Neural data analysis platform for perception and continuous-event electrophysiology

Ross McCrann

April 19, 2022

A dissertation submitted in partial fulfilment of the requirements for the degree of MSC (Computer Science)
Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism ‘Ready Steady Write’, located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ____________________________  Date: ____________________________
Abstract

Understanding the complex underpinnings of how the human brain processes the environment around it is one of the most important goals of modern-day neuroscience. While the ability to communicate and speak is one of the most fundamentally unique aspects of being human, the mechanisms by which it is enabled on a neurological level are relatively unknown. Advancements in brain imaging technology such as electroencephalography and magnetoencephalography (EEG & MEG) have greatly empowered researchers to isolate neural activity at the various stages of speech processing. However, the unstandardised and proprietary nature of brain data collection stands as an inhibitor to the use of modern data-driven analytic techniques such as machine learning.

The work presented in this dissertation, aims to develop a web-based neural analysis platform for both expert and non-expert users to analyse neural data (e.g., brain signals measured using EEG) using publically available, high quality data to simulate experimental conditions. Such a platform will allow the rapid prototyping and development of experiments in a user friendly, graphical environment that enables the user to determine the optimal configuration of new experiments. The brain imaging data stored on the platform will use the Continuous-event Neural Data format (CND) in order to encourage standardisation and replicability of results. In turn, this will give create the opportunity to explore the potential for big data analysis and more advanced machine learning techniques on brain signals in the future.

Additionally, this dissertation begins the development of a framework for implementing MATLAB code within a lightweight web application. This framework utilises modern web technologies in order to interact with the MATLAB code-base and serves as a proof-of-concept for future applications.
Acknowledgements

My dissertation is the culmination of the amazing support I have received over the academic year and would not have been possible without the guidance and encouragement of those around me.

My sincere gratitude to my supervisor Giovanni, whom without this work would not be possible. He has been a constant source of knowledge, direction and friendship which has carried me throughout the academic year.

My warmest thanks to the entire Di Liberto Lab Team, who offered a great sense of camaraderie throughout the year, especially during our Tuesday morning discussions and pizza dinner out in Bray.

I would like to thank my parents, Mark and Rebecca for their unwavering support throughout the year.
# Glossary of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEG</td>
<td>ElectroEncephaloGraphy</td>
</tr>
<tr>
<td>MEG</td>
<td>MagnetoEncephaloGraphy</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>FMRI</td>
<td>functional Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>ECoG</td>
<td>ElectroCorticoGraphy</td>
</tr>
<tr>
<td>BCI</td>
<td>Brain-Computer Interface</td>
</tr>
<tr>
<td>TRF</td>
<td>Temporal Response Function</td>
</tr>
<tr>
<td>mTRF</td>
<td>multivariate Temporal Response Function</td>
</tr>
<tr>
<td>STRF</td>
<td>Spectro-Temporal Response Function</td>
</tr>
<tr>
<td>ERP</td>
<td>Event-Related Potentials</td>
</tr>
<tr>
<td>ERF</td>
<td>Event-Related Field</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>GFP</td>
<td>Global Field Power</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear-time Invariant</td>
</tr>
</tbody>
</table>
# Contents

## 1 Introduction
  1.1 Motivation ........................................... 1  
  1.2 Research Goals .................................... 2  
    1.2.1 Research Aim .................................. 2  
    1.2.2 Research Objectives ............................ 2  
  1.3 Dissertation Structure ............................ 3

## 2 Background
  2.1 Modelling Speech Perception ...................... 4  
  2.2 Linear systems .................................... 6  
  2.3 Encoding & Decoding Models ...................... 7  
  2.4 Modelling the Brain Response Framework .......... 8  
  2.5 Event-related Potentials (ERP) .................. 12  
  2.6 Temporal Response Function ..................... 14  
  2.7 Continuous-event Neural Data Format ............ 16

## 3 Method
  3.1 Purpose of Neural Analysis Platform ............. 20  
  3.2 Development of platform .......................... 22  
  3.3 Front-End & User Interface Mock-ups ............. 25  
  3.4 System Architecture .............................. 25  
  3.5 Application Back-end & API ...................... 26  
  3.6 Model Training & Testing ........................ 31  
    3.6.1 Univariate TRF/STRF estimation ............... 32  
    3.6.2 Multivariate TRF/STRF estimation ............ 33  
    3.6.3 Stimulus Reconstruction ...................... 33

## 4 Neural Analysis Platform Use Case Scenarios
  4.1 Scenario 1: Proposed study of cognitive development in babies using audio-visual stimuli .................. 34
4.2 Scenario 2: Study comparing patients with and without learning deficits as biomarkers for research ........................................ 38
4.3 Scenario 3: Future non-verbal patient imagined speech decoding ........ 41

5 Conclusion ........................................ 45
5.1 Project Overview .................................. 45
5.2 Future Work ........................................ 46
5.3 Closing Remarks ................................... 47

A Appendix ........................................ 53
A.1 Encoding MATLAB Code ....................... 53
A.2 Decoding MATLAB Code ....................... 62
# List of Figures

2.1 A linear system characterised by impulse response. Figure illustrates two methods of measuring impulse response. (a) measured as the response to a brief pulse or (b) the impulse response of a system. Adapted from Ringach and Shapley, 2004. ................................................................. 5

2.2 A and B, schematic diagram of the apparatus used by Erich Sutter to measure, for the first time, the full spatiotemporal receptive field of simple cells in a cat. Reproduced from experiment by Erich Sutter, 1975; figure adapted from (Ringach, 2004). ................................................................. 6

2.3 EEG signals recorded from a relaxed subject. Four seconds of data are shown from four scalp locations. The corresponding amplitude spectra reveal activity in the 8-13 Hz band. Adapted from (Nunez et al., 2006). ................. 9

2.4 Magneto-encephalography (MEG) system at the Neuro-imaging research centre in Pitie Salpetriere hospital in Paris, France. 306 sensors spread over 102 areas enable both near and far magnetic fields to be measured in addition to visualising the anatomical structure of the brain. Subject receives visual stimulation from a video(faces with different expressions). ............... 10

2.5 Electroencephalographic recordings using 128 electrodes making scalp contact with a conductive gel. One of several kinds of EEG recording methods. Adapted from (Nunez and Srinivasan, 2007). ......................... 11

2.6 Event Related Potential waveform. ........................................... 13

2.7 Temporal Response Function waveform. .................................... 15

3.1 Unified Model Language Diagram (UML) visually representing the use case of neural analysis platform for expert and non-expert users. ...................... 23

3.2 Encoding and Decoding page user interface mock-up: features output graphs based on user input, cross-validation accuracy as well as scalp topographies at varying time lags. ......................................................... 24

3.3 Multi-Encoding and Single-Lag page user interface mock-up. Button activates cross data-set testing. ......................................................... 25
3.4 Home (A) and Login page (B) user interface mock-up: (A) An informative home page explaining what the Neural Analysis Platform is, including featured use case scenarios and a registration page for the platform. (B) Requires users to login to platform using email or G-mail credentials for increased security. 26

3.5 Encoding within the neural analysis platform ................................. 27
3.6 Decoding within the neural analysis platform ................................. 28
3.7 Encoding API in Flask ....................................................... 29
3.8 Cloud Architecture using MATLAB Production Server .................. 30
3.9 Schematic of forward and backward modeling approaches implemented using mTRF Toolbox ......................................................... 31
3.10 mTRFtrain and mTRFPredict code using optimal hyperparameter (lambda) 32
3.11 Stimulus reconstruction code .................................................. 33

4.1 Scenario in which a research uses the neural analysis platform in order to conduct a preliminary study on cognitive development in babies using cartoons as a continuous audio-visual stimuli. ........................................ 34

4.2 Experiment from which the Natural Speech dataset was gathered, assessing the encoding of speech features in EEG. Adapted from Di Liberto et al., 2015 36

4.3 Scenario in which a psychologist or neuroscientist uses the EEG and the Neural Analysis Platform to aid in the diagnoses of learning deficits such as dyslexia by providing quantifiable metric based on actual brain response data compared to those without the deficit. .................................................... 38

4.4 Multi-encoding within the neural analysis platform used to compare across datasets such as in scenario 2 ................................................. 40

4.5 Scenario in which a neuroscientist uses the EEG and the Neural Analysis Platform to create a BCI in which a patient with locked-in syndrome or ALS imagined speech can be decoded to audio. ........................................ 41

4.6 Resulting decoder graphs featuring: CV Accuracy, CV Error, Reconstruction Accuracy, Reconstruction Error as a function of time lags, Brain topographies at varying time lags. ......................................................... 42

4.7 Spoken speech used to train and test neural decoding model. During imagined speech, the model trained on spoken speech is used to decode the imagined speech to neural activity. Imagined speech reconstruction is them compared to original spoken speech using STRFs. Adapted from Martin et al., 2014 . 43
1 Introduction

1.1 Motivation

The human brain has over 100 billion neurons interconnected within a complex network designed to interpret the world around us. The mechanisms enabling us to perceive our environment is made possible through multivariate cascading processes of hierarchical cognitive structures in the brain. For example, the human ability to perceive speech is enabled through the transformation of auditory speech envelopes into smaller tokens of speech such as phonemes and syllables that are processed by the brain through lexical and semantic analysis to derive meaning. Neurophysiological experiments have revealed many of the fundamental mechanisms and infrastructures underpinning the brain’s ability to perceive the environment around it. However, such investigations are limited in that they are unable to effectively reveal the deeper mechanisms of how the human brain processes anthropological stimulus in the environment such as music and speech. Additionally, most neurophysiology research on human perception has used simple, controlled experimental paradigms that are generally far removed from reality. Recent work with non-invasive neurophysiology has led to a new realistic analysis framework where neural data can be recorded and studied as participants listen to continuous natural sounds, such as speech and music.

This dissertation describes the development of a machine learning platform for both expert and non-expert users to analyse neural data (e.g., brain signals measured using electroencephalography – EEG). This platform will use publicly available EEG data to simulate the working conditions targeted by the users and rapidly determine the optimal configuration for new experiments. This project aims to bridge the gap between cutting edge neural data analysis and the limited high-quality neural data publicly available. EEG experiments on continuous natural sensory stimuli processing are generally collected by each research group using their own methodological procedures and data format. The aforementioned platform will utilise the recently created continuous-event neural format to standardise the publicly available neural data imported into the platform for use by it’s users. In this respect, the datasets will be made readily available for use in data analysis by the
wider neurocognitive community. It is hoped that this will create a foundation upon which there is greater opportunity to explore the potential for big data analysis as well as more advanced machine learning techniques on neural data.

The resulting web application will support various existing neural data analysis algorithms. The primary library that will be used to implement the algorithms is mTRF-Toolbox, supplemented with EEGLAB. Both are libraries for modelling neural data with differing use cases for both. Through the use of the platform both expert and non-expert users (doctors, psychologists, etc.) will be able to conduct cognitive experiments using electrophysiological data without the need to know how to program or configure computational environments. By conducting such neurocognitive experiments it is hoped that users will be able to conduct further research into diagnosing conditions and deficits of neurological origin and will ultimately have greater access to objective measures of cognition (e.g., attention, speech processing, memory). Such measurements have great potential in the future for many practical applications such as use as biomarkers for diagnosing neurodevelopmental deficits (dyslexia).

1.2 Research Goals

1.2.1 Research Aim

The primary aim of the dissertation is to develop an internet application that enables both expert and non-expert users to conduct cognitive experiments as predictive computational models and evaluate those predictions against electrophysiological brain responses using publically available datasets.

1.2.2 Research Objectives

The dissertation objectives are as follows:

1. Conduct a review of the relevant literature within the neuroscience field as well as ancillary fields such as psychology, medicine etc. in order to understand the current state-of-the-art for using neurophysiological data to conduct experiments.

2. Develop a web application framework that enables the processing of electrophysiological data within an online platform using advanced analysis techniques prevalent within studies in the neurocognitive domain.

3. Develop an interface for interacting with the web application framework that can be used to conduct cognitive studies and rapidly configure experiment parameters. It is imperative that the user interface is interactive and visually appealing such that it can be easily used by both expert and non-expert users.
1.3 Dissertation Structure

Chapter 2 - Background
This chapter introduces and explores the theoretical foundations of the dissertation as well as the historical origins of cognitive neuroscience emerging from neuroscience and psychology. A review of key literature in the field is presented which serve as the basis for the later chapters of the dissertation. Additionally, the state-of-the-art technology used in neurocognitive experiments is detailed as well as prevalent analytical techniques.

Chapter 3 - Implementation
Chapter 3 presents the implementation of a lightweight web application that can be used to conduct cognitive experimentation. The design process and system architecture is detailed throughout with visualisations of the resulting platform. Technical methods involved in analytical processes based on theory from the previous chapter are described in detail with reference to libraries and algorithms.

Chapter 4 - Use Case Scenarios
This chapter demonstrates the use of the resulting neural analysis platform within the context of possible real world situations that the platform was been designed for. In total, three hypothetical use cases scenarios are presented in detail including background information with regards to each scenario.

