12-2020

A Python-based Brain-Computer Interface Package for Neural Data Analysis

Md Hasan Anowar
The University of Texas Rio Grande Valley

A PYTHON BASED BRAIN-COMPUTER INTERFACE PACKAGE FOR NEURAL DATA ANALYSIS

A Thesis

by

MD HASAN ANOWAR

Submitted to the Graduate College of
The University of Texas Rio Grande Valley
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE in ENGINEERING

December 2020

Major Subject: Electrical Engineering

A PYTHON BASED BRAIN-COMPUTER

INTERFACE PACKAGE FOR

NEURAL DATA ANALYSIS

A Thesis
by
MD HASAN ANOWAR

COMMITTEE MEMBERS

Dr. Paul Choi
Chair of Committee

Dr. Jae Sok Son
Committee Member

Dr. Nantakan Wongkasem
Committee Member

December 2020

ABSTRACT

Anowar, Md Hasan, A Python-based Brain-Computer Interface Package for Neural Data Analysis. Master of Science (MS), December, 2020, 70 pp., 4 tables, 23 figures, 74 references.

Although a growing amount of research has been dedicated to neural engineering, only a handful of software packages are available for brain signal processing. Popular brain-computer interface packages depend on commercial software products such as MATLAB. Moreover, almost every brain-computer interface software is designed for a specific neuro-biological signal; there is no single Python-based package that supports motor imagery, sleep, and stimulated brain signal analysis. The necessity to introduce a brain-computer interface package that can be a free alternative for commercial software has motivated me to develop a toolbox using the python platform. In this thesis, the structure of MEDUSA, a brain-computer interface toolbox, is presented. The features of the toolbox are demonstrated with publicly available data sources. The MEDUSA toolbox provides a valuable tool to biomedical engineers and computational neuroscience researchers.

DEDICATION

The completion of my master’s studies would not have been possible without my family's love and support. My parents and my wife wholeheartedly inspired, motivated, and supported me by all means to accomplish this degree. Thank you for your love and patience.

ACKNOWLEDGMENTS

I will always be grateful to Dr. Paul Choi, chair of my thesis committee, for all his mentoring and advice. He encouraged me to complete this process through his infinite patience and guidance from research design, coding, and manuscript editing. My thanks go to my thesis committee members: Dr. Nantakan Wongkasem and Dr. Jae Sok Son. Their advice, input, and comments on my dissertation ensured the quality of my intellectual work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Motivation

Brain signal analysis has become a significant research field in biomedical engineering. The development of the brain-computer interface makes it possible to regulate an external device only by thought. A brain-computer interface system transforms the brain's electrical activities to control signals used for moving prosthetic limbs or a computer cursor. Studies are going on different brain signal acquisition techniques, feature extraction methods, classification algorithms, and many other aspects of brain-computer interface systems. Although Python has become the most dynamic platform for scientific computing, MATLAB is still prevalent in brain-computer interface paradigms. There are not enough studies on python-based software packages for brain signal processing in the literature.

Being an interactive, object-oriented programming language, Python is the ultimate choice for data analysis. Python has become the state-of-the-art programming language in this era of data science with its full-fledged data analysis libraries like NumPy, SciPy, Pandas, Matplotlib. Python has a noncommercial open source license. Python is also a portable language, which means the code can run on any machine irrespective of the operating system. Python has no fixed data types of variables; all the variables are dynamic. That means the variables can update their types during program execution. The Python interpreter needs to check the data types of the variables repeatedly during the program execution, making Python execution speed

much slower than compiled languages such as C. However, the NumPy library of Python resolves this drawback of Python by assigning static arrays. Moreover, with the help of data visualization libraries such as Matplotlib and Seaborn, Python can display complex graphs, charts, and histograms.

Machine learning and Deep learning have emerged as the most efficient tools for big data analysis in this era of data science. The scikit-learn library of Python provides several classical and advanced machine learning algorithms for both supervised and unsupervised learning (Pedregosa, et al., 2011). Scikit-learn is distributed under a free software license, BSD, which supports the philosophy of open-source knowledge. Though it is written in a high-level language, it is an ease-of-use module. Scikit-learn can successfully harness Python's rich environment for object classification, statistical analysis, and pattern recognition. Moreover, Python also provides a deep neural network library called Keras. Keras usually runs on top of TensorFlow or Theano.

Contribution

In this research, a brain-computer interface toolbox, MEDUSA, is designed using Python. MEDUSA can analyze brain signals to understand the spontaneous and stimulated neural activities. The toolbox can perform motor imagery data analysis, sleep analysis, and speech signal processing. It works with several types of neurophysiological signals, i.e., Electroencephalogram (EEG), Electrocorticogram (ECoG), and Electrooculogram (EOG). MEDUSA can overcome commercial software limitations.

A Graphical User Interface (GUI) is also developed for the toolbox to bring all the features under a common platform. An open-source Python framework, Kivy, is used as the graphical interface for taking full advantage of Python's dynamic nature. The user-friendly

graphic interface allows the users to select the neural signal and feature to be analyzed. The users can visualize their selected type of neural signal and extract biological information from them.

Several machine algorithms from the Scikit-learn library are implemented for classification and regression analysis. The Linear Discriminant Analysis classifier and Common Spatial Pattern filter are utilized for motor imagery classifications. In Steady-State Visual Evoked Potential (SSVEP) epoch classification, the Support Vector Machine classifier is implemented to separate the visual stimulated signal. The Ridge Regression model is used to model the relationship between the speech stimulus signal and EEG.

The thesis is composed as follows: chapter 2 discusses the research background for neural signal analysis. Chapter 3 describes existing Brain-Computer Interfaces toolboxes. Chapter 4 illustrates the proposed python-based Brain-Computer Interface toolbox (MEDUSA). Chapter 5 discusses the features of the developed MEDUSA toolbox. Chapter 6 provides a conclusion.

CHAPTER II

RESEARCH BACKGROUND

Brain Electrophysiological Signals

The human brain produces electrical activities from different parts of the brain. Neurons use these electrical signals to receive and transmit information. Neurons communicate with each other by generating action potentials. A stimulated neuron fires an action potential, and this action potential takes the message signal to the axon terminal. Electrical impulses due to the brain's electrical activities enable the communication process.

The human brain is the most vital part of our nervous system. It consists of two parts, i.e., the central nervous system and the peripheral nervous system. The brain and spinal cord together make up the central nervous system. The peripheral nervous system contains the autonomic nervous system and the somatic nervous system. Again, the brain can be classified into two major regions: cerebral cortex and subcortical areas. The cerebral cortex controls motor and sensory processing. Besides, it also involves pattern recognition, language processing, reasoning, and planning. The ventral temporal cortex of the brain performs complex object recognition. High-level visual processing, i.e., face and scene recognition, is also carried out in the ventral temporal lobe.

Brain Signals Measurement Techniques

There are different techniques to measure the electrical activity of brain signals. The neural data recording system consists of an amplifier, analog to digital converter, and software

for storing the data (Guzman, Schlögl, & Schmidt-Hieber, 2014). Among the recording systems, the most popular methods are Electroencephalography (EEG), Electrocorticography (ECoG), Magnetoencephalography (MEG), and Functional Magnetic Resonance Imaging (fMRI).

Electroencephalography (EEG)

Electroencephalography (EEG) is a non-invasive method of measuring brain electrical activity. Due to cheap measuring devices, most brain-computer interface systems use EEG as the input signal. The electrodes of the EEG measuring devices are placed on the surface of the scalp. These electrodes are not in direct contact with the nerves or muscles; hence, EEG is called a non-invasive method. The EEG recording devices are lightweight, comparatively cheap, and easy to use (Graimann, Allison, & Pfurtscheller, 2009). It makes EEG the most popular technique for brain signal recording.

The EEG electrodes are positioned using an internationally recognized 10-20 system to get a consistent recording of EEG. 10-20 system refers to the electrodes' positions across the scalp and the underlying area of the scalp. The distance of nasion to inion is divided into 10%, 20%, 20%, 20%, and 10% parts. Figure 1 shows the electrodes that are placed at these points (Nicolas-Alonso & Gomez-Gil, 2012).

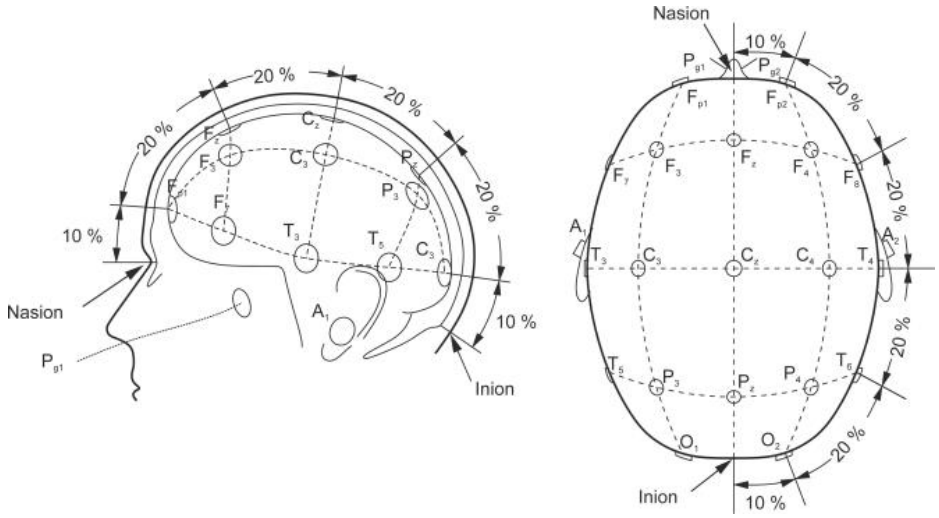


Figure 1: 10-20 EEG Recording System

Table 1 represents the information related to different EEG spectra.

Band	**Frequency (Hz)**	**Amplitude (µV)**	**Location**	**State**
Delta	0.5–4	100–200	The frontal region of the head	Deep sleep
Theta	4–8	5–10	Various regions of the head	Drowsiness, light sleep
Alpha	8–13	20–80	Posterior region of the head	Relaxed
Beta	13–30	1–5	Most evident in the frontal region	Active thinking
Gamma	>30	0.5–2	Somatosensory cortex	Hyperactivity

Table 1: EEG Frequency Bands

The electrodes to capture the EEG signals accumulate action potentials from different sources. It makes EEG signal decoding tricky as the recorded signals are susceptible to noises. The noises in the accumulated EEG signal are known as signal artifacts. Artifacts can be introduced by the subjects, environment, and equipment. As artifacts are not related to the

specified brain activity, different preprocessing techniques are applied to the recorded EEG signal to remove the artifacts.

Heuristically, EEG wave patterns can be distinguished in synchronized and desynchronized activities (Steriade, Gloor, Llinas, Lopes da Silva, & Mesulam, 1990). The synchronized activities are defined as two or more oscillations having the same frequency. Usually, synchronized waves have higher amplitudes and lower frequencies. Reticular thalamic neurons impact these synchronized activities (Steriade, Jones, & Llinls, Thalamic oscillations and signaling, 1990). On the other hand, desynchronized waves have lower amplitudes and higher frequency. An increase of the oscillatory activity in a particular frequency band is known as Event-related synchronization (ERS). Similarly, a decrease of the oscillatory activity in a fixed frequency band is called Event-related desynchronization (ERD). Event-related synchronization (ERS) and Event-related desynchronization (ERD) are essential parts of motor imagery decoding.

The drawbacks of the EEG signal are limited spatial and temporal resolution. As the electrodes of the EEG measuring device are placed on the scalp's surface, these electrodes are not in direct contact with the cortex. The distance from the source affects the recording of the EEG signals. This phenomenon is called volume conduction, and it results in a low spatial resolution in EEG potentials. Suppose the EEG signal's desired signal portion is weak and has other strong signal components having the same frequency range. In that case, low spatial resolution causes a severe problem in signal analysis. The frequency range of visual and sensorimotor signals overlaps most subjects, giving rise to this type of situation (Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2008).

Electrocorticography (ECoG)

Electrocorticography (ECoG) is an invasive method of neural signal recording. It is called an invasive procedure because it requires surgery to place the ECoG electrodes inside the brain. ECoG recording method achieves a better signal to noise ratio and higher frequency spectra than EEG signals.

Electrooculography (EOG)

Electrooculography (EOG) is an eye movement measurement technique where the difference of electrical potential between the front and back of the eye is recorded. The potential change of the anterior and posterior part of the eye is known as standing potential (Constable, Bach, Frishman, Jeffrey, & Robson, Feb 2017). EOG measures this standing corneal-retinal potential. Two electrodes are placed on either left and right side of the eye or above and below the eye. Another electrode is also placed on the forehead to make the ground. Figure 2 shows the placement of electrodes to record Electrooculogram data.

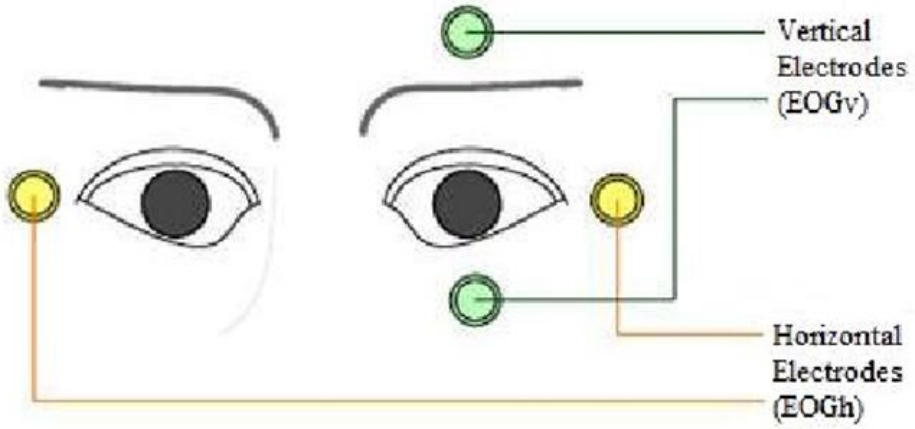


Figure 2: Electrodes Placement for EOG Recording

Any movement of the eye corresponds to a potential change between the retina and cornea. When the eye moves from the center position to the direction of one of the electrodes,

the electrode measures the potential of the retina's positive side. At that moment, the other electrode takes the potential of the negative side of the retina. Rapid eye movement (REM) measurement during sleep can be performed using EOG. Figure 3 presents the EOG data measurement technique.

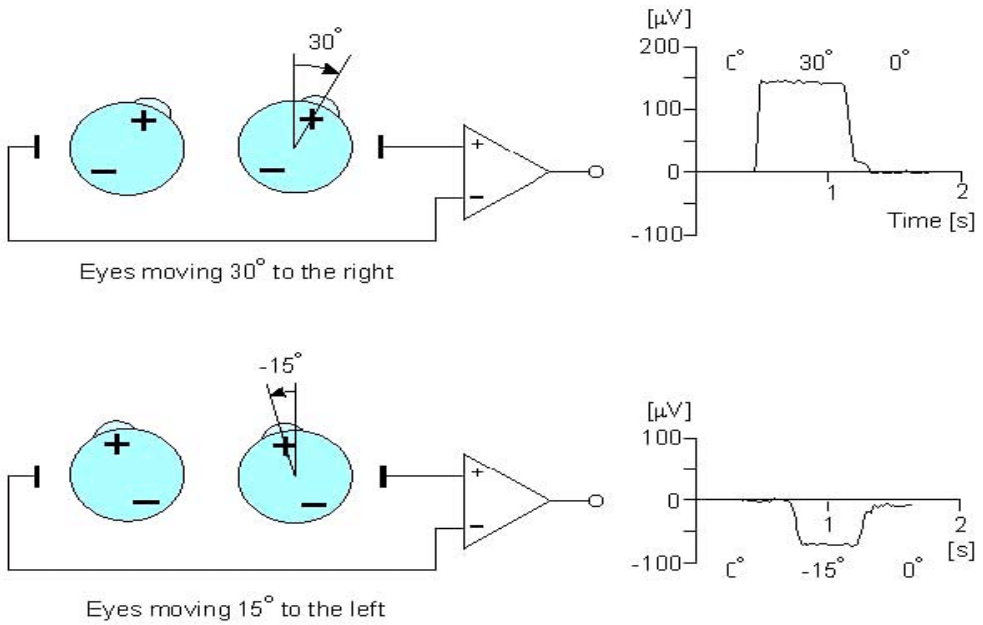


Figure 3: EOG Measurement Method

The values of EOG signals are usually in the microvolts range. The range from 50 µV to 3500 µV. The change of EOG values due to eye movement is almost linear for gaze angles up to 30 degrees (Navarro, Vázquez, & López-Guillén, 2018). The advantage of the EOG method is that it can record large eye movements. EOG can also measure eye movements when the eyelids are closed (Florea, Florea, & Vertan, 2018).

Functional Magnetic Resonance Imaging (fMRI)

Brain activity and blood circulation are related. The Functional Magnetic Resonance Imaging (fMRI) corresponds to neural activity by measuring the change in blood flow near the

active brain cells. When a neuron is active, it needs some energy to fire an action potential. The required energy is released by the blood, which is known as the hemodynamic response. The hemoglobin of oxygenated blood is diamagnetic, and deoxygenated blood hemoglobin is paramagnetic. The method for detecting magnetic signal variation due to blood oxygen level is called blood oxygen level-dependent (BOLD) contrast. fMRI method uses the BOLD contrast imaging technique to map the neural activity.

