DEEP NEURAL NETWORKS FOR REAL TIME MOTOR-IMAGERY EEG SIGNAL CLASSIFICATION

by

Ahmed Bahaaeldin Mohamed Selim

A thesis submitted in partial fulfilment of the degree of Doctor of Philosophy

School of Computing and Information Science
Faculty of Science & Engineering
Anglia Ruskin University,
Cambridge CB1 1PT, United Kingdom

March 2021

Acknowledgments

I would like to thank Dr Ian van der Linde for his support through one of the longest and hardest journeys of my life on both levels, academically and personally. I can't emphasise enough his assistance and guidance.

I am also very thankful to Dr Ken Revette who made me fall in love with AI and EEG when I was doing my bachelor's degree and inspired me to follow his footsteps. I would also like to thank Dr Doaa el Zanafly for her advice many years ago to believe in myself and chase my dreams.

It goes without mentioning, my father who supported me throughout my journey from the very beginning, the day I was born, and my mother for all her psychological support and advice to keep going.

I am also grateful for Anglia Ruskin University for providing me with the PhD studentship that allowed me to pursue my passion and provided me with a once in a lifetime opportunity. Finally, I couldn't be more grateful of having such amazing friends and their belief in me always provided me with the energy, motivation and will.

The truest of clichés stands, the most valuable thing in life are the people that you care about and they care about you. I have been blessed that way and wouldn't have been able to finish this thesis or even start without having all those people in my life. Forever grateful and always in my heart.

ABSTRACT

FACULTY OF SCIENCE AND ENGINEERING

DOCTOR OF PHILOSOPHY

DEEP NEURAL NETWORKS FOR REAL TIME MOTOR-IMAGERY EEG SIGNAL CLASSIFICATION

AHMED BAHAAELDIN MOHAMED SELIM

March 2021

The aim of this research is to develop a high-performance Motor Imagery (MI) classifier capable of using short signal intervals (0.8s) in an effort to move towards real-time performance for Brain-Computer Interfaces (BCIs). First, classification accuracy was investigated with different windows sizes and intervals and compared with baseline levels of performance with common existing methods, Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA), using both spatial and spectral features. It was found that spectral features could produce higher performance using shorter windows compared to spatial features. Next, a state-of-the-art Convolutional Neural Networks (CNN) was developed using the Continuous Wavelet Transformation (CWT), producing a novel Point-wise Convolutional Neural Network (PWCNN) that achieves performance very close to the state-of-the-art, namely 80% classification accuracy using the BCI IV 2b dataset operating on 2s intervals; however, random chance performance was found with the BCI IV 2a dataset. Next, to address the limitations of the PWCNN, a hybrid deep model was developed based on best practice CNNs and Recurrent Neural Networks (RNN). It incorporated novel spatial and temporal attention mechanisms, and is called Convolutional Recurrent Neural Network with Double Attention (CRNN-DA). This model was found to yield 73% classification accuracy and 60% kappa using the BCI IV 2a dataset, which is 3% higher than the winner of the BCI IV 2a competition. A generalisation of the Guided Grad-CAM method suited for EEG signals is also proposed to provide model decision interpretability, which may enable further optimisations to be made. In addition, a novel EEG augmentation technique, to be called *shuffled-crossover*, is proposed to address the issue of having small datasets for network training. As a consequence of increasing the number of training samples, this approach was found to elicit a further 3% increase in classification accuracy using the CRNN-DA. The suggested model (CRNN-DA) and methods move us closer to realising the aim of practical BCIs capable of responding to multiple input classes in real-time. The proposed double attention mechanism can serve as a feedback loop for data collection, enabling data reflecting user inattention, that may otherwise reduce training efficiency, to be rejected pre-emptively. The proposed augmentation technique can be used to reduce the quantity of training data required. The proposed modified Grad-CAM technique offers an insight into model decisions (viz., model interpretability) that may enable future performance enhancements to be identified more easily.

TABLE OF CONTENTS

Copyright Declaration

DEEP NEURAL NETWORKS FOR REAL TIME MOTOR-IMAGERY EEG SIGNAL CLASSIFICATION

Ahmed Selim

2021

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
BCI	Brain Computer Interface
BP	Band Power
CNN	Convolutional Neural Network
CSP	Common Spatial Patterns
CRNN-DA	Convolutional Recurrent Neural Network with Double Attention
CWT	Continous Wavelet Transform
DWT	Discrete Wavelet Transform
EEG	Electroencephalogram
ERD	Event Related Desynchronisation
ERP	Event Related Potential
ERS	Event Related Synchronisation
FBCSP	Filter Bank Common Spatial Pattern
FFT	Fast Fourier Transform
Grad-CAM	Gradient-weighted Class Activation Mapping
GRU	Gated Recurrent Unit
ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
LSTM	Long-Short Term Memory
MI	Motor Imagery
NBPW	Naïve Bayes Parzen Window
OVR	One Versus Rest
PCA	Principle Component Analysis
PSD	Power Spectral Density
PWCNN	Point-wise Convolutional Neural Network
QPP	Quadratic Programming Problem
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
SBCSP	Sub-Band Common Spatial Pattern
SBM	Stacked Boltzmann Machine

SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
WPD	Wavelet Packet Decomposition

List of Equations

Chapter 1:

Background and Related Work

New computer interfaces are continuously developed to make interacting with machines faster and easier. Most interface modalities have not been designed with disability in mind; for instance, mice, trackpads, keyboards, and the currently popular approach of interpreting finger strokes on touch screens (Preece, 2002). These interfaces require physical interaction, precluding their use by those with significant motor impairments, such as those who have impaired motor control as a consequence of neurological disease, brain trauma, limb injury or loss.

A Brain-Computer Interface (BCI) is a system that enables communication between humans and electronic devices in which the input is acquired directly from the brain activity of the user. Since these systems don't require any motor activity, they can provide a means for those who suffer from motor impairments (including, in the most extreme case, those who are entirely locked that have no other method to interact with the external world). BCIs provide a means for motor impaired or otherwise disabled users to communicate with the external world and control devices with their thoughts (Yi et al., 2013).

Traditional robotic prosthetics are effective when the nerve connections of the muscles are mostly functional since they operate by recording the electrical signals in muscles. These technologies are complicated and expensive, and may only be used in patients with functional nerve connections (Bright, Nair, Salvekar and Bhisikar, 2016). Conversely, the technology that underpins BCIs in inexpensive, and where electroencephalogram (EEG) devices are used, signals are recorded directly from the scalp meaning that distal muscular electricity activity does not need to be preserved. Muller-Putz and Pfurtscheller (2008) implemented a BCI-controlled robotic arm with four control signals to represent the four different motor functions of the arm: left movement, right movement, hand open, and hand close. More recently, a low cost BCI controlled prosthetic was proposed by Elstob and Lindo Secco (2016) that has five movements and uses a 3D-printed prosthetic arm. BCI systems have shown encouraging results in rehabilitating those who have suffered a stroke. In Gomez-Rodriguez et al. (2011) and Abiri et al. (2017), patients were instructed to imagine performing movements that were then executed by robotic arms to provide haptic feedback to the patients (with a small delay), which was found to stimulate recovery. Luu, Nakagome, He and Contreras-Vidal (2017) and McMahon and Schukat (2018) combined BCI and virtual reality for post-stroke rehabilitation, rather than using a robotic arm, thereby providing visual rather than haptic feedback, but found this to be similarly useful.

Even though several BCI systems have been developed, there are significant practical challenges that limit their reliability and responsiveness. There is usually a trade-off between speed and classification performance; *viz*., for accurate classification, computationally intensive analyses of high-resolution signals is required, but for high performance (i.e., real-time responsiveness), a smaller quantity of data and relatively computational inexpensive analysis methods are needed. The majority of current BCI systems use EEG signals, which are inherently non-linear, non-stationary and prone to a number of artefacts. Processing these signals is challenging and requires that a constellation of procedures are used for even modest classification accuracy to be achieved. This is in addition to other practical challenges, such as the need for dimensionality reduction, source separation, identifying suitable features, and overcoming the difficulties of effective training caused by small quantities of training data and signals that differ markedly from person to person.

The overarching objective of this thesis is to improve the accuracy and speed of EEG-based BCI systems that use the motor imagery (MI) approach. Prior to presenting new methods, in the following sections previous literature is summarised in terms of methods employed (including discussing method limitations or shortcomings), focussing particularly (but not exclusively) on methods that have previously be used for EEG-based MI classification for BCI systems. Where available, the features concentrated upon by each existing method (e.g., spatial, spectral, spatiotemporal), the core classification method (e.g., SVM, LDA, ANN), and key performance metrics will be discussed. The performance of newly developed models/systems, to be presented in subsequent chapters, will be compared back to the systems described in this literature review.

1.1 Brain-Computer Interface and Motor Imagery

Depending upon the specific purpose of a BCI system (e.g., text input, wheelchair control, playing games), different electrophysiological signals will be used to classify user intentions. For instance, in the case of text input systems (sometimes called *word spellers*), the P300 component has been extracted from the EEG signals as it is feature that appears in the signal when a user focusses their attention on a letter and makes a decision. The P300 component is an example of what is known as an Event-Related Potential (ERP). Other transient signal features include Visual Evoked Potentials (VEPs), which are elicited when a light flash or pattern is seen (Pfurtscheller and Neuper, 2001). However, VEPs will not be discussed further in this thesis, since they do not typically correspond to decision making (intention) and are therefore not well suited for interface control.

In addition to the P300, another particularly useful ERP that can be readily captured in EEG signals is Event Related Desynchronisation (ERD) and Event Related Synchronisation (ERS). These events occur when a motor movement is performed but also when a motor movement is merely *imagined*. ERP/ERS BCI-based systems have been used for the rehabilitation of motor functions in a medical context (Graimann et al., 2002a; Lu et al., 2017; Pfurtscheller and Neuper, 1997), and provide a very promising avenue for further development to increase their real-time operability and classification accuracy for general-purpose human-computer interfaces.

1.2 Event Related Potentials (ERPs), Event Related Synchronisation (ERS) and Event Related Desynchronisation (ERD)

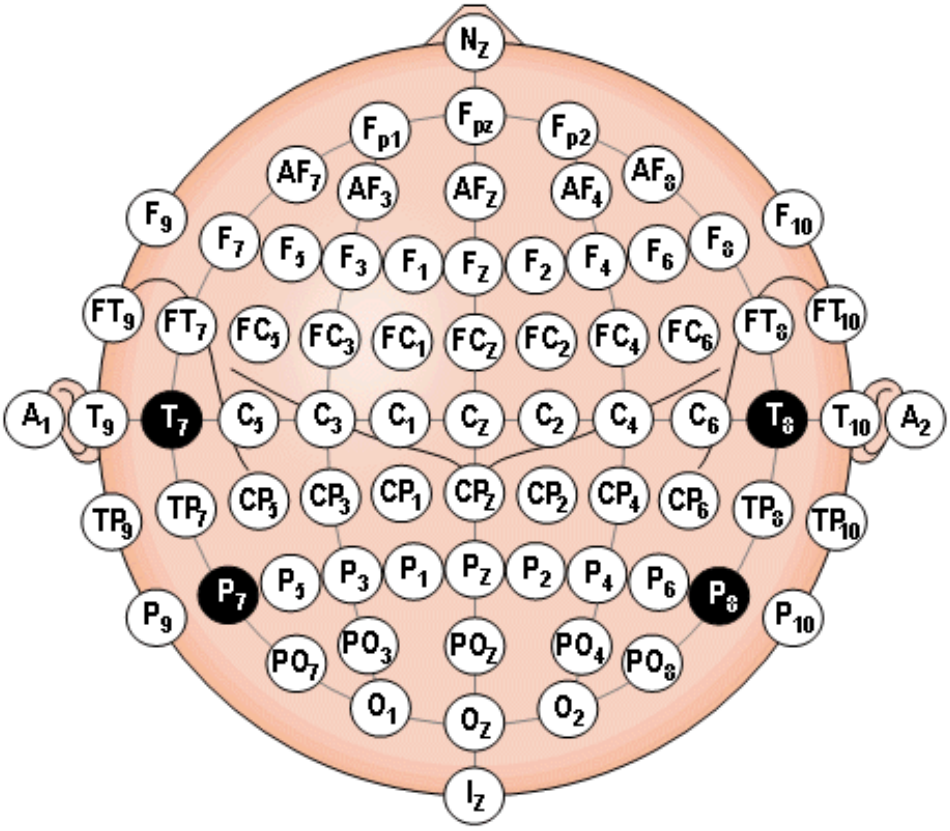


Figure 1.1: The 10-20 international system for electrode placement (from Sharbrough, 1991).

ERPs are time-locked events that correspond to a significant change in the activity of a population of neurons. There is a fixed time delay between the stimulus (such as an instruction or stimulus) and the evoked signal, and concurrent brain activity is considered noise that usually needs to be removed prior to signal interpretation. Averaging multiple signals may improve Signal-to-Noise Ratio (SNR).

Some stimuli, such as those in the visual domain, can attenuate the amplitude of the evoked signal, so the ERP approach is usable only in certain circumstances. The ERP model assumes two signals are added to each other, one is the signal of interest and the other is the noise (Pfurtscheller and Lopes, 1999).

The events that produce α (8-13 Hz) and β (13-30 Hz) signals can be detected as increases in power at these specific narrow frequency bands. However, analysing ERS/ERD for monitoring or controlling applications entails the identification of the frequency components most closely linked to the mental task performed, which also requires the cortical areas where it is more distinctive for the specific task to acquire unambiguous results that can be used as features by a classifier. Executing or imagining a movement has been found to induce an ERD in the sensorimotor cortex; the ERD is usually produced when planning the movement. Furthermore, the ERD of a left/right hand movement can be localised over the contralateral part of the sensorimotor cortex (left hand movement ERD would be measured over the right cortical region and *vice versa*) and can be detected in the α and β bands (Pfurtscheller and Neuper, 2001; Soman and Jayadeva, 2015).

The main sources of noise present in EEG signals originate from muscular activity, eye blinks, and nearby electrical devices. Capturing a signal of interest with good SNR is achieved by applying pre-processing procedures such as filtering, averaging, principle component analysis (PCA), and independent component analysis (ICA). These methods aim to eliminate unwanted artefacts and reveal a less contaminated approximation of the signal of interest. Next, feature extraction methods will aim to identify the most diagnostic signal characteristics (i.e., here, those that reveal user intentions), referred to as features (Al-Fahoum and Al-Fraihat, 2014; Lu and Yin, 2015).

1.3 Common Feature types

Powered samples

The most fundamental technique for extracting ERDs and ERSs from a signal to be used as the features for MI classification is averaging and obtaining power samples for selected frequency bands. After filtering the signal with the frequencies in interest, the filtered signal will only contain these frequencies; then the energy of the signal can be calculated as the square of the magnitude of the time domain samples (Graimann et al., 2002b) and (Pfurtscheller and Lopes, 1999).

One of the first studies classifying real movements online from a single trial was conducted by (Pfurtscheller et al., 1996). The following experimental task has been used since then as the base of MI experiments, and it can be used to visualise the experiments of the studies in the MI domain. The experimental task required that participants perform a fast right or left wrist movement. Each trial was 15s long, the cue was show at second 2 and was presented for 3s, followed by beeps that are 3s away from each other, the beep was for the participant to perform the required movement. The signals were recorded using 2 electrodes (C3 and C4) with Fz as a reference (international 10-20 system electrodes positions as shown in Fig. 1.1), 64 Hz as sampling rate and band-pass filtered between 0.5 Hz and 30 Hz. The 15s epochs were then band-pass filtered for the selected frequencies, then the samples are squared, followed by averaging over trials and consecutive averaging of the samples (8 samples per iteration) to reduce the variance, finally the ERD/ERS as shown in Fig. 1.2 was calculated as the normalised percentage of power change in relation to a reference period (Graimann et al., 2002b) and (Pfurtscheller and Lopes, 1999).

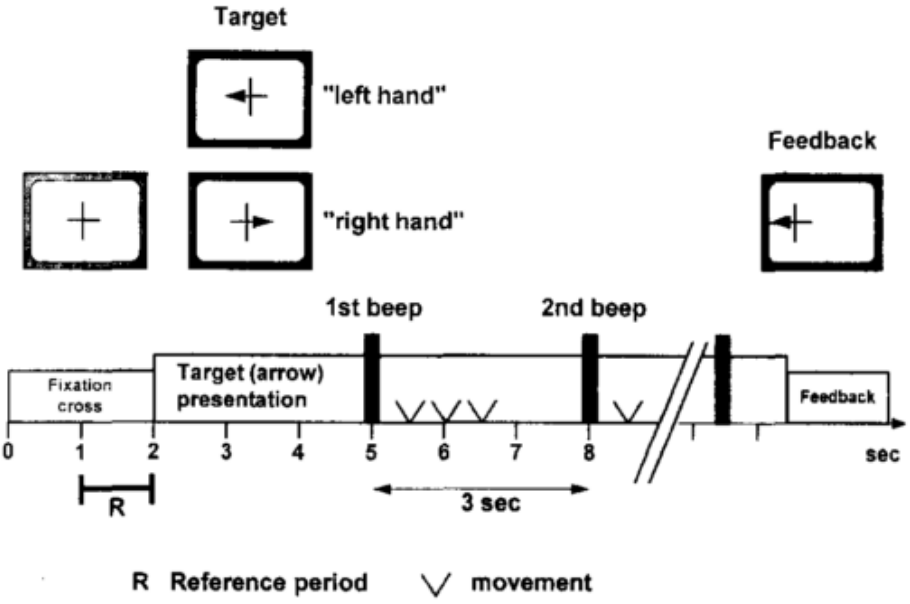


Figure 1.2: Example of the experimental task used in Pfurtscheller et al. (1996), showing the reference period (R) where the participant isn't performing the movement instructed yet. Source (Pfurtscheller et al., 1996).