Chapter 5 - Summary
The final chapter presents discussion and commentary on the dissertation in its totality with regards to the initial research aim and objectives established in the opening chapter. Additionally, possible future works are examined in addition to reflection on aspects of the dissertation. The dissertation concludes with final remarks on the project.
2 Background

2.1 Modelling Speech Perception

As humans our entire perception of the world around us is the result of our brain processing and interpreting sensory information in our environment. Perception is the procedure by which the brain makes sense of and registers the world around it through the processing of continuous natural stimuli. The human ability to perceive at a higher level than other species is one of, if not the greatest differentiators between humans and other species at the cognitive level. Through perception humans can understand speech in a variety of situations with relative ease even as the complexity rises in settings with multiple speakers, various accents, intonations and volumes of speech. Antithetical to the importance of the procedures underlying the complex neural systems that allow for the brain to perceive the environment, relatively little is known about the system that enables the brain to parse and process continuous stimuli like speech and music. Through the action of receiving and processing input the brain produces measurable and quantifiable electrical brain activity. The dynamics of such electrical activity vary based on the different environments, scenarios and stimuli that the brain receives as input; By recording the electrical brain response to the stimuli we can uncover information about the inner workings and mechanisms of the brain to reveal how the brain processes continuous neural stimuli. However, exposing the sophisticated inner workings of the neural system in the brain is an increasingly difficult and complex task (Casserly and Pisoni, 2010).

Within the field of neuroscience, the classic experimental paradigm to measure the brains response to stimuli has involved using a strict trial based structure, where each trial consists of a brief and discrete series of events which are regarded as independent from other trials (Huk et al., 2018). Such a discrete series of events could be short high frequency sounds used as a stimulus to detect how the brain’s electrical activity differs in response to different frequencies of sound. However, while this paradigm does allow the experimenter to adjust certain input parameters and measure the resulting outputs from the neural system, it overlooks that in reality, stimuli in the natural human environment is rarely, if ever, discrete (i.e individually separate and distinct). In reality, the natural human environment contains
dynamic visual stimuli which exhibit multiple spatial frequencies as well as altering auditory soundscapes consisting of multiple temporal frequencies (Murray et al., 2014). Such stimuli are unsuited to conventional trial based structures and reflect an over-dependence or preference for rigid, simplistic and controlled experiments which do not accurately capture the qualities and dynamics of stimuli in the natural environment. Through the use of continuous natural stimuli within experimentation, it is possible to gain a deeper, more comprehensive understanding of the underpinnings of the neural response process supporting perception while simultaneously maintaining high levels of experimental and quantitative rigor. While discrete inputs like single sinusoidal sounds have offered glimpses into the neural process, utilising more complex continuous stimuli such as human speech offers far more ecological validity and leverages advancements in analytical capabilities to offer insight into sensory input and neural activity.

From a methodological point of view, regardless of the stimuli that are used to invoke a neural response, the objective remains to identify the transformation by which neural system processes input. The upper section (a) of (Fig 2.1) can be said to be a classical, trial based experiment measuring a discreet input. The bottom section (b) visualises a realistic continuous stimulus as an input in order to identify the system involved in neural processing.

![Figure 2.1](image)

Figure 2.1: A linear system characterised by impulse response. Figure illustrates two methods of measuring impulse response. (a) measured as the response to a brief pulse or (b) the impulse response of a system. Adapted from Ringach and Shapley, 2004.
2.2 Linear systems

By delving into the historical origins of how linear systems are used to model stimulus and brain response we arrive at a "reverse correlation" experiment created by Erich Sutter in 1975. (Ringach, 2004). Sutter aimed to understand how the neural system processed input by recording the first evidence of the relationship between stimulus and response in the brain.

![Diagram of apparatus used by Erich Sutter](image)

Figure 2.2: A and B, schematic diagram of the apparatus used by Erich Sutter to measure, for the first time, the full spatiotemporal receptive field of simple cells in a cat. Reproduced from experiment by Erich Sutter, 1975; figure adapted from (Ringach, 2004).

Sutter did this by conducting an experiment in which he recorded the brain response to a stimulus within the neural system of a cat. While it may seem archaic to modern audiences, Sutter’s experiment bares a significant resemblance to the state-of-the-art experiments conducted by researchers in the field of cognitive neuroscience today and stands as an important milestone within the field by being the first recording of the full spatio-temporal spectrum. Sutter defined a stimulus as any input to the neural system. Such an input could be anything that evokes a neural response such as a sound, or an image entering your eyes. Sutter wanted to measure the brain’s response to this input so he set up an experiment in which a stimulus was produced by playing white noise from a tape. Sutter then arranged a second recorder a distance away from the feline subject’s head. The second recorder received the tape arriving from the first recorder as well as the amplified signals arriving from a micro electrode (attached to the subject’s head) onto another track on the tape. With this relatively simple, yet elegant experiment Sutter was able to capture the temporal relationship between the stimulus and the neural response. In order to analyse the data, the apparatus visible in Fig 2.2 B was used. The recorder on the left played the tape with the audio of the response while the other recorder played the track with the stimulus. Similarly to the time lags configuration of modern day experiments, the time lag (a period of time between the stimulus event and the response event) was varied by increasing the distance.
between the subject and the tapes. When a time lag was selected, the stimulus was replayed on an oscilloscope such that the intensity of the signal was multiplied in the presence of a spike in the neural response. When there was no neural response present for the stimulus the intensity of the signal was unchanged by the amplifier. Sutter’s seminal study of the temporal relationship between stimulus and response established the feasibility of applying novel techniques to capture how neural computation supports perception in mammals as well as inspiring work in the field of human cognitive neuroscience as we know it today.

2.3 Encoding & Decoding Models

In recent times, cognitive neuroscience has witnessed an increase in the quality and complexity of the data recorded from the human brain through advancements in invasive and non-invasive technologies as well as sophisticated computational analytics tools to investigate recorded data. While this data is often recorded in lab studies in which the results are unavailable to the wider scientific community, the increase in the quality of data has nonetheless enabled researchers to use more complex continuous stimuli that enable the neural response system to be studied under more naturalistic conditions.

Within cognitive neuroscience there are two main paradigms of analysis, neural encoding and neural decoding. These are complementary operations in which encoding uses stimuli to predict neural activity while decoding uses neural activity to predict information about stimuli (Naselaris et al., 2011). In this sense, encoding investigates what neural process transforms a stimulus to a response. For example, the neural system by which speaking evokes a specific brain response in the listener. Conversely, decoding investigates how a stimulus can be deciphered from a neural response. For example, deciphering what a words a person spoke by analysing the brain response of the listener. Encoding models have seen a significant increase in popularity within research literature using electrocorticography (ECoG) (Mesgarani et al., 2014), functional magnetic resonance imaging (fMRI) (Nishimoto et al., 2011) and EEG/MEG (Di Liberto et al., 2015). Additionally, studies have been conducted in which decoding models have been used in order to predict stimulus features using the neural activity response data (Mesgarani and Chang, 2012; (Pasley et al., 2012; Stéphanie et al., 2014). While differing in process, both the encoding and decoding paradigms are said to be predictive models and are generally represented mathematically in the form of a linear regression.

The advent of non-invasive and invasive electrophysiological recording technologies has enabled the detection of electrical activity in the brain synchronizing to specific properties present within the stimulus (Lalor et al., 2006; Ding and Simon, 2012; Gross et al., 2013). This phenomenon has been referred to as neural tracking and has enabled the analysis of perception within tasks with higher levels of complexity such as those involving continuous
The cocktail party scenario, describes a person who is at a party with multiple people immersed in conversation around the subject (Power et al., 2012). The subject’s goal is to listen to and communicate with people around them who are engaging in conversation. While at times challenging, for people with normal hearing this is a relatively simple task. However, for those who are hearing impaired or for any artificial intelligence system, the task can quickly descend into a muddled, disordered mess. The field of cognitive neuroscience has focused on the brain’s ability to resolve this multi-speaker cocktail party dilemma in a process referred to as neural entrainment. The term neural entrainment can often be used interchangeably with neural tracking, however, it is important to note that each refers to a specific aspect of the neural process and are distinct features. Neural entrainment in the narrow sense is said to refer specifically to the time-locked synchronisation of single, individual cells with a continuously occurring stimuli, operating on a microscopic level. While neural tracking refers to a broader sense of the neural system mapping to a continuous stimuli on a macroscopic level. To this end, neural tracking of multiple acoustic and linguistic features helps to divulge the complex underpinnings of how each feature is encoded in the higher order neural system.

While recent investigations detecting how the brain synchronises to certain properties present within exogenous stimuli have uncovered the phenomenon of neural tracking (Obleser and Kayser, 2019) as well as establishing many of the fundamental aspects of the method by which the brain enables perception, this work is limited in that it cannot shed light on brain functions that are unique to humans, such as speech and music perception. Most neurophysiological research on human perception has used simple, controlled experimental paradigms which are generally far removed from reality. However, in recent times studies have demonstrated that both invasive and non-invasive electrophysiology recordings can robustly detect neural tracking and as such reveal objective measurements in which perception can be studied via more intricate and complex experiments using continuous real-life stimuli like speech and music. (Zoefel and VanRullen, 2016; Zuk et al., 2020; de Taillez et al., 2020; Golubic et al., 2013)

2.4 Modelling the Brain Response Framework

The human brain comprises of a complex network of spatio-temporal processes and cross signal interactions that have proven difficult to capture. The sophisticated nature of how the neural system responds to a stimulus has led to development of a variety of technologies with the ability to capture the unique inner network dynamics of the brain. The two most ubiquitous non-invasive (not involving the introduction of medical instruments into the body) technologies to investigate the complex systems involved with processing natural
continuous stimuli and speech perception have been scalp electro-encephalography (EEG) and magneto-encephalography (MEG). Both technologies enabled researchers to develop a methodology that is capable of isolating neural activity during speech processing in response to continuous speech stimuli.

![Figure 2.3: EEG signals recorded from a relaxed subject. Four seconds of data are shown from four scalp locations. The corresponding amplitude spectra reveal activity in the 8-13 Hz band. Adapted from (Nunez et al., 2006).](image)

Before non-invasive procedures were developed, electrodes were placed inside of non-human skulls in order to study models of the neural system within animals such as primates (Rauschecker and Scott, 2009). While such procedures represented significant progress in understanding the neural biology of the brain’s response to stimuli, the cerebrum of a primate has not developed the mechanisms for interpreting speech such as as those that are evident in human brains and as such are not similar enough to be representative of the human neural system. Subsequently, invasive brain recording technologies such as electrocorticography (ECoG) have been used in human patients with severe cases of epilepsy which require brain surgery intervention. These recordings have resulted in reliable neural data with clear temporal and spatial resolution (Mesgarani et al., 2014). However, as a result of the ECoG recordings sampling patients with severe epilepsy, it could not be ascertained whether or not the neural system was affected by the aforementioned condition. Furthermore, the ECoG scan was limited in scope as only a specific area of the cerebral cortex could be scanned. The cerebral cortex is the outer most tissue of the largest portion of the brain (the cerebrum) and is largely responsible for the higher level processing within the brain such as speech, reasoning and memory. The invention of the functional magnetic resonance imaging (fMRI) in 1991 was able to surmount the shortcomings of ECoG by removing the restriction of only being able to be performed on certain groups (e.g. pre-operation epileptic patients) in that anyone could have an fMRI scan to effectively map their brain (Overath et al., 2015). Additionally, fMRI scans were not limited to a specific
area of the brain and could map the cerebral cortex and neural network in its entirety (Logothetis, 2008; Tuennerhoff and Noppeney, 2016). However, fMRI has a disadvantage in that the temporal resolution (how closely the measured activity corresponds to the timing of the actual neuronal activity) is not very high in quality and as a result, makes it difficult to record the rapid dynamics and intricacies of speech.

By improving over the shortcomings of invasive procedures such as ECoG (e.g. limited cortical mapping and highly limited patient group) as well as non-invasive alternatives such as fMRI (poor quality temporal resolution), EEG and MEG have become the primary methods of capturing spatio-temporal processes in the neural system. Furthermore, EEG has become the brain imaging technology of choice for many advanced research studies given that it is a widely available technology that is capable of effectively capturing dynamic neural processes with a relatively high level of temporal resolution. EEG was first discovered in 1929 by the German psychiatrist Hans Berger (Tudor et al., 2005). Since its inception, the EEG system has advanced significantly and now recorded signals are transmitted to EEG systems composed of a electrodes, a computer and an amplifier. EEG systems detect electrical activity in the brain through the utilisation of small metal discs called electrodes that are attached non-invasively to the scalp. An example of the resulting EEG channels and their corresponding amplitude can be seen in Figure 2.3. The signal from each electrode on the scalp is recorded and averaged over the total inputs. Typically the wave-forms from the resulting input occur less than 500 milliseconds after the stimulus is received. This allows the amplitude and latency from the stimulus to be analysed in order to uncover how the brain response corresponds to the presented stimulus.

Figure 2.4: Magneto-encephalography (MEG) system at the Neuro-imaging research centre in Pitie Salpetriere hospital in Paris, France. 306 sensors spread over 102 areas enable both near and far magnetic fields to be measured in addition to visualising the anatomical structure of the brain. Subject receives visual stimulation from a video(faces with different expressions).