Neuroimaging techniques like fMRI can understand neural circuitry abnormalities related to emotion regulation (Passarotti, Sweeney, & Pavuluri, 2009). Abnormalities in these neural systems may be associated with different types of mood disorders, i.e., bipolar disorder, depression, and anxiety disorders (Phillips, Ladouceur, & W.C., 2008). Machine learning in neuroimaging can help to discriminate the individuals who are at future risk of psychiatric disorders in the future (Mourão-Miranda, et al., 2012).

Magnetoencephalogram (MEG)

Magnetoencephalography (MEG) is a functional neuroimaging technique to capture the magnetic field produced in the brain. The main difference between MEG from EEG is that MEG uses a magnetic signal instead electrical signal to represent neural activity. MEG can record brain signals with high spatial and temporal resolution. MEG works outside of the skull, so this technique does not need surgery. As the MEG signals are weak, sensitive magnetometers are required to acquire the signal. Recently, superconducting quantum unit interference devices (SQUID) are being used as sensitive sensors. MEG also requires magnetic shielding to avoid interference from the earth's magnetic field. Usually, magnetically shielded rooms (MSR) made of aluminum and nickel-iron soft ferromagnetic alloy are used for the MEG recording

experiment (Ramadan & Vasilakos, 2017). MEG does not have any operational noise, which allows the subject to move its head during MEG recording.

Brain-Computer Interface

Brain-Computer Interface (BCI) is a non-muscular channel between the brain and an external device for communication and control. BCI, also known as the brain-machine interface (BMI), creates a direct pathway between the human body and a computer (Vidal, 1973). Each human body requires peripheral nerves as a communication pathway to control muscles. BCI bypasses the body's original communication pathway with an artificial pathway.

Electrophysiological measures of brain activity are used to develop BCI. BCI identifies the user's intent from brain activities such as slow cortical potentials, mu, and beta rhythms [11]. BCI decodes various scalp recorded rhythms recorded from the scalp. The decoded signal is then used to control an external electrical/mechanical device such as a prosthetic limb, a speller device, etc. The significant application of BCI is clinical BCI for patients who lost muscular control. Since the brain directly regulates the movements of the hands, feet, or similar organs, it is possible to control an external device using only brain signals; without using relevant muscles (Choi & Min, 2015). BCI system enables people with neuromuscular disorders to control sensorimotor activities. In a BCI system, the input is the neural signal, and the output is the control signal. The objective of BCI is to infer values from a person's neurophysiological signal (usually EEG). So, the BCI system can be regarded as an inference problem.

Every BCI system needs to have four features. Firstly, a BCI needs to record direct brain signals, which can be either invasive or non-invasive. Secondly, a BCI should give feedback to the user. Thirdly, user feedback needs to be in real-time. Finally, a BCI must be associated with the user's intentional control (Graimann, Allison, & Pfurtscheller, 2009). A user needs to

imagine performing a mental task to reflect it with a BCI system. Unintentional or passive activities of the brain can not be translated with BCIs.

Based on invasiveness, BCI can be divided into two types: non-invasive and invasive. In non-invasive techniques, signals are recorded by placing electrodes over the scalp. Non-invasive methods, i.e., EEG, do not require any penetration in the scalp. In invasive BCI methods like ECoG, recordings are taken directly from the brain. The invasive techniques record the brain signal underneath the skull. The microelectrodes are inserted on the brain surface. The electrodes are implanted directly into the cortex under the skull for single-cell or multi-unit recording. So, it needs surgery in the brain to put the electrodes inside the head. Invasive methods have improved signal strength than non-invasive methods (Ramadan & Vasilakos, 2017). However, invasive techniques have surgery risks, which is certainly a disadvantage. Due to higher surgical risks and ethical issues, invasive procedures are not usually applied to human subjects (Fouad, Amin, El-Bendary, & Hassanien, 2015). There are different types of non-invasive BCI based on the recording modality, including EEG (Electroencephalography), fMRI (functional Magnetic Resonance Imaging), MEG (Magnetoencephalography), EOG (Electrooculography), NIRS (Near-Infrared Spectroscopy), and PET (Positron Emission Tomography) (Hassanien & Azar, 2015). However, EEG has become the most common technology for BCI applications because of its affordable price and clinical use portability.

BCI systems can be divided into two categories based on signal acquisition methods: Online or Real-time and Offline analysis. Online analysis means real-time data processing. During the online study, EEG or ECoG devices are connected to the subject's scalp during the experiment. The offline analysis is performed using existing prerecorded data. The offline

analysis does not require the subjects' presence or the acquisition device all the time (Das, Tripathy, & Raheja, 2019).

BCI systems can be operated under asynchronous and synchronous modes. They are also known as self-paced and cue-based. In self-paced mode, the user of the BCI has control of the system. The user does not have to wait for the cue from the system. The terminating signal also comes from the user. So, the system needs to be always online and takes input from the brain. This type of system is practical for real-life implementation. But due to continuous operation, the performance of the self-paced system can be affected. It requires a specific change in neural activity from continuous data to detect user intention. On the other hand, the cue-based BCI system works during limited time frames. It needs a cue presented to the user to activate the system. Cue-based systems can be either real-time or offline systems. The implementation of cue-based BCI systems is relatively easy. However, because of user dependency on specific time frames, they are challenging to use in real-life applications (Jochumsen, et al., 2019).

Figure 4 shows brain-computer interface systems classification based on invasiveness, signal acquisition method, and mode of operation.

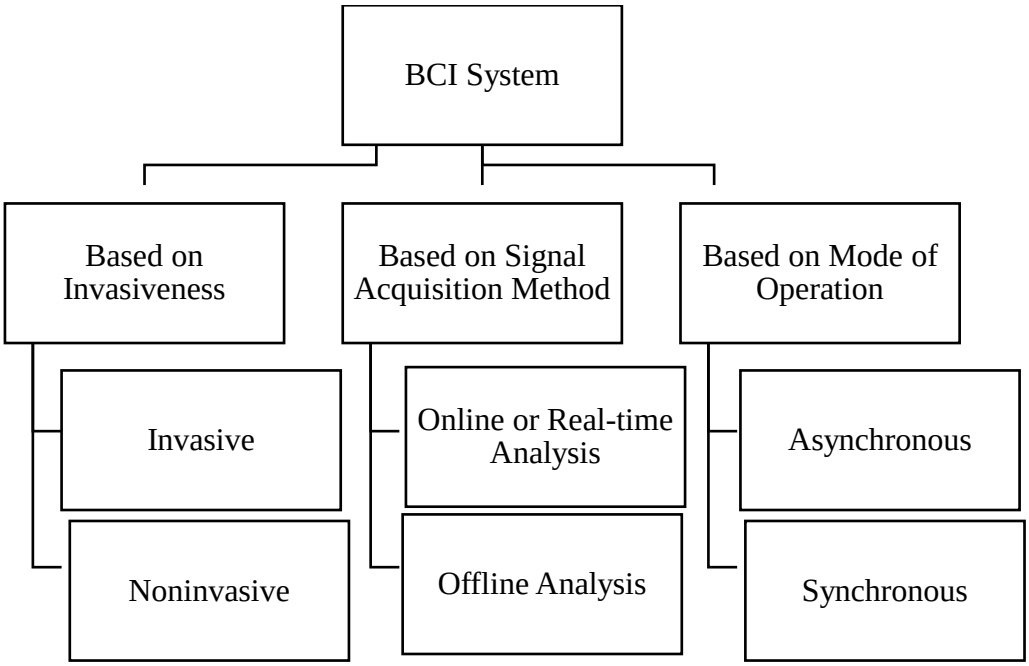


Figure 4: Classification of BCI systems

Figure 5 presents the brain-computer interface systems based on different types of electrophysiological signals.

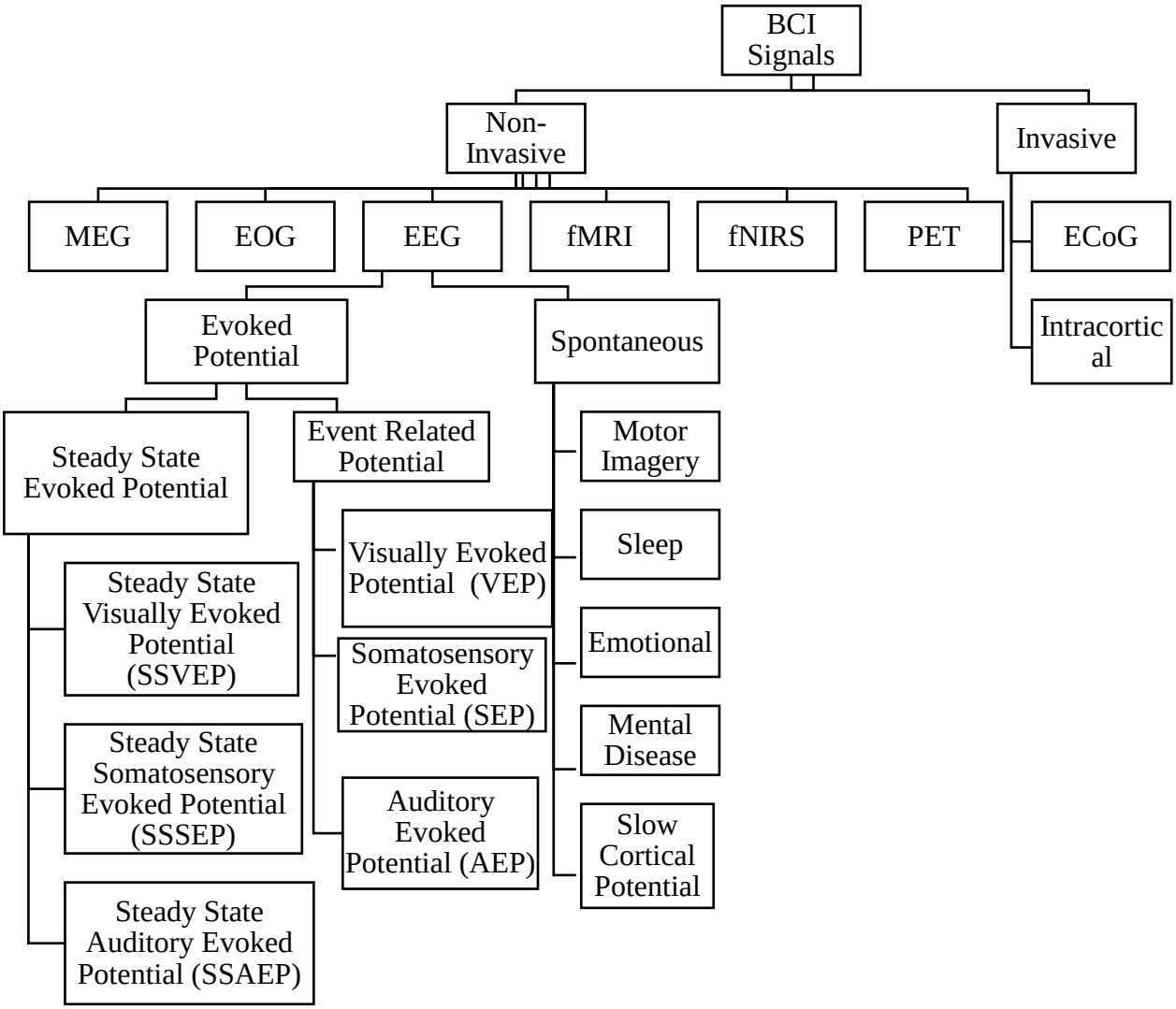
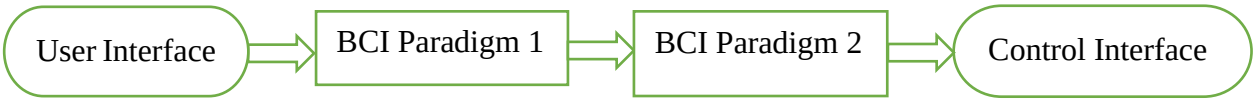


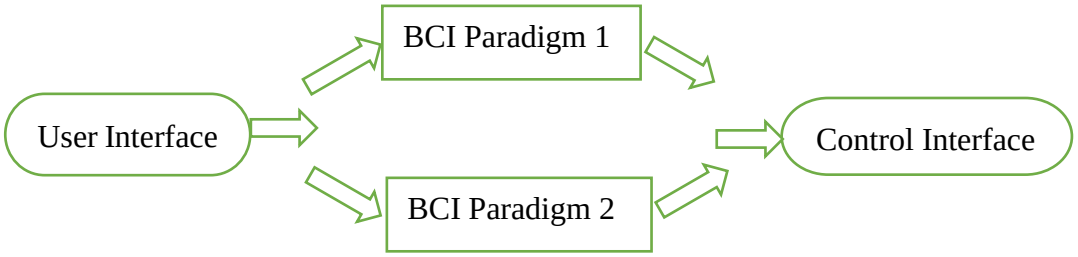
Figure 5: Types of BCI based on Neurophysiological signals

The most suitable technique for implementing BCI in a larger population is using EEG modality. EEG based BCI system can be divided into two categories: (i) externally stimulated paradigms resulting in evoked potential, and (ii) internally induced paradigms known as spontaneous activity. BCI also works with Event-Related Potentials (ERP).

EEG-based BCI has limitations due to the low spatial resolution of the EEG signal. Multimodal methods, along with EEG, are used to overcome this limitation. The multimodal approach is known as the Hybrid BCI system, which combines multiple biological signals (Müller-Putz, et al., 2011). Among the two or more physiological modalities in the hybrid BCI paradigm, at least one needs to be EEG (Amiri, Fazel-Rezai, & Asadpour, 2013). The structures of Hybrid BCI systems are shown in Figure 6.



(a)



(b)

Figure 6: Hybrid BCI Structures

(a) Sequential form (b) Simultaneous form

Brain-Computer Interface (BCI) tasks are two types: Classification and Regression analysis. Motor imagery-based brain-computer interface tasks are usually considered as classification problems. However, they can be analyzed using the regression technique too. Implementation of classification or regression depends on the problem. For example, in Figure 7 (Sellers, Krusienski, McFarland, & Wolpaw, 2007), both regression and classification analysis

need one function for two targets case. Although regression needs only one function for five target cases, the classification approach requires four functions. Figure 7 shows the difference between classification and regression analysis.

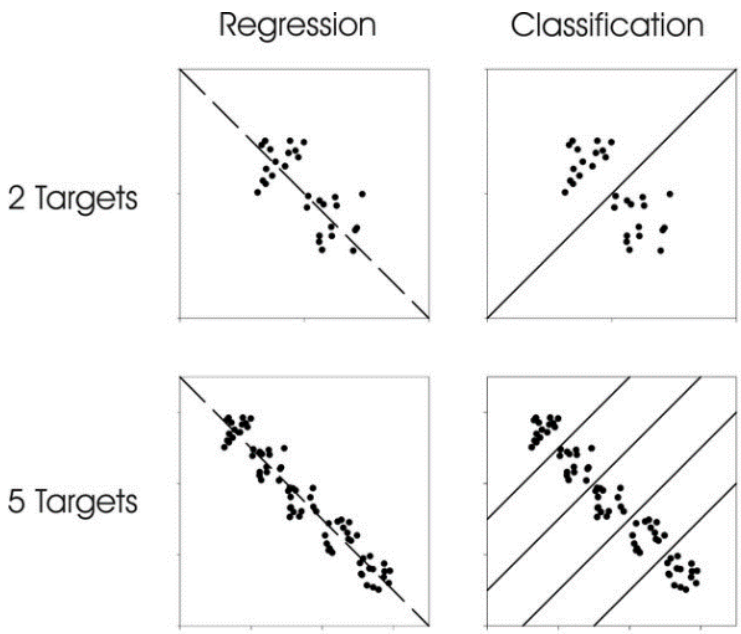


Figure 7: Classification Vs. Regression Analysis

Machine Learning Applications in Brain Signals Analysis

Linear discriminant analysis (LDA), Support Vector Machine (SVM), and K Nearest Neighbor (KNN) perform well as machine learning decision support systems for physiological signals. The machine learning algorithm which calculates a prediction function from the labeled data is called supervised learning. In supervised learning, training data is analyzed to develop a relation between data. This relation is used to map new data samples. This supervised learning algorithm needs to be tested after trained with data.

Machine learning needs relevant feature selection, which consumes a lot of effort. Conventional machine learning techniques to classify EEG signals depend on hand-crafted

features. These methods use information that is already present in the image itself. Besides, it takes lots of time to perform trial and error to find the best feature extraction algorithm. Deep learning overcomes this feature selection and extraction part. Deep learning methods can be applied to physiological signals, i.e., EEG, ECoG, EOG. The convolutional neural network model (CNN) is the most popular deep learning algorithm. However, deep learning is a lot more complicated than machine learning, resulting in longer training time (Faust, Hagiwara, Hong, Lih, & Acharya, 2018).

EEG has a non-stationarity property, which means the average position of an EEG signal defined over one interval is different in another interval. For this reason, feature extraction from the EEG signal is a difficult task. Tayeb et al. (2019) developed three deep learning models for imagery movements analysis directly from raw EEG signals. These are long short-term memory (LSTM), recurrent convolutional neural network (RCNN), and a spectrogram-based convolutional neural network model (CNN). The LSTM model is based on replacing neurons with LSTM units having feedback connections. The spectrogram-based CNN is called a Pragmatic convolutional neural network (p-CNN). As p-CNN shows much better classification performance than both LSTM and RCNN methods, it can be used for Motor Imagery decoding (Tayeb, et al., 2019).