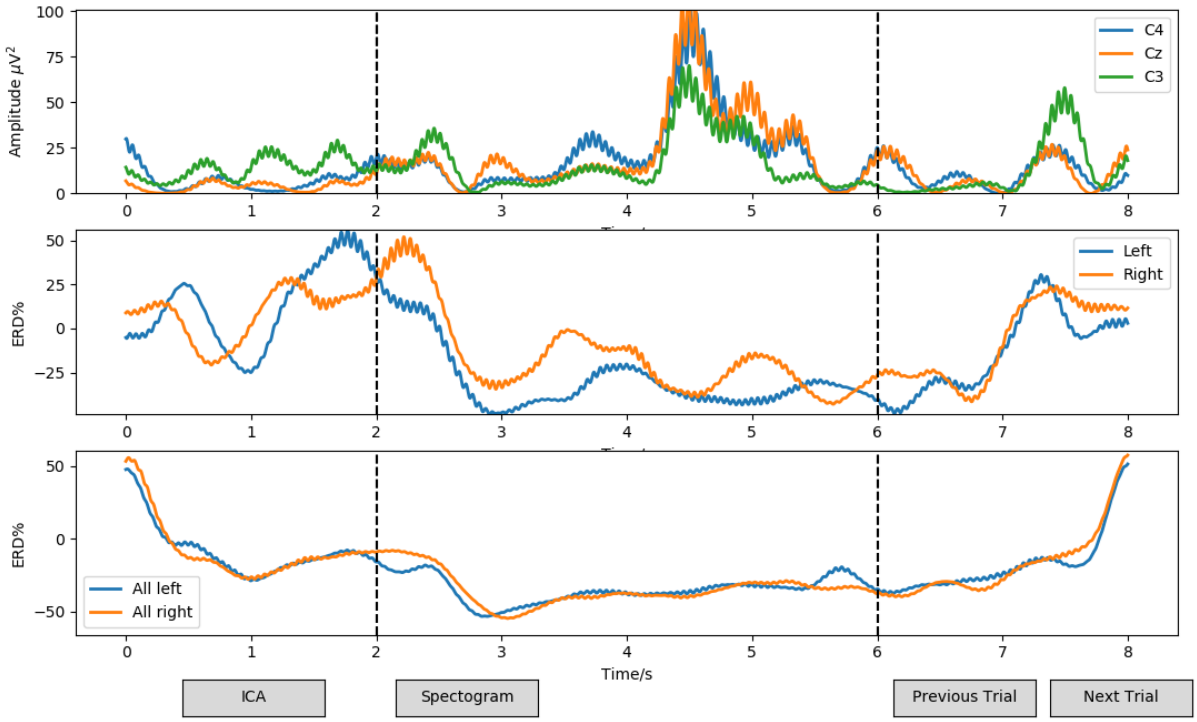


Figure 1.3: An example of an ERD analysis from Chapter 2. The first row shows the squared samples signal for a single trial in three channels (C4, Cz, C3). The second row shows the average ERD (right in blue and left in orange) over all trials for a single participant, over a single channel (indicated in the diagram which channel is chosen). The final row shows the average ERD over all trials over all participants for a selected channel. The dashed lines indicate the start and the end of the imagined movement (2s to 6s).

Spectral features

One of the earliest methods for extracting features for MI classification is the analysis of FFT components, after a pre-processing phase, since there are a limited number of frequency bands (α and β) that contain useful information. The process entails segmenting the full signal into temporal epochs corresponding to the external stimulus cue before estimating Band Power (BP) or Power Spectral Density (PSD) features. Since the classification of the estimated BP and PSD features occur at the end of the epoch, FFT signal decomposition is considered to be a convenient method of extracting specific frequency power features (Al-Fahoum and Al-Fraihat, 2014; Bashivan et al., 2015; Brodu et al., 2011; Carreiras and Sanches, 2011; Pfurtscheller et al., 1996; Tang et al., 2017).

In a study by Das et al. (2015), participants were instructed to perform a (real) right or left hand movement on the onset of a visual cue that instructed them which hand was to be used. Channels C3 and C4 were selected for analysis, since they seemed to provide the most distinctive signals, presumably as a consequence of their proximity to the sensorimotor cortex. The features were estimated as PSD with Welch's method (Padfield et al., 2019). The authors argued that FFT decomposition is the best way to divide the signal into segments and extract the power of the frequencies of interest. The PSD components of the frequencies of interest, α and β are extracted, a hamming window was applied, and the features were defined as the difference of PSD values calculated from the opposite electrodes (C3 and C4) and the average power (Eq. 1.1 and Eq. 1.2).

$$Y_{PSD} = \sum_{f1}^{f2} PSD_{C4}(f) - \sum_{f1}^{f2} PSD_{C3}(f) \qquad \text{Eq. 1.1}$$

$$Y_{POW} = PW_{C4} - PW_{C3} \qquad \text{Eq. 1.2}$$

Where $PSD_{c4/c3}$ are the PSD values of the two electrodes (C4 and C3) for the interval between $f1$ to $f2$ (the frequency intervals used in this study are mentioned above), and $PW_{c3/c4}$ are the average instantaneous powers of the same frequency intervals for both electrodes (Kalcher and Pfurtscheller, 1995). Finally, the features were fed to their proposed Adaboost based classifier and compared with traditional classifiers. 200 trials were randomly chosen for training the classifiers, and 100 remaining unseen trials were used for testing. A maximum classification accuracy of 89% for their novel method was reported compared to 83% using SVM and 81% using LDA.

Brodu et al. (2011) describe an approach that used a number of techniques for extracting band-power features in MI tasks, including spectrogram-based methods, Wigner-Ville distribution, Morlet Wavelet Scalogram, full signal periodogram, an auto-regressive model and Butterworth band-pass infinite impulse response (IIR) filtering. A ten-fold cross-validation accuracy of training trials from the BCI II set III and 2b (Lemm et al., 2004), and BCI IV 2b (Wang et al., 2004) was performed for evaluation. These datasets, used by many researchers to facilitate comparability, are described in detail in the following chapter. The experimental task is similar in the three datasets, but with minor

changes. Following a cue that indicated which movement to be imagined, participants imagined moving either their right hand or left hand for a short period of time. A total of thirteen datasets were used for the evaluation. LDA classification was used for all the techniques for fair comparison. The Morlet Wavelet Scalogram achieved higher classification accuracy over the mentioned techniques in six datasets, with a maximum accuracy of 96%, followed by the auto-regressive model in four datasets, with a maximum accuracy of 95%, spectrogram in two of the datasets, with a maximum accuracy of 95%, and finally the Wigner-Ville method for one data set and a maximum accuracy achieved of 93%. Nevertheless, the superiority of one technique over the other is not significant, for instance the auto-regressive for BCI IV subject four scored 95% while the Morlet-wavelet and the Wigner-Ville scored 93%. On the other hand, for some other participants the difference between two techniques might be significant but isn't for the rest of the techniques, such as BCI IV participant 7 for whom the auto-regressive approach scored 74%, the Morlet wavelet scored 73%, and the Wigner-Ville scored 70%. In addition, fine tuning the parameters for some techniques might have led to similar or better performance over others, for example choosing the mother wavelet and decomposition levels (Jahankhani et al., 2006).

Challenges and limitations using spectral features

Even though, frequency domain features are adequate for two-class MI BCI systems, the method requires a number of steps to extract salient features. A time-frequency transformation results in a loss of information, especially with non-stationary signals like EEG (Al-Fahoum and Al-Fraihat, 2014). Furthermore, the best bands vary from person to person (Kalcher and Pfurtscheller, 1995; Pfurtscheller and Aranibar, 1979). Also, consecutive trials of the same class are needed for efficient classification, especially in online systems; in other words, a movement has to be performed for more than 1s for accurate classification of the movement, usually between 4-9s would be needed using these techniques to classify an imagined movement with a high confidence (Brodu et al., 2011; Lu et al., 2017; Pfurtscheller et al., 1996).

Spatial features

Common Spatial Patterns (CSP)-based algorithms are considered to be one of the most successful approaches for two class (left *vs* right) MI problems. Kevric and Subasi (2017) reported an average of 89%, Pfurtscheller and Neuper (2001) and Soman and Jayadeva (2015) reported as high as 100%,

Wang et al. (2004) reported a maximum of 98%. This approach was first described by Mueller-Gerking et al. (1999). The basic CSP method aims to construct spatial filters for two different EEG populations; the spatial filters should be optimal for the classification between the two movements. The method is driven from two matrices simultaneous diagonalisation (Fukunaga, 1972). The Sub-Band Common Spatial Pattern (SBCSP) and Filter Bank CSP (FBCSP) based methods address the non-stationarity drawback of the CSP algorithm (Kai Keng Ang et al., 2008; Novi et al., 2007; Zheng Yang Chin et al., 2009). Since the features extracted by CSP provide spatial information by constructing spatial filters, useful spectral information goes to waste as it is disregarded. Thus, a more robust CSP methods are obtained by filtering the EEG signals into sub-bands and identifying optimum bands autonomously. Identifying the best bands to extract and localise the ERD/ERS vary from one individual to another, the process is time consuming and the bands have to be carefully analysed to avoid poor selection, which would otherwise lead to poor performance.

SBCSP was developed by Novi et al. (2007), in which CSP was applied to decompose sub-bands of the EEG signals. Gabor filters were used for signal decomposition, rather than FIR band-pass filters. LDA was used to compute a score for each band and these scores represented the feature space of the proposed method. SVM was used for the classification task. The BCI competition III dataset IVa was used for testing, in which the experimental task entailed imagining a right hand movement or a right leg movement. The ten-fold cross validation accuracy was used for the performance evaluation. The authors reported $86.3 \pm 1.1\%$ classification accuracy. The method was not found to improve the classification performance significantly over earlier methods, did have the benefit of a simpler autonomous method for identifying the optimal frequencies for each participant.

Extending the SBCSP approach, Kai Keng Ang et al. (2008) demonstrated the FBCSP method that aims to improve SBCSP by identifying and using the best subject-specific sub-band features automatically. It is worth mentioning that FBCSP won the BCI Competition IV 2a, 2b and has been the most widely used algorithm of CSP-based methods. FBCSP comprises four steps: the signal is band-passed filtered into frequency bands covering the range of frequencies containing the α and β bands. CSP is then applied to the resulting signals of each band separately and the salient pairs of these bands are combined and used as the features for the classification task. Ten-fold cross-validations was performed on the same data set BCI competition III dataset IVa for the evaluation of their proposed algorithm. A SVM was used for classification. The authors claimed higher

performance of FBCSP over SBCSP and CSP, since the FBCSP achieved 90.3±0.7% as opposed to SBCSP with 86.3±1.1% and 86.6± 0.7% for regular CSP.

Soman and Jayadeva (2015) extended FBCSP using an approach that entails an ensemble of SVMs for the classification, called the twin SVM approach. Their approach aimed to enhance the FBCSP by utilising the classification task. In FBCSP the classifier is trained for each of the CSP features of the various bands, the features of these bands that corresponds to the highest classification accuracies are selected and the classifier is trained again. Soman and Jayadeva argued that it is computationly expensive and suggested an alternative of identifying the most prominent features by computing a classifiability measure for the various bands before training the classifier. Finally, the twin SVM which aims to provide a better class separability by solving a smaller but double quadratic programming problem (QPP), as opposed to a single QPP for regular SVM, was used for the classification. A maximum accuracy of 100% (with ten-fold cross validation) was reported for the BCI IIIa dataset.

FBCSP-based algorithms have shown promising results in single trial classification and they are arguably simple enough to be used for real-time systems. For example, Soman and Jayadeva (2015) were able to train the model in 37s, and the classification task would take approximately 1.8s. However, classification accuracy decreased as a result of shortening the trial duration, significantly if the most prominent features was found to be in the α band, as opposed to faster oscillatory rhythms which recover from the desynchronised state within 1s (Pfurtscheller and Neuper, 2001).

Zheng Yang Chin et al. (2009) used FBCSP to classify four imaginary movements: right hand, left hand, foot and tongue. Since FBCSP is designed for binary problems (e.g., right *vs*. left), a One-Versus-Rest (OVR) approach entailed training each class against the rest. Four OVR classifiers were therefore used and binary Naïve Bayes Parzen Window (NBPW) classifier on the top of them to select or identify the movement class. The authors report a mean of 0.57 kappa for the testing set. In addition, Naeem et al. (2006) investigated four imaginary movements: right hand, left hand, foot, or tongue, comparing different methods for the feature extraction including the same OVR FBCSP approach for classification and a fast Independent Component Analysis (ICA) and infomax algorithm. An in-house experiment was conducted to collect the EEG data, limiting the ability to compare with other work, but the OVR FBCSP was reported to perform relatively poorly, but better than infomax

and ICA, with a mean accuracy of 64%, the Fast ICA and infomax achieved 58% and 59% respectively.

A study conducted by Shiman et al. (2015) aiming to classify same-limb movements using a CSP-based approach in which participants were instructed to move their hand in one of four directions: towards them, away from them, downwards-away, and downwards-towards. An SVM was used for classification, and the pre-processing of the signal included artifacts removal such as the eye-blinks and muscles artefacts. Average classification accuracy reported to be 36%. Another similar study by Woo et al. (2015) was more succesful classifying four directions of movements (from among right up, right down, left, left down, left up, and right) where the data was collected by the authors in lab. The features were extracted using CSP and LDA was used for the classification. The classification task was binary, such that only two of the movements were classified against each other at a time. The average reported accuracy was 74%.

Challenges and limitations using spatial features

CSP-based algorithms are not end-to-end solutions, requiring several stages to achieve high performance. The CSP approach is highly susceptible to noise (Devlaminck et al., 2011) and requires a priori feature selection (such as optimal bands), whether identified through autonomous and more manual methods. Furthermore, useful information could be lost due to the nature of the algorithm, in which signal decomposition into covariance matrices and eigenvectors is performed. This lost data could potentially be used to improve performance.

The previous studies summarise the ability of CSP to incease the input space of BCI systems. Although high performance in two class BCI systems was observed, the performance when classifying four classes significantly reduces, suggesting that this approach may not be suitable for more sophisticated multi-class BCI systems. Shiman et al. (2015) and Woo et al. (2015) argue that CSP can be used for identification of same limb movements, although the results of their studies don't support this claim. The CSP-based approaches used were seen to incur a significnant decline in accuracy when the number of movements increased, or the areas of sensorimotor cortex locations corresponding to the movements performed are closely (in the motor humunculus).

Spectro-temporal features

The use of spectro-temporal features to estimate energy in sub-bands has shown promising results for multi-class MI. In Abdalsalam et al. (2018) the four class problem was investigated, wherein participants were instructed to imagine right hand, left hand, foot and tongue when instructed to do so by a visual cue. Energy was estimated after decomposing the signal using a Discrete Wavelet Transform (DWT) and Empirical Mode Decomposition (EMD), and a comparison of performance between the two methods was performed. An ANN was used for classification, and different sized networks were empirically tested to identify the best parameters and number of hidden layers. The average classification accuracy reported of the DWT was 84% and an average of 90% was reported for the EMD. Moreover, a study conducted by Vijayendra et al. (2018) used a multi-class BCI system that aimed to control a unmanned aerial vehicle. The imaginary movements used in this study were unconventional, such that the particpants were instructed to imagine left hand, right hand, left hand with finger and elbow and right hand with finger and elbow. The EEG signals were pre-processed for electrical noise, muscles-related artifacts, and eye-blinks. Features were extracted by applying DWT and differential entropy. Finally, a simple ANN with one hidden layer was used for the classification, with 98% classification accuracy reported for the network with 500 hidden neurons after empirical testing for the optimal neurons number.

Behri et al. (2018), Kevric and Subasi (2017), and Qiu et al. (2016) attemted to classify two-class MI (left hand or a right hand vs foot) with the same dataset (BCI IVa) and found that specto-temporal features performed better than spatial features. Behri et al. (2018) reported average accuracy of 94.5% using DWT for features extraction and a k-nearest neighbour classifier. Kevric and Subasi (2017) reported an average of 94.5% using Wavelet packet decomposition (WPD) features, which is an extensiton of DWT and KNN classifier. Qiu et al. (2016) reported an average accuracy of 84% using a CSP-based method that improved channel selection by Sequential Floating Forward Selection (SFFS) and a SVM for classification. Additionaly, Abdalsalam M et al., (2018) and Vijayendra et al., (2018) imply that temporal and spectro-temporal feature extraction methods are better suited to multi-class MI classification problems, and conjecture that discriminant patterns can be extracted from the temporal domain when the correct methods and classifiers are adjusted.

Challenges and limitations of using specto-temporal features

The results of spectro-temporal based methods, and particularly the DWT, for multi-class MI BCI systems are promising, yielding hgh accuracy in real-time applications in Vijayendra et al. (2018) and (Jahankhani et al., 2006). However, DWT still requires meticulous selection of the mother wavelet. In addition, it is usually combined with a dimensionality reduction process, or features are calculated as metrices of the decomposition coefficients - this implies a loss of useful information and the inability to provide an end-to-end solution. Finally, the DWT features are most effective with longer trial windows, and shorter trial duration would lead to worsen classification accuracies, suggesting a limit to responsiveness in real-time systems.

1.4 Artificial neural networks for EEG classification

Artificial Neural Networks (ANNs) have excellent potential in pattern classification and recognition compared to traditional model-based methods. With a sufficient number of observations, ANNs are able to learn subtle relationships of functions that are hard to describe or model using traditional methods. Moreover, after training, a network is often sufficiently generalisable and robust that it can identify the discriminative patterns even if the provided new data, even where noise is present. After training the model, an ANN acts as a predictor and forecaster for real world problems. They are also non-linear function approximators. More importantly, ANNs have shown the ability to extract features without feature engineering (Tang et al., 2017), automatically remove artefacts (Qiu et al., 2018; Yang et al., 2016), and applying convolutional filters, subsampling, and transformations in a non-linear automated fashion (Tang et al., 2017).

ANNs have been combined with the traditional techniques and used as a classifier in many studies. In Hung et al. (2005), an ANN was used to discriminate between two imaginary movements (lifting right *vs*. left finger) using four participants, along ICA for features extraction. Two simple feed-forward neural networks were used: a Back-Propagation (BP-NN) and a Radial-Basis Function (RBF-NN) with one hidden layer, input and output were used for the classification of two mental imagery movements (left vs. right index). The BP-NN achieved a mean of 76% and a mean of 80% for the RBF-NN with the ICA for feature extraction, and achieved 60% and 62% without applying ICA for the BP-NN and RBF-NN respectively. Tavakolian et al. (2004) used a Genetic Algorithm (GA) with a feed-forward Multi-Perceptron Neural Network (MPNN) for the classification of three mental tasks:

multiplication, geometric figure rotation and neutral. The GA and MPNN were combined to select the optimum channels. In other words, the least number of combinations of channels that provide the highest classification accuracy. In this experiment, the MPNN was considered the fitness function of the GA. A mean classification accuracy of 100% was achieved for the mutated set of channels. Subasi (2007) used MPNN for detecting epileptic seizures, the features were extracted by applying wavelets and the system achieved an average of 93% classification accuracy.