MEG measures brain activity by recording the magnetic fields generated by electrical activity
in the brain (Gross, 2019). Similar to EEG, MEG offers the distinct advantage of being able to record non-invasively while maintaining high levels of temporal resolution. MEG was originally developed in the 1960s and the brains magnetic fields were first recorded using a single sensor (Cohen, 1968). Subsequently, MEG technology has advanced and deepened in it’s level of technical sophistication. The current state of the art leverages over 300 sensors that envelop the subjects scalp in a helmet shaped design. Furthermore, MEG offers a considerably higher accuracy over EEG at locating sources in the brain with the caveat that this solely applies to currents visible to the MEG. This can be portrayed as either advantageous or disadvantageous as MEGs could have greater potential for identifying single brain processes while EEGs identify an amalgam of temporally overlapping processes (Pulvermüller et al., 2001).

![Figure 2.5: Electroencephalographic recordings using 128 electrodes making scalp contact with a conductive gel. One of several kinds of EEG recording methods. Adapted from (Nunez and Srinivasan, 2007).](image)

While both EEG and MEG systems offer high quality temporal resolution (within milliseconds) there are distinct differences separating the technologies. As a result of EEG systems making physical contact with the scalp through the electrodes and conductive gel, EEGs are highly influenced by variance in electrical conductivity through the different mediums present in the equipment (i.e. electrodes, conductive gel, scalp skin, human tissue and the skull). Additionally, EEG can be affected by electrical noise present in the environment such as in electronic devices. As the MEG does not make physical contact with the subject it is not affected by differences in electrical conductivity and as a result the signals from MEG are less altered by electrical interference than EEG electrical potentials. In this respect, MEG permits greater anatomical analysis of the sections of the brain from where the neural activity is generated. Additionally, MEGs are highly dependant on the location and orientation of the sensor apparatus in respect to the source location of the
neural activity in the brain. For this reason, the orientation of the subjects head as well as the position of the head are transcribed for each subject between recording sessions in order to maintain integrity of the recorded signal data. As a matter of course, EEGs are unaffected by surrounding electromagnetic interference present in the environment. MEG systems are relatively expensive to acquire and maintain in addition to requiring a magnetically shielded room in order to avoid over contamination from extraneous magnetic fields present in the environment. Due to the cost involved with MEG systems, this paper has focused on the significantly more affordable, albeit variable in quality, EEG System. In evaluating the role of MEG and EEG in cognitive brain research, it is important to note that while not always practical due to budgetary constraints, both EEG and MEG should be utilised together to conduct more comprehensive brain scans where possible. Both non-invasive technologies provide precise temporal information about the brain activation patterns in its interaction with the continuous natural stimuli present in the environment. While not always practical due to budgetary constraints, both EEG and MEG should be utilised together where possible.

2.5 Event-related Potentials (ERP)

Non-invasive EEG and mMEG systems have proven a viable method to understand the brains response to continuous natural stimuli. EEG response data has played an important role in unlocking the underlying system of how the brain responds to a specific event of interest (a individual stimulus). For example the appearance of the stimulus in the brains response. Minuscule voltages appear in the EEG data and are referred to as event-related potentials (ERP) or event-related fields (ERF) when appearing in MEG data. ERPs and ERFs are generated in the brain in response to specific events or stimuli (Blackwood and Muir, 1990). In EEGs the subjects brain response to an event of interest or stimulus are generally not evident given that the EEG detects thousands of differing brain processes occurring simultaneously. In this respect the signal-to-noise ratio (SNR) is generally insufficient to accurately measure the response to a singular event.

The SNR has an extensive history within the field of neuroscience for its use as a measurement of the fidelity of signal transmission and detection by neurons and synapses (Schultz, 2007). In terms of it’s context within neuroscience experiments detecting ERPs, the SNR allows us to compare the quality of non-invasive EEG and MEG recordings containing events that are recorded in the presence of noise. The ERP signals are intertwined with signal noise from various sources such as electrical activity emanating from the muscles when the subject moves, background electrical activity within the environment as well as arbitrary brain activity that occurs naturally. Combined, these sources of noise create a difficult environment for discerning the relatively small voltages from ERP signals.
In studies involving ERPs, subjects are generally asked to perform tasks while being subject to a number of stimuli. A method of minimising high SNR’s is to extract a series of EEG epochs (particular times), time-locked to a specific stimulus. By conducting multiple trials, an experimenter is able to average the results of the experiments, this has the effect of averaging out superfluous brain activity while the relevant brain signal activity is brought to the fore (Coles and Rugg, 1995). This is based on the assumption that a specific stimulus will elicit the same ERP each time and that the noise in the experiment averages out to a mean of zero. This technique of averaging out the noise is an effective method of reducing noise in the SNR while preserving the response to the stimulus (Luck, 2014). While ERPs have seen extensive use within the domain of cognitive psychology and researchers have found stimuli that can reliably elicit ERPs from participating subjects, many of the resulting discoveries and advancements from cognitive psychology have been adopted within cognitive neuroscience research domain.

Within the natural environments complex auditory stimulus is ubiquitous. The human auditory system has evolved to process the multivariate forms of sound from quiet to loud noises, short to long bursts and layered complex human speech (Lalor et al., 2009). Significant advancements in temporal processing utilising ERPs has generated important insights into the underlying system behind how the auditory system processes speech. However, the ERP method is confined by the necessity for uncomplex, discrete stimuli in order to illicit a noteworthy response from a subject. The ERP framework is significantly more efficacious when modelling the response to isolated, short and frequent stimuli. In
relation to speech processing within this framework, the discrete stimuli that are used are generally composed of words, syllables or shorter units of speech called phonemes. In order to see how the brain responds to a more naturalistic continuous stimuli a new method was proposed called the Temporal Response Function (TRF) which circumvented the limitations of the ERP model. The TRF has enabled the study of complex continuous stimuli in addition to the system that enables the brain to distinguish between speakers and focus attention on an individual speaker in a multi-speaker scenario i.e. the aforementioned cocktail party scenario (Power et al., 2012).

2.6 Temporal Response Function

One of the foremost objectives of modern twenty-first century neuroscience research is to interpret how the brain processes sensory information and signals in natural environments. Invasive electrophysiology and brain imaging techniques have played an important role in this endeavour and will continue into the future in conjunction with noninvasive, macroscopic techniques such as EEG and MEG. The best way to analyse sophisticated neural responses to complex multivariate natural sensory stimuli derived from these aforementioned invasive and non-invasive techniques has been contested without general consensus within the neuroscience community. EEG and MEG data have seen a considerable increase in the application of system identification techniques to find a correlation between the firing activity of neurons to the sophisticated multivariate sensory stimuli; Such techniques have an extensive history within neuroscience research. System identification is a mathematically modelled function which describes the method by which a specific property of the stimulus is related to the neural response(Marmarelis, 2004). Specifically, the temporal response function, a filter mapping between features of sensory stimulus and neural response has been used on EEG and MEG data to find the correlation between neural activity and natural sensory stimulus.

The temporal response function (TRF) is a time varying stimulus feature characterised by data over time. TRFs measures how the change in stimulus features presents themselves in electrophysiological data such as EEG, MEG, ECoG etc. While it has been established that the human brain functions non-linearly and is time variant (response of the system to an input is a function of time), advancements within the field have been made by assuming that the neural system behaves similarly to a linear-time invariant (LTI) system (Mesgarani and Chang, 2012). The defining properties of an LTI system are linearity and time-invariance. The term linearity refers to the relationship between the input and output (both regarded as functions) and is a linear mapping. Time-invariance references that if we are to apply an input to the neural system either in real time or in T seconds, the output will be equivalent except for the time lag of T seconds. While it is a strong assumption that the
brain adheres to the LTI system, it enables the use of linear regression as an approach for modelling the relationship between stimulus and response. An additional advantage of using a linear model is that it is relatively simple to fit with limited amounts of data. For example, using EEG there are typically 128 channels and multiple time-points. Each channel is highly correlated with its neighbour and each time point is correlated with its neighbouring time point. In this respect, while EEG data may appear to have a relatively high degree of dimensionality, in practice the dimensions of the data can be very similar to each other. For this reason fitting the data with more complex models like deep neural networks (DNN) is not generally performed due to the high probability of over-fitting the data.

The interpretability and practicality of fitting a robust model with limited amounts of data has made linear models a mainstay of neuroscience research for decades. Additionally, using the LTI framework with linear regression allows the neural system to be modelled based on its impulse response. The impulse response or impulse response function (IRF) is the output of the system when given an input signal referred to as an impulse. In this respect, the impulse response can be used as a measure of continuous stimuli processing which can be estimated using the input and output signals. By using an estimate of the impulse response an output given an input can predicted. This is referred to as forward modelling (e.g. a brain response given input of speech). Likewise, the input that evokes an output can be reconstructed (e.g. reconstructing the speech that evoked a brain response). This is referred to as backward modelling. Furthermore, the estimated system response is referred to as the temporal response function (TRF) (Crosse et al., 2016). To model how the human brain supports speech and sound perception using the TRF framework, a sound envelope is used as a stimulus while the brain response is the neural signal measured by invasive or non-invasive recording methods i.e. ECoG, EEG, MEG, etc. TRF is a predictive model.
which enables the use of quantitative measurements that are obtained based on the prediction of the neural response as well as reconstruction of the stimulus. Furthermore, linear regressions frameworks such as the TRF have been able to accurately model the neural response to continuous human speech.

2.7 Continuous-event Neural Data Format

The Continuous-event Neural Data Format is a proposed standardised data format created by the CNSP workshop in order to promote standardisation, replicability and reusability of code within the field of neurocognitive study. The low amount of high quality, publicly available electrophysiological data and the unstandardised nature of the data collected in studies necessitates a structure to improve conformity and interoperability between neuroscience researchers around the world. Due to the prohibitive and costly nature of high-end EEG recording equipment, there is a diverse range of equipment currently used in studies that vary greatly in quality and performance. This variance in quality of recording equipment has precipitated the rise of wide range of formats and structures being propagated within the cognitive neuroscience community without a predominant structure to format neural data. This lack of structure creates additional friction and complexity when trying to run analysis on data collected from differing recording equipment that has been stored in varying formats. The CND format can alleviate such obstacles, enabling the cognitive neuroscience community to have greater accessibility and ease of use with regard to neural data analysis. Additionally, it is hope that through the usage and promotion of the CND format within the neural analysis platform that the wider community will be encouraged to make the data gathered from their studies publically available for use. All datasets used within the Neural Analysis Platform have been structured in the CND format prior to use within the site. The CND format is a relatively simple standardisation and will facilitate working across different teams, datasets and analysis methodology.

In order to save a dataset in the CND format the following file organisation/structure is required (based on CNSP Workshop 2021 Data Preparation guidelines:

```plaintext
1. Dataset folder (e.g., LalorNatSpeech)
2. eeg
   - raw data file 1 (e.g., bdf, xdf)
   - raw data file 2
   - ...
3. stim
   - wavfile1.wav
   - ...
```
In this folder structure each `dataSub.mat` file includes all of the neural data (e.g., EEG) from a given subject or session without any separate files for a distinct/trial or run. In order to protect the identity of each subject and for increased anonymisation of the data, each subject is indexed numerically. In this respect, the subject index should not reflect the order of their participation in a given recording session. Furthermore, the data in the CND format should be synchronised to the stimulus i.e., sample 1 in `dataSub.mat` corresponds to sample 1 in `dataStim.mat`. The `dataStim.mat` file is a file containing all the stimulus features for each of the trials. The features are preprocessed and additional fields can be optionally saved. Such an example of an additional fields includes a researcher saving labels indicating what the nFeatures in the 'data' field correspond to, such as frequency-bands or phonemes). The 'data' and 'fs' fields are required to be formatted as presented below.

```matlab
stim: struct with fields:
   names: {'speech envelope vectors', 'word onsets'}
   data: {2x20 cell}
```

The `stim` is a cell array containing either univariate or multivariate stimulus. It contains feature vectors for each feature and trial. In this particular case, it contains two stimulus features (envelope and word onset) and twenty trials. Each cell is a (timeSample x nFeatures) matrix and corresponds to one trial/run (e.g., 1 chapter of an audiobook). Different trials can have varying lengths.

```matlab
stimIdxs: 1:20
```

The above Stimulus indexes are recorded as it is possible that in a different experiment that a stimulus can be repeated any number of times. Each recording session may vary the order of the stimuli that are used so the 'eeg' structure can be used in order to validate the original presentation order.

Each `datasub.mat` file from `dataSub1.mat`, ..., `dataSubN.mat` contains one variable per recording modality, in this case one variable called 'eeg'.
The variable ‘eeg’ is a Matlab structure with the following fields:

1. **struct with fields:**
   - `dataType`: `'EEG'`
   - `deviceName`: `'BioSemi'` % Name of the recording system (if available)
   - `fs`: 128 % sampling frequency of the EEG signal
   - `data`: `{1x20 cell}`

Within each cell array, each cell is equivalent to a (timesample x channels) matrix. Ideally there is one trial/run per cell (e.g.) chapter one of an audio book. Different trials/runs can have different lengths and are sorted so that CND data from all recording sessions can use the same stimulus order (e.g., 1:20). The field `origTrialPosition` indicates the position of a given EEG segment (trial) in the presentation order of the original experiment, in case it is relevant for the analysis. In this experiment, trials were presented in order (audio-book). If the experiment had 2 conditions (e.g., listening vs. imagery), with 20 trials, then `data` would have dimensionality 1x40 cell and we would need to include the condition indices of each trial in an additional field `stim.'condIdxs'`.