Common Spatial Pattern (CSP)

Common spatial pattern (CSP) is a special type of spatial filter used to decompose a multivariate signal into additive components. CSP maximizes the variance in two-class signal matrices. This method can be implemented for extracting features in a two-class decoding problem. It is mainly used for the brain-computer interface to get the component signals that reflect the cerebral activities related to motor imagery functions (Pfurtscheller, Guger, &

Ramoser, 1999). Since event-related desynchronization (ERD) or event-related synchronization (ERS) are the concerned parts for motor imagery decoding, the CSP method is highly efficient in calculating spatial filters for detecting these segments (Blankertz, Dornhege, Krauledat, Muller, & Curio, 2007).

The common spatial pattern can be adapted for event-related potentials (ERP) analysis (Congedo, Korczowski, Delorme, & Lopes da Silva, 2016). Distinct regions of the brain control the movement of certain parts of the body. For example, the cortex's left hemisphere controls the right hand, and the cortex's right hemisphere controls the left hand (Lemm, Blankertz, Curio, & Muller, 2005). So, every spatial pattern is related to the cortex's specific region that controls the motor activity (Blankertz, Dornhege, Krauledat, Muller, & Curio, 2007). The microscopic model of EEG generation described by Nunez et al. (Nunez & Srinivasan, 2006) is,

$$x(t) = \boldsymbol{A}s(t) + n(t) \tag{1}$$

Where, x(t) = scalp surface potential, **A** = propagation vector, s(t) = source signal, n(t) = noise

The model connects the source activities to the surface electrodes' potentials; hence it is called the forward model. The propagation vector **A** of the forward model represents the coupling strength of each source to the electrodes. **A** is known as the spatial patterns of the sources. By reversing the model, we get,

$$\hat{S}(t) = W^T\, x(t) \tag{2}$$

This backward model relates the acquired signals from the sensors to the originating sources. The rows of the $\mathbf{W^T}$ matrix is called spatial filters. Common Spatial Pattern analysis yields a data-driven signal decomposition represented by $\mathbf{W^T}$ matrix.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a well-known method for classification purposes. PCA is an unsupervised learning algorithm to compress the data. It minimizes the dimensions of high-dimensional datasets. In PCA, correlated variables are expressed in a smaller number of variables to maximize the variance. These smaller numbers of variables are called principal components. But the shape and location of the data sets change if PCA is applied.

Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) calculates a linear combination of features to discriminate between two or more events. To classify between these two classes (both hands movement vs. both feet movement), LDA is implemented. LDA is computationally cheap, easy to implement, and provides almost perfect results as other complex classification techniques [4]. It mathematically models the difference between the classes of data. It tries to maximize the ratio of between-class variance to the within-class variance of a dataset. LDA is vastly used in pattern recognition and machine learning algorithm. In Figure 8, LDA provides a decision region between set 1 and set 2.

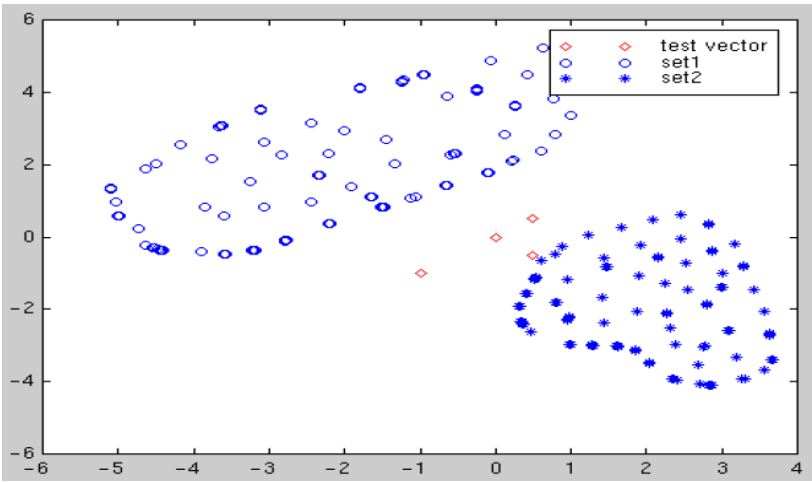


Figure 8: Application of Linear Discriminant Analysis

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning technique that supports both classification and regression analysis. SVM performs classification tasks by constructing hyperplanes to distinguish between objects of different classes. It draws separating lines to define decision boundaries. SVM provides higher accuracy for image classification and image segmentation tasks. The most used kernel functions for SVM are the Linear and the Radial Basis Function (RBF) kernels.

$$K(x, x') = \exp\left(-\gamma||x - x'||^2\right) \tag{3}$$

Where γ = influence of a single training example and $\gamma > 0$.

Cross Validation

In the K-Fold cross-validation method, the original data is split n times, and n-fold cross-validation is performed on those n different splits. At first, each split of training data is cut into n pieces. Then for a single piece of a split, the classifier is trained on the other n-1 pieces. The process is continued for each piece of the split data. The average is then determined to get the accuracy of this split. The whole procedure can be visualized in Figure 9 (Estermann, 2017). In this figure, 10-fold cross-validation is performed.

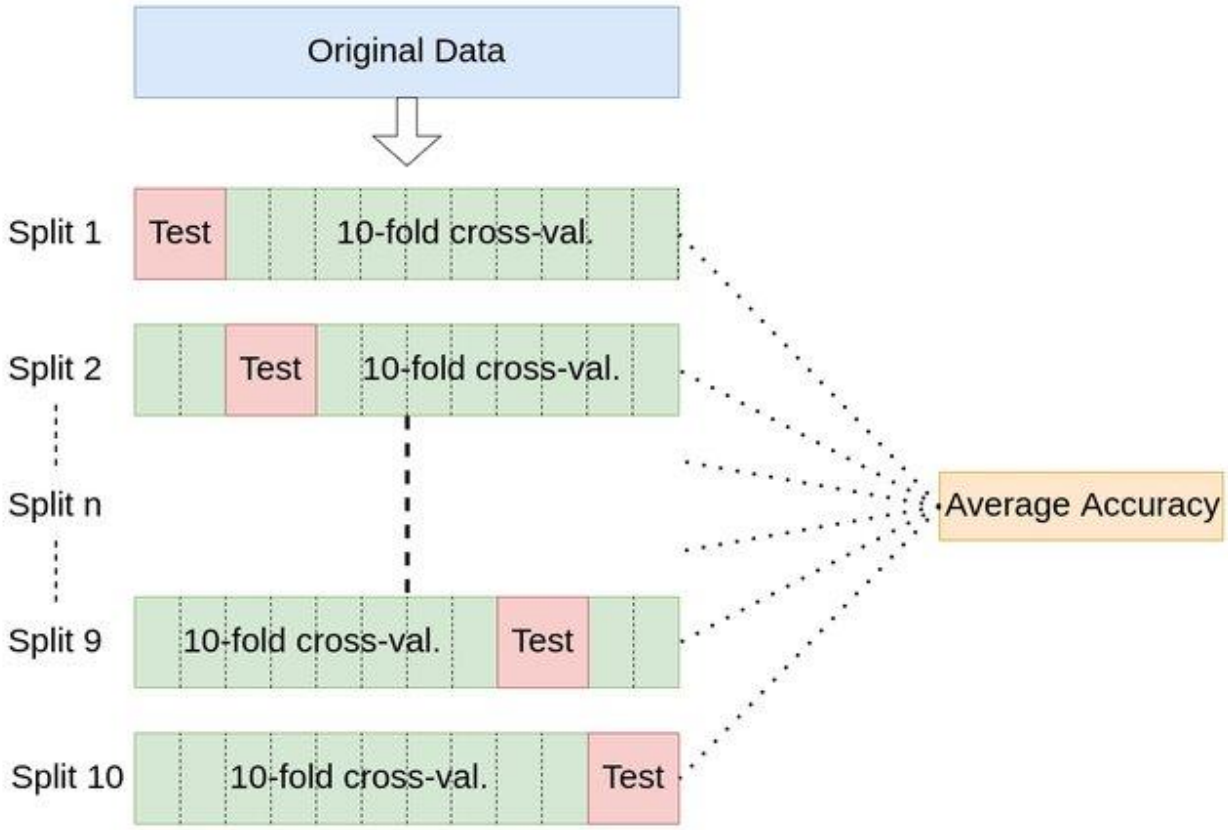


Figure 9: Cross-Validation Workflow

Automatic Event Detection

Automatic event detection detects micro-events, i.e., sleep spindles, k complex, slow-wave, rapid eye movement, etc., from sleep data. Automatic event detection methods are categorized into three types. The first one is filtering the signal to extract the envelope and thresholding with a fixed or tunable step. The second method starts with decomposing the signal into transient and oscillatory components, then filtering, and thresholding are applied. The third method relies on machine learning techniques, filtering, spectral and temporal feature extraction. Then the events from the sleep data can be predicted using binary Support Vector Machine.

Fourier Analysis Vs. Time-Frequency Analysis

The Fourier transform (FFT) is the most common technique to get frequency domain data from time-domain data. Fourier transform gives the frequency content of a signal, but the time information is lost in the process. Fourier analysis considers signal as infinite duration sinusoids; it represents the signal by a sum of continuous sinusoids. However, in practical cases, many signals are of short duration and change properties with time. Although Fourier transform is good at representing stationary signals, it is not suitable to detect the essential characteristics of non-stationary signals like EEG or speech data. The period and frequency of non-stationary signals change substantially over their duration. As Fourier transform is not reliable for transient data analysis, time-frequency analysis can overcome this deficiency. Time-frequency analysis studies a signal in a two-dimensional way, both in time and frequency domains. Most efficient time-frequency analyses are Short-time Fourier transform (STFT) and wavelet transform.

Short-time Fourier Transform (STFT)

Short-time Fourier transform (STFT) performs a local spectral analysis. STFT divides a longer time signal into shorter segments to compute the Fourier transform on each segment. The main disadvantage of the Short-time Fourier Transform is that it can provide only uniform resolution. In STFT, a wide window in the time domain results in more points in the frequency domain. So, the spectrum has more detail, which means it has a higher frequency resolution. But a wide time window means less precision in the time domain, which gives lower time resolution. Similarly, a narrower window in the time domain provides fair time resolution but low-frequency resolution. This relatively poor time-frequency resolution trade-off is a considerable downfall of STFT.

Wavelet Transform

Neuro-biological signals are usually non-periodic and contain fast transients features. Wavelet transform is more suitable than STFT for analyzing these signals. Wavelet transform decomposes a signal in a sum of small parts called wavelets. Wavelets are a special kind of function that shows oscillatory behavior for a short period with an average value of zero. Wavelet transform offers several advantages for transient signal analysis. The most significant advantage of wavelet analysis is its adaptive resolution property. Unlike the STFT, the Wavelet transform does not perform signal analysis at a fixed resolution at different frequencies. As the length of a wavelet is inversely proportional to the frequency, the time-frequency product remains the same. That means the number of cycles of oscillations within a wavelet is constant. In the case of low frequencies, frequency resolution increases while time resolution decreases. Thus, wavelet transform provides high-frequency resolution at low frequencies. Similarly, it provides high time resolution at high frequencies. Overall, the wavelet transform yields better time-frequency resolution comparing with STFT. Moreover, the computational time for EEG analysis using wavelet transform is significantly lower than with Fourier analysis (Schiff, Aldroubi, Unser, & Sato, 1994). In Figure 10, the Fourier Transform, STFT, and Wavelet Transform are compared.

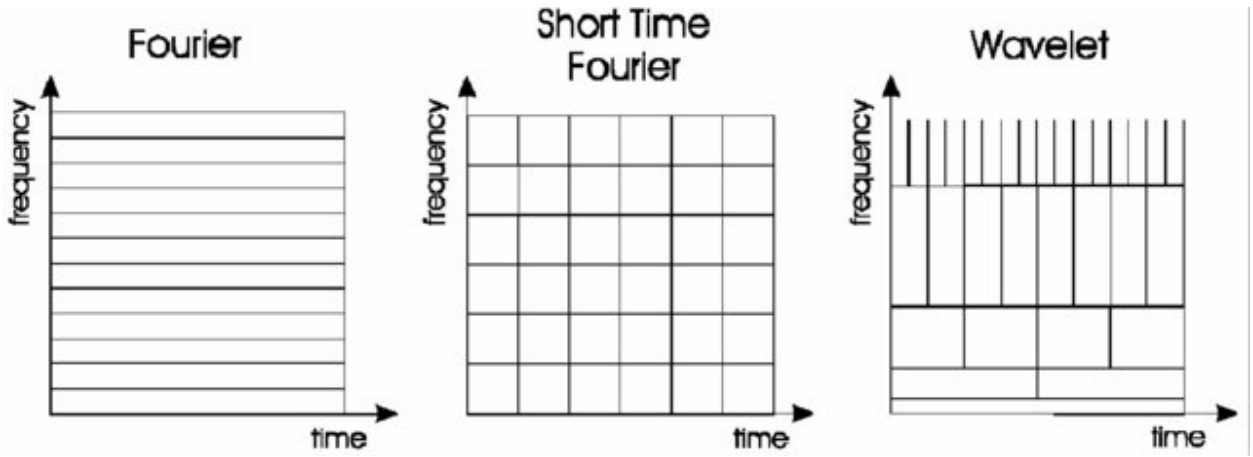


Figure 10: Comparison of Fourier Transform, STFT, and Wavelet Transform

Morlet wavelet, derived from Gabor wavelet, gives the best time-frequency resolution trade-off. That is why the wavelet transformation using Morlet wavelet has become an essential part of event-related brain activity analysis (Sinkkonen, Tiitinen, & Näätänen, 1995). The real-valued Morlet Wavelet is shown in Figure 11.

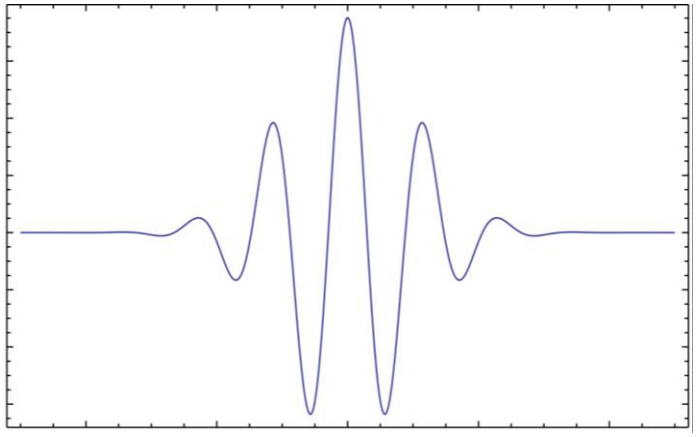


Figure 11: Real-valued Morlet Wavelet

CHAPTER III

RELATED WORKS

Brain-Computer Interface Toolboxes using MATLAB and C/C++

There are several Brain-Computer Interface (BCI) software platforms for researchers and general users. However, most of the BCI software require third party commercial software like MATLAB. The MATLAB development environment is a rapid prototyping tool. It has gained popularity as a computing platform due to its vast computational tools for high-performance prototyping. Apart from MATLAB, C, and C++ language based BCI tools are also available.

EEGLAB

EEGLAB is the most well-known software to process neurophysiological data. It has been widely used as a signal processing environment for EEG signal analysis. It is written in the MATLAB language. EEGLAB provides continuous and event-related EEG and MEG data analysis. EEGLAB has advanced signal preprocessing methods like time-frequency analysis, artifact rejection, independent component analysis (ICA), etc. It supports 20 different binary file formats of neuro signal data. It also comes with a user interface. However, EEGLAB runs on the MATLAB software environment, which requires a commercial MATLAB license.

BCILAB

BCILAB is a MATLAB toolbox for building and testing human-computer interfaces. It works as a plugin for EEGLAB. It is open-sourced and distributed under a GPL license. BCILAB provides a rich graphical interface that supports all the functionality in the toolbox.

g.BCIsys

g.BCIsys is a commercial BCI research tool by Gtec which supports several data acquisition systems. It is a MATLAB/Simulink based BCI research and development system. g.BCIsys processes brain signals to record slow cortical potentials and P300 waves. It has Simulink blocks for data visualizing and storing as well as for parameter extraction and classification. g.BCIsys has g.HIsys library for biosignal data acquisition. Besides, providing standard Simulink blocks, g.HIsys library allows the users to write customized code using MATLAB or C. The g.BSanalyze library of g.BCIsys allows performing multimodal signal processing and analysis. g.BSanalyze also has a graphical user interface for EEG, ECoG, EOG, EMG data analyses.

BCI2000

BCI2000 is a general-purpose BCI research software. It was first released in 2001 and written in C/C++ language. It is free for education and non-profit research purposes. It incorporates biosignals, signal processing techniques, output devices, and operating protocols. BCI2000 contains four modules. The source module, signal processing module, user application module, and operator module perform brain signal acquisition, process the brain signals, provide user feedback, and interface with the user, respectively. It comes with an interactive user interface and supports online signal processing. BCI2000 has a stimulus presentation program to present auditory and visual stimuli to the BCI system user. This stimulus presentation module is suitable for psychophysiological experiments, i.e., implementing event-related potential (ERP) paradigms. The real-time performance of the BCI2000 platform was tested using three BCI systems: cursor control using sensorimotor rhythms, cursor control using slow cortical potentials,

and spelling using P300 potential (Schalk, McFarland, Hinterberger, Birbaumer, & Wolpaw, 2004).