Semi-supervised neural networks

In addition, in the past few years, Deep Belief neural Networks (DBNs) have gained popularity due to the increases in available computing power. DBNs provide the ability to combine two or more types of networks into one architecture. In particular, supervised and unsupervised neural networks are combined for feature extraction or dimensionality reduction and classification in a single network. EEG data has been analysed using DBNs; for instance, Tang et al. (2017) using a Stacked Boltzmann Machine (SBM; Hinton and Sejnowski, 1983) to extract features and a softmax layer (Bishop, 2006) on top for classification. The network was optimised with particle swarm algorithm, instead of more typical gradient descent, and accuracy of up to 90% was reported in a two-class MI classification (left vs. right hand).

In Lu et al. (2017), a SBM was used for classifying two motor imagery movements. Band power features were extracted and the network was trained using the conjugate gradient method (Meiller, 1991). An intermediate stage between the pre-tuning and the fine-tuning stage was also used to optimise the softmax layer (classification layer) before optimising the whole network or fine-tuning the weights. The best classification accuracy obtained was 96% and an average of 83% has been reported. Also, in Kobler and Scherer (2016), a Restricted Boltzmann Machine (RBM) was used for classifying a two-class motor imagery (right *vs*. left hand movements). The authors computed a Laplacian derivation of thirteen channels to obtain three signals (the channels around the motor sensory cortex) and the logarithm of the band-powers extracted as explained in (Kalcher and Pfurtscheller, 1995) were used as the features. The authors reported an average of 88.9% classification accuracy and a 98% maximum accuracy for an experienced participant (trained for MI).

Moreover, Li et al. (2015) explored the effectiveness of unsupervised learning and auto-encoders for feature extraction. Random data points (samples) were removed from the signal and fed to the auto-

encoder which tried to reconstruct the signal and feed it to a classifier, either an SVM or adding a softmax layer at the top of the network. They found that the classification accuracy was not significantly reduced compared to where the signal was complete; their findings suggests that auto-encoders are able to reconstruct salient features of the signal with less information, and that bad trials or noisy trials don't have to be eliminated from the training set to achieve good performance.

Convolutional neural networks

CNNs have been implemented that can automatically find convolutional filters similar to those extracted by CSP, requiring a number of steps and filter banks as described above. These filters were able to extract the prominent features of the EEG signal in one layer, subsample and transform the extracted features to time domain in the second layer and finally classify the features as in Tang et al. (2017) with enhanced classification accuracy over conventional methods with a maximum observed accuracy of 92%. Furthermore, unsupervised-supervised (semi-supervised) systems have been used for the classification task in P300 spelling applications, achieving relatively high accuracies compared to the state-of-art methods from single trials, where the task entails participants trying to focus on characters and letters presented on the screen (Gareis et al., 2017).

Two significant studies using CNNs based architectures introduced end-to-end deep networks for classifying MI by Schirrmeister et al. (2017) where a Shallow Convolutional Network (SCN) and a Deep Convolutional Network (DCN) were introduced achieving mean accuracies of 67.6% and 67.8% classifying 4-class MI data (right-hand, left-hand, feet and tongue). The shallow architecture had a layer which explicitly calculated the band-power as spontaneous power (squaring the samples), while the deep architecture consisted of classic convolution network blocks. The reported performance was the result of a 10-cross-fold using the BCI IV 2a dataset and the provided unseen test set. Even though the authors achieved state-of-art results, 4s signals were still required for their deep model and 2.5s for their shallow model. The second study was conducted by Lawhern et al. (2018) achieving near state-of-art performance with a mean accuracy of 65% in a four-fold on the same dataset (BCI IV 2a), with reported accuracy is for the unseen test set. The authors focused on having a general network that is used for all types of BCI based applications and aimed to provide a compact CNN architecture with fewer parameters to improve the training speed by using separable convolutional layers (further details in Chapter 3); however, at least 2s of the signals are required as input to achieve the performance reported.

Recurrent neural networks

More recently, RNNs have been gaining popularity in EEG classification as they were specifically developed to deal with data in the time domain. RNNs have an interconnected parallel and non-linear structure, and are considered dynamic because they are able to employ nonlinear filters (Güler, Übeyli, & Güler, 2005) that are more flexible than the common linear methods found in other ANN architectures, and are better suited to the nature of the EEG signals elicited in MI. In addition, RNNs can 'remember' events from previous steps and learn dependencies between layers that are not directly connected. RNNs do have some drawbacks; they need to learn long-term dependencies, especially when trained using standard gradient-descent and back-propagation methods (Sutskever & Hinton, 2010), and can thus be difficult to construct and train. In one study, Forney & Anderson (2011) reported a maximum accuracy of 99% (mean = 98%, SD = 0.8) when differentiating between an imaginary right hand movement and the cognitive task of counting backwards using data from three participants that used nearly-raw (i.e., minimally pre-processed) EEG signals as input.

In Hema et al. (2007) RNNs were used to discriminate between four mental tasks (complex problem solving, geometric figure rotation, visual counting, and resting). In Güler, Übeyli, & Güler (2005), seizures were classified with up to 96% accuracy. However, the system was not tested on the task of discriminating between MI movements, but on two quite distinct mental tasks that are unsuitable as control signals for BCIs. Another subsequent study by Ko et al. (2018) aimed to classify MI signals (right hand, left hand, foot and tongue) in the BCI Competition IV2a dataset. The authors used a network scheme which included convolutional layers that are connected recurrently, claiming the proposed network is able to extract spatial and temporal features and finally classify the movements. However, the their approach was not described in sufficient detail to be replicated, and their reported kappa value didn't show an improvement in classification over other methods discussed. Zhang, Yao, Chen, & Monaghan, (2019), combined RNN and CNN architectures to classify a four class MI (left hand, right hand, feet and tongue), achieving state-of-art classification accuracy of 59%±0.1 on the BCI IV 2a dataset aiming to have a generalised model that is trained with all the participants (across participants) instead of training for each participant separately. The authors used Long-Short-Term-Memory (LSTM) to address the issue of the vanishing gradients, one of the limitations of more basic RNN models (Sutskever and Hinton, 2010). The architecture extracts the features in the CNN layers and then these features are fed to the LSTM layer. An additional attention mechanism was employed

to highlight the temporal slice (samples in a point of time, discussed further in Chapter 4) with the highest contribution. The authors also argue that the the attention mechanism should provide a level of interpretability obtained from the attention vector. The authors claim that their model extracts spatiotemporal information from the signal, but didn't validate this claim with further analysis on what the model is learning.

A subsequent study by Ma et al. (2018) suggested an architecture based on two parallel LSTMs, with one trying to learn the from the temporal information and one from the spatial information. A mean accuracy of 68% on classifying five MI movements is reported (eyes closed, both feet, both fists, left fist and right fist) from eegmmidb dataset, which represented an approx. 8% improvement over the state-of-art methods at the time. The suggested architecture has an impressive classification performance, but lacked any interpretability or feedback to improve on the training; e.g., providing a feedback loop for participants to indicate if they lost focus, and provide an approximation of the duration required for participants to succesfuly imagine the MI movement. In Wang, Jiang, Liu, Shang, & Zhang (2018), a combination between LSTM and One Dimension-Aggregate Approximation (1d-AX) which aims to reduce feature dimensionality operating on time-series and channels weighting was employed for the classification of the four MI in the BCIIV 2a dataset. The authors claimed a significant improvement of contemporary approaches, obtaining 76% mean accuracy ± 5.92 which is between 4-8% improvement over the state-of-art. However, the accuracies reported didn't show confusion matrices or report a second measure like Kappa values to compare with the winning methods of the BCI IV2a competition. Furthermore, again, the suggested architecture doesn't provide any interpretatbility either and required about 3s signals to achieve the reported performance limiting its usefulness in real-time classification.

1.5 Suggested methods

The studies summarised imply that it may be possible to design an end-to-end ANN capable of extracting salient features for classifying MI movements. This could be achieved by combining unsupervised and convolutional layers, similar to those in Li et al., (2015) and Ko et al. (2018). A RNN integrating unsupervised layers to obtain high classification accuracy from 1s trials would enhance the applications of BCI system in for real-time applications, such that discussed in Forney and Anderson (2011). Last but not least, RNNs have an interconnected parallel and non-linear structure and are considered dynamic, since they are able to employ non-linear filters (Güler et al.,

2005) that are more flexible than the common linear methods found in other ANN architectures, and are therefore better suited to the nature of the EEG signals elicited in MI. In addition, RNNs can 'remember' events from previous steps and learn dependencies between layers that are not directly connected. RNNs do have some drawbacks: since they need to learn long-term dependencies, when trained using standard gradient-descent and back-propagation methods (Sutskever and Hinton, 2010) they can be difficult to construct and train. Furthermore, to test and evaluate the suggested methods, a baseline is accquired employing different window sizes with tradional naïve methods for classification.

Guided grad-cam

The objective of model *interpretability* is gaining traction with researchers, especially for the deep models employed in domains that require visibility and an understanding of what the model is learning (e.g., medical imaging). It also provides the researcher with a more complete understanding of how the model is learning, and a chance to address the limitations that become apparent. Very few studies have employed interpretability with deep in EEG signal classification. Schirrmeister et al., (2017) tried to interpret their models by visualising activations, perturbing the input and the output of the convolutional layers (referred to their approach as input-perturbation). Lawhern et al. (2018) followed a similar approach where they were investigating the activation maps of the convolutional layers. Zubarev, Zetter, Halme, & Parkkonen (2019) suggested a linear layer to find a reduced representation of the spatial correlation where the parameters are learnable (optimised by gradient descent). The authors claim that investigating the weights of these layers will provide an interpretation of the spatial correlations (i.e., which channels are contributing more), which can help in method refinement. As previously discussed, Zhang, Yao, Chen, & Monaghan (2019) argue that the attention vectors can provide insights into which time-slices are contributing most. A number of techniques suitable for deep models have been proposed over the last decade. Early work proposed feedback loops for activation maps suggested (Cao et al., 2015) which was developed further by Andreotti, Phan, & De Vos (2018), Selvaraju et al. (2017) and Springenberg, Dosovitskiy, Brox, & Riedmiller (2015). These studies provide the framework and the basis for visibility of the deep models to be discussed in Chapter 4, which addresses the fact that interpretability hasn't been explored thoroughly in the EEG domain because visualising these signals is inherently difficult.

Attention Mechanisms

Another interesting direction of research is the use of attention mechanisms. These improve classification performance and are able to provide an attention vector, such that the features with higher scores are identified as having a greater contribution to classification accuracy, which adds interpretability. The attention mechanism in Bahdanau, Cho, & Bengio (2015) provided an insight on how words in Natural Language Processing (NLP) system correlate, and the authors were able to gain an understanding of what words the model is focusing on, and which words provide better translation from English to French. In Vaswani et al. (2017), the authors introduced the Transformer Model, in which they observed improved results by employing a simple model with an attention mechanism, arguing that this architecture can replace the recurrent layers and provide a level of visibility. Finally, the Squeeze and Excitation (S&E) method introduced by Hu (2018) provides an attention mechanism that can emphasise the feature maps with better feature explanation by acquiring a global average for each channel and adding a small fully connected network to assign the attention weights (higher weights means better features).

Summary

In summary, current classification accuracy using non-invasive EEG-based BCIs is insufficiently high to be used for practical applications (particularly those entailing neuroprosthetics), and a marked reduction in accuracy is observed when the number of limbs to be differentiated between is increased. The most common method for extracting features in the literature is CSP, which has high performance in classifying two movements but falls dramatically when this number is increased. Subsequently, Semi-Supervised DBNs and RNNs have shown promising results (including classifying EEG data relating to seizures, and in differentiating distinct cognitive tasks). The average intervals, to the best of my knowledge until the time of writing this study, is between 2 to 4s to obtain the state-of-art performance. Moreover, methods for deep model interpretation have not been thoroughly developed and tested in the context of motor imagery classification. Finally, the dataset sizes and the fact that each subject requires a separate model to be trained, limits the performance of deep models which performs better with larger datasets.

Accordingly, it is hypothesised that the features extracted from the time domain and processed using a new supervised RNN architecture with data augmentation could lead to faster, higher accuracy

classification, and will enable multiple classes to be discriminated (such as within-limb imagery). Investigating the deep models with state-of-art interpretation techniques may provide insights that can be exploited to reducing signal time intervals for real-time applications. Furthermore, adding visibility to the models is useful for researchers trying to understand the correlations and causations learned by a deep architecture.

A RNN is theoretically capable of learning the relationships or dependencies between the ERD and ERS signal events, since these events occur at different times. This research aims to improve classification accuracy for motor imagery in BCI systems by developing and testing a dynamic RNN architecture that exploits both temporal and time-frequency (spectro-temporal) features, which will be compared with major extant approaches. In addition, classification accuracy for compound, goal-oriented, speed and weight-contingent, using a new RNN classifier will be examined to assess the hitherto unexplored role of participant instructions.

1.6 Summary of Contributions to Knowledge

- Investigation the classification performance of MI over different windows and intervals to find optimum settings;
- Development of a novel deep architecture based on GRU and convolution units;
- Development of a novel temporal and a novel spatial attention mechanism;
- Development of a generalised guided Grad-Cam for EEG for higher interpretability;
- Development of a novel EEG data augmentation technique.

1.7 Objectives

- Obtain a baseline performance of MI classification employing different window sizes;
- Identify which of the feature's types (spectral, spatial or temporal) provide higher performance using shorter windows;
- Investigate CNN based neural network architectures (EEGNet and Shallow net) performance using spectral (CWT) features versus raw EEG signals;
- Investigate the performance of the models applying the suggested data augmentation to the training data;
- Investigate the impact of the number of CWT scales on the performance of the network;

- To establish whether the outcome of the model can be fully interpretable in terms of spatial, temporal and spectral analysis;
- Identify if adding a spatial attention mechanism will provide insights on the spatial correlation.
- Identify if a temporal attention mechanism with improve analysis for the signal intervals with higher contribution to the performance;
- Establish whether a grad-CAM can be used for extracting and interpreting the prominent EEG features;
- Identify what is the shortest signal period that be used while maintaining the state-of-art classification accuracy;
- Establish whether data augmentation will improve the performance of the models.

1.8 Structure of the Thesis

In Chapter 1, the most common methods and techniques used in the literature for EEG analysis and classification were described, and an introduction to Brain-Computer Interfaces was provided that also outlined the types of features that are extracted from the EEG signals used for classification between classes. In addition, related work was summarised, including both historical and new methods that provide state-of-art performance until the time of writing.

In Chapter 2, an experiment is conducted the determined the optimum time intervals and window sizes for EEG signals with spectral and spatial features using basic classifiers obtaining a baseline for the following experiments. In Chapter 3, Convolutional Neural Networks (CNNs) based classifiers are investigated with Continuous Wavelet Transform (CWT) based features. The section contains a comparison between using CNNs with raw EEG signals vs. CNNs with CWT features. The suggested Point-Wise Convolutional Neural Network (PWCNN) is discussed and interpretation of what the models are learning using scalograms and guided Grad-CAMs. In Chapter 4, a novel end-to-end architecture is suggested that is proposed to increase classification performance over short intervals, which is compared with the best performing methods in the literature. In addition, the suggested modification of guided-Grad-CAM and the novel attention mechanisms are discussed. Chapter 5 presents a general discussion of the findings from this thesis and outline the next steps that could be taken to further this work.

Chapter 2:

Datasets and evaluation method

2.1 Datasets

Motivation

The BCI competition IV dataset 2b (Leeb et al., 2007) and BCI competition IV dataset 2a were used for the experiments in this thesis. These are the most common datasets used in the literature for motor imagery classification tasks, enabling the performance of work presented herein to be compared directly with already published methods. Most of the results in the literature use accuracy and kappa to measure and report their methods performance, so these metrics are used in the present research.

BCI IV 2b:

This dataset contains MI data collected from for nine participants. For each participant there were five sessions. There were two types of session: with feedback or without. The first two sessions for all participants were without feedback, the next three sessions had feedback. The data were initially recorded with 22 channels, but was then reduced to three bipolar recordings that correspond to (Cz, C3 and C4). The data was recorded at a sample rate of 250 Hz with a 0.5 Hz to 100 Hz analogue filter, followed by a 50 Hz notch filter to remove power-line noise. There are two different classes (imagined movements): left hand and right hand. The first two sessions had 120 trials and the remaining sessions had more repetitions leading to a total of 160 trials (including bad trials, which were manually identified and labelled in the dataset files).