- `origTrialPosition`: 1:20

Eeg.data cells have a 1-to-1 correspondence with the stim.data cells. This means that the eeg.datai corresponds to stim.datai and that data from all subjects is sorted in the same way. However, the actual stimulus presentation order in the experiment may have been different. This field is used to remember the stimulus presentation order and indicates the original position of each trial in that order. Handy suggestion. To recover the EEG data with the original presentation order use:

1. `clear X; X(eeg.origTrialPosition) = eeg.data;`
2. `chanlocs: [1x128 struct]`

Channel location variable (EEGLAB format)

1. `extChan: {[1x1 struct]}

By converting their data to the CND format, researchers will be able to analyse their data without requiring reformatting in order to run models and scripts. Additionally, code and script reusability will increase across future experiments and projects saving the researcher time spent re-writing and modifying code. Using the CND format will make it possible to compare different configurations of experiments as well as varying analysis methodologies on different datasets. For example, TRFs can be created for a music dataset as well as speech dataset in addition to being able to configure the input parameters of each experiment individually. It is hoped that using the CND format will increase standardisation and
collaboration across a relatively disjointed industry of research teams working in the field of cognitive neuroscience. This will foster increased collaboration and less unilateral workflows within leading edge research teams.
3 Method

3.1 Purpose of Neural Analysis Platform

Advancements in neuroscience and analytical technologies have allowed us to understand the underlying systems in the brain in significantly more detail than previously possible. The brain's ability to interpret the environment around it is one of the most important aspects of human nature and understanding how the brain operates is one of the keys goals in modern cognitive neuroscience. To this end, researchers around the world are utilising advancements in invasive electrophysiology and non-invasive brain imaging technology such as EEG and MEG with high temporal resolution as well as leveraging the power of modern machine learning and data analytics techniques. However, due to the complexity of neural data, challenges arise in deciding how best to analyse time varying, continuous neural responses to multivariate natural stimuli.

The domain of neuroscience comprises of a cross-section of neurologists, scientists, psychologists, programmers and researchers. Furthermore, there is a varying degree of technical fluency in programming as well as modern data analysis techniques such as machine learning amongst the cohort of professionals within the cognitive neuroscience community. In order to circumvent the time and cost involved in developing proprietary modern data analytics tools for individual labs and users, multiple libraries for conducting data analysis on electrophysiological data have been developed by different groups around the world and have become widely available for the public to use online. Such projects are developed open-source with different implementations available in different programming languages such as Python and MATLAB. Popular online libraries and toolkits include: mTRF-Toolbox, Eelbrain, NoiseTools and EEGLAB. The advent of such technologies has been a boon to the cognitive neuroscience community with programming knowledge, enabling them to easily express cognitive hypothesis in the form of computational models and evaluate their predictions with electrophysiological brain response data. However, in order to use the aforementioned libraries and toolkits, users must be well versed in programming and setting up development environments which can prove a time consuming and arduous challenge to those with little to no programming experience. This has the effect
of excluding swathes of non-programming users such as many researchers, psychologists and neuroscientists from having access to modern state-of-the-art data analysis tools to conduct research as well as make sense of their electrophysiological response data. The inherent challenge and inaccessibility of configuring programming libraries and toolboxes for non-technical users necessitates the development of a novel online platform in which both expert and non-expert users can have access to the prevalent data analysis techniques that are most commonly used for experimenting with stimuli and response data.

This paper introduces a new online web application called the neural analysis platform that gives users access to the functionality and extensibility of the aforementioned programming libraries and tool-kits available online without the prerequisite of knowing how to set up and configure development environments or knowing how to program. Specifically, the neural analysis platform will enable users to extract useful information from electrophysiological brain responses such as EEG and MEG data by generating the distinct responses associated with different predictor variables. In order to accomplish this, the platform will automatically estimate a multivariate temporal response function (mTRF) based on the user-defined in-app configuration, which maps the predictors to brain responses elicited by continuous natural stimuli.

The ongoing challenge of standardisation and reproducibility has become a topic of paramount importance within the cognitive neuroscience domain, previously held paradigms in which researchers publish studies are being reconsidered in a bid to resolve inconsistencies across research and academic studies. This is known as the replication crisis, in reference to the ongoing methodological predicament in which a substantial amount of results from scientific studies have been found to be very difficult or near impossible to reproduce. This is a problem found frequently in the medicine and psychology fields as well as in cognitive neuroscience where researchers are increasingly designing and carrying out their own proprietary studies, without providing the data that was gathered in the experiment to the wider research community for validation. This data is generally a prerequisite to being able to reproduce the results of a study which researchers need in order to verify the accuracy and validity of claims made within a study. Additionally, replicability with regards to EEG experiments can be affected by the nature of EEG signals which are subject to change based on a multitude of factors. The electrodes that are attached to the subject’s scalp during EEG experiments can be affected by sources of interference such as electrical signals in the environment where the experiment is taking place (e.g. a flickering light on the ceiling). Movement from the subject triggering electrical signals firing from the brain to the muscles can also add noise to the experiment, hindering the ability to replicate results. Such examples of unwanted to noise in the experimental environment can compound and create a cumulative affect which is often larger than the signal that is trying to be observed from the neural response. While the neural analysis platform will not alleviate these challenges, by
having the data standardised in the CND format, inconsistencies in the electrophysiological data will more readily be able to be identified and corrected. The neural analysis platform will aim to alleviate some of the challenges of modern experimental cognitive neuroscience with regard to the interpretability and reliability of its findings through encouraging the sharing of datasets used within experiments for use by the public within the platform.

Currently in research using brain imaging techniques the researcher conducting the experiment is afforded high degrees of flexibility with regard to the methodological approach they use to record the data, the equipment that they use and the analysis techniques that are used to evaluate their results. Studies involving EEG experiments can have variations of electrodes on the scalp, frequency bands, preprocessing, filtering and time lags that vary greatly depending on the study and all contribute to a distinct methodological workflow for each research team or individual. Many researchers and research groups use different preprocessing specifications and data analysis techniques on the electrophysiological data as well as alternative methods for artefact removal within the data. Such distinctions in processes are generally for perfectly credible reasons but nevertheless are a hurdle in standardising the EEG format. This impracticality has greatly hindered the prospects of combining datasets from different EEG studies that have not been preprocessed to an identical specification. Through encouraging the use of a homogeneous framework for preprocessing electrophysiological data, the neural analysis platform hopes to enable the combining of data from different sources, conducted with varying equipment. The prior mentioned Continuous-event Neural Data (CND) format will be a catalyst for promoting the sharing and standardisation of electrophysiological data.

### 3.2 Development of platform

The design of the platform has been an iterative process that has developed over time through careful consideration of the most useful workflow for both experts and non-expert users to allow them to rapidly configure cognitive experiments. A Unified Model Language (UML) diagram was created and gradually improved in order to best depict how the user will interact with the web application in different scenarios which can be seen in figure XXX. The primary use of the application was initially to enable users to study how continuous natural speech is processed by the human brain through allowing them to configure a forward TRF. Furthermore, users are able to experiment with how neural responses to speech can be extracted with high levels of temporal resolution. After discussion with my supervisor as well as potential users of the application, it became apparent that it would be highly beneficial to add both forward and backward modelling capabilities to the neural analysis platform to enable users to further investigate continuous stimuli such as audiovisual speech processing.
Figure 3.1: Unified Model Language Diagram (UML) visually representing the use case of neural analysis platform for expert and non-expert users.

User Interface mock-ups of each of the platform’s features were created in order to visualise the aesthetic of the platform as well as the user experience and functional components that
would need to be developed in React. Various components such as the 'Confirm Button' and 'Reset' buttons were added to increase ease of use for the user. While initially there were plans to integrate additional functionality in terms of adding values and measures of statistical significance to the platform, it was decided that users would be best served by enabling them to download the output data using the 'Download' button. After downloading the output data, the user can conduct additional analysis that is unique to their experimentation in addition to creating their own tests for statistical significance.

Additionally, users can use interactive slider components to adjust parameters such as the 'Length of the Trials', the 'Number of Trials', the 'Number of Subjects', 'Lags' and range of 'Lambda' values to their desired configurations based on the experiment they are conducting. Once the user clicks the 'Confirm' button the web-page will indicate that the back-end is generating the output by displaying loading spinners that disable the 'Confirm' button from being clicked again until it has completed the initial request. Due to the nature of the computations that are taking place on the neural data such as training and testing, the application can take a significant amount of time to generate outputs depending on the operations that are being performed and the dataset/s that are being used. In order to mitigate user wait times, once a configuration has been run by a single user, the output is saved to the server so that it can be instantly served to all users with no wait times or requirement to be computed again. Furthermore, by saving the resulting outputs when a unique configuration is run, the platform will not need to spin up an instance of MATLAB production server thus saving on costs of running the platform.
3.3 Front-End & User Interface Mock-ups

The user-interface of the neural analysis platform was designed in order to offer a complete range of functionality that expert users would expect when conducting experiments while not being overly intimidating to non-expert users. The UI plays a vital role in the success of the neural analysis platform moving forward as its design is the primary method of conveying information to the user. It was important to maintain involvement with the intended users of the platform throughout the design of the system. In this respect, the platform received constant feedback from intended users throughout the design process in order to iterate on the initial design specification. How information was conveyed was especially important throughout designing the neural analysis platform as the intended use cases of the platform are relatively esoteric and require a significantly large amount of background knowledge.

The front-end was designed using reusable React components in order to reduce code debt and increase maintainability of the codebase. Additionally, consistency of notation, abbreviations and naming conventions were maintained throughout the platforms code base.

3.4 System Architecture

In order to build the neural analysis application I used the micro web framework Flask. Flask is a lightweight Web Server Gateway Interface (WSGI) written in Python. Flask was
selected in order to make setting up quick and easy while also enabling the application to scale up to become more complex and handle a larger amount of users overtime. Flask enables the developer to select the tools and libraries that will be used in the application and makes adding new functionality to web applications simple. The decision was made to implement the front-end of the Neural Analysis Platform in React. React is an open-source JavaScript library for building user interfaces and was originally developed by Meta (formerly Facebook). React can be used in conjunction with the Flask Python framework and enables state management of UI components as well as rendering to the Document Object Model (DOM), a cross platform interface ubiquitous in online web applications.

3.5 Application Back-end & API

In a data driven world, machine learning has become the primary means of gaining insights to create accurate models and make informed decisions. In this respect, two of the most ubiquitous programming languages for implementing machine learning models are MATLAB and Python. In terms of cognitive neuroscience, a package called the mTRF-Toolbox (multivariate Temporal Response Function) has been developed in MATLAB as a tool for modelling multivariate stimulus-response data. The mTRF-Toolbox can be used to model the functional relationship between the neural response and dynamic sensory input such as continuous natural stimuli like speech and music as well as reconstruct stimulus features. In this respect, the mTRF-Toolbox package enables the analysis of a variety of different...
neurophysiological data types such as MEG, EEG and ECoG. By using MATLAB and the mTRF-Toolbox package a developer can conduct analysis and configure cognitive experiments in a development environment. However, the absence of suitable software architectures for deploying MATLAB models to fully functional web applications has been a significant limiting factor that has restricted the benefits of the mTRF-Toolbox to technology savvy, academic audiences with MATLAB programming experience. Additionally, the use of MATLAB requires downloading and configuring a MATLAB environment which requires purchasing a MATLAB license and can be a time consuming and challenging ordeal for those who are not experienced in setting up development environments on their local computer. Creating an online hosted web application that allows both expert and non-expert users to utilise the benefits of the MATLAB in a visual real-time environment, will make high quality cutting edge neural data analysis available to significantly wider audience. Additionally, creating the neural analysis platform will establish a framework for deploying MATLAB programs to web applications so that future projects can gain the benefit of directly deploying MATLAB code to enhance the user experience.

In order to integrate the Python/Flask back-end, the React front-end and MATLAB (specifically the mTRF-Toolbox), I developed an Application Programming Interface (API) within Flask. An API refers to a computer program that is designed to be utilised by another
The decision to implement an API was made based on a number of factors: The datasets that would be accessed by the neural analysis platform were quite large (> 3GB) making download via File Transfer Protocol (FTP) unwieldy as well as extremely resource intensive. In order for the platform to be useful to users and have a satisfying user experience, it was paramount that the user was able to perform neural analysis as well as adjust configurations in real time and for the resulting output to be displayed within the platform interface. In this respect, the changes and updates to the configuration of experiments are likely to be very frequent requiring recurrent calls to update the user interface. The API will accept HTTP GET requests containing the data and parameters selected by the user in the app. The API will then respond to the requests with corresponding graphs (Temporal Response Function, Mean Error). The API endpoint will accept parameter values in a URL string. The values will be based on the different parameters selected by the users within the user-interface e.g. Dataset, Lags, Lambda Value, Graph etc.

To connect the API that was developed with MATLAB, I used MATLAB Engine which provides a MATLAB package that can be called from Python to be used as a computational instance. The MATLAB Python Engine API was added to the project and after the dependency was added to the environment, the MATLAB Engine API library was used to connect to a MATLAB session, send the user selected dataset and parameters to the session, run an mTRF model, compute a graph on the data using mTRF-Toolbox functions and then store the graph on the server. This provided a flexible two way integration between Python and MATLAB allowing the execution of MATLAB commands from within the
Flask/Python environment without needing to switch programming environment and create a desktop session for MATLAB. This allows the computational functions implemented within the mTRF-Toolbox package to be run in a MATLAB environment. The output’s generated by the mTRF-Toolbox are then returned to the Python/Flask environment and saved to the server. The full extent of the MATLAB built in plotting functions are used to visualise the data and then directly exported to PNG files which are displayed in the user interface. In terms of the front-end user interface of the neural analysis platform, a responsive web-page was developed to collect the parameters selected by the user data, interact with the RESTful API, and display the resulting graphs. The design of the user interface went through multiple iterations over time as feedback was gained from both expert and non-expert users. The development of the UI required the use of JavaScript, HTML and CSS to create custom react components that could be dynamically populated and altered depending on the data-set that the user selected to be analysed on the platform.