BCI++

BCI++ is an open-source and free application for brain-computer interfaces. BCI++ consists of two modules: the first module is called the hardware interface module. The hardware interface module performs signal acquisition and visualization. It allows real-time analysis either in C/C++ or MATLAB environment. It is written in C/C++ and built for Microsoft Windows only. It can work both for home applications and laboratory experiments. The hardware interface module supports Kimera II, G.Mobilab, Neuroscan, Brain Product devices. BCI++ communicates with these devices via both Bluetooth and TCP/IP. The second module is a graphical user interface module known as AEnima. Similar to the hardware interface module, AEnima is also written in C/C++ language. It was developed using a multiplatform graphics engine. It is responsible for the management of different protocols based on 2D/3D graphic engine. The hardware interface module and AEnima communicate via TCP/IP connection. Steady-state visual evoked potential (SSVEP) and motor imagery BCI systems are presented as example applications to show the capabilities of BCI++ (Perego, Maggi, Parini, & Andreoni, 2009).

BioSig

The BioSig is one of the oldest MATLAB-based toolboxes for offline analysis of EEG and ECoG signals. It has a free and open-source comprehensive library for biomedical signal processing. Along with MATLAB, it is also compatible with Octave. Most of the functions can be used in both MATLAB and Octave. The library has many statistics and time series analysis

tools, including classifiers, cross-validation, adaptive autoregression, blind source separation, common spatial pattern, etc. (Schlögl & Brunner, 2008).

xBCI

xBCI is a generic platform to build real-time BCI systems. The platform was developed using C/C++ language. It supports multithreaded parallel processing and online data classification. xBCI provides a graphical user interface-based diagram editor to design the systems efficiently. xBCI has a dependency on BioSig for C/C++ toolbox, which saves and loads EEG data. The framework was validated using two different BCI systems: motor imagery based BCI and steady-state visual evoked potential (SSVEP) protocol based BCI (Susila, Kanoh, Miyamoto, & Yoshinobu, 2010).

OpenViBE

OpenViBE is a general-purpose software which allows user to build and test BCI systems. It has a unique approach called visual programming, which distinguishes it from other BCI platforms. This visual programming enables users to use the platform without the knowledge of programming. OpenViBE has dedicated tools for virtual reality applications. It contains tools for visualization and feedback based on virtual reality and 3D display. The graphic interface is easy to use for developing BCI systems. The key features of the software are illustrated in two virtual reality-based BCI applications. In these experiments, the user moves a virtual object through real or imagery motor movements (Renard, et al., 2010).

FieldTrip

FieldTrip is a comparatively new MATLAB based EEG, MEG, and other electrophysiological signal processing toolbox (Oostenveld, Fries, Maris, & Schoffelen, 2011). FieldTrip provides algorithms for time-frequency analysis using multitapers, source

reconstruction, event-related potential study, and statistical inference. However, it does not have a graphical user interface; the user needs to write scripts to use the functions.

Brainstorm

Brainstorm is a platform for the real-time analysis of brain signals. This toolbox contains an interactive user-interface that facilitates non-programming users to use the software. Brainstorm allows multimodal signal recording, artifact detection and correction, visualization along with 2D and 3D surface mapping, source modeling, time-frequency analysis using Morlet wavelet, Fast Fourier Transform and Hilbert transform, functional connectivity, pattern analysis using machine learning, etc. It can remove common artifacts such as eye blinks, heartbeats, powerlines, etc. Brainstorm applies signal space projection to remove ocular and cardiac artifacts. The software has been developed using MATLAB and Java. However, researchers can use it without having a commercial MATLAB license (Tadel, Baillet, Mosher, & Pantazis, 2011).

Table 2 lists the most common BCI software tools written in MATLAB and C/C++ languages.

Toolbox Name	Brain Signal	Applications	Online/Offline Analysis	User Interface	Platform	License	Operating System
EEGLAB	EEG, MEG	Continuous and Event-related analysis, Independent Component Analysis, Time-Frequency Analysis, Artifact Rejection, Data Visualization	Realtime	Yes	MATLAB	BSD	Windows, MAC, Linux
BCILAB	EEG, MEG	Signal Processing, Feature Extraction,	Realtime	Yes	MATLAB	GPL	Windows, MAC, Linux

		and Machine Learning					
g.BCIsys	EEG, ECoG	Simulink blocks for Feature Extraction and Classification	Realtime	Yes	MATLAB/ Simulink	Proprietary	Windows, Linux
BCI2000	EEG	Data Acquisition, Stimulus Presentation, Brain Monitoring	Realtime	Yes	C++	GPL	Windows, MAC, Linux
OpenViBE	EEG	Visual Dataflow Programming, 2-D Visualization, Virtual Reality	Realtime	Yes	C++	LGPG	Windows, MAC, Linux
BCI++	EEG	Signal Acquisition, Storage, Visualization, Real-Time Execution	Realtime	Yes	C/C++	GPL	Windows
xBCI	EEG	Data Acquisition (DAQ), Data Processing, Classifiers, Visualization	Realtime	Yes	C++	GPL	Windows, Linux
BioSig	EEG, ECoG, ECG, EOG, EMG	Data Acquisition, Artifact Rejection, Feature Extraction, Classification, Data Visualization	Offline	No	MATLAB/ OCTAVE	GPL	Windows, macOS, Linux
FieldTrip	EEG, MEG, iEEG, NIRS	Time-Frequency Analysis Using Multitapers, Source Reconstruction, Event-Related Potential Analysis, And Statistical Inference	Realtime	No	MATLAB	GPL	Windows, macOS, Linux
Brainstorm	EEG, MEG, fNIRS, ECoG	Signal recording, Visualization, Source modeling, Time-Frequency analysis, Functional Connectivity	Realtime	Yes	MATLAB	GPL	Windows, macOS, Linux

Table 2: MATLAB and C/C++ based Brain-Computer Interface Toolboxes

Brain-Computer Interface Toolboxes based on Python

The following BCI toolboxes are based on the Python platform. They are listed in PyPI and GitHub repositories.

MNE-Python

MNE-python is a software package for Electroencephalography (EEG) and Magnetoencephalography (MEG) signal processing. MNE data analysis tool covers all phases of EEG and MEG analysis. MNE-python is a branch of MNE software projects which also includes MNE-C, MNE-CPP, MNE-MATLAB. The default data format of MNE for FIF, it also supports various other types of data formats, i.e., European Data Format (.edf), Biosemi (.bdf), BrainVision, etc. MNE-python has dependency on SciPy and NumPy. However, to achieve full functionality, Matplotlib, Mayavi, PySurfer, Scikit-learn, Numba, NiBabel, Pandas, Picard, DIPY, Imageio, PyVista are required. MNE provides an Independent Component Analysis (ICA) class that implements the ICA algorithm to remove the EEG or MEG artifacts and noise. ICA decomposes a multivariate signal into statistically independent subcomponents. A significant application of ICA is the reduction of signal noises by making the corresponding subcomponent to zero. MNE-Python supports FastICA, which is an ICA technique available in Scikit-Learn. Filtering the raw signal into a specific frequency of interest is a significant part of the EEG & MEG signal preprocessing. MNE's filter function supports both finite impulse response (FIR) and infinite impulse response (IIR) filters. MNE provides an EEG and MEG dataset collected from Martinos Center of Massachusetts General Hospital. In the recording experiment, both visual and auditory stimulation were presented in a random sequence. The EEG data were collected using 60 electrodes simultaneously, and the MEG data were collected using the 306 channel Neuromag Vector View MEG system. MNE-Python works with three types of data

structures: raw, epoch, evoked. The recorded data without any preprocessing is called raw data. Epoch data are the segmented raw data; they are extracted from the raw format in each stimulation event. Averaging epoch data generates evoked data (Gramfort A. , et al., 2014).

Wyrm

Wyrm is a brain-computer interface software written in Python. It has a set of functions for preprocessing, feature extraction, and classification. It can be used for the motor imagery movement decoding and event-related potential analysis (Venthur, Dahne, Hohne, Heller, & Blankertz, 2015). It supports both real-time and offline research. Wyrm works together with two different python packages named Mushu and Pyff. The Pyff works as a cross-platform framework to perform neuroscientific experiments. Pyff is under the General Public License (GNU). Pyff provides a pythonic platform for easy development of feedback applications and a set of useful, high-quality stimulus presentations (Venthur, et al., 2010). Mushu is another python-based signal acquisition software for EEG data streaming. Mushu serves as a unified interface and is placed between the amplifier for the raw EEG data acquisition. After reading and converting the data, it sends the data to the BCI system (Venthur & Blankertz, Mushu, a free- and open source BCI signal acquisition, written in Python, 2012).

Eelbrain

Eelbrain is a python-based toolbox for statistical analysis of EEG and MEG signals (Brodbeck, Brooks, Das, & Reddigari, 2019). Eelbrain package has a Var class that works as a container for one-dimensional scalar data. For multidimensional data, the Eelbrain software uses NDVar. NDVar associates the description of dimensions with the data. Eelbrain provides a user interface application developed using a cross-platform user interface library called wxPython.

YASA

Yet another spindle application (YASA) is a python tool for sleep analysis. YASA has functions for the spindle, slow-wave, and rapid eye movement detection. It works with EEG and EOG signals.

Visbrain

Visbrain is a python package developed for brain signal visualization. The sleep module of the Visbrain package is dedicated to visualizing and analyzing polysomnographic data (Combrisson, et al., 2017). It provides spindle and slow wave detection techniques from sleep rhythms. In Visbrain's sleep module, sleep signals are measured in 30 seconds of epochs (McGrogan, Braithwaite, & Tarassenko, 2001). The package supports European Data Format (.edf), Micromed (.trc), Brain Vision (.eeg), and Elan (.eeg).

Wonambi

Wonambi is a package for electrophysiological data analysis written in Python. Wonambi package requires the use of NumPy and SciPy libraries. It also has optional dependencies on PyQt5, python-vlc, vispy, h5py, mne, nibabel, tensorpac, and fooof 1.0. Wonambi comes with a graphic interface that enables users to score sleep stages. Wonambi has functions for time-frequency analysis like short-time Fourier transform and wavelet transform.

SSVEPY

SSVEPY is a package for steady-state evoked EEG data processing. SSVEPY can perform offline analysis on visual, auditory, thermo-sensory signals. It has a dependency on MNE and h5py.

Nilearn

Nilearn is a machine learning-based python toolbox for multivariate analysis of neuroimaging data. Nilearn takes advantage of the scikit-learn module for pattern analysis, classification, and decoding. Machine learning in neuroimaging can help to discriminate the individuals who are at future risk of psychiatric disorders in the future (Mourão-Miranda, et al., 2012). Nilearn applies a machine learning model for Multi-Voxel Pattern Analysis (MVPA) and fMRI data decoding. Nilearn performs data masking by using a mask to get the time series data from the NiftiImage object. Data masking is the process of generating a similar substitute for the actual data. Data masking is to protect the original data and reduce the risk of data breaches.

Nitime

Nitime is a python package for analyzing experimental fMRI data in time series. Nitime can perform spectral estimation from fMRI data. The experimental fMRI data usually contain a wide range of the spectrum. So, the data preprocessing is necessary to extract the vital part from the signal.

Table 3 lists the existing python-based brain-computer interface software tools.

Toolbox Name	Brain Signals	Applications
MNE Python	EEG, MEG, sEEG, ECoG, fNIRS	Data Visualization and Preprocessing, Source Estimation, Time-Frequency Analysis, Connectivity Analysis
WYRM	EEG	Filtering, Feature Extraction, Classification
EELBRAIN	EEG, MEG	Statistical Analysis of EEG and MEG
YASA (Yet Another Spindle Algorithm)	EEG, EOG, Sleep	Sleep analysis, Spectral Analysis, Spindle, Slow-wave, REM detection
Visbrain	EEG, Sleep	Sleep Analysis, Spindle, Slow-wave, REM detection
Wonambi	EEG, ECoG, iEEG, sleep	Sleep Staging, Visualization, Slow-wave detection

SSVEPY	EEG	Steady-State Visually Evoked Potential Analysis
Nilearn	fMRI	Mutli-Voxel Pattern Analysis
Nitime	fMRI	Spectral Estimation

Table 3: Existing Python based Brain-Computer Interface Packages

CHAPTER IV

PROPOSED BRAIN-COMPUTER INTERFACE TOOLBOX

A Brain-Computer Interface toolbox called MEDUSA is proposed. Table 4 describes the features of the proposed toolbox.

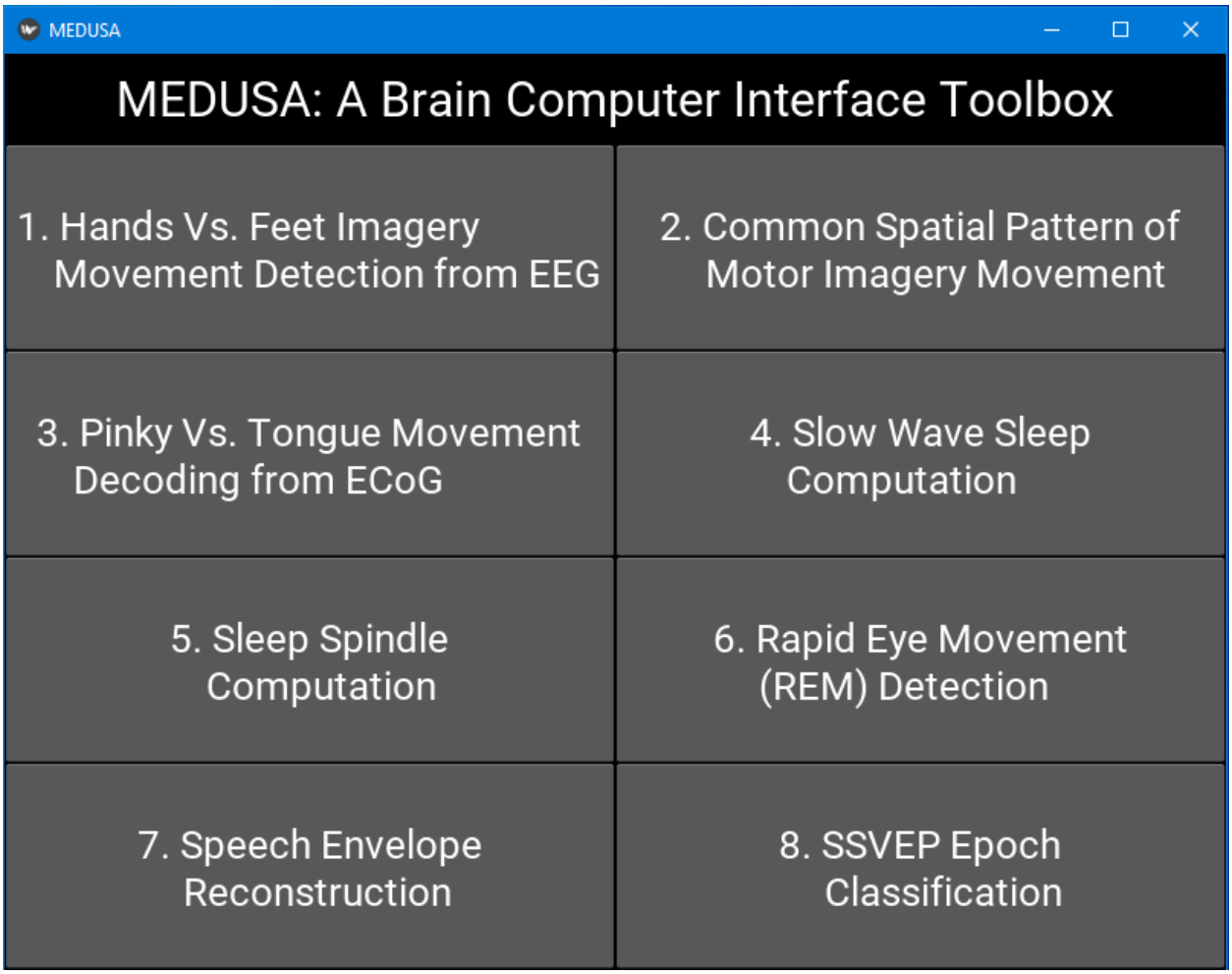
Features	Descriptions	Data type
Hands Vs. Feet Imagery Movement Detection from EEG	Classification accuracy of hands vs. feet motor imagery movement detection	EEG
Common Spatial Pattern of Motor Imagery Movement	Plotting of common spatial pattern of pinky vs. tongue imagery movement	ECoG
Pinky Vs. Tongue Imagery Movement Decoding from ECoG	Decoding of pinky vs. tongue motor imagery movement	ECoG
Slow Wave Sleep Computation	Estimating slow waves from sleep rhythms	EEG
Sleep Spindle Identification	Sleep spindle detection	EEG
Rapid Eye Movement (REM) Detection	Detection of rapid eye movements	EOG
Speech Envelope Reconstruction from EEG	Speech reconstruction from EEG	EEG
SSVEP Epoch Classification	Epoch classification from visual stimulation	EEG