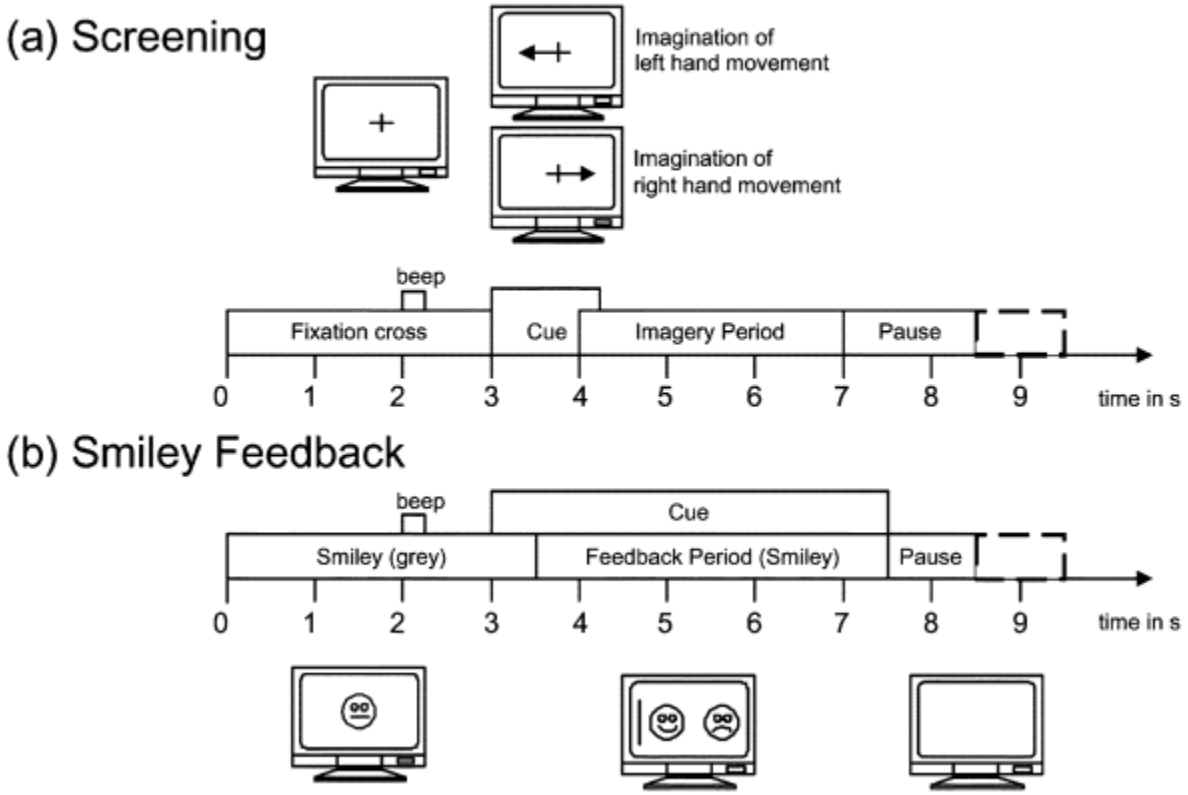


Figure 2.1: (a) The experimental task for sessions 1 and 2 (without feedback), (b) experimental task for sessions 3 to 5 (with feedback).

The experimental task was as follows. For data collection sessions 1 and 2 sessions (Fig. 2.1a), at the start of each trial, a fixation cross was presented for 3s. A beep was sounded at 2s to alert participants to a forthcoming instruction. Next, a 1.5s visual cue was presented (an arrow pointing either left or right). Next, participants were instructed to imagine the movement corresponding to the direction of the arrow (left hand for or right hand) for 4.5s. A blank screen was then shown for 1.5s as a break between trials. Sessions 3-5 (Fig. 2.1b) incorporated feedback. A grey smiley face of neutral expression was presented at the beginning of each trial instead of the fixation cross. The beep was still sounded also played at 2s, the cue was presented at 3s indicating which side the participant should move the smiley to. During the feedback period (3.5s to 7.5s), the smiley would turn green if moved to the right direction and red if otherwise. At 7.5s, the participant was instructed to keep imagining the movement as much as possible and the screen went blank for an interval between 1 and 2s that were added to the trial.

BCI IV 2a:

The BCI Competition IV 2A. The data was collected from nine participants performing four Motor-Imagery movements. The four movements are defined as: left hand, right hand, feet and tongue. Each trial was about 9s long where at the begging a fixation cross is shown at $(t = 0)$ then at 2s $(t = 2)$, an arrow was presented on the screen (cue onset) instructing the participant to perform one of the four movements (arrow pointing left, right, down and up for left hand, right hand, feet and tongue respectively). Participants were instructed to keep imagining the movement until the fixation cross disappeared, which it did at 6s. Finally, a short break with a blank screen until the next trials was shown. The data was collected from 22 Ag/AgCl electrodes and samples with 250Hz. The data was then band-pass filtered between 0.5 Hz and 100Hz, followed by a 50Hz notch filter to reduce noise generated from line. Electrode placement followed the international 10-20 system.

Validation

Both datasets BCI Competition IV 2A and 2B provide additional sets for testing. The testing sets were recorded at different sessions (not a subset of the original dataset). The testing set is also referred to as the unseen testing sets throughout the thesis since they were not used in the training or validation of the models. The final reported performance metrics (as described in the next section) are evaluated on the unseen test sets to address overfitting problems (where the model is not generalisable and performs well on the specific dataset that it was trained on).

Limitations

Both datasets have only nine participants, where the number of epochs per participant ranged between 600 to 720 epochs. Noting that the models are trained on each participant separately. A larger dataset with thousands of epochs would be more ideal for training deep networks and would in turn lead to more confident models. The limitation is addressed by data augmentation as described in Chapter 4 and 5.

The environment of where the data was collected is not described. Hence, the surrounding environment of the participants might have been less than optimum. There could have been some unidentified interference with the recordings which could affect the analysis unfavourably.

2.2 Evaluation metrics

2.2.1 Accuracy

The first metric is accuracy, is defined as in Eq. 2.1 and Eq. 2.2:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. 2.1}$$

$$Accuracy = \frac{No.of\ correct\ predictions}{Total\ number\ of\ predictions} \quad \text{Eq. 2.2}$$

Where TP is true positive, TN is true negative, FP is false positive and FN is false negative. For binary classification, the accuracy is considered as a good metric. However, when the problem we are facing contains multi class labels (more than 2 classes), the accuracy doesn't represent how good the model is classifying each class, it might be very good classifying two of the classes while neglecting the other classes.

2.2.2 Cohen Kappa

The second metric is the Cohen kappa Landis & Koch (1977), and implemented using the library Sklearn (Varoquaux et al., 2015), which is defined as in 2.3:

$$k = \frac{p_o - p_e}{1 - p_e} \quad \text{Eq. 2.3}$$

Where p_o is the observed probability of agreement between raters and p_e is the hypothetical probability of random agreement (e.g. the probability of the labels being assigned randomly). A value of $k = 1$ implies a perfect agreement, anything below that value implies less than perfect agreement.

The Cohen kappa coefficient is useful since it provides both a degree of accuracy and reliability in a statistical classification.

2.2.3 Confusion matrix and F-β

Confusion matrices are used to record the number of TP, FP, TN and FN results. Precision and recall of the model are to be expressed as F-β values and calculated using the library sklearn (Varoquaux et al., 2015). Such that precision is the percentage of the correct positives predictions that belongs to the positive class as in Eq.2.4 and recall is the percentage of the correct positive predictions out of all the positive classes in the data set as shown in Eq.2.5.

$$Precision = \frac{True\ positives}{(True\ positives + False\ positives)} \quad \text{Eq. 2.4}$$

$$Recall = \frac{True\ positives}{(True\ positives + False\ Negatives)} \quad \text{Eq. 2.5}$$

Finally, the F-β is a quantified metric combining both precision and recall as in Eq.2.6:

$$\mathrm{F} - \beta = \frac{(1 + \beta^2) * precision * recall}{(\beta^2 * precision + recall)} \quad \text{Eq. 2.6}$$

Where the β is an adjustment factor (to weight precision and recall) and it was set to 1 which gives equal balance to both metrics.

Chapter 3:

The impact of window size on classification accuracy

3.1. Introduction

No prior study has exhaustively evaluated the effect of window size on the BCI classification accuracy. Accordingly, it is hypothesized that optimising window sizes will improve performance. To address the hypothesis, in this chapter, two motor imagery classes will be discriminated between using two methods that have not previously been compared when window size and window start time are manipulated.

Most MI-based BCI research aim to achieve high classification accuracy whilst having the potential to operate in real-time; shorter window sizes are therefore favourable provided that high classification accuracy can be maintained. In this experiment, we measure the performance of the aforementioned methods over different window sizes and window start times. Furthermore, the classification accuracies will be assessed employing the most common classifiers in more recent BCI systems which are LDA (McMullen et al., 2014; Pfurtscheller et al., 1998; Yu et al., 2016), SVM (Al-Fahoum and Al-Fraihat, 2014; Hung et al., 2005; Soman and Jayadeva, 2015; Subasi and Gursoy, 2010), and ANN (Forney and Anderson, 2011; Senior et al., 2007; Subasi, 2007; Übeyli, 2009). ANN-based classifiers will be developed and compared with LDA and SVM in subsequent experiments.

The Python 3 software framework implemented to support this experiment will be used as the basis for subsequent experiments, since no suitable extant package for MI classification that contains all necessary features and classifier cores is available.

3.2 Method

3.2.1 Apparatus

Software written in Python 3 (van Rossum, 1995) was run on a MacBook Pro (Apple Corp., Cupertino CA) with a 2.9 GHz Intel Core i7, and 8 GB 1600 MHz DDR3 of memory. The following additional Python 3 software libraries were used: numpy for array operations and arithmetic functions (docs.scipy.org, 2017); scipy for signal operations and filtering (docs.scipy.org, 2017); sklearn for the SVM and LDA classifiers (scikit-learn.org, 2017); Theano for neural networks (Al-rfou et al., n.d.); and matplotlib library (matplotlib.org, 2017) for graphing.

3.2.2 Dataset

Dataset is described in section 2.1.

3.2.3 Algorithm and Procedures

Pre-processing and Windowing

These following pre-processing steps were applied to the signal before applying the feature extraction methods (BP, CSP). As mentioned above, there were a total of five sessions for each participant, for each of the participants, the first three sessions were used for training and validation and the remaining two sessions were kept for unseen data testing as they were provided for that purpose in the BCI competition. The total signal was further divided into 9s epochs and then band-pass filtered from 8Hz to 30Hz to extract the frequencies of interest as discussed in the literature and for more details refer to (Pfurtscheller and Lopes, 1999).

The windows used for the signals were: 0.5s, 1s, 2s, 4s. A step of 10% of the total trial length to measure the performance from different time points (window start time). In addition, for each window, classification accuracy was assessed in two ways where the features used for the training are calculated according to the following:

1. The features are calculated for each of the windows and for all window starting times. Validation and testing were assessed at the corresponding window and window starting seconds.

2. The features are calculated for the best performing window and window starting time. Validation and testing were assessed for all the trial where the window is fixed as the best performing window while all the window start points were assessed.

FFT: Band-Power (BP)

Signals corresponding to each of the nine participants were (separately) subject to a sequence of procedures, including pre-processing and filtering, ERD/ERS quantification, and transformation to a

feature space for classification. Band-power features are calculated using different methods and the underlying procedure was described by (Pfurtscheller and Lopes, 1999).

First, to extract the ERD/ERS, the frequencies in which they occur must be identified. Signals corresponding to each trial were Butterworth band-passed filtered between 9-30Hz, using a 3rd order filter (Pfurtscheller and Lopes, 1999). The spectrogram method convolves the signal with a localised windowing function in time t. A time-frequency representation $X(f,t)$ is obtained by applying Fast Fourier Transform (FFT) (Marchant, 2003) on the convolved signal $X(t)$. The power spectrum is then calculated as in Eq. 3.1.

$$P_f = \frac{1}{N}\sum_t |X(f,t)|^2 \qquad \text{Eq. 3.1}$$

Where P_f represents the power marginal in the frequency domain, and finally the power in each band is calculated as in Eq. 3.2.

$$P_B = \sum_{f\in B} P_f \qquad \text{Eq. 3.2}$$

Where P_B is the power in each band and calculated as the sum of marginal in the each of the bands of interest. The powers in each band are then normalised and the logarithm of the outcome forms the features vector.

Common Spatial Patterns (CSP)

The basic CSP method aims to construct spatial filters for two different EEG populations; the spatial filters should be optimal for the classification between the two movements. The method is driven from two matrices simultaneous diagonalisation (Pfurtscheller and Neuper, 2001; Shiman et al., 2015; Soman and Jayadeva, 2015; Wang et al., 2004; Woo et al., 2015). The EEG signal is first band-passed filtered, usually from 8Hz to 30Hz (discussed in the previous section). The filtered EEG signal of a trial is represented as matrix E with NT dimensions, where N is the number of channels (electrodes)

and T as the number of samples in each trial. The covariance matrix is calculated and normalised as shown in Eq. 3.3.

$$C = \frac{EE^T}{tr(EE^T)} \quad \text{Eq. 3.3}$$

Where trace (tr) is the sum of the diagonal elements of the matrix. The spatial filters obtained of the two populations are averaged (i.e., right and left MI) producing the two matrices C_l and C_r and the composite covariance matrix represented as shown in Eq. 3.4.

$$C_c = \overline{C_l} + \overline{C_r} \quad \text{Eq. 3.4}$$

Then C_c is further decomposed into eigenvectors Uc and eigenvalues λ_c, as shown in Eq. 3.5:

$$C_c = U_c \lambda_c U_c^T \quad \text{Eq. 3.5}$$

The variances are equalised by the whitening matrix as shown in Eq. 3.6:

$$P = \lambda_c^{-0.5} U_c^T \quad \text{Eq. 3.6}$$

The transformation $S_l$and S_r are calculated as shown in Eq. 3.7.

$$S_l = PC_lP^T \text{ and } S_r = PC_rP^T \quad \text{Eq. 3.7}$$

Then S_l and S_r share common eigenvectors as shown in Eq. 3.8.

$$\text{If } S_l = \text{B } \lambda_l B^T \text{ then } S_r = \text{B } \lambda_r B^T \text{ and } \lambda_l + \lambda_r = I \quad \text{Eq. 3.8}$$

From Eq. 3.6, the transformation of the combined distribution by P is isotropic which means the union of the individually transformed distributions are also isotropic. Hence, the eigenvector in $\overline{S_l}$ holding the largest eigenvalue would have the least eigenvalue in $\overline{S_r}$ and vice versa. Furthermore, the first and last eigenvectors in B of the projected whitened EEG are the discriminative features of the two EEG populations.

Finally, the projection matrix W and the mapping Z of trial E are defined as shown in Eq. 3.9

$$W = (B^T P)^T \; and \; Z = WE \qquad \text{Eq. 3.9}$$

The common spatial patterns are found in the columns of W^{-1}. Finally, for the classification task, the first and last rows of Z are log-transformed for a normalised distribution in most of the cases and a linear classifier would be sufficient to for the classification task.

3.2.4. Statistical Methods and Cross Validation (CV)

A one-way ANOVA was used to specify if there were significant differences in classification accuracy (dependent variable) for each window size (factor with 4 levels: 0.5 s, 1 s, 2 s, and 4 s). The null hypothesis is that the classification accuracy obtained by changing the window sizes and window starting seconds are similar and there should be no significant difference between the population means.

Cross-fold-validation (CV) (Stone, 1974) was used to measure the performance of the classification. The aim of using this approach is to ensure the performance of the model without over-fitting in the training and achieve a less biased more generalised performance estimate of the model. The dataset is randomly shuffled and split into ten groups, for each of the groups, the samples are divided into a training set and an unseen test set which was set to 80% to 20% respectively in this experiment, the model is then trained and evaluated on the test set. The same process is applied for all the ten groups and the final classification accuracies are then calculated as the average scores across the ten groups.

3.3. Results

The results of the performance are obtained after employing a 10-fold cross-validation (CV) approach. The process is carried out ten times (10 repetitions) such that nine folds are used for the training and the remaining fold is used for the testing. The overall classification accuracy (CV accuracy) is the average accuracy for each iteration. Table 3.1 shows the best performing window sizes and their corresponding window start for CSP and Band-Power (BP) features employing an LDA for classification. Table 3.2 shows the best performing window sizes and their corresponding window start for CSP and BP features employing an SVM for classification. The resulting classification accuracies for each of the windows and the highest achieved classification accuracy for all participants are shown in Figs. 3.1, 3.2, 3.3, 3.4. The average classification accuracy for all the window sizes and window starting points are shown in Fig. 3.5.

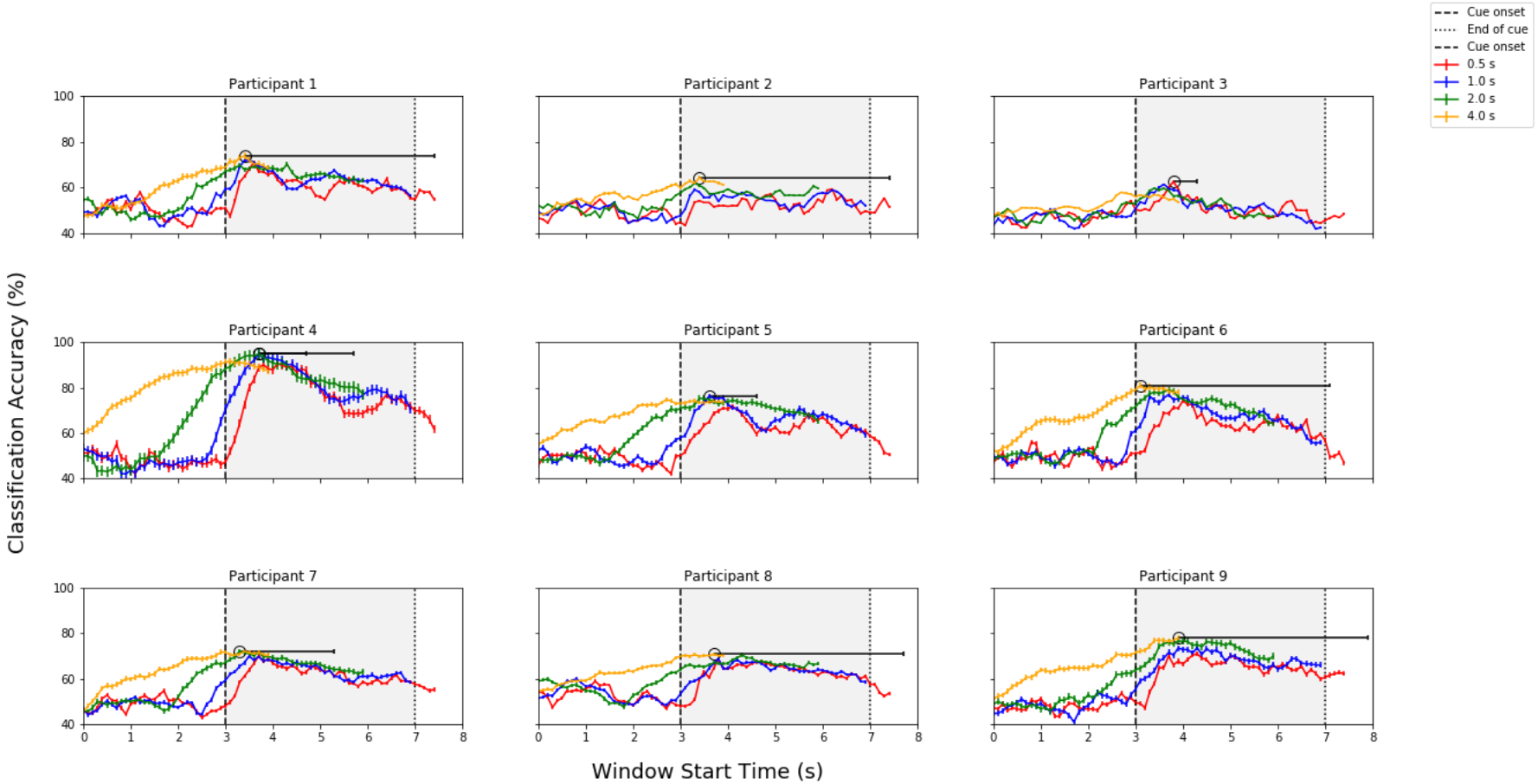


Figure 3.1: The classification accuracy of all nine participants using CSP features and an LDA classifier. The Window Start second is on the x-axis and the corresponding classification accuracy on the y-axis. The grey shaded area represents the period between the cue on set and end of cue. The highest accuracy achieved over all windows sizes and start times is marked by a circle. The horizontal line represents the window size that achieved the highest classification accuracy. The error bars show ± 1 standard error of the mean.