In order to put the neural analysis platform into production, changes were made to the initial architecture in order to deploy it online. AWS Machines with MATLAB pre-installed on the Ubuntu Linux operating system were a requirement in order to run the platform. To deploy the architecture, a server with an AMD CPU was first provisioned from AWS. Specifically, the AWS EC2 Cloud service was used to provision a m5.large EC2 instance type with 16 GiB memory, a 64 virtual core CPU and 20 GB of ethernet. The provisioned server was available as a MATLAB Production Server, a paid application server offered by MathWorks for integrating MATLAB analytics into web applications. This enabled the deployment of the neural Analysis platform to the server without the need to create a custom infrastructure. By combining the elasticity of the AWS infrastructure with MATLAB Production Server the platform has the capabilities to support large numbers of users as well as a sizable amount of requests simultaneously. The software cost charged for the use of the MATLAB production server is ran on a cVPU/per hour basis. The total amount that is charged for running the web app is based on the number of vCPUs active across the EC2 instance. The benefit of using AWS is that it allows the creation, configuration, and
deployment of a scalable, highly-available MATLAB Production Server environment. It should be noted that in order to minimise the cost of running the web application the author will obtain use of the Trinity College Dublin, Campus-Wide License (TAH License) in order to avoid incurring the costs of running the server.

In order to store large neurophysiological data files such as MEG and EEG files, storage will need to be provisioned for the project. Around 20.5 GB will be used after the server is fully configured (based on estimations of initial dataset offerings). Larger amounts of additional storage will be required for storing EEG data to be used in the platform as more users make their high quality neural data publically available for use on the platform and the data is converted to the CND format. To this end, additional storage can be provisioned in the future for an added cost.

In order to connect to the server, an automatically generated key pair was used to make the initial ssh connection to the server, and an additional user was created with password authentication enabled. The developer’s IP Address was added to the allow list for RDP connections to enable graphical remote access to the server which is required for interacting with MATLAB activation prompts. Windows Remote Desktop Connection was used to connect to the server and activate MATLAB after opening it for the first time and to complete the license check after starting new instances of the API.

![Cloud Architecture using MATLAB Production Server](image)

Figure 3.8: Cloud Architecture using MATLAB Production Server

The Neural Analysis Platform enables users to access many of the functionalities available within the mTRF-Toolbox MATLAB package. The features included in the Neural Analysis Platform include forward/backward modelling, efficient cross-validation, decoder-encoder transformations and single-lag analysis. In order to model the stimulus-response mapping of
the user selected dataset within the platform, the backend calls the mTRF-Toolbox packages built-in-function `mTRFtrain`. The `mTRFtrain` computes either univariate or multivariate ridge regression. Also known as Tikhonov regularization, ridge regression is useful for mitigating the problem of multicollinearity in linear regression by estimating the coefficients of multiple-regression models in scenarios where linearly independent variables are highly correlated such as mapping the stimulus and the response. There are two distinct methods by which the model can be trained on the selected dataset which both yield the same result because they are based on a linear assumption. The first is by training the model separately on each individual trial and then averaging over the total number of trials. The second method is by training on a concatenation (series of joined trials) of trials.

Figure 3.9: Schematic of forward and backward modeling approaches implemented using mTRF Toolbox

### 3.6 Model Training & Testing

The `mTRFtrain` function required that both the stimulus and response data have equivalent lengths, sampling rates (Hz) as well as be the same lengths of time. The `mTRFtrain` function simultaneously trains on all of the features within the data such as frequency bands or response channels. Additionally, the function requires minimum and maximum time lags entered as input parameters which are converted to samples based on the aforementioned sampling rate. Without user input the platform automatically optimises the stimulus-response mapping by using cross validation implemented with the mTRF-Toolboxes `mTRFcrossval` function. To this end the function identifies the aforementioned ridge regression parameter that optimises the stimulus-response mapping. Additionally, both the input and output data are normalised to aid in optimising the cross-validation function. Through the process of z-scoring the data, also known as standard score, the range of values required to optimise parameters can be significantly reduced. A z-score is a numerical
indication of how far the from the mean a data point and requires the knowledge of the mean and standard deviation in order to be computed. An optimised model is generated after tuning the model parameters using cross validation. Subsequently testing is performed on data that was held out from the training using the \textit{mTRFpred} function. A benefit of utilising the cross validation metrics averaged across trials is that the model is not influenced by individual trials and as a result the model performance when tested on new unseen data has a very high probability of resulting in similar results to that of the training results. To this end, when the \textit{mTRFpred} function generates performance metrics, it outputs a measure for each individual feature in a multivariate signal. In the case of the neural analysis platform implementation, the multivariate signal will generally be EEG electrophysiology data and the primary feature will be an EEG channel. In this respect the decision process on how to examine the models performance is determined by the choice of feature that the user selects to examine.

```
model{sub} = mTRFtrain(env(train_trials), alleeg.data(train_trials), alleeg.fs, ...
    dirTRF, tmin, tmax, lambdas(bestLambda), 'verbose', 0);
subAverages=[subAverages, model{sub}]
[pred, stats] = mTRFpredict(env(test_trials), alleeg.data(test_trials), model{sub});
test_r(:,:,sub) = stats.r;
```

Figure 3.10: \textit{mTRFtrain} and \textit{mTRFpredict} code using optimal hyperparameter (lambda)

### 3.6.1 Univariate TRF/STRF estimation

Methods of mapping sensory stimuli, such as speech or any other stimulus to neural data such as EEG data include generating spectro-temporal response functions (STRF) and temporal response functions (TRF). Univariate TRF estimation is offered by neural analysis platform and implemented with the \textit{mTRFplot} function. Additionally, the user has the option to graph the global field power (GFP), an index which has become a commonly used parameter for the temporal analysis of EEG sequences, either to find latencies or identify the maximum electric field strength across the scalp at each time lag (Michel et al., 1993). Based on the temporal profile of the GFP, the scalp typography can be analysed in order to ascertain where the strongest source of the evoked response is across the brain at different time lags. GFP can be computed as the mean of all of the potential differences in the field corresponding to the spatial standard deviation (Skrandies, 1990).
3.6.2 Multivariate TRF/STRF estimation

In addition to univariate TRF estimation, the neural analysis platform allows users to select multivariate estimations of continuous neural data in which is referred to as an mTRF. The mTRF analyses the data by filtering it logarithmically in order to generate a frequency analysis of the auditory periphery. The mTRF is calculated using user defined lags between -0.5s to 1s. Using the mTRF graph visualisations, the user can see the dominant encoded speech information at every frequency band up to 6 kHz, the frequency band which generally contains the most encoded information. By averaging the mTRF data across each of the frequency bands, a univariate TRF measurement is generated which closely approximates the TRF that was calculated using the stimulus envelope as input. Studies have demonstrated that while multivariate TRF models have increased sensitivity to regularisation and require more computational power, they have a greater performance than univariate TRF Models for predicting the evoked response from the stimulus (Di Liberto et al., 2015).

3.6.3 Stimulus Reconstruction

As previously mentioned, the neural analysis platform can also model the stimulus response mapping in the opposite direction (backwards modelling). The ability to construct backwards models offers users an alternative method to analyse how stimulus features can be encoded in a neural response. The decoding framework creates a model to translate the data from the response to the stimulus and then use it to estimate the stimulus input that created the response. This method is known as stimulus reconstruction and enables the user to decode or rebuild stimulus features from the evoked brain response. The decoder is calculated using ridge values and user determined lag values from -0.5s to 1s. In this case the EEG response data is the input of the model rather than the stimulus. Likewise the stimulus is the output of the model and the direction of the user determined lags are reversed (-1s to 0.5s).

```matlab
% Train model
model = mTRFtrain(stim_train, eeg_train, eeg.fs, Dir, tmin, tmax, lambda, 'zeropad', 0);

% Test model
[pred, test] = mTRFpredict(stim_test, eeg_test, model, 'zeropad', 0);

% Plot reconstruction
subplot(2,2,3)
plot((1:length(stim_test(1)))/eeg.fs, stim_test(1), 'linewidth', 2), hold on
plot((1:length(pred(1)))/eeg.fs, pred(1), 'linewidth', 2), hold off
xlim([0,10]) title('Reconstruction') xlabel('Time (s)') ylabel('Amplitude (a.u.)') axis square, grid on
legend('Orig', 'Pred')
```

Figure 3.11: Stimulus reconstruction code
4 Neural Analysis Platform Use Case Scenarios

In order to demonstrate the use of the Neural Analysis Platform within the context of the real world environments in which the platform has been designed for, three hypothetical use cases scenarios are described below.

4.1 Scenario 1: Proposed study of cognitive development in babies using audio-visual stimuli

Figure 4.1: Scenario in which a research uses the neural analysis platform in order to conduct a preliminary study on cognitive development in babies using cartoons as a continuous audio-visual stimuli.

Early identification of babies and children who are at risk of having developmental or learning deficits is paramount to ensuring that all children can have fulfilling lives and attain their full cognitive potential. EEG has an extensive history of use within the cognitive neuroscience field in order to measure the cognitive and social development of babies and children while remaining a non-invasive, direct measure of brain activity with high temporal resolution. In recent years EEG has seen an increase in use due to the growth in availability of more portable, low-cost EEG systems enabling the cognitive and social development of
children to be analysed at a much greater scale than previously possible. Additionally, advancements in data analysis tools such as those available in the mTRF-toolbox MATLAB package allow those with programming experience to have access to industry leading models and analysis. However, due to disparities in research funding, these studies, which can require considerable sample sizes and follow up can be difficult to conduct and extract statistically significant results from.

To this end, A model scenario for the use of the Neural Analysis Platform is one in which a neurologist without programming knowledge would like to conduct a study on the cognitive development of babies. It is expedient and useful to synthesise existing publicly available high quality neural data obtained from past EEG studies in order to understand how much data will be required to conduct the study, how long the stimulus should be, how many subjects are required for statistically significant results etc. Unfortunately, there are very few high quality publically available EEG data sets with infant subjects as the participants. This is partly due to the challenges involved with assessing children of this age such as getting children to cooperate when coaxing them to complete an activity over an extended period of time. Additionally, the process of setting up the EEG electrodes and apparatus on younger participants can prove to be challenging over longer durations and babies require parental oversight over the course of the study. In respect of this the neurologist decides to study the cognitive development of the baby through having them watch cartoons on a monitor as a continuous audio-visual stimulus and then measure their neural response through using an EEG system. Having babies watch cartoons can be a very effective measure when conducting longer experiments as babies are much more likely to sit still and cooperate when they are being entertained by visual cues such as cartoons. As previously mentioned, the lack of publically available high quality EEG data online compounded with the specificity of the cartoon continuous stimulus has created the opportunity for the neurologist to use the neural analysis platform to create a simulated experiment before conducting the study. The neurologist would like to apply for a grant in order to fund and conduct the study as well as seek ethical approval. In order to assimilate the requirements of the study for a grant application the neurologist needs to run a pilot experiment using the EEG data that is available on the Neural Analysis Platform.

After gaining access the online website the neurologist can login to neural analysis platform. The homepage contains a navigation bar as well as an instructional page on how to use the platform with information on each of the configurable parameters available within the website. The neurologist can select the encoding page from the navigation bar and is presented with the configurable parameters for an encoding analysis. The neurologist can then select the most similar dataset available on the platform to that of the audio visual cartoons that will be presented to the subjects of the study. In this particular case the dataset which is most similar, while not equivalent is the natural speech dataset. The
natural speech dataset uses data from a study that measured the EEG responses of adult humans subject to continuous natural speech (Di Liberto et al., 2015). Specifically, the stimulus included the subject listening to 2 minute chunks of audio from an audio-book of a classic work of fiction read by a speaker. In this case EEG data was recorded using a 128-channel ActiveTwo system (BioSemi). Details on the study that generated the dataset can be found in Figure 4.2.

The speech dataset uses 2 minute chunks of audio book as the length of each trial. However, this length of trial is unsuitable for use in babies as their attention span can not normally be held for 2 minutes. In the neural analysis platform the author can select a different trial length such as 10/15 seconds. In this respect the neurologist can experiment with different trial lengths using the simulated data in order to find out the optimal trial length for use in a real life experiment as well as how many trials are required in order to yield statistically significant results. This could be referred to as a power analysis for sample size in which the statistical power of a test is the probability that the experiment will detect an effect that does actually exist. When performing hypothesis testing, a considerable amount of preplanning is required before conducting experiments to collect data. A crucial stage of the planning process is determining how much data needs to be collected by estimating the sample size of the study. Ideally, the neurologist would prefer a smaller sample size due to prohibitive cost and strain of conducting experiments with larger samples.
while still staying within the constraints of satisfying statistical significance criteria. In this case it would be preferential to spend less time with the subjects in order to yield the best cognitive response data while also obtaining statistically significant results. Of course, the experiment has never been run so it is very difficult to show the optimal number of subjects that is required in order to yield a sufficient level of statistical significance.