Table 4: Features of the Proposed BCI Package

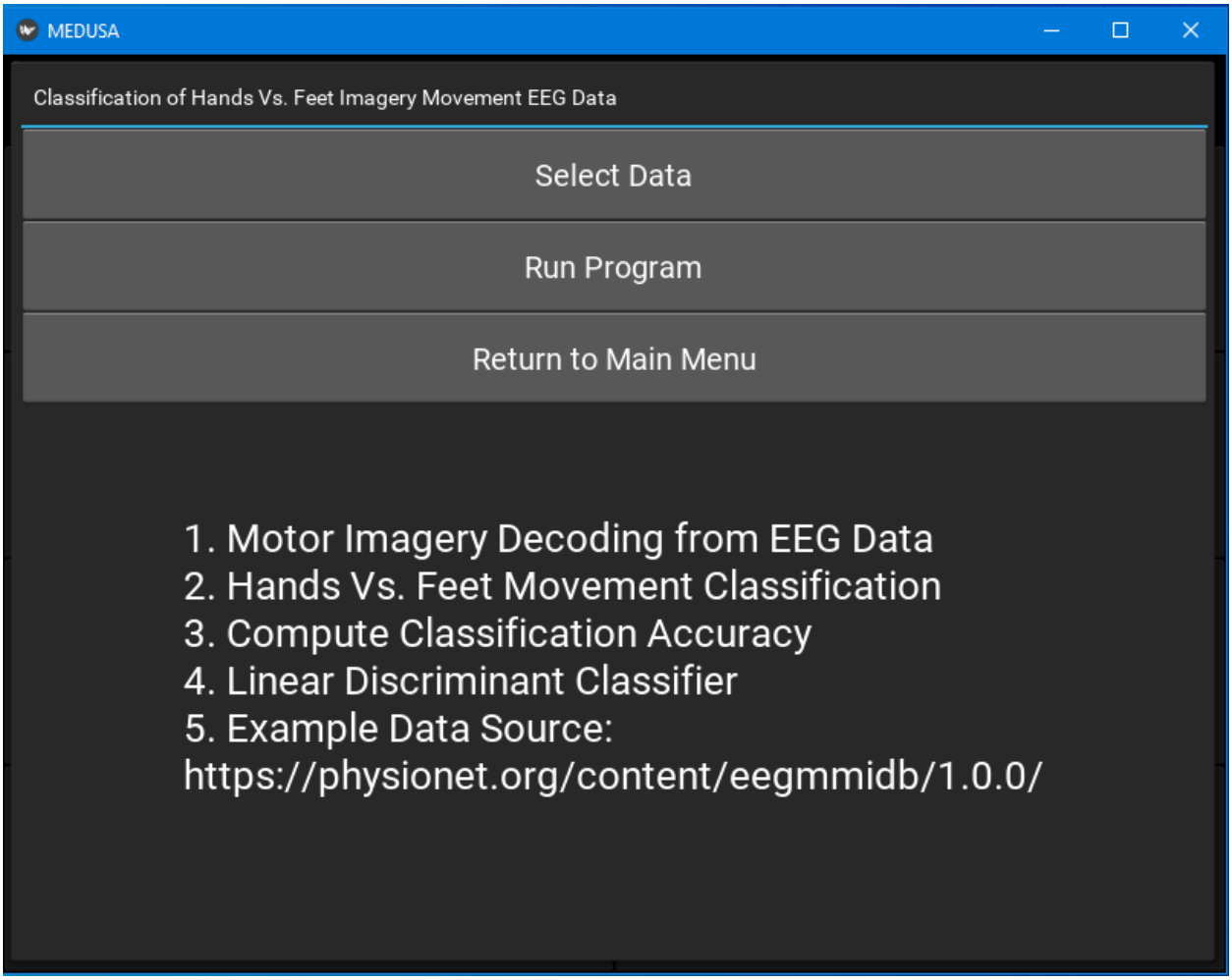
Graphical User Interface

Graphical User Interface works as a flexible tool for conveniently implementing a new operating protocol. So, a user interface is necessary to simplify user applications. The popular Python-based Graphical User Interface (GUI) frameworks are PyQT, Tkinter, Kivy, wxPython, etc. Among these frameworks, Kivy can take full advantage of the dynamic nature of Python. It is an Open Graphics Library (OpenGL) released under MIT license. Kivy also enables the developers to design a cross-language and cross-platform user interface.

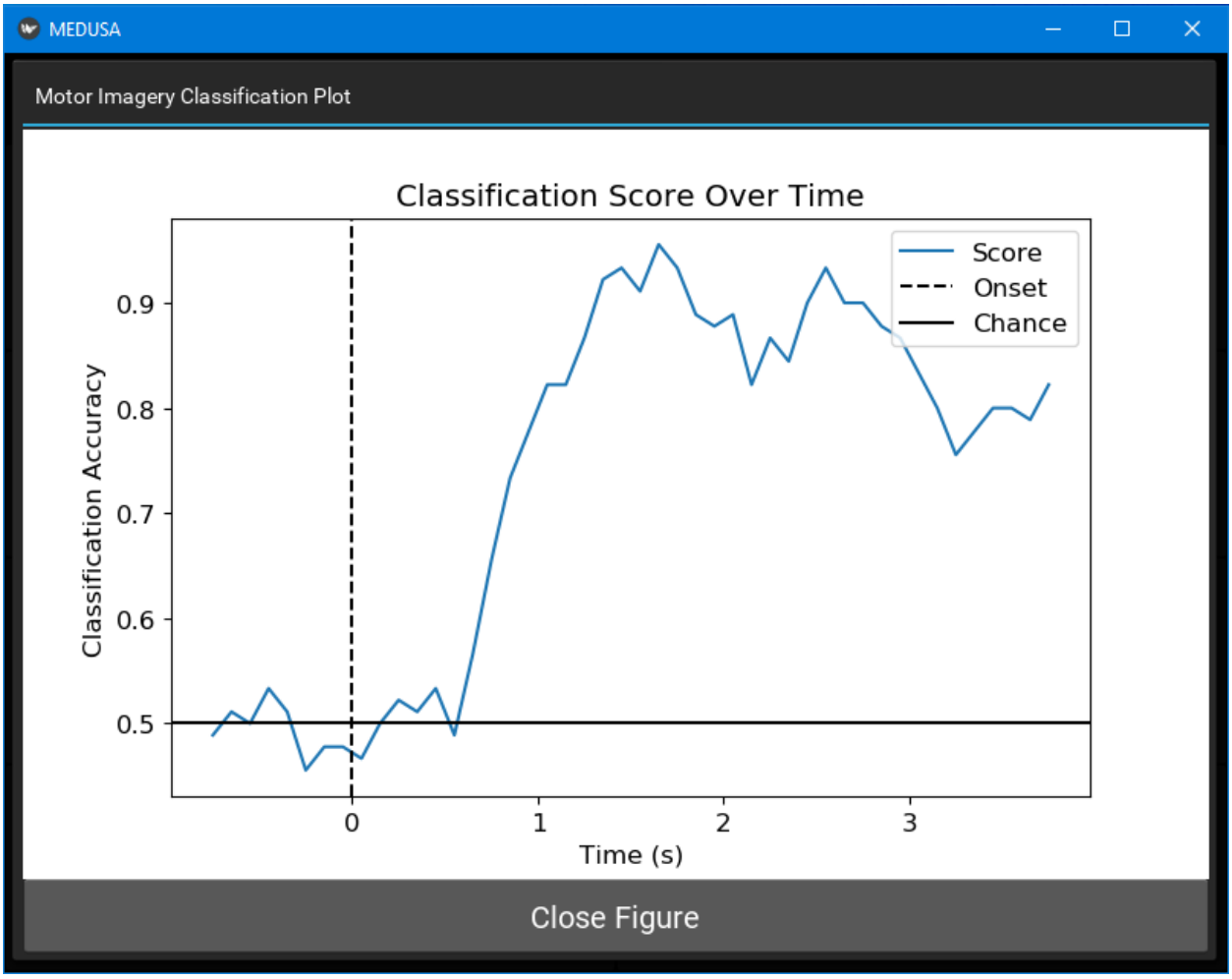
The brain-computer interface toolbox MEDUSA has an interactive Graphical User Interface for neural signal visualization and analysis. The user interface has been designed using Kivy graphic library in the Python platform. The user interface includes classical user interface widgets, i.e., Button, Label, and complex user interface widgets, i.e., FileChooser, Popup. Figure 12 shows the MEDUSA, the developed brain-computer interface toolbox.


MEDUSA
MEDUSA: A Brain Computer Interface Toolbox
1. Hands Vs. Feet Imagery Movement Detection from EEG
2. Common Spatial Pattern of Motor Imagery Movement
3. Pinky Vs. Tongue Movement Decoding from ECoG
4. Slow Wave Sleep Computation
5. Sleep Spindle Computation
6. Rapid Eye Movement (REM) Detection
7. Speech Envelope Reconstruction
8. SSVEP Epoch Classification

(a)


MEDUSA
Classification of Hands Vs. Feet Imagery Movement EEG Data
Select Data
Run Program
Return to Main Menu
1. Motor Imagery Decoding from EEG Data
2. Hands Vs. Feet Movement Classification
3. Compute Classification Accuracy
4. Linear Discriminant Classifier
5. Example Data Source:
https://physionet.org/content/eegmmidb/1.0.0/

(b)



(c)

Figure 12: Developed Python-Based Brain-Computer Interface Toolbox (MEDUSA)

In the above figures, only the hands vs. feet imagery movement detection from EEG feature is shown. (a) The MEDUSA toolbox's top window shows eight different features (b) Popped up window when the corresponding feature is selected (c) Output of the feature after running the program.

Performance Analysis

To compare the performance of the algorithms, some criteria are needed. These are several parameters for performance evaluation.

$$\text{Sensitivity} = \frac{No\ of\ True\ Positive}{No\ of\ True\ Positive + No\ of\ False\ Negative}$$

$$\text{Specificity} = \frac{No\ of\ True\ Negative}{No\ of\ False\ Positive + No\ of\ True\ Negative}$$

$$\text{False Positive Rate} = 1\text{-Specificity} = \frac{No\ of\ False\ Positive}{No\ of\ False\ Positive + No\ of\ True\ Negative}$$

$$\text{False Positive Proportion} = \frac{No\ of\ False\ Positive}{No\ of\ True\ Positive + No\ of\ False\ Negative}$$

$$\text{False Positive Amount} = \frac{No\ of\ False\ Positive}{No\ of\ True\ Positive + No\ of\ False\ Positive}$$

CHAPTER V

RESULT ANALYSIS

Motor Imagery Movement Analysis

The most common brain-computer interface task is the motor imagery movement decoding. Motor imagery is defined as mentally imagining a given body action, i.e., hand, feet, finger, or tongue movement. In motor imagery, the subjects imagine the movements of body parts instead of doing them. Experimental results show that motor imagery exhibits the same neural mechanisms involved in motor control of actual actions (Decety, 1996). Motor imagery movement generates sensorimotor rhythms (SMR). It is possible to decode the activities from the left hand, right hand, feet, and tongue imagery movements data. The respective cortical areas of the signals generated by the left hand, right hand, feet, and tongue imagery movements are comparatively large. For example, the left hand, right hand, and foot motor imagery data originates from C3, C4, and Cz areas of the brain, respectively. These cortical areas are easily distinguishable. So, motor imagery BCI applications are usually developed based on these body parts imagery movements (Schlögl, Lee, Bischof, & Pfurtscheller, 2005).

Hands Vs. Feet Imagery Movement Detection from EEG

For this decoding, both fists and both feet imagery movements are used. The EEG data used here is taken from PhysioNet (Goldberger, et al., 2000). The data is in European Data Format (EDF+). The 64 channel EEG signals were recorded using 64 electrodes. The electrodes were placed following the international 10-10 system.

Each data is sampled at 160 Hz. Data durations are 120 seconds (Schalk, McFarland, Hinterberger, Birbaumer, & Wolpaw, 2004). Both fists and both feet movements are recorded in runs 5, 6, 9, 10, 13, and 14. T0, T1, or T2 annotate each motor imagery task. T0 means that the subject is resting. T1 represents both fists' onset of motion, and T2 represents both feet' onset of movement.

This program takes the first subject and trial numbers 6, 10, and 14 as input. It determines whether the person imagined movement of either both hands or both feet on that particular event. Part of the original data was divided into test sets (X_test, y_test) and train sets (X_train, y_train). ShuffleSplit function from python sci-kit learn library is used to split the dataset. ShuffleSplit function shuffles the data samples and then split the raw data into train and test sets. The model is trained using train data, the parameters are optimized using cross-validation, and lastly, performance is tested using test data.

The Linear discriminant analysis (LDA) classifier is implemented to classify these two classes (both hands movement vs. both feet movement). The transformer uses the training data and training labels and returns a transformed version of training data. LDA classifier is trained with the feature vector computed from the train data. When the training is finished, the trained LDA classifier is used to classify the test data's feature vector to obtain the final result. Finally, each trial is classified from the results of the LDA classifier. Figure 13 represents the machine learning model of the decoding algorithm.

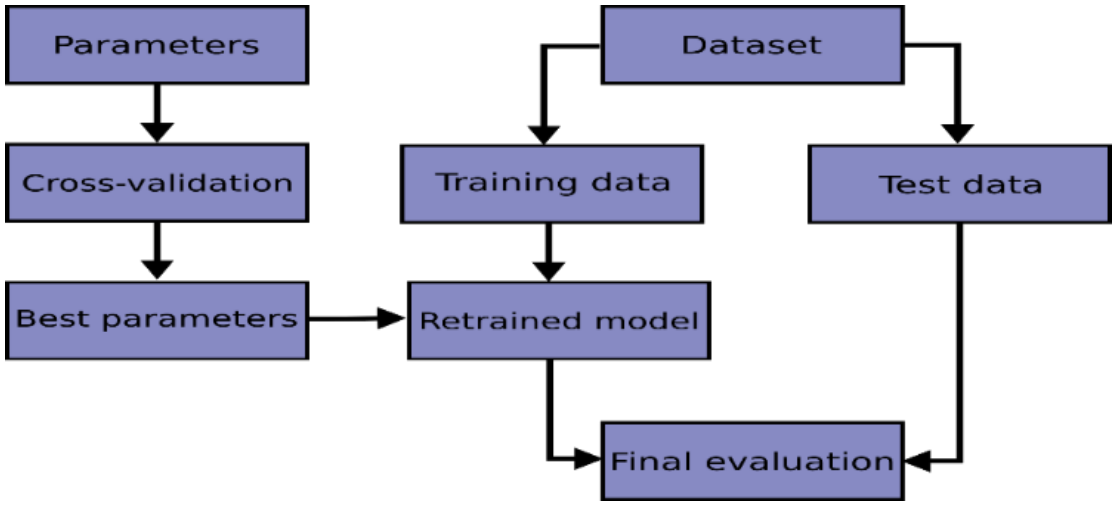


Figure 13: Machine Learning Model of the Decoding Algorithm

Figure 14 shows the classification accuracy score of the hands vs. feet imagery movement data.

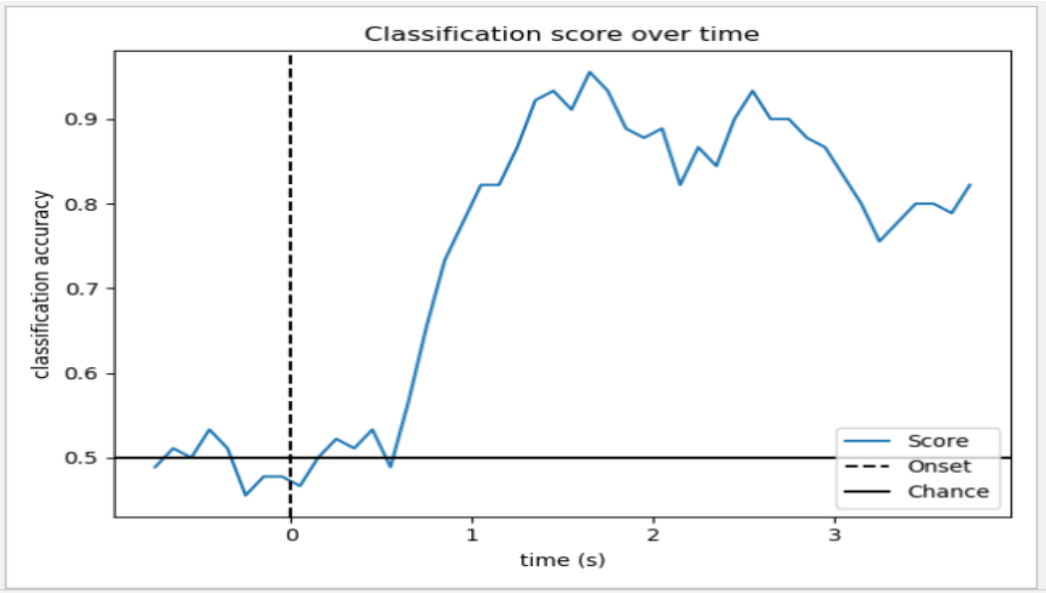


Figure 14: Classification Score of the Hands Vs. Feet Motor Imagery Movement Data

Common Spatial Pattern of Motor Imagery Movement

The feature calculates the common spatial pattern of motor imagery movement using a color bar. The Electrocorticography (ECoG) dataset was taken from BCI Competition III. The experiment was designed to record the ECoG signal on the right motor cortex of the subject. In this recording experiment, the person had to imagine that he/she was moving his/her small finger of the left hand (pinky) or tongue according to the cue displayed on the screen. The data contains 278 trials of training data with labels and 100 trials of test data in which labels are unknown. Each trial included either an imagined finger or tongue movement and was recorded for 3 seconds. The ECoG data was in the microvolts range and was sampled in 1 kHz sampling frequency. As a visual cue is shown to the subject, there could be a possibility of the visual evoked potential in the recordings. That is why the recording was taken after 0.5 s of the visual cue's end (Lal, et al., 2004).