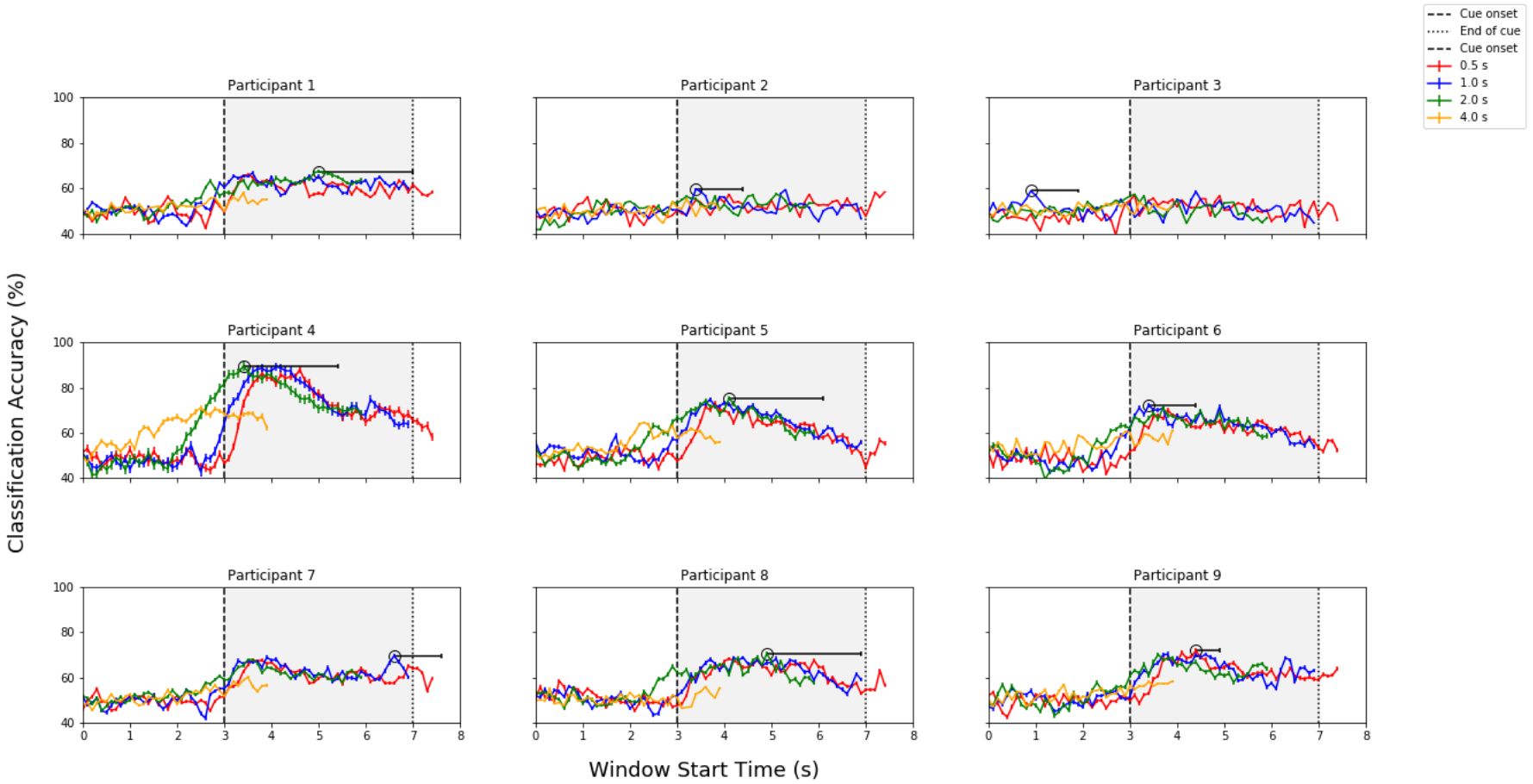


Figure 3.2: The classification accuracy of all nine participants using BP features with a LDA classifier. The Window Start second is on the x-axis and the corresponding classification accuracy on the y-axis. The grey shaded area represents the period between the cue on set and end of cue. The highest accuracy achieved across over all window sizes and start times is marked by a circle. The horizontal line represents the window size that achieved the highest classification accuracy. The error bars show ± 1 standard error of the mean.

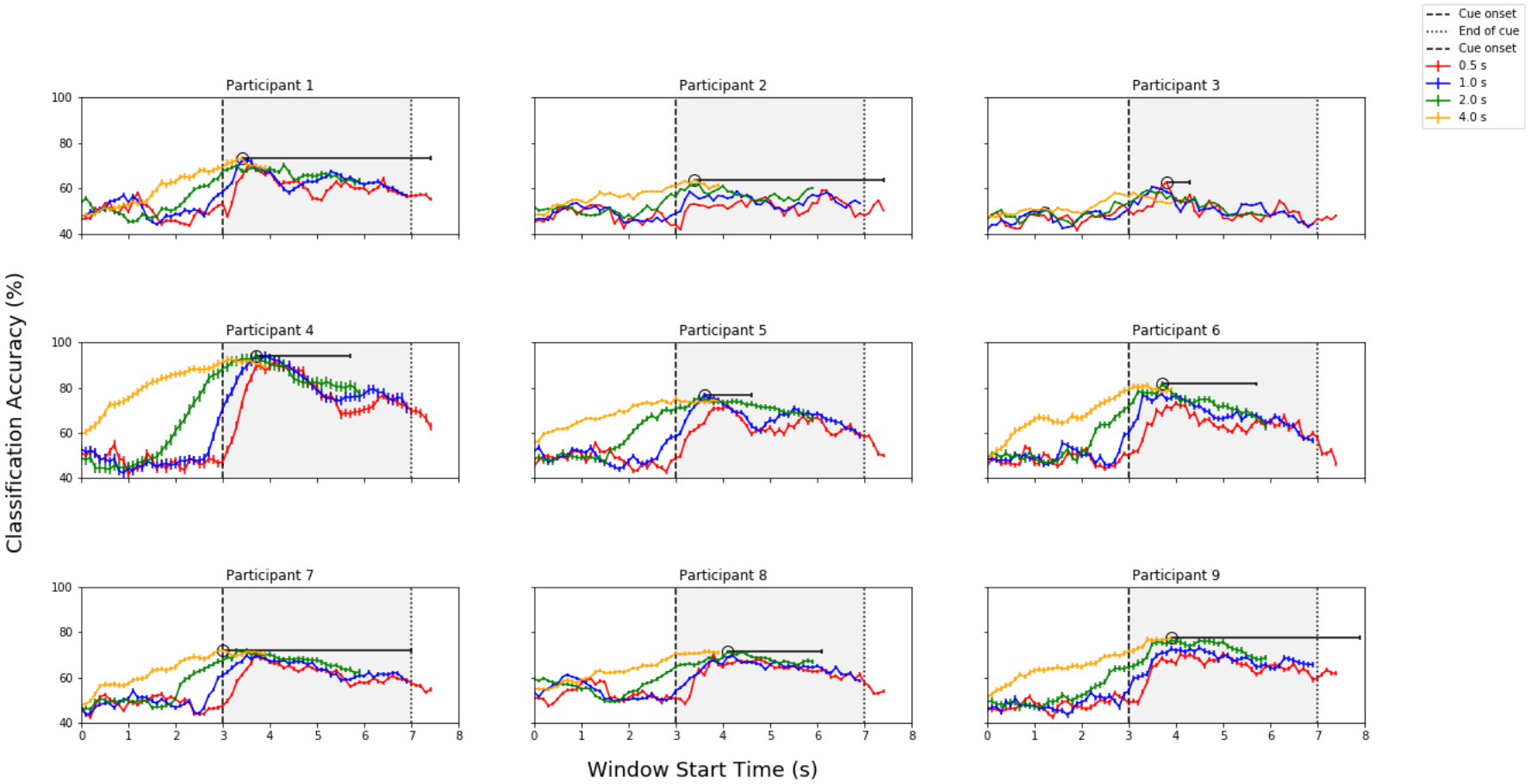


Figure 3.3: The classification accuracy of all nine participants using CSP features with a SVM classifier. The Window Start second is on the x-axis and the corresponding classification accuracy on the y-axis. The grey shaded area represents the period between the cue on set and end of cue. The highest accuracy achieved across over all window sizes and start times is marked by a circle. The horizontal line represents the window size that achieved the highest classification accuracy. The error bars show ± 1 standard error of the mean.

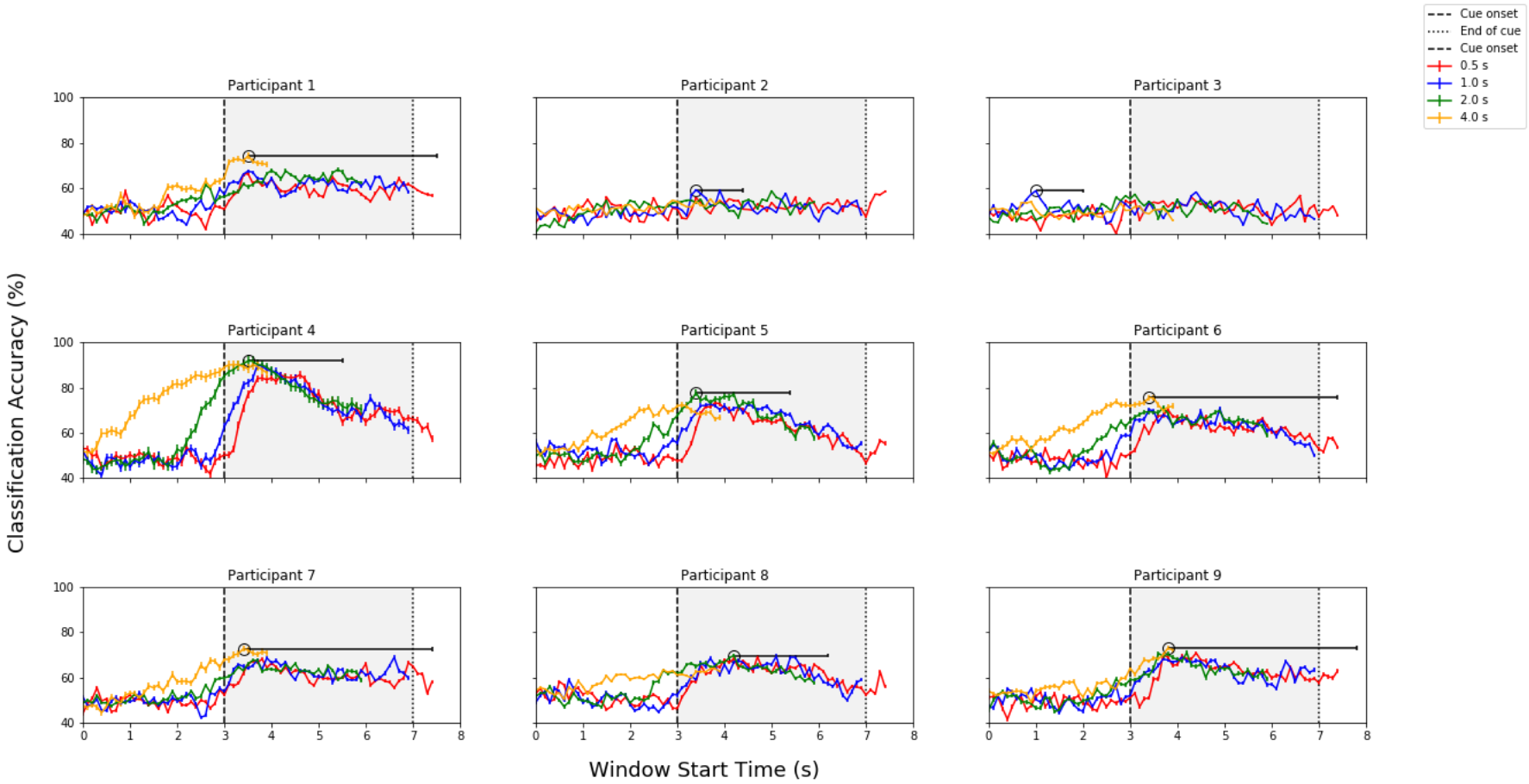


Figure 3.4: The classification accuracy of all nine participants using BP features with a SVM classifier. The Window Start second is on the x-axis and the corresponding classification accuracy on the y-axis. The grey shaded area represents the period between the cue on set and end of cue. The highest accuracy achieved over all window sizes and start times is marked by a circle. The horizontal line represents the window size that achieved the highest classification accuracy. The error bars show ± 1 standard error of the mean.

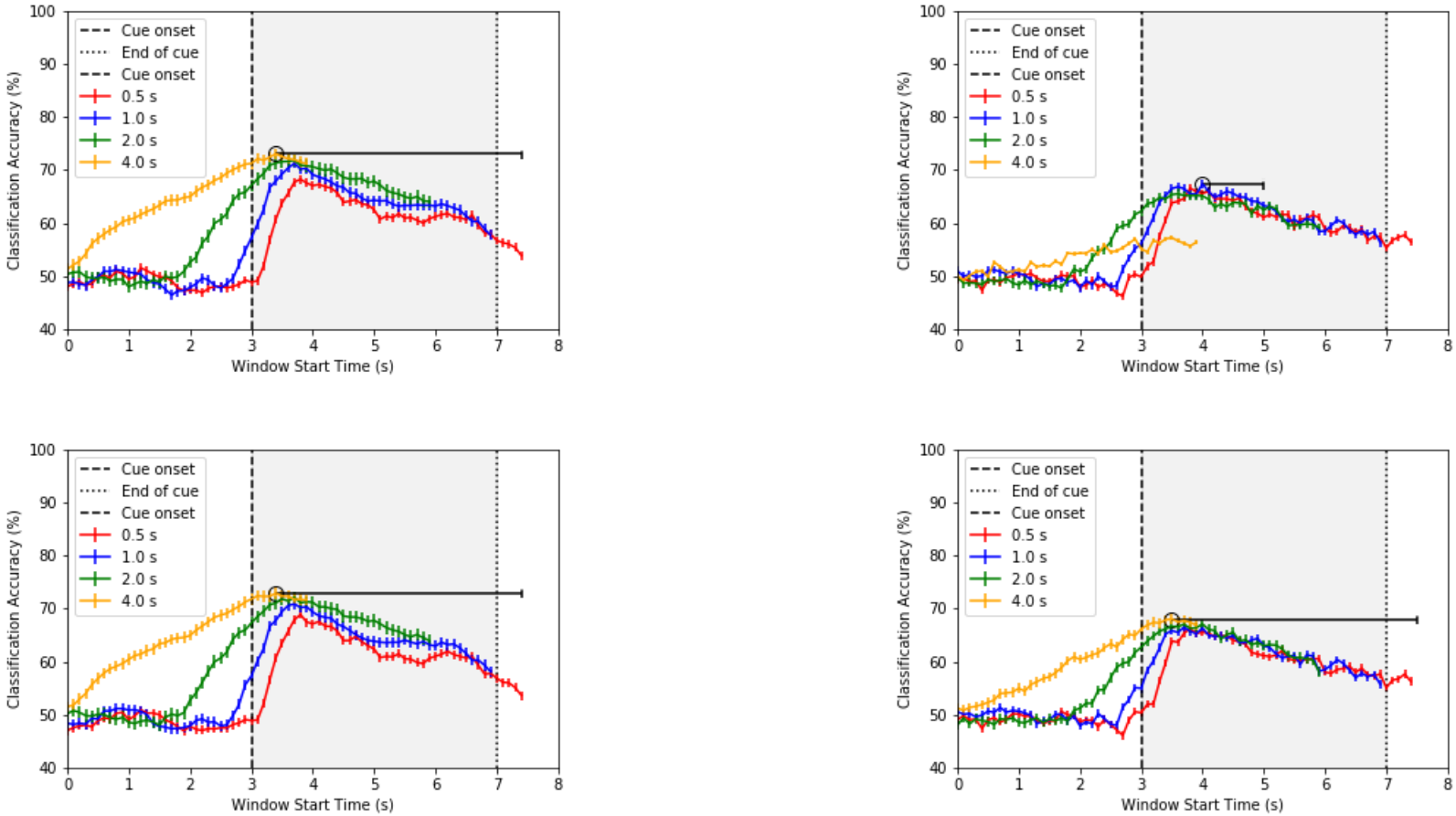


Figure 3.5: The average classification accuracy over participants. The top two figures are the results of using a LDA classifier and the bottom two figures employ SVM for classification, where the left column holds the CSP features and the right columns holds the BP features. The Window Start second is on the x-axis and the corresponding classification accuracy on the y-axis. The grey shaded area represents the period between the cue on set and end of

cue. The highest accuracy achieved over all windows is marked by a circle. The horizontal line represents the window size that achieved the highest classification accuracy. Error bars show ± 1 standard error of the mean.

	CSP		BP	
Participant	Best Window Size (s)	Best Window Start (s)	Best Window Size (s)	Best Window Start (s)
01	4.00	3.40	2.00	5.00
02	4.00	3.40	1.00	3.40
03	0.50	3.80	1.00	0.90
04	2.00	3.70	2.00	3.40
05	1.00	3.60	2.00	4.10
06	4.00	3.10	1.00	3.40
07	2.00	3.30	1.00	6.60
08	4.00	3.70	2.00	4.90
09	4.00	3.90	0.50	4.40
Mean	4.00	3.40	1.00	4.00

Table 3.1: Best windows size and window start for each participant using CSP and BP with LDA classification.