In the neural analysis platform the neurologist can run an experiment on 10 adult subjects. Through running the experiment in the platform the neurologist can then obtain the prediction correlation envelope and compare that to the mismatch correlation. This can then be adapted and continuously improved through modifying the number of subjects, the trial length and the number of trials to select the parameters that are the most statistically correlated. A very important consideration within this experiment is the SNR, comparing the level of a desired signal to the level of background noise. Extracting useful data from infant subjects in the presence of various amounts of irrelevant or distracting information (noise) is one of the most important elements of measuring cognitive development in children. Due to the SNR affecting contextual cuing differently in children and adults (Yang and Merrill, 2015) it is imperative to consider the potential that data recorded from babies may be noisier than data from adults. Furthermore the neurologist must evaluate would happen in a real world experiment if the SNR was slightly or significantly worse as the current experiment is based on the SNR of adults. In order to simulate real world conditions of capturing EEG data from infant subjects using the neural analysis platform the neurologist can download the data. They can then generate artificial noise in the data by using random EEG data and summing it to the EEG data of the subjects. This has the effect of making the SNR within the experiment three times smaller so that the author can identify the best parameters to achieve statistical significance in a real world trial with infant subjects. The neural analysis platform greatly benefits the neurologist in this case by enabling them to quickly test configurations for their experiment using high quality publicly available data.
4.2 Scenario 2: Study comparing patients with and without learning deficits as biomarkers for research

Figure 4.3: Scenario in which a psychologist or neuroscientist uses the EEG and the Neural Analysis Platform to aid in the diagnoses of learning deficits such as dyslexia by providing quantifiable metric based on actual brain response data compared to those without the deficit.

Invasive and non-invasive brain imaging techniques such as EEG can be used in order to identify unique patterns in the firing of neurons amongst cohorts with learning or developmental deficits that are of neurological origin. Learning deficits such as dyslexia can cause limitations in reading and writing ability while not affecting the intellect of the person with the deficit. Such hidden disabilities can very often be undetected throughout a persons lifetime as someone with dyslexia could be normal and healthy in every other observable way. Currently the diagnosing process of such learning deficits primarily focus on observing behavioural symptoms of the deficit in order to recognise the condition amongst the population. This method utilises standardised testing and has become ubiquitous as the primary method of dyslexia diagnosis. In recent years, invasive and non-invasive brain imaging techniques such as fMRI, EEG and MEG have been used in research as an alternative to behavioural methods in order to recognise patterns in the brains of those with learning deficits that differ from the neurological activity in the majority of the population (Perera et al., 2018). Dyslexia fundamentally affects the brain’s ability to decode words, which in turn impacts spelling performance as well as the development of reading fluency. It is imperative that such neurological conditions are identified at an early age in order to allow for interventions to be implemented early in the child’s development and learning. As those
with dyslexia have difficulties with learning due to impairment of the left hemisphere of the brain associated with language processing. EEG can provide insight into the unique dynamics of the human brain function (Lyon et al., 2003 Mohamad et al., 2016). Due to the increased affordability and portability of EEG systems, EEG has become one of the most popular brain imaging techniques used to identify indicators of dyslexia. It should be noted that brain imaging results are currently used purely for researcher purposes and are unable to act as a replacement for standardised techniques and practices used by medical practitioners in the domain. In this respect, another suitable use case of the neural analysis platform to aid researchers in identifying potential biomarkers for learning deficits like dyslexia.

Researchers can use the neural analysis platform to conduct investigations into how the neural response differs in those with dyslexia versus a control group who do not have dyslexia. While not currently possible, it is hoped that in the future medical professionals can make use of objective, quantifiable measurements of learning deficits that are not solely reliant on behavioural or standardised testing methods. The researcher can use a dataset comprising of brain response data from a study in which children between the ages of 4 and 10 with dyslexia listened to an audio of continuous natural speech for 5 minutes while non-invasive EEG data was recorded and then in a second session the child performed a multitude of cognitive tests on an iPad. Cognitive tests measure the subjects language level, their memory, intelligence as well as general attention skills. It is important to note that in order for the highest level of ecological validity, the subjects that participated in the experiment were a similar reading level and age in order to mitigate the potential for unrepresentative results. The researcher can then select a dataset from the same study except conducted on children in the control group. It would be expected that if a subject were to have an impairment such as dyslexia that their EEG data would reveal lower levels of entrainment related to processing speech. After logging on to the neural analysis platform, the researcher can navigate to the multi-encoding page. They can then select the aforementioned dyslexia study of children with dyslexia as the first dataset parameter and the same dyslexia study with the non-dyslexic control group as the second dataset parameter. The researcher can then input parameters such as how many subjects they would like to compare in each dataset as well as the time lags, sampling frequency etc.

After inputting the parameters of the experiment, the researcher can then use the neural analysis platform to generate a linear regression for a multivariate temporal response function on both the dyslexia and control groups, which is displayed within the user interface. Additionally, the cross-validation prediction accuracies will be displayed as well as the scalp topographies at varying time lags. The scalp topographies can be used to see the measurements of localisation of brain electrical activity and how they differ between both the dyslexia and control groups. Downloading the topographies and using them to conduct further analysis based on the type of stimulus that is used can reveal much deeper insights.
into the inner workings of the brain localised to specific spatio-temporal locations. For example we would anticipate that the area of the brain engaged when the subject is reading words would have a higher degree of difference between subjects with and without dyslexia. However, there may also be subtle differences in stimulus that test memory, recall and awareness. An additional approach available to the researcher in order to compare datasets is to obtain the EEG prediction accuracies for an individual subject, calculated using the other subjects within the dataset. This can be done by selecting the same dataset for both dataset parameters and then altering the subjects that are used in the experiment to select an individual participant. In this respect, the same could be done across datasets for comparisons of the dyslexia group and the control group by training the data on the dataset with dyslexia and then testing it on the dataset without dyslexia.

By using the neural analysis platform the researcher has the benefit of access to many distinct datasets of electrophysiological data gathered from studies on participants with dyslexia as well as control groups. In this scenario the neural analysis platform benefits the user by providing instant access to electrophysiological datasets of groups with and without a learning deficit that can be rapidly analysed and compared without the requirement of knowing how to code. In this respect, research into electrophysiological data as potential biomarkers for learning deficit can be accelerated while not yet a possibility.
4.3 Scenario 3: Future non-verbal patient imagined speech decoding

Figure 4.5: Scenario in which a neuroscientist uses the EEG and the Neural Analysis Platform to create a BCI in which a patient with locked-in syndrome or ALS imagined speech can be decoded to audio.

The ability to produce speech is a hierarchical mechanism that utilises the intention to speak with the synchronisation of the brain and vocal chords to produce articulable sounds.

Locked-in syndrome is a rare neurological disorder affecting the body in which there is total paralysis of all voluntary muscles except for the muscles that control the movements of the eyes. Individuals with locked-in syndrome are both conscious and awake, yet are unable to communicate through conventional methods due to their inability to produce movements (other than eye movement) or speak (aphonia) (Giacino et al., 1995; Branco et al., 2021). Locked-in syndrome is a neurological condition in which the inability to control muscles is caused by damage to the brain-stem (bottom part of the brain) which contains nerve fibers that control the ability to send signals from the brain to the rest of the body. Additionally, those who suffer from progressive neuromuscular diseases such as Amyotrophic Lateral Sclerosis (ALS), commonly known as Lou Gehrig’s disease, face similar issues in communication. Current methods of communicating with those who are affected by such ailments involve utilising brain-computer interfaces (BCI) to track eye-ball movements as well as visual and attentional correlations to build communication. This is a slow and tedious process generally operating at a few words per minute. It has been shown that mental imagery produces neural activation patterns that are similar to actual perception. For example, imagining that you are speaking activates the motor and posterior inferior frontal regions (Fiez and Petersen, 1998). Similarly, imagining that you are moving limb activates
the motor cortex and imagining an object or image activates the visual cortex etc. (Martin et al., 2014; Pylyshyn, 2002; Roth et al., 1996). Speech imagery in the form of imagined speech, inner speech etc. refers to the ability to "hear" speech internally through imagining activating the parts of the body that contribute to speech such as the tongue, lips and vocal chords (Brigham and Kumar, 2010). The use of non invasive and invasive brain imaging techniques has the potential to create a much faster speed of communication through the direct decoding of imagined speech from neural signals in the brain connected to a speech synthesizer. In this respect, the use of neural decoding models can predict auditory features that are experienced from imagined speech and have been used previously to predict continuous spectro-temporal features of speech (Mesgarani et al., 2009).

Figure 4.6: Resulting decoder graphs featuring: CV Accuracy, CV Error, Reconstruction Accuracy, Reconstruction Error as a function of time lags, Brain topographies at varying time lags.

While it is not yet possible to reconstruct the speech envelope in real time due to current technological limitations, a potential future use case of the neural analysis platform is one in which a neurologist or researcher uses the data that is publically available on the platform in order to see how stimulus features can be reconstructed from the neural response, to create a speech envelope which can be directly synthesised to sound. Instead, users can utilise neurophysiological data in order to train a neural decoding model from imagined speech in order to predict acoustic speech features. Studies have shown that auditory features of speech can be accurately reconstructed and used to identify individual words during listening activities (Pasley et al., 2012). In a possible use case scenario, a neurologist whom has a patient who is unable to communicate through verbal or non-verbal cues would be greatly
benefited by conducting research into the patient’s brain response evoked by a stimuli such as visual text. The user can select the ‘imagined speech’ dataset comprising of publicly available ECoG data collected from subdural electrodes on epilepsy patients before undergoing neurological procedures. During this study, text was visually displayed on a screen and subjects were asked to read the text aloud so that the spoken speech was recorded. The subjects were subsequently asked to read the same speech and imagine saying it aloud in their heads. Each trial lasted between 6 and 8 minutes and was repeated at the discretion of the subject. This study example is adapted from Martin et al., 2014. While patients with LIS cannot speak, the same experiment can be conducted on the patient by recording their imagined speech and a control using either invasive or non-invasive brain imaging techniques. To this end, within the platform the neurologist can configure a linear mapping between the neural response activity and the speech envelope using a backwards model. They can select the configuration parameters relevant to the experiment such as the time lag range, the sampling rate and the range of lambda values. By running the experiment with different configurations the neurologist can assess the reconstruction accuracy and error compared to the original speech as well as evaluate the model performance and cross-validation accuracy. Additionally, the neurologist can evaluate the predictive power of

![Figure 4.7: Spoken speech used to train and test neural decoding model. During imagined speech, the model trained on spoken speech is used to decode the imagined speech to neural activity. Imagined speech reconstruction is then compared to original spoken speech using STRFs. Adapted from Martin et al., 2014](image)

the neural decoding model to reconstruct the auditory features of speech. In this case the
A neurologist uses the spoken speech to train and test the accuracy of the neural decoding model's ability to reconstruct the features of speech. The reconstructed speeches' features and patterns are then compared to the original spoken speech envelope. During imagined speech, there is no sound output which prevents the building of decoding models directly from the imagined speech data. Alternatively, the decoding model that was trained on the spoken speech is used to decode the neural activity evoked from the imagined speech. After fitting the spoken speech model to the imagined speech data the performance of the imagined speech reconstruction is compared to the original spoken speech to evaluate it's accuracy. Comparisons can be conducted using spatio temporal response function (STRF) graphs available within the neural analysis platform which can be seen in Figure 4.7.

The results of the experiment can be assessed through the reconstruction accuracy produced by the platform. Additionally, the user can download the correlation coefficient between the reconstructed speech and the original speech. A neurologist may wish to further analyse the reconstruction by evaluating its ability to identify specific speech. For example, individual speech streams including important and meaningful words or phrases that do not require full sentences such as 'hungry' or 'go outside' might be more pertinent to evaluate initially than complex sentences. In this case the neural analysis platform benefits a neurologist by enabling them to measure the performance of decoding models on high quality electrophysiological data consisting of spoken and imagined speech. This allows the user to adjust configuration parameters and measure performance across different datasets.
5 Conclusion

5.1 Project Overview

This dissertation presents a web application framework that enables the processing of neurophysiological data within an online platform using advanced analysis techniques that are prevalent in the field. Through the use of the platform both expert and non-expert users are able to conduct cognitive experiments as predictive computational models and evaluate those predictions against electrophysiological brain responses. The outputs of such models are then displayed graphically to the user within the platform. Users can opt to download the models with their selected configurations in order to conduct further analysis as well as statistical tests. The data that has been made available in the platform is high-quality, publicly available, neural data that has been formatted in the CND format in order to promote standardisation of the datasets.

The objectives set out in the introduction of the dissertation are as follows:

1. Conduct a review of the relevant literature within the neuroscience field as well as ancillary fields such as psychology, medicine etc. in order to understand the current state-of-the-art for using neurophysiological data to conduct experiments.

2. Develop a web application framework that enables the processing neurophysiological data within an online platform using advanced analysis techniques prevalent in studies of neurocognitive studies.

3. Develop an interface for interacting with the web application framework that can be used to conduct cognitive experimentation. It is imperative that the user interface is interactive and visually appealing such that it can be easily used by both expert and non-expert users.