This algorithm performs offline data processing, which means the complete data set is available before filtering. This offline processing allows for a non-causal filtering approach. Although Infinite Impulse Response (IIR) filters do not have a linear phase, the zero-phase digital filtering can eliminate the nonlinear phase distortion. The zero-phase transfer function has no phase component; however, it must be a non-causal filter. A non-causal forward and backward filter is used for the analysis of the dataset. A classical Butterworth IIR filter generates the filter coefficients. After forward filtering, the filtered sequence is time-reversed. Then it runs back through the same filter and is reversed again. The combined filter performs zero-phase filtering, and the final output sequence has zero phase distortion. So, the forward and backward filter can reduce transitional effects at the beginning and end of the signal. The common spatial

pattern method is adopted for both training and test data. Figure 15 shows the common spatial pattern of the pinky vs. tongue imagery movement ECoG data.

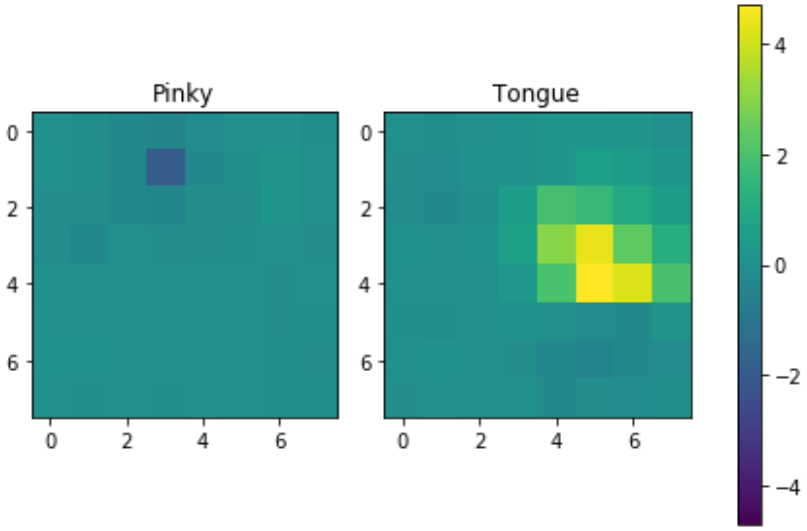


Figure 15: Common Spatial Pattern of Pinky vs. Tongue Movement

Pinky Vs. Tongue Imagery Movement Decoding from ECoG

Support Vector Machine Classifier is used to classify the pinky vs. tongue movement. The linear kernel is selected for the support vector machine. The dataset was taken from BCI competition III, where the winner achieved 91% accuracy. This method achieves an accuracy of 94 % for that data set, comparing the predicted labels with the actual labels. Finally, a confusion matrix is plotted shown in Figure 16.

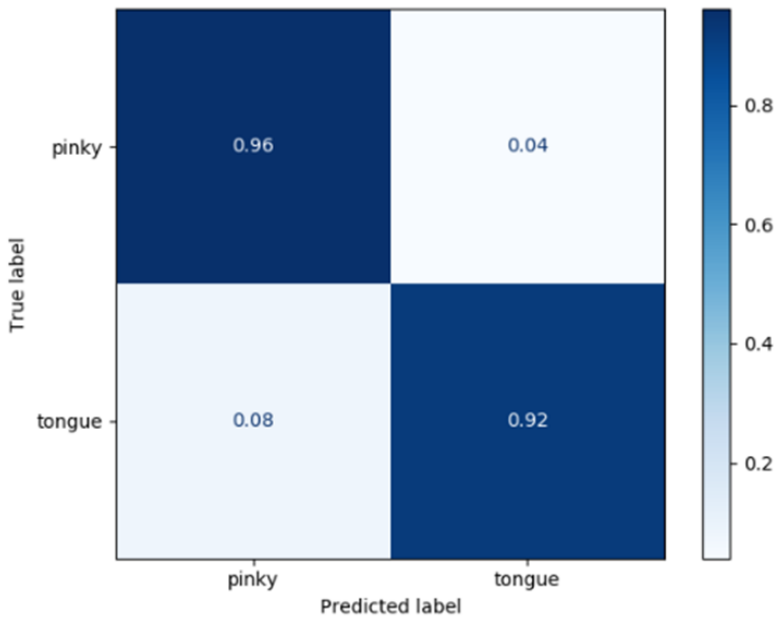


Figure 16: Confusion Matrix of Pinky vs. Tongue Imagery Movement Decoding

Sleep Signal Analysis

While our body is disconnected from the environment during sleep, but our brain remains active. Electroencephalography (EEG) can measure electrical signals that the brain produces while sleeping. These signals are in different magnitudes and frequency ranges. They can be categorized in Delta (<4 Hz), Theta (4-7 Hz), Alpha (8-13 Hz), Beta (14-30 Hz), and Gamma (>30 Hz) rhythms depending on their frequency spectra.

Sleep stages can be divided into two major parts: Non-rapid Eye Movement sleep (NREM) and Rapid Eye Movement (REM) sleep. The NREM stage's EEG signal frequency is less than 30 Hz, and the REM stage is greater than 30 Hz. Most of the portions of the sleep are NREM sleep. K-Complex, sleep spindle, and slow-wave constitute the NREM sleep part. K-complexes are defined as brief negative waves followed by a positive component and a final negative peak (Combrisson, et al., 2017). K-complex occurs in the 2nd stage of NREM sleep.

Non-rapid Eye Movement sleep (NREM) can be further divided into three parts: stage 1 (N1), stage 2 (N2), and stage 3 (N3). In stage 1, subjects are relaxed and have slow eye movements. EEG signal shows Alpha rhythm (8-13 Hz) in stage 1. This stage appears when the eyes are closed and disappears at eye-opening. EEG recordings in stage 2 show theta activity (4-7 Hz). Sleep spindles and k complexes begin to occur in this stage. Stage 3 is the deepest part of NREM sleep; it is also known as slow-wave sleep (SWS). This stage consists of the delta rhythm, which is less than 4 Hz. Figure 17 shows the different stages of sleep, known as hypnogram. The EEG, EOG, EMG recordings are visually scored to get hypnogram data. Qualitative data, such as the duration of each sleep stage, can be measured from a hypnogram. The hypnogram consists of Wake, N1, N2, N3, REM, and artifacts.

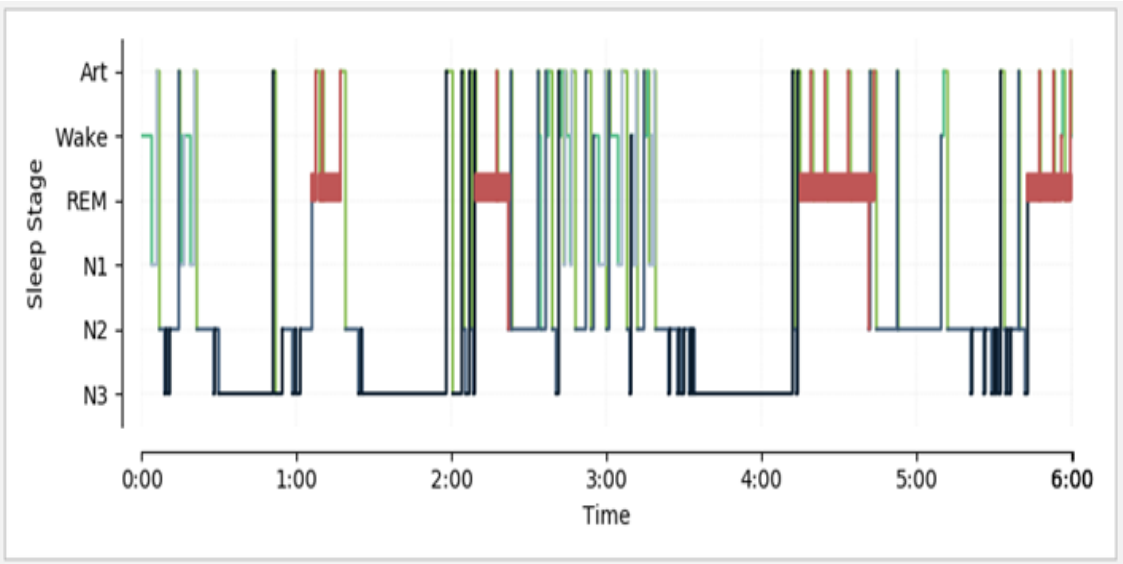


Figure 17: Sleep Stages Hypnogram

Studies suggest that adequate sleep hours play a significant role in learning and memory consolidation. Memory consolidation is defined as saving new information in memory. Consolidation during sleep brings changes in-memory representations quantitatively and

qualitatively (Diekelmann & Born, 2010). Autism is also associated with abnormalities in sleep patterns (Tessier, et al., 2015).

Slow Wave Computation

Slow-wave oscillation refers to the neural activity of synchronized slow oscillations. Slow-wave sleep is found in non-rapid eye movement stage 3 sleep (N3), the deepest sleep stage. The EEG signal of slow-wave has characteristics of high amplitude and low frequency. Delta rhythm, prominent in young adults, is found in slow-wave sleep. The frequency of delta waves ranges from 0.5 Hz to 4 Hz.

Slow-wave oscillation is comprised of a hyperpolarized phase and a depolarized phase. The hyperpolarization phase is known as the downstate. All cortical neurons are silent during that period. The depolarization phase, which is known as the upstate, comes after the hyperpolarization state. At this state, the thalamocortical system is busy with synaptic activity. Cortical neurons have increased firing rates during slow-wave oscillation are as high as during waking or REM sleep. The membrane potential fluctuations are also higher than wakefulness (Steriade & Timofeev, Natural waking and sleep states: a view from inside neocortical neurons, 2001). Slow waves are generated due to the intrinsic properties of thalamocortical neurons. Timofeev et al.'s in vivo study show the origin of slow-wave activity due to the interplay of intrinsic currents (Timofeev, Grenier, Bazhenov, Sejnowski, & Steriade, 2000).

The algorithm used here to identify slow waves is based on the 'Massimini2004' algorithm (Massimini, Huber, Ferrarelli, Hill, & Tononi, 2004). Massimini et al. designed an automatic detection algorithm for slow-wave detection. In this algorithm, the scalp is divided into four non-overlapping gray areas. Each area consists of several electrodes. The recorded signals of these electrodes of each region are averaged. The detection algorithm applied some

criteria over these four locally averaged signals. The first criterion is that the slow-wave portion should have two zero crossings separated by more than 300 ms but less than 1000 ms. Secondly, the slow-wave's negative peak voltage between the negative and positive zero crossings should be less than -80 microvolts. Thirdly, the peak to peak amplitude should be higher than 140 microvolts. When any of these four averaged signals pass all the criteria, the algorithm counts the event as a slow wave oscillation. In Figure 18, the left portion shows the four gray areas consisting of electrodes using the 10-20 system. The figure's right side presents the three criteria (a, b, c) for detecting slow waves.

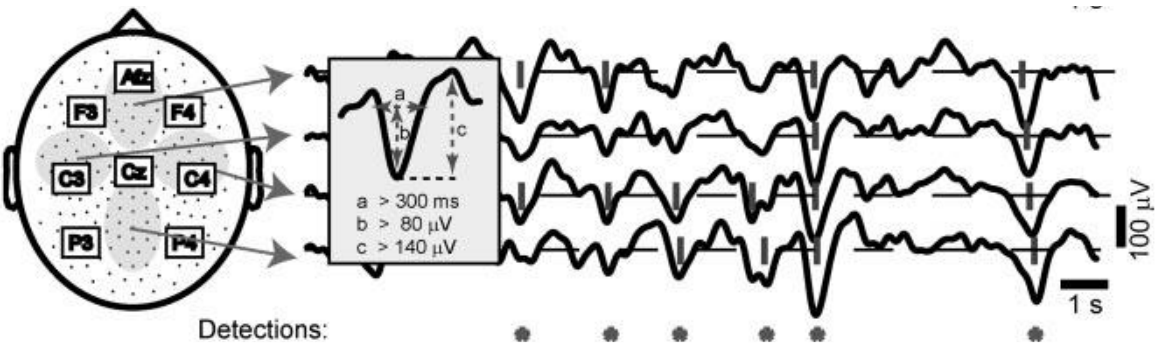


Figure 18: Slow Wave Detection Method

This algorithm detects slow waves on a single-channel EEG signal. The sleep is recorded for 6 hours in Cz, Fz, and Pz channels. The data is sampled at 100 Hz. A 30 seconds epoch from the Fz channel is taken for analysis. The slow-wave detection method uses a butter-worth filter as a detection filter. The Bandpass frequency is set to 0.1 Hz - 4 Hz. The duration for the trough is set to 0.25 s to 1 s. The negative trough amplitude is set to -80 microvolts, and peak to peak magnitude is kept at 140 microvolts. When a zero crossing is found with a subsequent peak, the algorithm adds it to an event. The algorithm checks whether the event is within the amplitude limit. The event is discarded if another zero crossing is not found within the set duration or the

peak to peak amplitude is less than the set value. When the event passes all criteria, the event is counted as a slow wave. The computed slow waves are shown in Figure 19. A total of 21 instances of slow-wave are found during the 30 seconds epoch of non-rapid eye movement sleep.

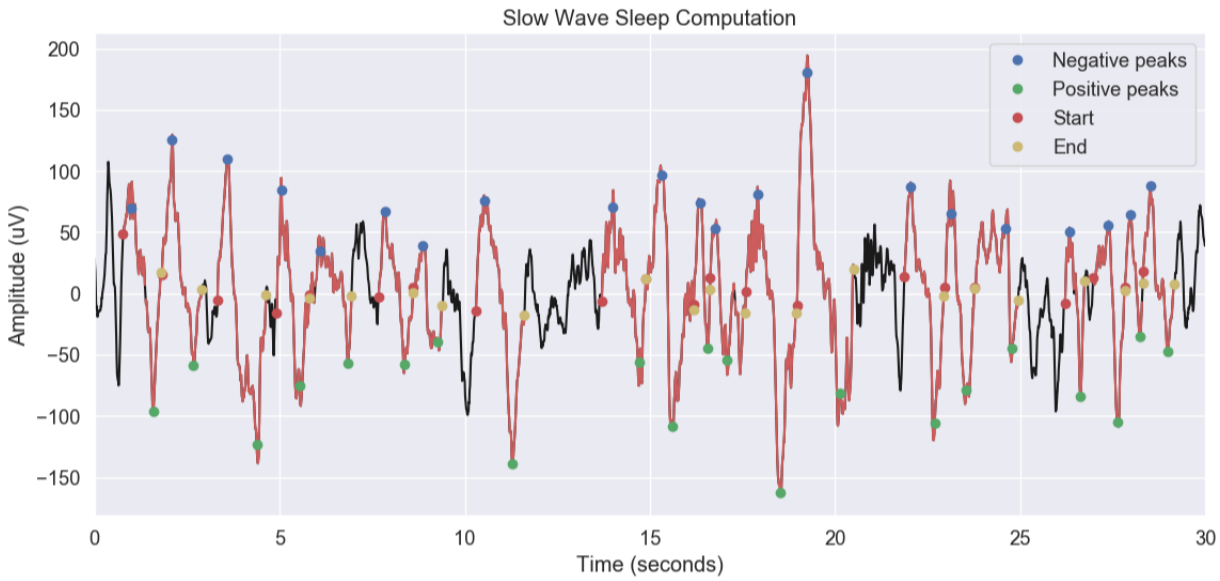


Figure 19: Computed Slow Waves from EEG Signal

Sleep Spindle Computation

Sleep spindles are defined as bursts of neural activity in 12-14 Hz frequency (Rechtschaffen & Kales, 1968). Thalamic neurons generate spindle waves. Sleep spindles have a profound impact on cognitive learning and memory. Thalamocortical and neuro-modulatory dysfunction, which occurs in schizophrenia, is linked to sleep spindle alteration. Research conducted on schizophrenic patients shows an unusual reduction in sleep spindle activity. Sleep spindle number, amplitude, duration have decreased significantly in schizophrenic patients (Ferrarelli, et al., 2007).