	CSP		BP	
Participant	Best Window Size (s)	Best Window Start (s)	Best Window Size (s)	Best Window Start (s)
01	4.00	3.40	4.00	3.50
02	4.00	3.40	1.00	3.40
03	0.50	3.80	1.00	1.00
04	2.00	3.70	2.00	3.50
05	1.00	3.60	2.00	3.40
06	2.00	3.70	4.00	3.40
07	4.00	3.00	4.00	3.40
08	2.00	4.10	2.00	4.20
09	4.00	3.90	4.00	3.80
Mean	4.00	3.40	4.00	3.50

Table 3.2: Best windows size and window start for each participant for CSP and BP using SVM classification.

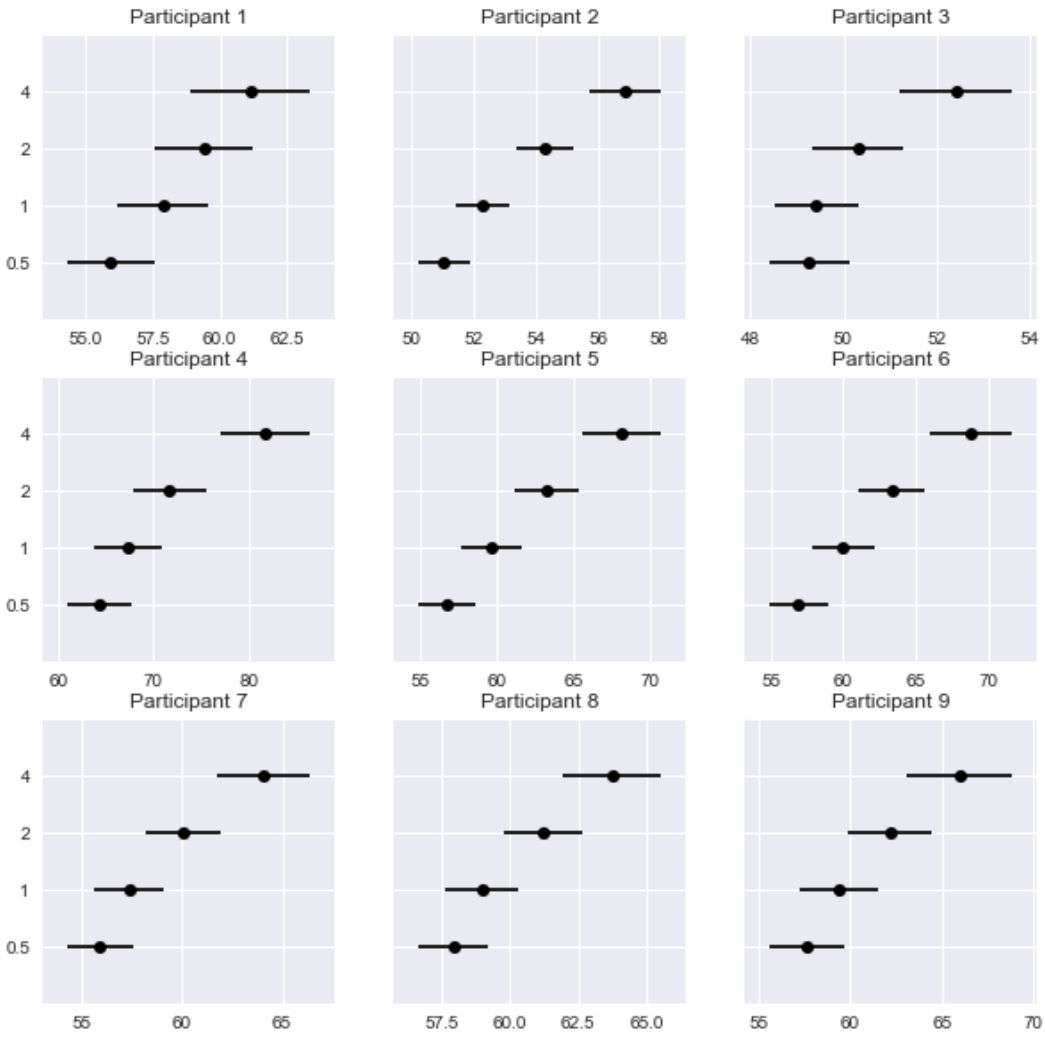


Figure 3.6: The error bars with confidence intervals for each group separately employing LDA and CSP features. The y-axis is the window-size and the x-axis is the average across trials.

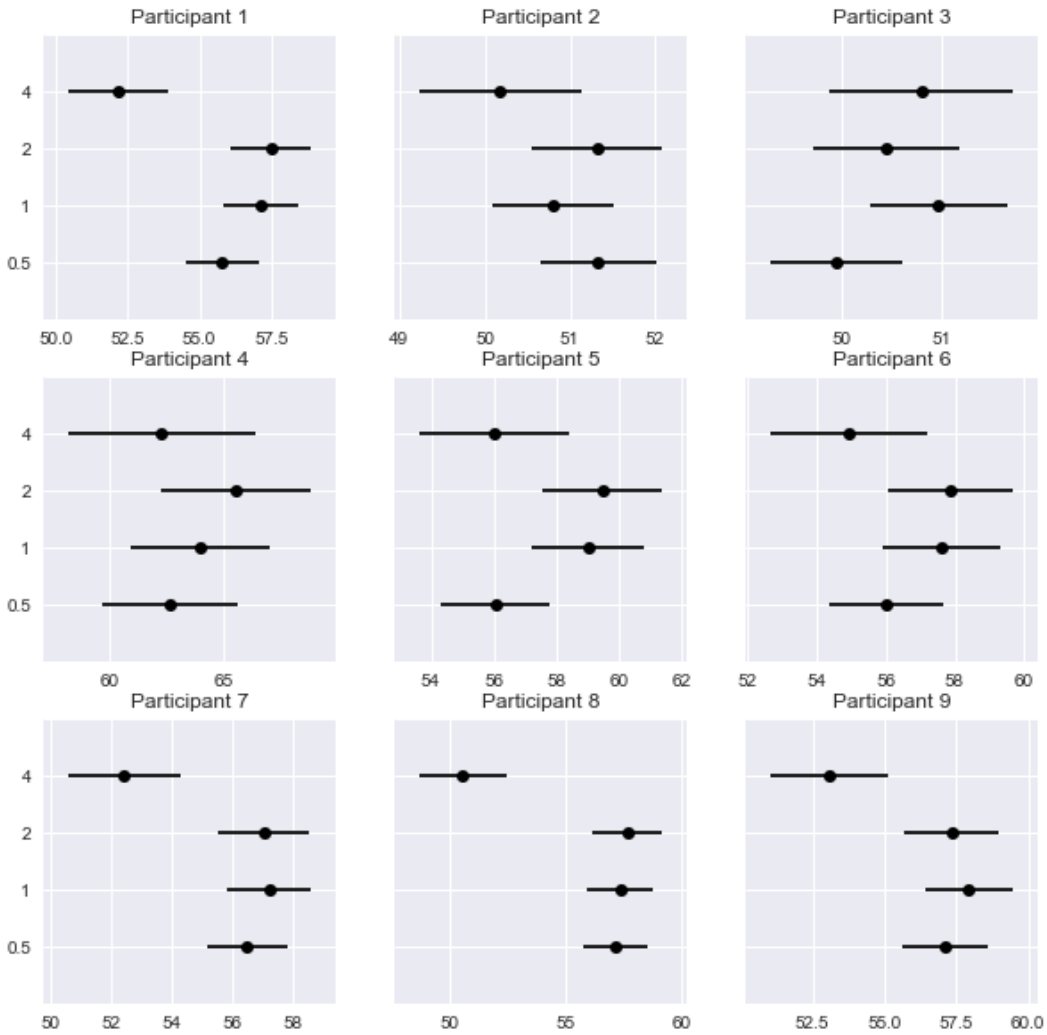


Figure 3.7: The error bars with confidence intervals for each group separately employing LDA and FFT features. The y-axis is the window-size and the x-axis is the average across trials.

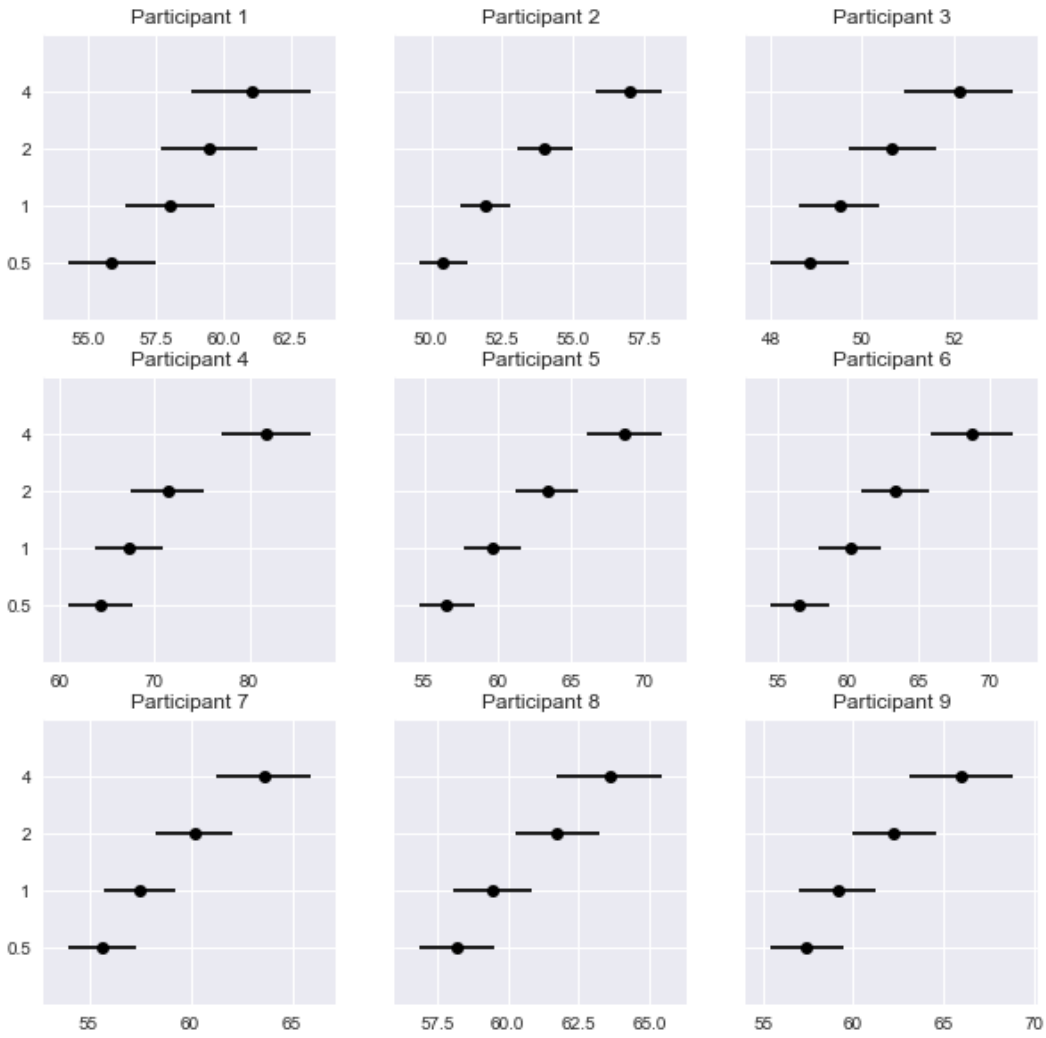


Figure 3.8: The error bars with confidence intervals for each group separately employing SVM and CSP features. The y-axis is the window-size and the x-axis is the average across trials.

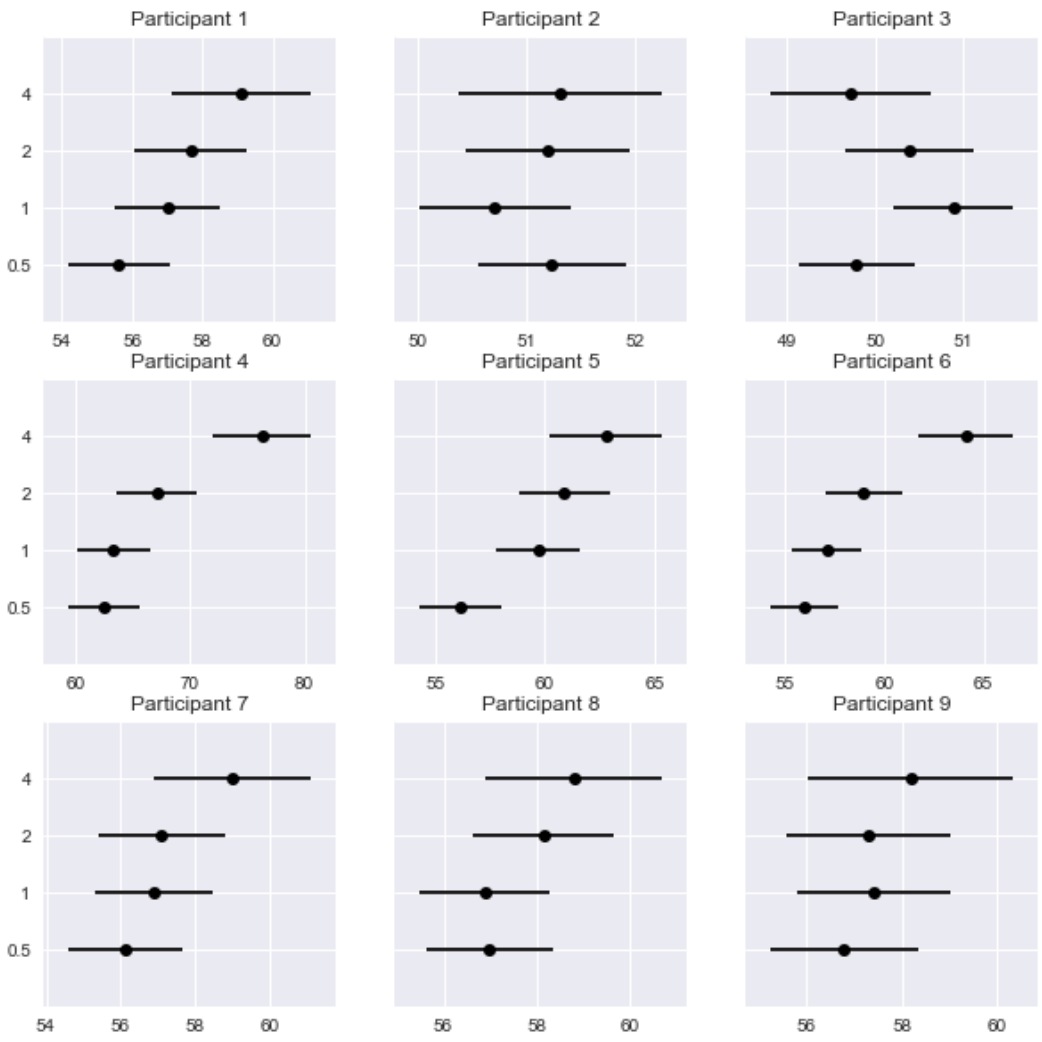


Figure 3.9: The error bars with confidence intervals for each group separately employing SVM and FFT features. The y-axis is the window-size and the x-axis is the average across trials.

The CSP features with a LDA yielded an average accuracy of 76.80 % (SD 10.03 %), and a maximum accuracy of 95.00 % for participant 4. On the other hand, the BP features with a LDA yielded an average accuracy of 70.50 % (SD 8.50 %) and a maximum accuracy of 90.00 % for participant 4. Furthermore, The CSP features with a SVM yielded an average accuracy of 74.70% (SD 8.91 %) and a maximum accuracy 93.87 % for participant 4. While the BP features with a SVM yielded an average accuracy of 72.52 % (SD 9.31 %) and a maximum accuracy of 91.75 % for participant 4.

First, the main effect of window size in the CSP and LDA Tukey HSD method for Post-hoc pairwise tests between individual window sizes as shown in Table 3.3 and Fig. 3.6

Participant	ANOVA *F*(3,241)	All	0.5 vs 1.0	0.5 vs 2.0	0.5 vs 4.0	1.0 vs 2.0	1.0 vs 4.0	2.0 vs 4.0
01	4.68	< 0.01	ns	< 0.05	< 0.01	ns	ns	ns
02	22.13	< 0.01	ns	< .01	< .05	< .01	< .01	ns
03	6.00	< 0.01	ns	< .05	< 0.01	ns	< 0.01	ns
04	10.84	< 0.01	ns	ns	<.01	< .05	<.01	ns
05	16.35	< 0.01	ns	< .01	< .01	< .01	< .01	ns
06	14.19	< 0.01	ns	< .01	< .01	< .01	< .01	ns
07	10.65	< 0.01	ns	< .05	< .01	ns	< .01	ns
08	9.30	< 0.01	ns	< .05	.01	ns	.01	ns
09	7.26	< 0.01	ns	< .05	< .01	ns	< .01	ns

Table 3.3: The main effect of window size in the CSP and LDA. The p-value is only reported where the is a significance between the groups. Where there is no significance, (ns) is reported.

Second, the main effect of window size in the BP and LDA Tukey HSD method for Post-hoc pairwise tests between individual window sizes as shown in Table 3.4 and Fig. 3.7

Participant	ANOVA *F*(3,241)	All	0.5 vs 1.0	0.5 vs 2.0	0.5 vs 4.0	1.0 vs 2.0	1.0 vs 4.0	2.0 vs 4.0
01	7.71	< 0.01	ns	ns	< 0.05	ns	< 0.01	< 0.01
02	ns	ns	ns	ns	ns	ns	ns	ns
03	ns	ns	ns	ns	ns	ns	ns	ns
04	ns	ns	ns	ns	ns	ns	ns	ns
05	ns	ns	ns	ns	ns	ns	ns	ns
06	ns	ns	ns	ns	ns	ns	ns	ns
07	5.80	< 0.01	ns	ns	< 0.01	ns	< 0.01	< 0.01
08	12.31	< 0.01	ns	ns	< 0.01	ns	< 0.01	< 0.01
09	4.48	< 0.01	ns	ns	< 0.05	ns	< 0.01	< 0.05

Table 3.4: The main effect of window size in the the BP and LDA. The p-value is only reported where the is a significance between the groups. Where there is no significance, (ns) is reported.