Research objective #1 was achieved in chapter 2. This chapter explored the historical origins of linear system signal processing, encoding and decoding paradigms, invasive and non-invasive means of collecting neurophysiological data as well as frameworks for continuous event neurophysiology experimentation. Inspection of relevant literature in the
field revealed the methodologies that are used in state-of-the-art studies and experiments that would inform the development of the neural analysis platform.

Chapter 3 saw the development of neural analysis platform that would lead to the accomplishment of Research objective #2. The examination of relevant research material in the prior chapter revealed the necessary features that web application would need to facilitate. To this end, the neural analysis platform supports forward/backward modelling, efficient cross-validation, decoder-encoder transformations and single-lag analysis. The platform leverages the Flask micro-web framework in conjunction with React and MATLAB production server in order to deliver a comprehensive and lightweight application that can support scaling in the future if required.

Research objective #3 was demonstrated within chapter 4 of the dissertation by detailing the scenarios in which varying users could utilise the neural analysis platform to test hypothesis and conduct cognitive experiments on high quality, publicly available neurophysiological data. Such scenarios featured non-technical users with objectives varying from conducting preliminary studies on cognitive development in children to imagined speech decoding in non-verbal patients. Additionally, the use of the platform to act as an objective biomarker for learning deficits is a particularly salient use case that has significant potential for use in clinical practice.

5.2 Future Work

The neural analysis platform has potential to be extended to add additional functionality to be more useful to expert and non-expert users alike. Generative models would be a useful addition to the platform’s current features in order to generate new data instances for use on the platform. Generative models are a subdivision of unsupervised machine learning and have great potential for use in the platform by automatically revealing patterns such that the model can be used to generate new data that could have plausibly been taken from the original dataset. To this end, users on the platform could create new datasets from high quality neurophysiological data that is already available on the platform. The generation of increased amounts of simulated data creates the opportunity to identify appropriate algorithms for big data analysis that will improve the state-of-the-art in brain modelling as well speech decoding. Examples of generative models include Gaussian Mixture Modelling and Naive Bayes. The latter of which works by summarising probability distributions of each input variable and output class. A probability of each possible outcome is then calculated for each variable and the independent probabilities are combined to predict the most likely outcome. Alternatively, a Generative Adversarial Networks (GAN) could be used as an architecture for training a generative model using deep learning. Furthermore, an additional feature that could be added to the platform is the ability to conduct cluster analysis on the
neurophysiological datasets. Potential cluster analysis techniques that could be implemented include canonical correlation analysis (CCA) and multiple canonical correlation analysis (MCCA). Specifically, MCCA is a multivariate model that can be used to extract frequency features from electroencephalography data. As the platform acquires more electrophysiology data, cluster analysis has the potential to offer increased performance over alternate approaches to modelling data.

5.3 Closing Remarks
The development of the neural analysis platform presented in this dissertation was successful within the bounds of its established objectives. It is hoped by the author that the platform will prove useful to expert and non-expert users, enabling them to conduct cognitive experiments as predictive computational models and evaluate those predictions against electrophysiological brain responses. Additionally, it is hoped that the platform will encourage the sharing of more high quality electrophysiological data online as well as foster increased collaboration amongst the cognitive neuroscience community by promoting the use of the CND format.
Bibliography


A Appendix

Below are extracts of relevant code used within the Neural Analysis Platform to process neurophysiological data and output graphs from user selected parameters.

A.1 Encoding MATLAB Code

```matlab
function matlab_script(dataset_var, chart_var, lambda_var1, lambda_var2, lag_var1, lag_var2, stim_set_var, nsubjects_var)
    % Directory names
    dataMainFolder = ['./mTRF_Toolbox-master/datasets/' dataset_var '/'];
    dataCNDSubfolder = 'dataCND/';
    msubtrfFolder = 'MultiSubTRFs/';
    % Loading Stim data
    stimFilename = [dataMainFolder dataCNDSubfolder 'dataStim.mat'];
    disp('Loading stimulus data: dataStim.mat')
    load(stimFilename, 'stim')
    env = stim.data(stim_set_var,:);

    % Use an individual subject for testing
    nSubs = nsubjects_var;
    ntrials = 8; % restrict the number of trials (generic models can be used with less data)
    niter = 100; % number of iterations to compute the null distribution

    % restrict the number of trials
    env = env(1:ntrials);
    all_sbj_eegs = cell(nSubs,1);
```
for sub = 1:nSubs % iterate over subjects
  % Loading preprocessed EEG
  preproc_eeg_fl = sprintf('/dataSubjects/dataSub%d',sub);
  % preprocessed EEG filename
  eegPreFilename = [dataMainFolder dataCNDSubfolder
    preproc_eeg_fl '.mat'];
  disp(['Loading preprocessed EEG data: ' preproc_eeg_fl])
  load(eegPreFilename, 'eeg')
  % restrict the number of trials
  eeg.data = eeg.data(1:ntrials);
  % add to the all-subject EEG structure
  all_sbj_eegs{sub} = eeg;
end

% Setup the envelopes and EEG for multi-subject modeling
disp('Preparing the data for multi-subject modeling...');
[alleeg, env, sub_tag] = setup_multisubject_modeling(
  all_sbj_eegs, env);

% get the number of channels
nchan = length(eeg.chanlocs);

% clear eeg stim all_sbj_eegs

% Truncation and normalization
% Truncate the trial lengths so that, in each trial, the EEG
% and envelope
% are the same length
disp('Truncating EEG and stimulus so they are the same
length...');
for tr = 1:length(env)
  envLen = size(env{tr},1);
  eegLen = size(alleeg.data{tr},1);
  minLen = min(envLen, eegLen);
  env{tr} = double(env{tr}(1:minLen,:));
  alleeg.data{tr} = double(alleeg.data{tr}(1:minLen,:));
end

% Filter the envelope above 1 Hz
% Filtering the envelope...
% hd = getHPFilt(eeg.fs, 1);
% Filtering EEG data
env = cellfun(@(x) filtfilterd(hd, x), env, 'UniformOutput', false);

% Normalising EEG data by subject
% clear tmpEnv tmpEeg
disp('Normalizing the eeg data...');
for sub = 1:nSubs
    % get the trials for a particular subject
    sub_idx = find(sub_tag == sub);
    % concatenate the data together for this subject
    tmpEnv = env{sub_idx(1)};
    tmpEeg = alleeg.data{sub_idx(1)};
    for tr = 2:length(sub_idx) % getting all values
        tmpEnv = cat(1, tmpEnv, env{sub_idx(tr)});
        tmpEeg = cat(1, tmpEeg, alleeg.data{sub_idx(tr)});
    end
    normFactorEnv = std(tmpEnv(:)); clear tmpEnv;
    normFactorEeg = std(tmpEeg(:)); clear tmpEeg;
    for tr = 1:length(sub_idx) % normalisation
        env{sub_idx(tr)} = env{sub_idx(tr)}/normFactorEnv;
        alleeg.data{sub_idx(tr)} = alleeg.data{sub_idx(tr)}/normFactorEeg;
    end
end

%% Model setup
% TRF hyperparameters
Tmin = lag_var1*1000;
Tmax = lag_var2*1000;
disp('entering lambda loop')
lambda_temp = lambda_var1;
if lambda_var1 == 0
    lambda_temp = 0.001;
end
lambdaArray = [lambda_temp];
iterator = lambda_temp;

while (iterator <= lambda_var2)
    iterator=iterator*10;
    lambdaArray=[lambdaArray, iterator];
end

if lambda_var1 == 0
    lambdaArray=[0, lambdaArray]
end

lambdas=lambdaArray
disp(lambdas)

nlambda= length(lambdas)
dirTRF = 1; % Forward TRF model

% Use leave—one—out to get testing accuracies
disp('Leave—one—subject—out testing ... ');
model = cell(nSubs,1);
test_r = NaN(ntrials,nchan,nSubs);
nullr = NaN(niter,nchan,nSubs);
dpr = NaN(nSubs,nchan);
subAverages=[]
for sub = 1:nSubs
    fprintf('** Leaving out subject %d
','sub);
    % Get the testing trials (the trials for the particular subject)
test_trials = find(sub_tag==sub);
    % The rest are training trials
    train_trials = setxor(1:length(env),test_trials);

    % TRF — Compute model weights
    disp('Running mTRFcrossval ')
    [stats_cv, t] = mTRFcrossval(env(train_trials), alleeg.data(train_trials), alleeg.fs,...
        dirTRF, tmin, tmax, lambdas, 'verbose', 0);
% average r across channels and trial to find the optimal lambda value
[maxR, bestLambda] = max(squeeze(mean(mean(stats_cv.r,1),3)));
fprintf('Best r = %.3f
',maxR);

disp('Running mTRFtrain')
model{sub} = mTRFtrain(env(train_trials), alleeg.data(train_trials), alleeg.fs,
   dirTRF, tmin, tmax, lambdas(bestLambda),'verbose',0);  
subAverages={subAverages, model{sub}}

%%% mTRFpredict on left−out subject
% Testing on left−out trial
[pred, stats] = mTRFpredict(env(test_trials), alleeg.data(test_trials), model{sub});
test_r(:, :, sub) = stats.r;

% Compute the null distribution for this subject
nullr(:, :, sub) = compute_null_predacc(pred, alleeg.data(test_trials), tmin, tmax, alleeg.fs,
   'circshift', niter);
end
fprintf('
');

% Save the multisubject models
save([dataMainFolder msubtrfFolder 'MultiSubTRF'], 'model', 'test_r', 'nullr', ...
   'nSubs', 'ntrials', 'tmin', 'tmax');

% Get the testing trials (the trials for the particular subject)
test_trials = find(sub_tag==sub);
% The rest are training trials
train_trials = setxor(1:length(env), test_trials);
% c. Run fast cross−validation
disp('Running cross-validation...')
[stats_cv, t] = mTRFcrossval(env(train_trials), alleeg.
data(train_trials), alleeg.fs, ...
dirTRF, tmin, tmax, lambdas, 'verbose', 0);

% g. Plot CV accuracy
 cv_path = append('/Users/rossmccrann/5th Year/neural-
   platform-flask-react/neural-platform/src/Images/Graphs/
   Encoding/Two/cv_accuracy/', dataset_var, '_', chart_var,
   '_', string(lambda_var1), '_', string(lambda_var2), '_'
   , string(lag_var1), '_', string(lag_var2), '_', string(
   stim_set_var), '_', string(nsubjects_var), '.png')

hold on
ax = gca;
ax.FontSize = 16;
set(gcf,'Visible','off')
rr = mean(stats_cv.r,3); % rr: nSub x nLambdas

errorbar(1:numel(lambdas),mean(rr,1), std(rr)/sqrt(numel(
   env(train_trials))), 'linewidth',2)

set(gca,'xtick',1:nlambda,'xticklabel',lambdas), xlim([0,
   nlambda+1])

yl = ylim; % Get current limits.
ylim([0, yl(2)]); % Replace lower limit only with a y of 0.
title('Cross Validation Accuracy')
xlabel('Lambdas')
ylabel('Correlation')
axis square, grid on
my_plot_cv = gcf

disp(cv_path)
exportgraphics(my_plot_cv, cv_path, 'Resolution', 300)

% Plot the TRF fit to the training trials
modelAve = mTRFmodelAvg(subAverages)
% Plot the TRF fit to the training trials
chan_to_plot = 64; % this corresponds to Fz in the Natural Speech dataset

figure

% Generate plot
switch chart_var
  case {'mtrf', 'mgfp'}
    xlims = [modelAve.t(1), modelAve.t(end)];
    [~, lag_var1_] = min(abs(modelAve.t-xlims(1)));
    [~, lag_var2_] = min(abs(modelAve.t-xlims(2)));
    lags = lag_var1_:lag_var2_;
  end

feat = 1:size(modelAve.w,1);

switch chart_var
  case 'trf'
    h = plot(modelAve.t, squeeze(modelAve.w(:, :, chan_to_plot)), 'LineWidth',2); grid on
    title('TRF (Fz)'), ylabel('Amplitude (a.u.)'), xlabel('Time Lag (ms)');
    legend('Dataset 1')
    ax = gca;
    ax.FontSize = 16;
    set(gcf,'Visible','off')
  case 'gfp'
    h = area(modelAve.t, squeeze(modelAve.w(:, :, chan_to_plot)), 'edgecolor','none'); grid on
    title('Global Field Power'), ylabel('Amplitude (a.u.)'), xlabel('Time Lag (ms)');
    legend('Dataset 1')
    ax = gca;
    ax.FontSize = 16;
    set(gcf,'Visible','off')
  case {'mtrf'}
h = imagesc(modelAve.t(lags),1:numel(feat),squeeze(modelAve.w(:,:,chan_to_plot)));
set(gca,'ydir','normal')
title('STRF (Fz)'), ylabel('Frequency band'), xlabel('Time Lag (ms)')
legend('Dataset 1')
ax = gca;
ax.FontSize = 16;
set(gcf,'Visible','off')

case {'mgfp'}
h = imagesc(modelAve.t(lags),1:numel(feat),squeeze(modelAve.w(:,:,chan_to_plot)));
set(gca,'ydir','normal')
title('Global Field Power'), ylabel('Frequency band'), xlabel('Time Lag (ms)')
legend('Dataset 1')
ax = gca;
ax.FontSize = 16;
set(gcf,'Visible','off')
end