The automatic sleep spindle detection algorithm implemented here uses continuous wavelet transform followed by amplitude threshold and duration criteria. The frequency, power

threshold, and duration parameters are obtained from previous literature (Devuyst, Dutoit, Stenuit, & Kerkhofs, 2011) (Huupponen, et al., 2007) (Huupponen, et al., 2000). At first, the signal is wavelet transformed by convoluting with a Morlet wavelet. The central frequency band is kept at 12–14 Hz. After performing the wavelet transformation, the amplitude part is compared with the amplitude threshold. Here,

$$\text{Amplitude threshold} = \text{Mean amplitude} + 2.0 * \text{standard deviation of amplitude} \quad (4)$$

The time indices are computed where the signal amplitude exceeds the threshold. The sum of the absolute power of delta (<4Hz), theta (4–8Hz), alpha (8–12Hz), sigma (12–16Hz) waves are computed. Then each wave is divided by the resulting sum of absolute power so that the total sum of powers of these four waves in each instant is equal to 1. These resulting values are called normalized powers of the frequency bands. The normalized power of the Sigma band indicates the relative spindle power, which is obtained using the following equation.

$$\text{Normalized power of Sigma band} = \frac{\int_{12}^{16} S(f,t)df}{\int_{0.5}^{40} S(f,t)df} \quad (5)$$

Where S = Spectrogram of the signal

Similar to amplitude thresholding, a power threshold is also applied using the Sigma band's normalized power. Time indices are calculated where the normalized power exceeds the power threshold (= 0.2). Finally, a criterion is used to remove the events with a duration of <0.5 seconds or >2 seconds. Figure 20 shows the detected spindles from EEG data.

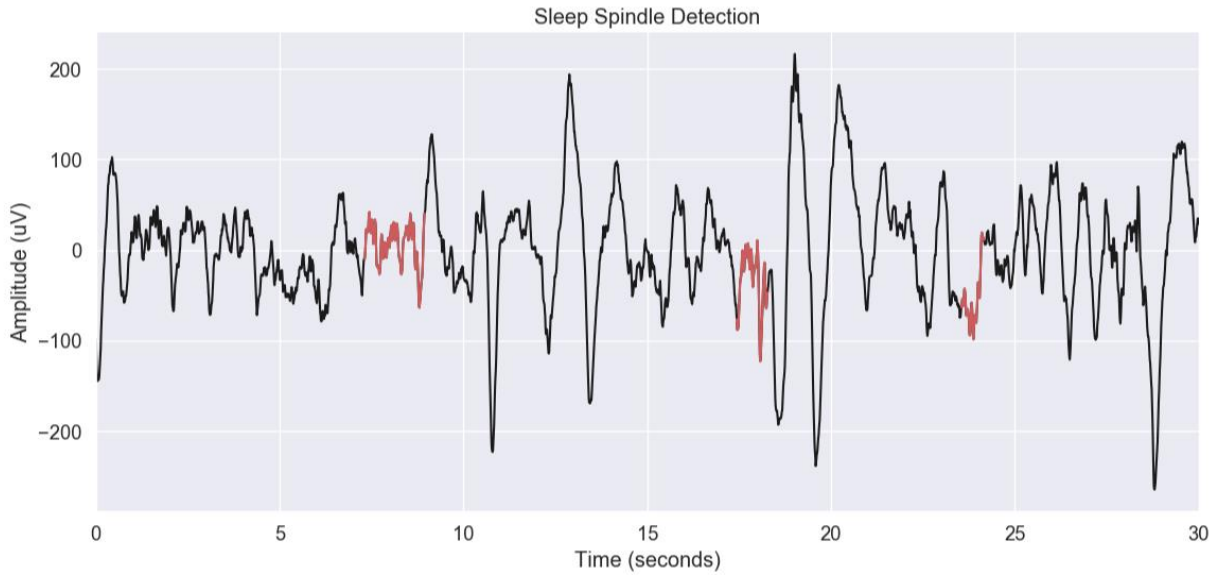


Figure 20: Spindle Detection from EEG

The algorithm detected three spindles during non-rapid eye movement stage-3 sleep. The average duration of spindles is 1.03 seconds. The first spindle starts at 7.31 second and ends at 8.94 seconds, the second spindle starts at 17.43 second and ends at 18.27 second, and the third spindle starts at 23.54 second and ends at 24.16 seconds.

Rapid Eye Movement Detection

The potentials generated due to Eye Movement while sleeping are two types: one consists of slower components known as slow eye movement (SEM), another consists of fast movements known as Rapid Eye Movements (REM). REM occurs after NREM sleep. During REM sleep, our brain is found as active as an awake condition. The EEG signal shows high Gamma rhythmic (>30 Hz) frequency at the REM stage. It is perceived that dream happens at this REM stage. A human spends 5-6% of his life in REM sleep on average. REM sleep has a deep impact on declarative and procedural memory consolidation. Declarative memory is defined as learning fact-based information, for example, the recipe of a steak. The memory-related to how to do a

specific task is known as procedural memory, like playing a musical instrument. REM performs synaptic consolidation through specific neuromodulation patterns and electric field oscillations (Diekelmann & Born, 2010).

Electrooculogram (EOG) signal measures the rapid eye movement during REM sleep. The duration of the EOG dataset is 50 seconds. The data was sampled at 256 Hz. As manually counting REM sleep is a time consuming and subjective process, an automated detection technique is used for REM. This REM detection method needs both left outer canthus EOG (LOC) and right outer canthus EOG (ROC) data. The method filters the LOC and ROC signal and performs thresholding on the negative product of LOC and ROC data (Agarwal, Takeuchi, Laroche, & Gotman, 2005). The threshold value is set to 10 $(\mu V)^2$. The frequency range for REM is set to 0.5 to 5 Hz. The output returns the REM characteristics, i.e., start, end, duration, etc. The amplitudes of LOC and ROC at REM peak are also detected. A total of 23 REM instances are detected. REM starts at 8.96 s and ends at 49.91 s. The duration of the REM parts ranges from 0.3 to 1.45 seconds. The computed rapid-eye movements are shown in Figure 21.

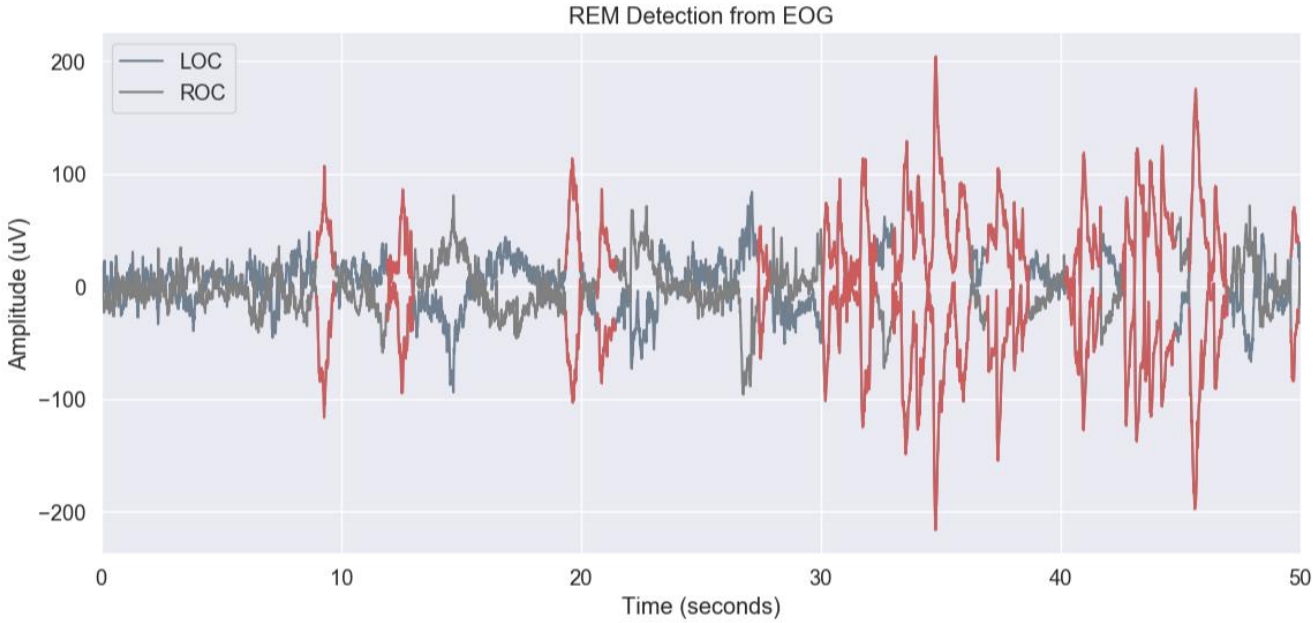


Figure 21: REM Detection from Electrooculography Data

Receptive Field Prediction

The receptive field is a part of sensory space that shows the neural response when stimulated. Receptive fields are known for neurons of the auditory system, the somatosensory system, and the visual system. The human auditory system can detect and identify the sound of 20 to 20,000 Hz range. The brain can process the speech signal continuously in a natural environment. If the stimulus and measured response are known, then it is possible to predict the measured response from the stimulus by computing a response function (Brodbeck, Presacco, & Simon, Neural source dynamics of brain responses to continuous stimuli: Speech processing from acoustics to comprehension, 2018). When an acoustic signal enters the human ear, it undergoes a complex series of transform. The vibration of the acoustic signal stimulates the Cochlea to produce nerve impulses. The signal is then transformed into a spectrogram of various frequency ranges (Yang, Wang, & Shamma, 1992). The auditory nervous system maps these frequency bands into the neural response. This process is a complex version of the Temporal

Response Function and requires multiple variables. Therefore, it can be represented by a multivariate Temporal Response Function (mTRF).

Temporal Response Function (TRF) is a filter that relates the neural response to sensory stimuli. EEG and MEG analysis are using TRF widely recently. Temporal Response Function (TRF) is modeled by developing a relationship between instantaneous neural response r(t,n) to stimulus property s(t). If we take the TRF as $\mathrm{w}(\tau, \mathrm{n})$,

$$\mathrm{r}(\mathrm{t}, \mathrm{n}) = \sum_{\tau} \mathrm{w}(\tau, \mathrm{n})\mathrm{s}(\mathrm{t} - \tau) + \epsilon(\mathrm{t}, \mathrm{n}) \tag{6}$$

Where, $\epsilon(\mathrm{t}, \mathrm{n}) =$ residual response.

Temporal Response Function is calculated by reducing the mean squared error between the actual neural response, $\mathrm{r}(\mathrm{t}, \mathrm{n})$ and predicted natural response, $\hat{\mathrm{r}}(\mathrm{t}, \mathrm{n})$.

$$\mathrm{Min}\ \epsilon(\mathrm{t}, \mathrm{n}) = \sum_{\mathrm{t}} [\mathrm{r}(\mathrm{t}, \mathrm{n}) - \hat{\mathrm{r}}(\mathrm{t}, \mathrm{n})]^2 \tag{7}$$

Sensory neurons work as stimulus-response transducers for sensory stimulation. Sensory neurons sum up the signals from different receptive fields and stimuli from a different time. This neurophysiological process is known as reverse correlation (Ringach & Shapley, 2004). The TRF is calculated by solving the equation using reverse correlation (DeBoer & Kuyper, 1968). Reverse correlation is solved by the following matrix operation, which is the ordinary least square equation.

$$w = (S^T S)^{-1} S^T r \tag{8}$$

Where, S = lagged time series of the stimulus property, s

$$S = \begin{bmatrix} s(1-\tau_{min}) & s(-\tau_{min}) & \cdots & s(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & s(1) & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & s(1) \\ s(T) & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & s(T) & \cdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & 0 & \cdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & s(T) & s(T-1) & \cdots & s(T-\tau_{max}) \end{bmatrix} \quad (9)$$

TRF is calculated throughout maximum and minimum time lags. The time window is defined as $\tau_{window} = \tau_{max} - \tau_{min}$. A $\tau_{window} \times N$ matrix will express the TRF. For multivariate analysis, frequency spectrum f = 1…. F will be added as a variable. So, speech signal spectrogram is represented by s(t, f) and mTRF by $w(f, \tau, n)$ respectively. The resulting mTRF becomes $F\tau_{window} \times N$ dimension matrix (Crosse, Di Liberto, Bednar, & Lalor, 2016)

Speech Envelope Reconstruction

Non-invasive studies can be used for speech processing applications to various hearing and language disorders like Dyslexia and Attention-deficit/hyperactivity disorder (ADHD) (Guttorm, et al., 2005) (Johnstone, Barry, & Clarke, 2013). In this feature, a speech envelope has been reconstructed from EEG. The stimulus activity of neurons is predicted from the EEG data. The decoding model establishes a relation between the EEG and speech signal. It provides a connection between the neural response and sensory stimulus by creating a stimulus reconstruction model.

This experiment's dataset is 128 channel EEG responses of human subjects while listening to an audiobook continuously. The subjects listened to both the natural speech and time-reversed version of the audiobook (Di Liberto, O'Sullivan, & Lalor, 2015). Total 28

segments of audio signals, each having 155s, were used as stimuli. The EEG data is sampled at 512 Hz.

At first, the speech dataset and EEG sensors' locations are loaded. Ridge regression model is used as a base estimator. Ridge regression is a type of linear regression. But the coefficients of Ridge regression are calculated by ridge estimator instead of ordinary least squares estimator. The ridge estimator is analogous to the ordinary least square equation with a regularization parameter λ added to the loss function.

$$w = (S^T S + \lambda I)^{-1} S^T r \tag{10}$$

Regularization, also known as a penalty, minimizes the variance of the estimates by introducing a small amount of bias. Although ridge regression is slightly biased, it has a lower variance than the ordinary least square estimator. So, the ridge estimation method provides improved efficiency in estimation problems. In the case of λ = 0, the ridge estimator becomes the ordinary least square method.

The regularization parameter λ is selected such that it generates the lowest Mean Squared Error. The Ridge regression model is fitted using the training EEG signal and training speech data. 3-Fold cross-validation is performed. Finally, the speech signal envelope is predicted from the test EEG signal. The actual stimulus signal and predicted envelopes are plotted side by side in Figure 22.

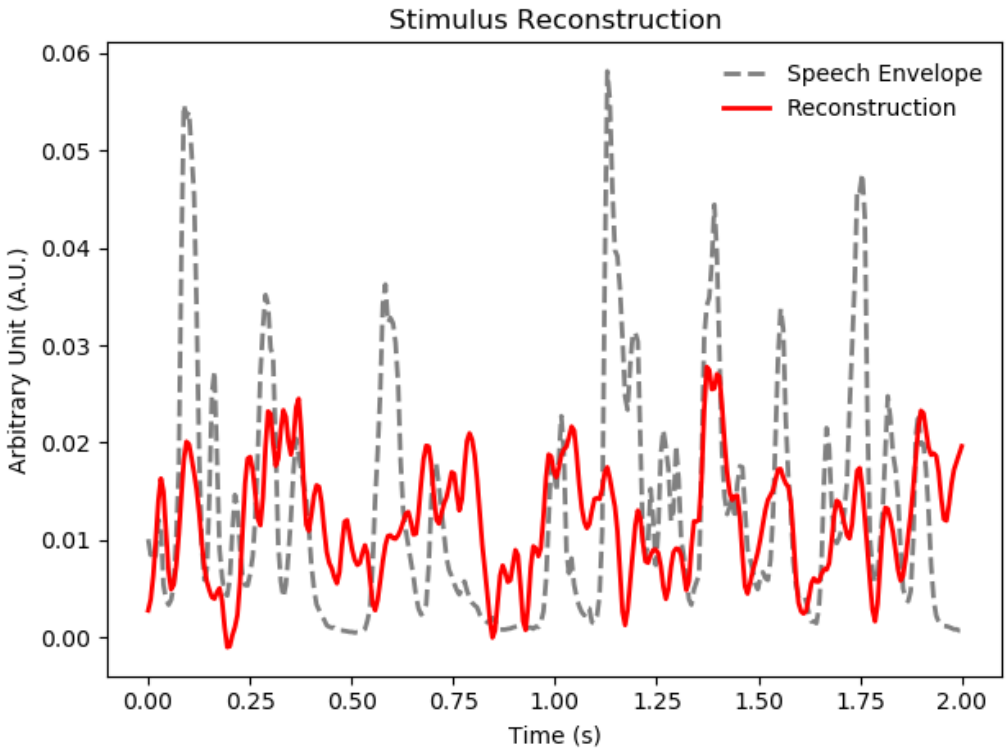


Figure 22: Speech Envelope Reconstruction from EEG

SSVEP Epoch Classification

This feature classifies epochs of EEG evoked frequency data. The data is visual evoked EEG signal while displaying human faces to the subject.

When the brain is given stimulation, i.e., light or sound, the brain responds by electrical activity. The response is known as an evoked response, and the generated potential is called evoked potential. Visual evoked potential, especially the steady-state visual evoked potential (SSVEP), is one of the most used signals of BCI analysis. Constant frequency visual stimulus signal affects the EEG signal coming from the brain cortex. SSVEP is the reflection of visual stimulation in brain activity.

The data used in this program was generated through visual stimulation using face images. The data consists of two classes. The first data class is visual stimulation at 1.2 Hz, and the second data class is random visual stimulation.

The bandpass frequency is set at 0.5 Hz-30 Hz. Support vector machine classifier is trained using training data and labels. The average signal-to-noise ratio for each epoch is used as the feature. Then the trained classifier is applied to test data. Finally, a confusion matrix is presented in Figure 23, which shows the accuracy of the prediction.