Third, the main effect of window size in the CSP and SVM using Tukey HSD method for Post-hoc pairwise tests between individual window sizes as shown in Table 3.5 and Fig. 3.8:

Participant	Anova $F(3,241)$	All	0.5 vs 1.0	0.5 vs 2.0	0.5 vs 4.0	1.0 vs 2.0	1.0 vs 4.0	2.0 vs 4.0
01	4.80	< 0.01	ns	< 0.05	p < 0.01	ns	nss	ns
02	25.77	< 0.01	ns	< 0.01	< 0.01	< 0.05	< 0.01	< 0.01
03	6.63	< 0.01	ns	< 0.05	< 0.01	ns	< 0.05	ns
04	10.92	< 0.01	ns		< 0.01	ns	< 0.01	< 0.01
05	18.31	< 0.01	ns	< 0.01	< 0.01	ns	< 0.01	< 0.05
06	13.93	< 0.01	ns	< 0.01	< 0.01	ns	< 0.01	< 0.05
07	9.82	< 0.01	ns	< 0.01	< 0.01	ns	< 0.01	ns
08	7.90	< 0.01	ns	< 0.01	< 0.01	ns	< 0.01	ns
09	7.88	< 0.01	nss	< 0.05	< 0.01	ns	< 0.01	ns

Table 3.5: The main effect of window size in the CSP and SVM. The p-value is only reported where the is a significance between the groups. Where there is no significance, (ns) is reported.

Fourth, the main effect of window size in the BP and SVM Tukey HSD method for Post-hoc pairwise tests between individual window sizes as shown in Table 3.6 and Figure 3.9:

Participant	Anova $F(3,241)$	All	0.5 vs 1.0	0.5 vs 2.0	0.5 vs 4.0	1.0 vs 2.0	1.0 vs 4.0	2.0 vs 4.0
01	ns	ns	ns	ns	ns	ns	ns	ns
02	ns	ns	ns	ns	ns	ns	ns	ns
03	ns	ns	ns	ns	ns	ns	ns	ns
04	8.86	ns	ns	ns	< 0.01	ns	< 0.01	< 0.05
05	6.06	ns	ns	< 0.05	< 0.01	ns	ns	ns
06	9.72	ns	ns	ns	< 0.01	ns	< 0.01	< 0.05
07	ns	ns	ns	ns	ns	ns	ns	ns
08	ns	ns	ns	ns	ns	ns	ns	ns
09	ns	ns	ns	ns	ns	ns	ns	ns

Table 3.6: The main effect of window size in the BP and SVM. The p-value is only reported where the is a significance between the groups. Where there is no significance, (ns) is reported.

Finally, the results of the best classification accuracies using a fixed window for the training and the testing for each participant's accuracy for each of the methods is shown in Table 3.7. A comparison between the best the classification accuracies of the best performing classifiers (LDA and SVM) with both features (CSP and BP) of this experiment in Table 3.8 with the classification accuracies of a selected state-of-art relevant methods: CSP (Wang et al., 2012), bispectrum (Shahid and Prasad, 2011) and FBCSP (Ang et al., 2012) and the top three performing methods for the BCI competition IV 2b (on the BCI-competition IV web page).

Participant	CSP/LDA	BP /LDA	CSP/ SVM	BP / SVM
01	72.00	68.00	71.00	66.00
02	63.00	52.00	63.00	55.00
03	62.00	60.00	62.00	59.00
04	97.00	95.00	93.00	91.00
05	75.00	73.00	76.00	77.00
06	81.00	54.00	80.00	56.00
07	72.00	66.00	71.00	66.00
08	70.00	68.00	71.00	68.00
09	78.00	64.00	78.00	63.00
Mean	74.44	66.67	73.89	66.78

Table 3.7: Classification accuracies of all participants training employing a fixed window size, determined previously. Performance was measured on the unseen sessions (4 and 5)

Part-icipant	CSP / LDA	BP/LDA	CSP	Chin	Gan	Coyle	Bi-spectrum	FBCSP
01	72.00	68.00	66.56	70.00	71.00	60.00	77.00	68
02	63.00	54.00	57.86	61.00	61.00	56.00	65.00	59
03	62.00	60.00	61.25	61.00	57.00	56.00	61.00	59
04	97.00	95.00	94.06	98.00	97.00	89.00	97.00	98
05	75.00	73.00	80.63	93.00	86.00	79.00	82.00	93
06	81.00	54.00	75.00	81.00	81.00	75.00	85.00	80
07	72.00	66.00	72.50	78.00	81.00	69.00	75.00	78
08	70.00	68.00	89.38	93.00	92.00	93.00	91.00	93
09	78.00	64.00	81.25	87.00	89.00	81.00	87.00	87
Mean	74.44	66.89	75.39	80.22	79.44	73.11	80.00	79.44

Table 3.8: Classification accuracies of the best performing methods using CSP and BP features in this experiment (CSP/LDA and BP/LDA), other state-of-the-art methods and the top three methods of the BCI competition IV 2b.

3.4. Discussion

The aim of this research is to optimise real-time BCI systems performance. To achieve a high performance real-time BCI system, the duration of the signals from where the features are extracted and classified needs to be as short as possible. Therefore, finding the methods that achieve higher performance with shorter windows might lead to an improvement in the current BCI systems. In addition, investigating if there are any useful information in other parts of the signal that can be used to improve the classification accuracy (e.g., if the last second increases the performance or it is just noise). Hence, this experiment was conducted to investigate and compare the classification accuracies of discriminating between two moto-imagery movements using different sized intervals. Moreover, combining methods to identify the performance of each of the two types of features CSP and BP with the most used classifiers SVM and LDA.

This analysis suggests that, using CSP features employing LDA and SVM for the majority of the participants (Figure 3.11) collapsing across window start position, the window size factor had a significant effect on classification accuracy (Figs. 3.7 and 3.9). However, post-hoc tests suggest that this effect is quite specific, with the significant pairwise comparisons being that 2s and 4s windows both outperform the shortest (0.5s and 1.0s) windows (Figs. 3.2 and 3.4), suggesting that longer windows do confer a performance advantage.

On the other hand, using BP features employing LDA and SVM collapsing across window start position, the window size factor had a significant effect on classification accuracy (Figures 3.8 and 3.10) for a subset of the participants (Figs. 3.11). The post-hoc tests suggest that where the effect is witnessed, the significant pairwise comparisons being that 0.5s, 1.0s and 2.0s windows both outperform the longest (4.0s) window (Figs. 3.3 and 3.5), suggesting that shorter windows do confer a performance advantage.

Even though, The CSP outperformed the BP in both applying LDA and SVM with a mean classification accuracy of 75% and 74% respectively as opposed to 70% and 72% for BP (Table 3.4). Where CSP showed a significant effect employing the 4.0s window, and using the BP features

showed no significance or favoured shorter windows (0.5s, 1.0s and 2.0) suggesting that using BP features based systems have the potential for achieving similar or better performance with shorter windows. In addition, there was no significance between the best accuracies achieved between the CSP and BP for both the LDA and SVM.

Furthermore, the second task was to identify if the intervals that scored the highest for each of the participants would result in more generalised features for the rest of the signal and the performance was measured on the two unseen sessions (4 and 5). The intervals that scored the highest in the previous task were used for the training of the model, the classification accuracy was measured on the full trial with overlapping windows. Comparing with the validation sessions (1,2 and 3), there was a decrease of 3% in the average classification accuracy of the BP features for both SVM and LDA, and a 1% decrease in the CSP features employing an LDA, while no change in the CSP features employing SVM, the final results are shown in Table 3.7 .

Accordingly, the findings imply that all the intervals after the cue (the highest accuracies obtained that are higher than chance level lies in the interval of the cue on set (Figs. 3.2 to 3.6) can be used to enforce the decision of the classifier assuming the signals are continuous and the system is operating in real-time. Moreover, in this experiment the pre-processing was minimal where it entailed a band-pass filter and calculating the features as opposed to some of the methods used in Ang et al. (2012), Wang et al. (2012) and Shahid & Prasad (2011) wherein extra steps in the pre-processing was done to enhance the features quality and choose the optimum features. The analysis also suggests that the BP features might be a better candidate for a real-time system and can be used to achieve higher accuracies by identifying the best bands for each of the participants. However, ideally the system should be able to identify and use the features autonomously and without losing information by adding extra pre-processing steps. Hence, further investigation was needed to address some of the findings and limitations of the experiment, such as more optimised feature extraction and classification methods to extract the salient features from all the intervals. The next experiments were conducted to identify the best feature extraction and classification methods for short period signals and suggesting a novel neural network to capture the relationship of the intervals and enforce the decision of the classifier.

3.4.2 Summary of Contributions

- Evaluated the traditional naïve over different window sizes and periods;
- Established the best window sizes and intervals for each of the discussed methods;
- Established that spectral features provide the highest performance for short windows;

3.4.3 Limitations

- Only naïve methods for extraction and classification were evaluated;
- The performance difference between participant subsets was not investigated;
- Performance was not evaluated d using raw EEG signals.

Chapter 4:

A pointwise convolutional neural network for two-class MI classification

4.1 Introduction

As noted in Chapter 1, the most commonly used methods in the BCI literature, such as Filter Bank Common Spatial Pattern (FBCSP) (Ang et al., 2012), bispectrum (Shahid and Prasad, 2011), and Deep Belief Networks (DBN) (Tang et al., 2017), rely upon the extraction and exploitation of features from several frequencies. In other words, filter banks are used and the most discriminatory features from each frequency are identified and weighted to support signal classification. Since, EEG signals are sampled in the time domain, they are noisy and non-stationary. Even though methods such as CSP and FFT yield relatively high accuracies in classification, the transformations and single processing functions used reduce signal resolution. Moreover, EEG signals are continuous and vary over time, features should ideally be extracted in the time or time-frequency domain with minimal transformations to limit resolution loss.

Two-dimensional (2D) and three-dimensional (3D) Convolutional Neural Networks (CNN) have been used very successfully in the image domain for recognition, detection, segmentation and other tasks. The images supplied as input are minimally 2D (e.g., greyscale) and 3D if RGB colour or some other trichromatic scheme. The models that excel in these tasks, such as ResNet (He, Zhang, Ren and Sun, 2015) and VGG (Simonyan and Zisserman, 2014), have a complex and deep structure with a correspondingly high number of parameters to learn. These networks were trained on millions of sample images spanning a broad context.

Newly proposed wavelet-based CNNs methods perform time-frequency classification. These methods have yielded higher rates of accuracy, but still typically entail a number of transformations in the pre-processing stages; for instanace, Tang et al., (2017) and Lee & Choi, (2018) render the EEG data into a 2D image-like form (analogous to scalograms, periodograms, or spectrograms). The wavelet transformation results in a 3D matrix, where the first dimension corresponds to frequency, the second dimension corresponds to the time samples and the third dimension corresponds to the number of electrodes. Thus, the acquired matrix structured like an image matrix (e.g. channels × width × height) and could be processed by CNN based models without fundamental changes of the models architectures.

However, these networks have deep architectures, leading to a relatively larger number of parameters. Deep neural networks favour large datasets, while EEG datasets are usually limited and are typically no bigger than one or two thousand samples. Furthermore, individual participants have unique signal properties that restricts the size of the dataset to a single person per set. In addition, the number of pre-processing steps was increased to include such operations as filter banks and correlation coefficients by trading off resolution and having using end-to-end networks. (Al Rahhal, Bazi, Al Zuair, Othman, & BenJdira, 2018)

Another practice that has shown to improve generalisation and harness the power of Deep Neural Networks for EEG signal classification is *cropped training* introduced by Schirrmeister et al., (2017), wherein a windowing function is applied to each trial to obtain more training samples, and as such can be considered a *data augmentation* approach. Furthermore, in Schirrmeister et al., (2017), the authors introduced three different CNN architectures: a Shallow, a Deep and a Hybrid architecture and the authors demonstrated that the suggested networks are able to learn and merge similar features to FBCSP and FFT in an end-to-end approach with minimal pre-processing (4Hz high pass filter to remove artefacts such as eye blinks) and use cropped training to increase the number of training samples achieving state-of-the-art performance. They demonstrate similarities between FBCSP and FFT features by visualising the filters generated and comparing them with FBCSP and FFT using spatial maps and spectrograms. Finally, a state-of-the-art network called EEGNet has been proposed by Lawhern et al.,(2018) in which the authors attempt to construct a generalised CNN architecture that is able to classify different BCI tasks, such as P300, VSCP and motor imagery using an end-to-end network that handles any EEG signal.

CNNs were initially designed for, and excel at, computer vision tasks. Using CNNs with time domain data, and more specifically EEG signals that are difficult to visually inspect and classify (as opposed to visually inspecting an image to determine if it is a cat or a car, for example), is difficult. Rigorous evaluation and testing are required. Subsequently, a well-defined and state-of-art baseline is typically used for this type of study. Here, Shallow Net and EEGNet have been used as the baseline, since:

(i) They have been shown to achieve high performance for MI classification;

(ii) They are rigorously tested against different datasets with high quality statistical and visual analyses;

(iii) The first two layers are relatively similar with small differences, which is useful for the analysis of this study.

Finally, the following methods exploiting time-domain features have been empirically tested using the models and window sizes discussed previously: Continuous Wavelet Transformation (CWT), Discrete Wavelet Transformation and Spectrograms. Using CWT resulted in the best performance without optimisation or feature engineering. Hence, CWT was chosen for extracting the time-frequency features. Noting that the results of the aforementioned methods were compared using a pilot analysis.

Accordingly, the following key questions are addressed to establish a baseline:

- What is the impact of training the Shallow net and EEGNet with CWT signals on classification accuracy?
- What is the impact of the number of scales and channels on the performance of the network?

These questions were tackled by first replicating the Shallow net and EEGNet, as described in the methods section, and evaluating it with BCI IV 2b to acquire a fair baseline for comparison. In the first task, the CWT signals were used as an input for the network as it is acquiring also a baseline for CWT features. In the second task, the network was rigorously tested by adapting the middle activation layers to reflect the imposed requirements by the CWT. In addition, increasing the number of wavelet scales for the transformation provides a wider spectrum at the cost of increasing the number of parameters, which in turn increases the model's complexity and increases the difficulty of minimising the error using a small dataset.

The problem of the small dataset has been tackled by the cropping training strategy that is described in detail in the methods section and described in detail in Schirrmeister et al., (2017).

However, a well-known problem in motor imagery tasks is the difficulty for participants in maintaining the visualisation of the movement for a long period of time, leading to a transient ERD/ERS in each trial. Even though cropped training (data augmentation) provides more samples from a sliding window, it assumes that the EEG signals are periodic over the 4s after the cue onset and as the authors used the full 4s after the cue onset for their reported accuracies and obtained lower classification accuracy using 2s. Nevertheless, the technique proves that it is very useful for event detection in MI classification. In Lawhern et al.,(2018), the authors used 2s for training and testing $[0.5s, 2.5s]$ after the cue onset for testing the model in a MI task, implying that the cropping strategy doesn't seem to support the claim that the model is able to learn generalised features from the full trial (4s after or at the cue onset).

Here, it is hypothesised that a suitable CNN-based architecture using CWT features with the appropriate data augmentation will improve the classification performance. The proposed architectures are designed and optimised in chronological order to investigate and address the limitations of the aforementioned methods. The proposed architectures are simple and with minor adaptations in the Shallow and EEGNet architectures aiming to combine the strengths of the CNNs and CWT. Tackling those problems leads to the main contributions of this chapter, which are as follows:

- Adapting the Shallow net and EEGNet to introduce a novel CNN structure that is able to learn the most discriminatory features from any number of provided scales. The structure is similar to the Shallow Net with one difference: the depth of the network are the scales of the CWT and the network uses separable deep-wise convolution layers (as in EEGNet) to learn scale-specific temporal and spatial filters, such that the final conv layer is applied to the merged output of the learned filters. Ideally, the network is able to achieve relatively close accuracies to the state-of-the-art methods.
- In Deep CNNs and EEGNet focused on the interpretation of narrow-frequency band-power and spatial features decoded by the networks. Visualising the learned spatial filters at each band in terms of hidden unit activations and highest contributing features (relevance of individual features). Following the previous experiments, in this study, the interpretation

focuses on the time-locality of the features learned by the network. This provides an insight into the second adaptation of the networks, where the temporal filters in Shallow net and EEGNet are replaced by the pre-calculated CWT filters. Finally, the validity of a proposed data-augmentation technique (described below) will be evaluated.

- A novel data-augmentation technique for EEG signals is introduced and evaluated which will be referred to as *shuffled-crossover crops*. As the name implies, the strategy reflects the intrinsic nature of EEG signals by shuffling crops within trials (e.g., switching a fixed number of samples between $1s$ and $3s$) and across trials with the same label (e.g. switching a fixed number of samples between trial N and trial $N + 10$). Clarifying that in a continuous recorded EEG signal, the event may take place in any second or period.

To summarise, the methods proposed in this chapter represent a self-sufficient end-to-end CNN that is able to learn CWT features providing an arbitrary number of scales or a mother wavelet as it was proved to be quite difficult to find the optimal configurations for CWTs. The suggested architecture PWCN (see methods) learns an average of the scaled provided regardless how many scales are used in the decomposition, providing sufficient computational power, a larger number of scales would provide higher resolution of the signals. Furthermore, it eliminates the pre-processing and interpolation techniques to reduce the number of parameters and acquire frequency bands as features by using separable deep-wise conv nets.

Shallow net and EEGNet have been used as the baseline, since these performed better than Deep and Hybrid architectures for MI tasks, as well as being rigorously tested against different datasets with high quality statistical and visual analysis. The study also includes the winner of the BCI IV 2b competition method FBCSP, which was discussed in a previous chapter.

In the first stage, the BCI IV 2b dataset has been used for comparison in order to maintain consistency with previous chapters. In addition, the available computing power makes it very time consuming to test the methods on larger datasets such as BCI IV 2a. However, later on, the most common datasets and methods are compared to directly in order to evaluate the proposed models.