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Encoding/One/','dataset_var','_','chart_var','_',string(lambda_var1),'_',string(lambda_var2),'_',string(lag_var1),'_',string(lag_var2),'_',string(stim_set_var),'_',string(nsubjects_var),'.png')

exportgraphics(my_plot, export_var, 'Resolution', 300)

%lim = max(max(abs(modelAve.w(:,7:14))));
%
subplot(2,2,3)
topoplot(modelAve.w(:,11), eeg.chanlocs)
%caxis([-lim, lim])
topoplot(modelAve.w(:,10), eeg.chanlocs, 'maplimits', [-lim, lim], 'whitebk', 'on')
title([num2str(modelAve.t(10)), ' ms'])

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Encoding/Three/', dataset_var, '_', chart_var, '_', string(lambda_var1), '_', string(lambda_var2), '_', string(lag_var1), '_', string(lag_var2), '_', string(stim_set_var), '_', string(nsubjects_var), '.png')

pause(3) % wait time to save image

plot_topography()

if download
download_string = append(dataset_var, csv)
writematrix(modelAve.t, '.csv')
end

clfall()

% for loop with labels of electrodes
% eeg.chanlocs.
% eeg.chanlocs
% c. Plot decoder weights
%lim = max(max(abs(modelAve.w(:,7:14))));
%figure(2)
%subplot(2,2,1)
%topoplot(modelAve.w(:,7), eeg.chanlocs, 'maplimits', [-lim, lim], 'whitebk', 'on')
%title([num2str(modelAve.t(7)), ' ms'])
A.2 Decoding MATLAB Code

```matlab
function matlab_script()

    % 1. Data ingestion
    close all;
    clear; clc;
```
% c. Load data

disp('Loading data...')
load('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/flask_api/custom_matlab/mTRF-Toolbox-master/datasets/LalorNatSpeech/dataCND/dataStim.mat', 'stim');
load('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/flask_api/custom_matlab/mTRF-Toolbox-master/datasets/LalorNatSpeech/dataCND/dataSub10.mat', 'eeg');

%% 8. DRYAD data preparation for Subject 10

% a. Use envelope feature
stim.data = stim.data(1,:);

% b. Crop EEG to match stim length
for i = 1:numel(eeg.data)
    eeg.data{i} = eeg.data{i}(1:length(stim.data{i}),:);
    eeg.extChan{1,1}.data{i} = eeg.extChan{1,1}.data{i}(1:length(stim.data{i}),:);
end

% c. Set up highpass filter
highpass_cutoff = 0.1;
highpass_order = 3;
hd_hp = getHPFilt(eeg.fs, highpass_cutoff, highpass_order);

% d. Set up lowpass filter
lowpass_cutoff = 8;
lowpass_order = 3;
hd_lp = getLPFilt(eeg.fs, lowpass_cutoff, lowpass_order);

% e. Filter EEG recording channels
disp('Filtering recording channels...')
eeg.data = cellfun(@(x) filtfilt(hd_hp, x), eeg.data, 'UniformOutput', false);
eeg.data = cellfun(@(x) filtfilt(hd_lp, x), eeg.data, 'UniformOutput', false);
% f. Filter EEG external channels

disp('Filtering external channels...')

eeg.extChan{1,1}.data = cellfun(@(x) filtfilt(hd_hpf, x), eeg.extChan{1,1}.data, 'UniformOutput', false);
eeg.extChan{1,1}.data = cellfun(@(x) filtfilt(hd_lpf, x), eeg.extChan{1,1}.data, 'UniformOutput', false);

% g. Downsample data

fs_new = 64;
disp('Downsampling data...')
eeg = cndDownsample(eeg, fs_new);
stim = cndDownsample(stim, fs_new);

% h. Interpolate bad channels
disp('Interpolating bad channels...')
if isfield(eeg, 'chanlocs')
    for i = 1:numel(eeg.data)
        eeg.data{i} = removeBadChannels(eeg.data{i}, eeg.chanlocs);
    end
end

% i. Re-reference EEG data
disp('Re-referencing EEG data...')
eeg = cndReref(eeg, 'Avg');

% j. Normalize EEG data
disp('Normalizing data...')
eeg_data_mat = cell2mat(eeg.data);
eeg_std = std(eeg_data_mat(:));
eeg.data = cellfun(@(x) x/eeg_std, eeg.data, 'UniformOutput', false);

%% 3. Cross-validation

% a. Define training and test sets
test_trials = 10:13; % 20% of data
stim_train = stim.data;
eeg_train = eeg.data;
stim_train(test_trials) = [];
eeg_train(test_trials) = [];
stim_test = stim.data(test_trials);
eeg_test = eeg.data(test_trials);

% b. Model hyperparameters
Dir = -1;
tmin = 0;
tmax = 250;
lambda_vals = 10.^(−2:2:8);
nlambda = numel(lambda_vals);

% c. Run fast cross-validation
disp('Running cross-validation...')
(cv = mTRFcrossval(stim_train, eeg_train, eeg.fs, Dir, tmin, tmax,
    lambda_vals, ...
    'zeropad',0,'fast',1));

% d. Plot CV accuracy
figure(1)
subplot(2,2,1)
errorbar(1:nlambda, mean(cv.r), std(cv.r)/sqrt(numel(stim_train)), 'linewidth',2)
set(gca, 'xtick', 1:nlambda, 'xticklabel', −2:2:8), xlim([0, nlambda+1])
title('CV Accuracy')
xlabel('Regularization (1\times10^{\lambda})')
ylabel('Correlation')
axis square, grid on

% e. Plot CV error
subplot(2,2,2)
errorbar(1:nlambda, mean(cv.err), std(cv.err)/sqrt(numel(stim_train)), 'linewidth',2)
set(gca, 'xtick', 1:nlambda, 'xticklabel', −2:2:8), xlim([0, nlambda+1])
title('CV Error')
xlabel('Regularization ($1\times10^\lambda$)')
ylabel('MSE')
axis square, grid on
set(gcf,'Visible','off')

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Decoding/One/1', '.png')

exportgraphics(my_plot, export_var, 'Resolution', 300)

%% 4. Model training

% a. Get optimal hyperparameters
[rmax, idx] = max(mean(cv.r));
lambda = lambda_vals(idx);

% b. Train model
disp('Training model...')
model = mTRFtrain(stim_train, eeg_train, eeg.fs, Dir, tmin, tmax,
          lambda,...
          'zeropad',0);

% c. Plot decoder weights
lim = max(max(abs(model.w(:,7:14))));
figure(2)
subplot(2,2,1)
topoplot(model.w(:,7),eeg.chanlocs,'maplimits',[-lim,lim],'
          whitebk','on')
title([num2str(model.t(7)),' ms'])
subplot(2,2,2)
topoplot(model.w(:,9),eeg.chanlocs,'maplimits',[-lim,lim],'
          whitebk','on')
title([num2str(model.t(9)),' ms'])
subplot(2,2,3)
topoplot(model.w(:,11),eeg.chanlocs,'maplimits',[-lim,lim],'
          whitebk','on')
title([num2str(model.t(11)),' ms'])
subplot(2,2,4)
topoplot(model.w(:,14),eeg.chanlocs,'maplimits',[-lim,lim],'whitebk','on')
title([num2str(model.t(14)),' ms'])
set(gcf,'Visible','off')

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Decoding/Two/2','.png')

exportgraphics(my_plot, export_var, 'Resolution', 300)

%% 5. Model testing

% a. Test model
disp('Testing model...')
[pred, test] = mTRFpredict(stim_test, eeg_test, model,'zeropad',0);

% b. Plot reconstruction
figure(1)
subplot(2,2,3)
plot((1:length(stim_test{1}))/eeg.fs,stim_test{1},'linewidth',2), hold on
plot((1:length(pred{1}))/eeg.fs,pred{1},'linewidth',2), hold off
xlim([0,10])
title('Reconstruction')
xlabel('Time (s)')
ylabel('Amplitude (a.u.)')
axis square, grid on
legend('Orig','Pred')

% c. Plot test correlation
subplot(2,2,4)
bar(1,rmax), hold on
bar(2,mean(test.r)), hold off
set(gca,'xtick',1:2,'xticklabel',{'Val.', 'Test'})
title('Model Performance')
xlabel('Dataset')
ylabel('Correlation')
axis square, grid on

%% 6. Single—lag stimulus reconstruction

% Run single—lag cross—validation

tmin = −250; tmax = 500;
[stats,t] = mTRFcrossval(stim_train, eeg_train, eeg.fs, Dir,
    tmin, tmax, lambda, ...
    'type', 'single', 'zeropad', 0);

% Compute mean and variance
macc = squeeze(mean(stats.r))'; vacc = squeeze(var(stats.r))';
merr = squeeze(mean(stats.err))'; verr = squeeze(var(stats.err))';

% Compute variance bound
num_folds = numel(stim_train);
xacc = [-flipud(t),-t]; yacc = [flipud(macc-sqrt(vacc/num_folds)),macc+sqrt(vacc/num_folds)];
xerr = [-flipud(t),-t]; yerr = [flipud(merr-sqrt(verr/num_folds)),merr+sqrt(verr/num_folds)];

% Plot accuracy
figure(3)
subplot(1,2,1), h = fill(xacc, yacc, 'b', 'edgecolor', 'none');
hold on
set(h, 'facealpha', 0.2), xlim([tmin, tmax]), axis square, grid on
plot(-flipud(t), flipud(macc), 'linewidth', 2), hold off
title('Reconstruction Accuracy'), xlabel('Time lag (ms)'), ylabel('Correlation')

% Plot error
subplot(1,2,2)
h = fill(xerr,yerr,'b','edgecolor','none'); hold on
set(h,'facealpha',0.2), xlim([tmin,tmax]), axis square, grid
plot(fliplr(t),fliplr(merr),'linewidth',2), hold off
title('Reconstruction Error'), xlabel('Time lag (ms)'), ylabel('MSE')

set(gcf,'Visible','off')

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Decoding/Three/3', '.png')
exportgraphics(my_plot,export_var,'Resolution',300)

7. Model performance with less data

% a. Define training and test sets
test_trials = 10:13;
stim_train = stim.data(1:3);
eeg_train = eeg.data(1:3);
stim_test = stim.data(test_trials);
eeg_test = eeg.data(test_trials);

% b. Model hyperparameters
Dir = -1;
tmin = 0;
tmax = 250;
lambda_vals = 10.^(−2:2:8);
nlambda = numel(lambda_vals);

% c. Run fast cross-validation
disp('Running cross-validation...')
cv = mTRFcrossval(stim_train,eeg_train,eeg.fs,Dir,tmin,tmax,
                 lambda_vals,...
                 'zeropad',0,'fast',1);

% d. Get optimal hyperparameters
\([\text{rmax}, \text{idx}] = \max(\text{mean(cv.r)})\);
\(\lambda = \text{lambda_vals(idx)}\);

\% e. Train model
\texttt{disp(’Training model...’)}
\(\text{model} = \text{mTRFtrain(stim_train, eeg_train, eeg.fs, Dir, tmin, tmax, lambda, ’zeropad’, 0)};\)

\% f. Test model
\texttt{disp(’Testing model...’)}
\([\text{pred}, \text{test}] = \text{mTRFpredict(stim_test, eeg_test, model, ’zeropad’, 0)};\)

\% g. Plot CV accuracy
\texttt{figure(4)}
\texttt{subplot(2,2,1)}
\texttt{errorbar(1:nlambda, mean(cv.r), std(cv.r)/sqrt(numel(stim_train)), ’linewidth’, 2)}
\texttt{set(gca, ’xtick’, 1:nlambda, ’xticklabel’, ’-2:2:8’), xlim([0, nlambda+1])}
\texttt{title(’CV Accuracy’)}
\texttt{xlabel(’Regularization (1\times10^\lambda)’)}
\texttt{ylabel(’Correlation’)}
\texttt{axis square, grid on}

\% h. Plot CV error
\texttt{subplot(2,2,2)}
\texttt{errorbar(1:nlambda, mean(cv.err), std(cv.err)/sqrt(numel(stim_train)), ’linewidth’, 2)}
\texttt{set(gca, ’xtick’, 1:nlambda, ’xticklabel’, ’-2:2:8’), xlim([0, nlambda+1])}
\texttt{title(’CV Error’)}
\texttt{xlabel(’Regularization (1\times10^\lambda)’)}
\texttt{ylabel(’MSE’)}
\texttt{axis square, grid on}

\% i. Plot reconstruction
\texttt{subplot(2,2,3)}
plot((1:length(stim_test{1}))/eeg.fs, stim_test{1}, 'linewidth',2), hold on
plot((1:length(pred{1}))/eeg.fs, pred{1}, 'linewidth',2), hold off
xlim([0,10])
title('Reconstruction')
xlabel('Time (s)')
ylabel('Amplitude (a.u.)')
axis square, grid on
legend('Orig','Pred')

% j. Plot test correlation
subplot(2,2,4)
bar(1,rmax), hold on
bar(2,mean(test.r)), hold off
set(gca, 'xtick',1:2, 'xticklabel', {'Val.','Test'})
title('Model Performance')
xlabel('Dataset')
ylabel('Correlation')
axis square, grid on

set(gcf, 'Visible', 'off')

my_plot=gcf
export_var = append('/Users/rossmccrann/5th_Year/neural-platform-flask-react/neural-platform/src/Images/Graphs/Decoding/Four/4', '.png')

exportgraphics(my_plot, export_var, 'Resolution', 300)