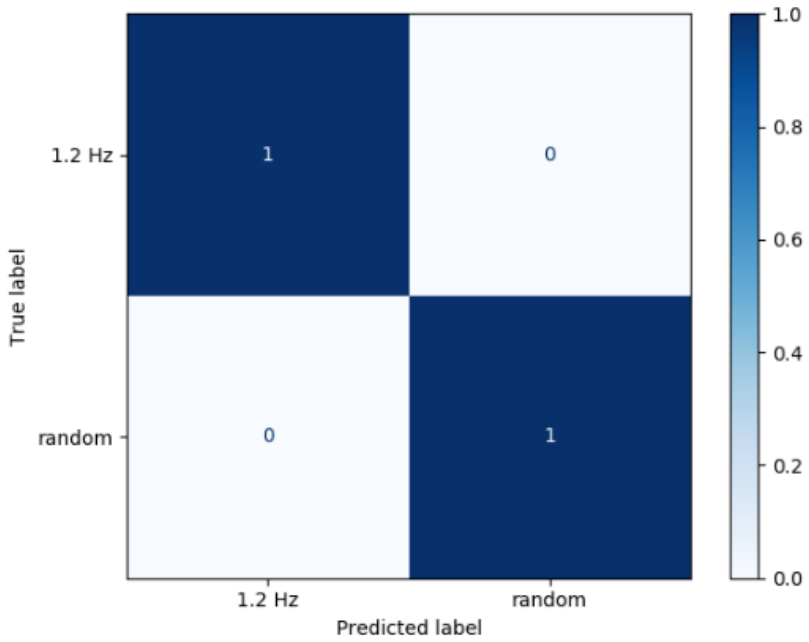


Figure 23: Confusion Matrix of Epoch Classification

CHAPTER VI

CONCLUSION

In this work, a Brain-Computer Interface toolbox, MEDUSA, is presented. MEDUSA can perform both invasive and non-invasive data analysis. The toolbox supports Electroencephalography (EEG), Electrocorticography (ECoG), and Electrooculography (EOG) data. Loading and reading data files from standard file formats, i.e., European Data Format (.edf) and .mat, are supported by the toolbox. The MEDUSA toolbox is tested using several publicly available data such as PhysioNet and Brain-Computer Interface competition III datasets. This project's significance is that it provides a common platform for neural signal processing for experimental scientists. This research connects the neuroscientist with brain signal analysis algorithms through an easy to use and user-friendly graphical user interface.

REFERENCE

Agarwal, R., Takeuchi, T., Laroche, S., & Gotman, J. (2005). Detection of rapid-eye movements in sleep studies. *IEEE Transactions on Bio-Medical Engineering, 52*(8), 1390–1396.

Amiri, S., Fazel-Rezai, R., & Asadpour, V. (2013). A Review of Hybrid Brain-Computer Interface Systems. *Advances in Human-Computer Interaction*, 1.

Blankertz, B., Dornhege, G., Krauledat, M., Muller, K., & Curio, G. (2007). The non-invasive Berlin Brain-Computer Interface: Fast Acquisition of Effective Performance in Untrained Subjects. *NeuroImage, 37*(2), 539-550.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. (2008). Optimizing spatial filters for robust EEG single trial analysis. *IEEE Signal Processing Magazine, 25*(1), 41-56.

Brodbeck, C., Brooks, T. L., Das, P., & Reddigari, S. (2019). Eelbrain: 0.30.

Brodbeck, C., Presacco, A., & Simon, J. (2018). Neural source dynamics of brain responses to continuous stimuli: Speech processing from acoustics to comprehension. *Neuroimage, 172*, 162-174.

Choi, K., & Min, B.-K. (2015). Future Directions for Brain-Machine Interfacing Technology. In *Lee SW., Bülthoff H., Müller KR. (eds) Recent Progress in Brain and Cognitive Engineering. Trends in Augmentation of Human Performance* (pp. 3-18). Netherlands: Springer.

Combrisson, E., Vallat, R., Eichenlaub, J., O'Reilly, C., Lajnef, T., Guillot, A., . . . Jerbi, K. (2017). Sleep: An Open-Source Python Software for Visualization, Analysis, and Staging of Sleep Data. *Frontiers in Neuroinformatics, 11*, 60.

Congedo, M., Korczowski, L., Delorme, A., & Lopes da Silva, F. (2016). Spatio-temporal common pattern: A companion method for ERP analysis in the time domain. *Journal of Neuroscience Methods, 267*, 74-88.

Constable, P., Bach, M., Frishman, L., Jeffrey, B., & Robson, A. (Feb 2017). ISCEV Standard for clinical electro-oculography (2017 update). *Doc Ophthalmol, 134*(1), 1-9.

Crosse, M., Di Liberto, G., Bednar, A., & Lalor, E. (2016). The Multivariate Temporal Response Function (mTRF) Toolbox : A MATLAB Toolbox for Relating Neural Signals to Continuous Stimuli. *Frontiers in Human Neuroscience*.

Das, S., Tripathy, D., & Raheja, J. (2019). A Review on Algorithms for EEG-Based BCIs. In *Real-Time BCI System Design to Control Arduino Based Speed Controllable Robot Using EEG* (pp. 25-56). Singapore: SpringerBriefs in Applied Sciences and Technology.

DeBoer, E., & Kuyper, P. (1968). Triggered Correlation. *IEEE Transaction of Biomed Eng, 15*, 169–179.

Decety, J. (1996). Do imagined and executed actions share the same neural substrate? *Cognitive Brain Research* , 87-93.

Devuyst, S., Dutoit, T., Stenuit, P., & Kerkhofs, M. (2011). Automatic Sleep Spindles Detection – Overview and Development of a Standard Proposal Assessment Method. *33rd Annual International Conference of the IEEE EMBS.* Boston, Massachusetts, USA.

Di Liberto, G. M., O'Sullivan, J. A., & Lalor, E. C. (2015). Low-Frequency Cortical Entrainment to Speech Reflects Phoneme-Level Processing. *Current biology, 25*(19), 2457-2465.

Diekelmann, S., & Born, J. (2010). The memory function of sleep. *Nature Reviews Neuroscience, 11*, 114–126.

Estermann, B. (2017). *Evaluation of Low-Cost EEG Hardware in a Motor Imagery Based Brain Computer Interface.* Zurich.

Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., & Acharya, U. R. (2018). Deep learning for healthcare applications based on physiological signals: A review. *Computer Methods and Programs in Biomedicine, 161*, 1-13.

Ferrarelli, F., Huber, R., Peterson, M., Massimini, M., Murphy, M., & Riedner, B. (2007). Reduced Sleep Spindle Activity in Schizophrenia Patients. *The American Journal of Psychiatry*, 483–492.

Florea, C., Florea, L., & Vertan, C. (2018). Computer Vision for Cognition. In *Computer Vision for Assistive Healthcare.*

Fouad, M. M., Amin, K. M., El-Bendary, N., & Hassanien, A. E. (2015). Brain Computer Interface: A Review. *Brain-Computer Interfaces, Intelligent Systems Reference Library, Springer, 74*, 3-30.

Goldberger, A. L., A. N. Amaral, L., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., . . . Stanley, H. E. (2000). Components of a New Research Resource for Complex Physiologic Signals. *PhysioBank, PhysioToolkit, and PhysioNet, 101*(23).

Graimann, B., Allison, B., & Pfurtscheller, G. (2009). Brain–Computer Interfaces: A Gentle Introduction. In *Brain-Computer Interfaces* (pp. 1-27). Berlin: Springer.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., . . . Hämäläinen, M. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience, 7*, 267.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., . . . Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. *Neuroimage, 86*, 446-460.

Greenwood, D. (1990). A cochlear frequency-position function for several species--29 years later. *Journal of the Acoustical Society of America, 87*(6), 2592-2605.

Guttorm, T., Leppänen, P., Poikkeus, A., Eklund, K., Lyytinen, P., & Lyytinen, H. (2005). Brain event-related potentials (ERPs) measured at birth predict later language development in children with and without familial risk for dyslexia. *Cortex, 41*(3), 291-303.

Guzman, S. J., Schlögl, A., & Schmidt-Hieber, C. (2014). Stimfit: quantifying electrophysiological data with Python. *Frontiers in Neuroinformatics, 8*, 16.

Hassanien, A., & Azar, A. (2015). Brain–Computer Interfaces: Current Trends and Applications. *Springer, 74*.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science, 293*(5539), 2425–2430.

Huupponen, E., Gómez-Herrero, G., Saastamoinen, A., Värri, A., Hasan, J., & Himanen, S.-L. (2007). Development and comparison of four sleep spindle detection methods. *Artificial Intelligence in Medicine, 40*(3), 157-170.

Huupponen, E., Värri, A., Himanen, S., Hasan, J., Lehtokangas, M., & Saarinen, J. (2000). Optimization of sigma amplitude threshold in sleep spindle detection. *Journal of Sleep Research, 9*(4), 327–334.

Jochumsen, M., Navid, M. S., Nedergaard, R. W., Signal, N., Rashid, U., Hassan, A., . . . Niazi, I. K. (2019). Self-Paced Online vs. Cue-Based Offline Brain-Computer Interfaces for Inducing Neural Plasticity. *Brain Sciences, 9(6), 127., 9*(6), 127.

Johnstone, S., Barry, R., & Clarke, A. (2013). Ten years on: a follow-up review of ERP research in attention-deficit/hyperactivity disorder. *Clinical Neurophysiology, 124*(4), 644-657.

Lal, T., Hinterberger, T., Widman, G., Schr¨oder, M., Hill, N., Rosenstiel, W., . . . Birbaumer, N. (2004). Methods Towards Invasive Human Brain Computer Interfaces. *17th International*

Conference on Neural Information Processing Systems. Vancouver, British Columbia, Canada.

Lemm, S., Blankertz, B., Curio, G., & Muller, K. R. (2005). Spatio-Spectral Filters for Improving the Classification of Single Trial EEG. *IEEE Transactions on Biomedical Engineering, 52*(9), 1541-1548.

Massimini, M., Huber, R., Ferrarelli, F., Hill, S., & Tononi, G. (2004). The sleep slow oscillation as a traveling wave. *The Journal of Neuroscience, 24*(31), 6862-6870.

McGrogan, N., Braithwaite, E., & Tarassenko, L. (2001). BioSleep: a comprehensive sleep analysis system. *Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Istanbul, Turkey.

Mourão-Miranda, J., Oliveira, L., Ladouceur C, .. D., Marquand, A., Brammer, M., Birmaher, B., . . . Phillips, M. L. (2012). Pattern Recognition and Functional Neuroimaging Help to Discriminate Healthy Adolescents at Risk for Mood Disorders from Low Risk Adolescents. *PLoS ONE, 7*(2).

Müller-Putz, G., Breitwieser, C., Cincotti, F., Leeb, R., Schreuder, M., Leotta, F., . . . Millán, J. R. (2011). Tools for Brain-Computer Interaction: A General Concept for a Hybrid BCI. *Frontiers in Neuroinformatics, 5*, 30.

Navarro, R. B., Vázquez, L. B., & López-Guillén, E. (2018). EOG-based wheelchair control. In *Smart Wheelchairs and Brain-Computer Interfaces.*

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors, 12*(2), 1211-1279.

Nunez, P., & Srinivasan, R. (2006). *Electric Fields of the Brain: The Neurophysics of EEG 2nd edition.* New York: Oxford University Press.

Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J. (2011). FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience*.

Passarotti, A., Sweeney, J., & Pavuluri, M. (2009). Neural correlates of incidental and directed facial emotion processing in adolescents and adults. *Social Cognitive and Affective Neuroscience, 4*(4), 387–398.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Perego, P., Maggi, L., Parini, S., & Andreoni, G. (2009). BCI++: A New Framework for Brain Computer Interface Application. *18th International Conference on Software Engineering and Data Engineering.* Las Vegas.

Pfurtscheller, G., Guger, C., & Ramoser, H. (1999). EEG-based brain-computer interface using subject-specific spatial filters. *Engineering applications of bio-inspired artificial neural networks, Lecture Notes in Computer Science, 1607*, 248-254.

Phillips, M., Ladouceur, C., & W.C., D. (2008). A neural model of voluntary and automatic emotion regulation: implications for understanding the pathophysiology and neurodevelopment of bipolar disorder. *Molecular Psychiatry , 13*, 833–857.

Ramadan, R., & Vasilakos, A. (2017). Brain Computer Interface: Control Signals Review. *Neurocomputing, 223*, 26-44.

Rechtschaffen, A., & Kales, A. (1968). *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects.* Washington DC: United States Government Printing Office.

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., . . . Lecuyer, A. (2010). OpenViBE: An Open-Source Software Platform to Design, Test, and Use Brain–Computer Interfaces in Real and Virtual Environments. *Presence, 19*(1), 35-53.

Ringach, D., & Shapley, R. (2004). Reverse correlation in neurophysiology. *Cognitive Science , 28*(2), 147–166.

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., & Wolpaw, J. R. (2004). BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering, 51*(6), 1034-1043.

Schiff, S. J., Aldroubi, A., Unser, M., & Sato, S. (1994). Fast wavelet transformation of EEG. *Electroencephalography and Clinical Neurophysiology, 91*(6), 442-455.

Schlögl, A., & Brunner, C. (2008). BioSig: A Free and Open Source Software Library for BCI Research. *Computer, 41*(10), 44-50.

Schlögl, A., Lee, F., Bischof, H., & Pfurtscheller, G. (2005). Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of Neural Engineering, 2*(4).

Sellers, E., Krusienski, D., McFarland, D., & Wolpaw, J. (2007). Non-Invasive Brain-Computer Interface Research at the Wadsworth Center. In *Toward Brain-Computer Interfacing* (pp. 31-42). Cambridge, MA: The MIT Press.

Sinkkonen, J., Tiitinen, H., & Näätänen, R. (1995). Gabor filters: an informative way for analysing event-related brain activity . *Journal of Neuroscience Methods, 56*(1), 99-104.

Steriade, M., & Timofeev, I. G. (2001). Natural waking and sleep states: a view from inside neocortical neurons. *Journal of Neurophysiology, 85*, 1969 –1985.

Steriade, M., Gloor, P., Llinas, R., Lopes da Silva, F., & Mesulam, M. (1990). Basic mechanisms of cerebral rhythmic activities. *Electroencephalography and Clinical Neurophysiology, 76*(6), 481 - 508.

Steriade, M., Jones, E., & Llinls, R. (1990). *Thalamic oscillations and signaling.* New York: John Wiley & Sons.

Susila, I., Kanoh, S., Miyamoto, K., & Yoshinobu, T. (2010). xBCI: a generic platform for development of an online BCI system. *IEEJ Transactions on Electrical and Electronic Engineering, 5*(4), 467-473.

Tadel, F., Baillet, S., Mosher, J., & Pantazis, D. L. (2011). Brainstorm: A User-Friendly Application for MEG/EEG Analysis. *Computational Intelligence and Neuroscience, 2011*.

Tayeb, Z., Fedjaev, J., Ghaboosi, N., Richter, C., Everding, L., Qu, X., . . . Conradt, J. (2019). Validating Deep Neural Networks for Online Decoding of Motor Imagery Movements from EEG Signals. *Sensors (Basel), 19*(1), 210.

Tessier, S., Lambert, A., Chicoine, M., Scherzer, P., Soulières, I., & Godbout, R. (2015). Intelligence measures and stage 2 sleep in typically-developing and autistic children. *International Journal of Psychophysiology, 97*(1), 58-65.

Timofeev, I., Grenier, F., Bazhenov, M., Sejnowski, T., & Steriade, M. (2000). Origin of Slow Cortical Oscillations in Deafferented Cortical Slabs. *Cerebral Cortex, 10*(12), 1185–1199.

Venthur, B., & Blankertz, B. (2012). Mushu, a free- and open source BCI signal acquisition, written in Python. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* San Diego, CA.

Venthur, B., Dahne, S., Hohne, J., Heller, H., & Blankertz, B. (2015). Wyrm: A Brain-Computer Interface Toolbox in Python. *Neuroinform, 13*(4), 471–486.

Venthur, B., Scholler, S., Williamson, J., Dahne, S., Treder, M., Kramarek, M. M., & Blankertz, B. (2010). Pyff – a pythonic framework for feedback applications and stimulus presentation in neuroscience. *Frontiers in Neuroinformatics, 4*(179).

Vidal, J. J. (1973). Toward Direct Brain-Computer Communication. *Annual Review of Biophysics and Bioengineering* , *2*, 157-180 .

Yang, X., Wang, K., & Shamma, S. (1992). Auditory Representations of Acoustic Signals. *IEEE Transactions on Information Theory*, *38*(2), 824 - 839.

Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, *33*(4), 1538-1579.

BIBLIOGRAPHICAL SKETCH

Md Hasan Anowar completed his Master's in Electrical Engineering from the University of Texas Rio Grande Valley (UTRGV) in 2020 and his Bachelor's in Electrical and Electronic Engineering from the Bangladesh University of Engineering and Technology (BUET) in 2015. His research interests include biomedical signal processing, neural engineering, and machine learning. He developed a brain-computer interface toolbox on the Python platform for EEG and ECoG signal processing for his Master's thesis. He maintained a GPA of 4.00/4.00 throughout his graduate study and was awarded "Presidential Graduate Research Assistantship" and "Margaret L. Draper Scholarship." He has four years of teaching experience as a teaching assistant and a lecturer. He is now enrolled in a Ph.D. program. Upon completing the Ph.D. degree, he plans to pursue a career in academia and research.

Permanent Mailing Address: 52/3 West Rajabazar, Panthapath, Tejgaon, Dhaka, Bangladesh, 1215. Personal Email: hasan92niloy@gmail.com