
- [40] L. Bellier et al. *Data from: Nonlinear decoding models enable music reconstruction from human auditory cortex activity*. Zenodo, Apr. 2023. DOI: 10.5281/zenodo.7876019. URL: <https://doi.org/10.5281/zenodo.7876019>.
- [41] T. Nakai, N. Koide-Majima, and S. Nishimoto. “Correspondence of categorical and feature-based representations of music in the human brain”. In: *Brain and Behavior* 11.1 (2021), e01936. DOI: 10.1002/brb3.1936.
- [42] S. Martin et al. “Neural Encoding of Auditory Features during Music Perception and Imagery”. In: *Cerebral Cortex* (2017). 1–12.
- [43] A. Evans et al. “3D statistical neuroanatomical models from 305 MRI volumes”. In: *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*. IEEE, 1993, pp. 1813–1817. ISBN: 978-0-7803-1487-0. DOI: 10.1109/NSSMIC.1993.373602.
- [44] NIST MNI Brain Atlases. *NIST MNI Brain Atlases*. <https://nist.mni.mcgill.ca/atlas/>. Accessed: 2024-06-14.
- [45] “Empirical Musicology Review”. In: 7.1-2 (Jan. 2012), pp. 5–11.
- [46] H. Honing. “Without it no music: beat induction as a fundamental musical trait”. In: *Annals of the New York Academy of Sciences* 1252 (2012), pp. 85–91. DOI: 10.1111/j.1749-6632.2011.06402.x. URL: <https://doi.org/10.1111/j.1749-6632.2011.06402.x>.

- [47] F. Lerdahl and R. Jackendoff. “An Overview of Hierarchical Structure in Music”. In: *Machine Models of Music*. The MIT Press, Jan. 1993. ISBN: 9780262290982. DOI: 10.7551/mitpress/4360.003.0019. eprint: https://direct.mit.edu/book/chapter-pdf/2305916/9780262290982_can.pdf. URL: <https://doi.org/10.7551/mitpress/4360.003.0019>.

bibliography with the title "Bibliography"