The methods are described briefly and concisely for the ease of reading and following the methods introduced.

Filter Bank Common Spatial Patterns (FBCSP)

The winner of the competition was the FBCSP method proposed in Ang et al. (2008) and Chin et al. (2009), and it is commonly used for methods for decoding MI EEG signals. The FBCSP method is based on Common Spatial Patterns (CSP) that has also been discussed in Chapter 3 (section 3.2) in detail in which it was compared with FFT features. Moreover, it was used in the discussed methods as a Benchmark in their studies. Finally, the suggested conv nets adapted for EEG classification as in Shallow net, Deep-Net, and EEGNet, contain a conv block that mimics the basis of the FBCSP pipeline. Thus, the following is a brief description of the steps of FBCSP:

- **Band-pass Filtering**: The signals are band-passed into the frequencies of interest usually between 4-40 Hz, a number of band-pass filters are used (in Ang et al. (2008) and Chin et al. (2009), 9 band-pass filters were used to obtain 9 banks each is 4 Hz wide).
- **CSP:** For each of the filter banks, CSP is applied and spatial filters are extracted in a supervised manner for training and for decoding the learned filters are applied to the test signals (refer to Chapter 2 for the detailed computation). Emphasizing that the feature vectors are the logged minimum and maximum variance between two classes for each filter bank.
- **Classification:** A classifier is trained using the log-variance spatial features, while for prediction and testing the CSP filters are applied in an unsupervised manner, which can be referred to as *decoding*.

In this study, the ShallowFBCSP and EEGNet will be discussed in detail to provide an understanding of the introduced architecture and the motivation behind it in this section. For details on the Deep and Hybrid Nets, refer to the study by Schirrmeister et al., (2017), since they will be mentioned briefly throughout the chapter. More importantly, the adaptation and adjustments of the networks to provide CWT as features are will be outlined in the relevant parts of the method.

I. Shallow net

The Shallow net was inspired by the FBCSP pipeline. The structure of the network aims to extract the band-power features at the first layer which corresponds to the band-pass filtering in the FBCSP by having a temporal kernel of size (Time Conv Filter x 1) (Fig. 3.1) operating as convolution over time for each channel (electrode). Looking at the kernel size, the width is set to one to restrict the filter learning to the temporal features. The second layer is analogous to a spatial filter and the kernel size is (1, number of channels) restricting the filter to learn the spatial relationship in the form of weights between the electrodes pairs. Inspecting the kernel size, the height is set to one. Emphasising that splitting the two layers and having two different kernels (temporal and spatial) as opposed to having one conv layers with a kernel of size (25, number of channels) was argued to provide better performance with larger number of channels. (Schirrmeister et al., 2017).

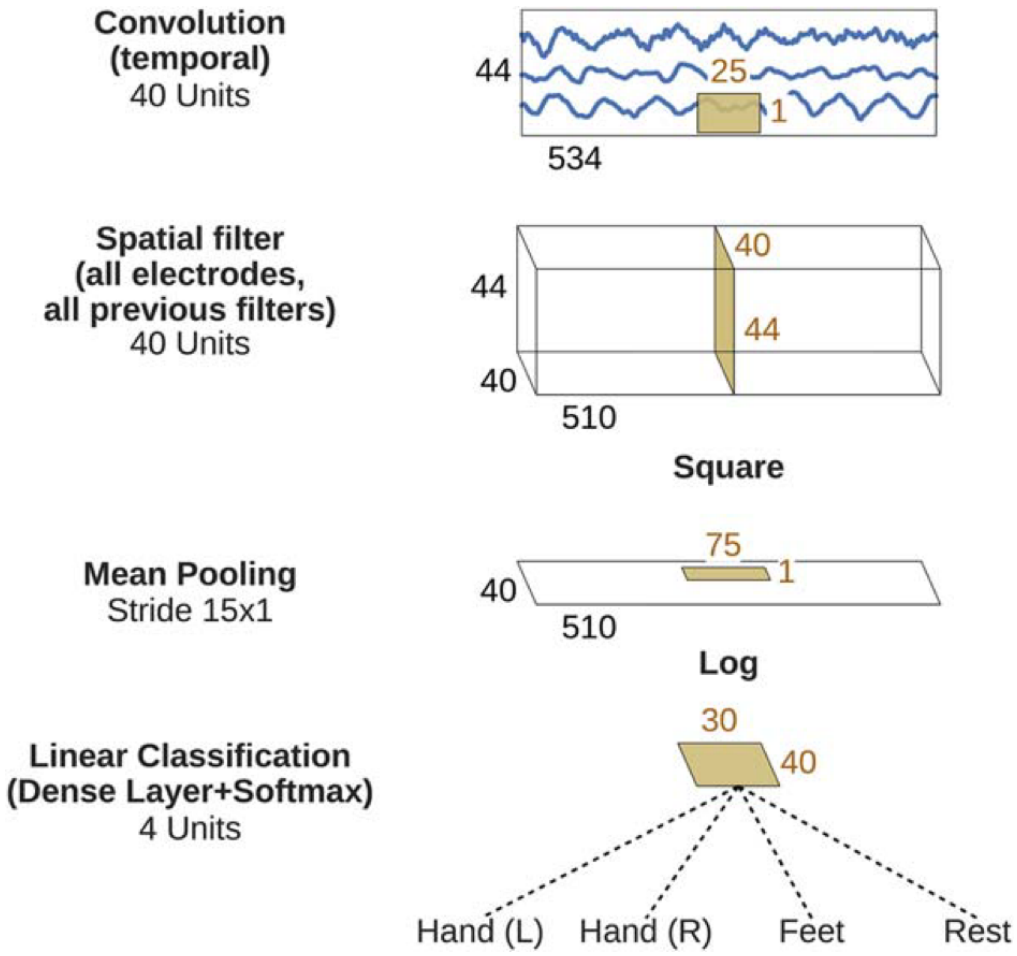


Figure 4.1: The original Shallow net architecture for four classes. The first temporal layer has a kernel of size (25,1), the spatial filter (second conv layer) has 44 channels and the kernel size is (1,44) and a depth of 40 (40 channels output of the temp filter). A mean pooling layer of the filters, and finally a fully connected layer for classification of 4 classes {Hand (L), Hand (R), Feet and Rest}.

A squaring non-linearity layer (squaring the output of the spatial filter) acquiring the instantaneous power is applied to the output of the temporal and spatial filters. Finally, the last steps of the FBCSP pipeline are to calculate the log-variance and to mirror this computation to a mean pooling layer, followed by a logarithmic activation function that is applied to the instantaneous power acquired from the previous layer.

Finally, for activation layers, an exponential linear unit (ELU) was used as the activation function, succeeding the output of the filters acquired from the spatial layer, noting that there are no activation layers between the temporal conv layer and the spatial layer and principally, those two layers can be combined into one conv layer. The ELU activation is defined as shown in Eq. 3.1, given by Clevert et al. (2016). The original network architecture is shown in Fig. 3.1.

$$f(x) = \begin{cases} e^x - 1 & if\ x \geq 0 \\ x & otherwise \end{cases} \qquad \text{Eq. 3.1}$$

II. EEGNet

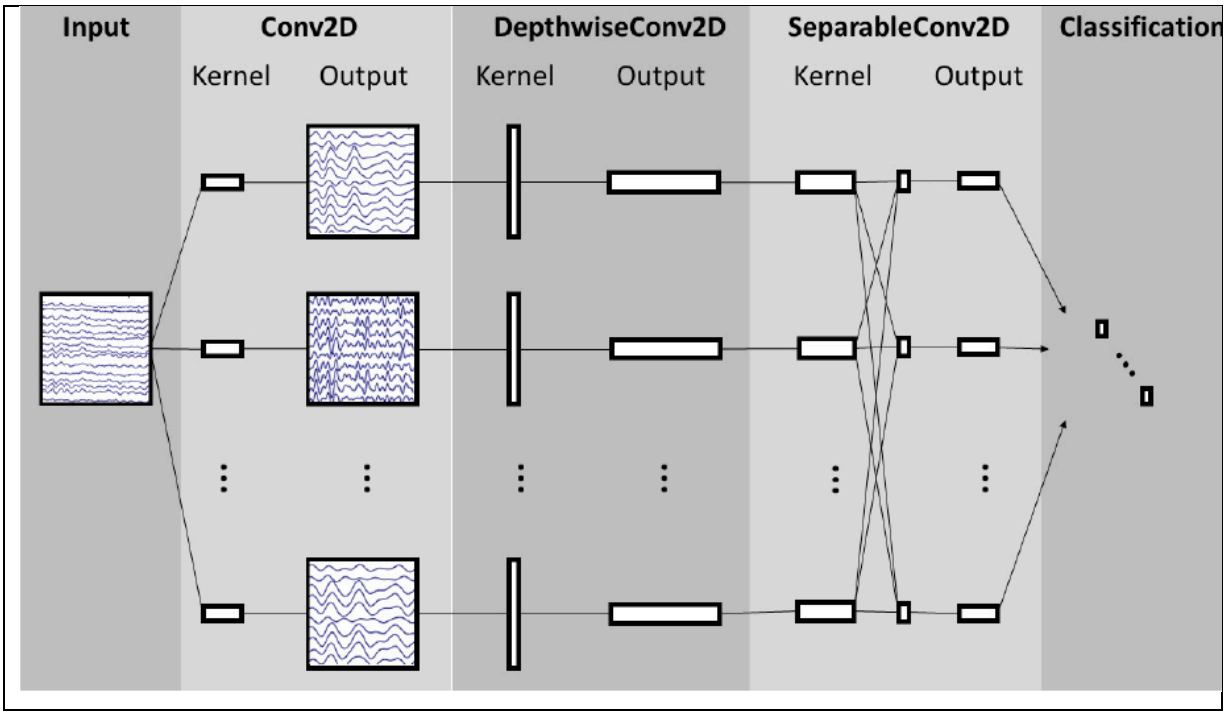


Figure 4.2: The original EEGNet architecture for four classes.

The EEGNet architecture was designed as a general-purpose model for different BCI paradigms. The architecture aims to reduce the number of trainable parameters and maintaining a compact (small) number of layers. The first two convolutional layers are serving the same purpose of the Shallow net that is fundamentally extracting the band-power features at different band-pass frequencies. The second layer namely Spatial-Conv in Shallow net aims to learn spatial maps

for each temporal filter using a depth-wise convolution filters allowing the filters to extract frequency-specific filters. The depth-wise convolution operates on each channel separately to decouple the relationship across the filters, in other words, for each channel a set of filters are learned independently where each channel corresponds to a specific like frequency. Finally, a point-wise convolution follows with the purpose of learning the optimal combined feature maps across all the channels (frequency like). Another difference between Shallow net and EEGNet, where Shallow net employed a squaring function as the non-linear activation, EEGNet employed Exponential Linear unit (ELU) activation functions after the temporal convolution layer and after the depth-wise and point-wise convolutional operation as shown in Figure 4.2. Finally, the output is flattened and fully connected layer is added for the classification. For full detail of the network refer to the study.

4.2 Method

4.2.1 Datasets and experiment procedure

To evaluate the performance of manipulating the proposed windows, the BCI competition IV dataset 2b and BCI Competition IV 2a descripted in section 2.1.

4.2.2 Apparatus

A NVIDIA 650 G-force GPU card was used for the calculations of the model parameters (Gradient descent and model updating), with CUDA 8 and Pytorch (Paszke et al., 2017) for the implementation of the proposed models and the Neural Networks based models evaluated (EEGNet, Shallow net) on a Linux based machine with 4 Quad-cores.

Formal representation of the input (for ease of reading throughout the chapter)

The datasets used in this experiment are defined as $D^i = \{ (X^1, y^1),\ldots,(X^{Ni}, y^{Ni}) \}$ such that D is the dataset of subject i, *and* N^i denotes the number of trials for subject i. X denotes the number

of trials which belongs to class y and $X^j \in \mathcal{R}^{ET}$ where j is the trial number and $1 < j < N_i$, E as the number of electrodes recorded and T as the total number of time samples per trial. The representation of the input to the Convnets are crucial, and as discussed in the literature review (Chapter 1), one common approach is to provide the input as an EEG 'image', whilst the Shallow Net was designed to operate on raw EEG signals aiming to learn spatially global filters and local temporal filters. Concretely, the models are evaluated on the BCI IV 2b competition dataset with the two classes: left hand, right hand, such that a trial j has a corresponding class $y_j \in \{l_1$= Hand left, l_2 = Hand right$\}$.

To feed the 2D matrix as an input to the ConvNet a third dimension is added to form a tensor, where the third dimension represents depth (also referred to as channels in conv nets literature). In this case, there are no depth-wise convolutions at the first layer.

4.2.3 Continuous Wavelet Transformation (CWT) for feature extraction

Wavelet transformation was introduced by Daubechies (1990) as an improvement over Fourier transformation to analyse signals with non-stationary instantaneous power found over several frequencies which has proved (as discussed in Chapter 1) to be an outstanding candidate for EEG feature extraction and analysis. Given the raw signal $X(t)$, a single trial $X^j(t)$, the wavelet transformation is applied to each trial at each electrode (will refer to the electrode number as e) $X_e^j(t)$, as:

$$CWT(\omega, s) = \frac{1}{\sqrt{|s|}} \int f(X_e^j) \psi * \left(\frac{X_e^j - \omega}{s} \right) dt \qquad \text{Eq. 4.2}$$

Where ψ is the mother wavelet and a Morlet wavelet (Eq. 4.3) is used in this study, $\psi *$ as the complex conjugate, ω and s are the scaling and shifting parameters respectively. The general definition of the Morlet wavelet is:

$$\psi(t, f_0) = \frac{1}{\sqrt{\sigma_t \sqrt{\pi}}} e^{-\frac{t^2}{\sigma_t^2}} e^{j2\pi f_0 t} \quad \text{Eq. 4.3}$$

Where f_0 is the central frequency of the wavelet function which was set to 10 Hz and σ_t is a term that is used for the trade-off between temporal resolution and spectral resolution. The resulting matrix after applying the CWT to the EEG signals, is the matrix $X^j \in \mathcal{R}^{SET}$ where N is the number of scales (frequencies of interest, 8 Hz < S < 50 Hz), T the transformed time steps (samples) and E is the number of electrodes. To this end, the first layer of the following models namely the temporal convolution layer becomes a depth-wise convolution layer operating on 2 dimensions (scales and samples).

4.2.4 Model Architectures and Adaptations

Adaptation of Shallow net for CWT

To acquire a baseline using CWT features, the first adaptation was to remove the squaring non-linearity layer and pass the features straight to the log-variance and mean pooling layers. The motivation behind that is CWT without squaring or calculating the power of each frequency showed to obtain remarkably high accuracies as shown in (Al Rahhal, Bazi, Al Zuair, Othman, & BenJdira, 2018). The first convolutional layer of the Shallow net and EEGNet aim to learn a temporal representation or more precisely the band features for different frequencies per feature map (artificial channels generated by the kernels). One way of testing the performance of the network is replacing the learned feature maps by pre-calculated CWT features. Hence, the temporal convolutional layer was removed and the the second convolutional layer aiming to learn the spatial correlations is now operating on the CWT features such that input $X \in \mathbb{R}^{STE}$. Where the S are the number of scales, T are the time-steps and E are the input channels (electrodes).

III. Point-wise Convolutional Network (PwCN)

Using CWT features as an input to the previous networks (Shallow net and EEGNet) increases the dimensionality of the input significantly by the factor S (the number of *scales)*, which in turn increases the number of parameters. Building on top of their architectures, one key factor is considered for the purpose of BCI, by keeping the spatial convolution layer to learn spatial maps across channels.

The following architecture is proposed to achieve similar or relatively close performance using CWT features. Noting that only a CWT is applied to the raw signals at the pre-processing stage using a Morlet wavelet as the mother wavelet covering the frequencies from 9 Hz to 31 Hz resulting in 32 scales. No interpolation or dimensionality reduction is applied.

- The first layer is a point-wise convolutional layer, the kernel size = (1,1). The input to the network is equal to the number of scales provided (32 for BCI 2b and 59 for BCI 2a). This layer aims to learn a summary of the band-power like features across the scales since the temporal representation has been already calculated by the CWT, assuming that the event takes place at different time steps in the different frequencies. The output of the layer behaves like a feature reduction at this level. If the output is given as one channel, the kernel will find the optimal combination of those scale features providing one channel for the next layer. In this experiment, the output channels were tested for $N = 40,20$ and 4, meaning that the point-wise layer is learning S number of optimally mixed scales (frequencies) and the N number of kernels represents different variations of N maximising the performance.
- A batch normalisation procedure is employed to speed training and reduce overfitting (Ioffe & Szegedy, 2015).
- The second layer is an average pool layer with a kernel size of $(16,1)$ and a stride of $(16,1)$ for further dimensionality reduction.
- Followed by the spatial convolution layer with a kernel size of (1, input channels) to learn the spatial maps.
- Batch normalisation.
- The linear activation function ELU (Eq.4.1) is applied.

- Another convolutional layer is added to deepen the model and providing another level of transformation. The kernel size was set to $(10,1)$. Noting that in the Spatial-Conv layer, the number of channels (the width of the input) is reduced to a value of 1.
- Another average pooling layer for dimensionality reduction of kernel size and stride size of $(32,1)$.
- A drop-out with 50% probability that enables the model regularize the mode by zeroing out 50% of the neurons while training which in turn prevent over-fitting especially with small sample sizes.
- Finally, a fully connected layer with a Softmax activation function (Gibbs, 1902) for the classification of the movements.

The model was fitted using an Adam optimiser (Kingman and Ba, 2014), where the parameters are kept as the default. The error was minimised by the categorical cross-entropy loss function. The early stopping is similar to the one used in Shallow Net for consistency, such that the model stops early according to the lowest validation loss. Another important detail on early stopping, the weights of the model are saved after finding a new best validation loss and the model is reset to those weights, the training continues from the checkpoint of the last best model.

4.2.5 Augmentation using Shuffled-Crossover crops

Data augmentation is a common technique used in training neural networks, where it aims to reduce model overfitting using existing information in the training generating new data samples. The widely generic practices entail cropping, flipping an image, rotating the image and colour changes or in other words geometric augmentation (Cireşan, Meier, Gambardella, & Schmidhuber, 2010; Yang, Zhao, Chan, & Yi, 2016). As discussed in the experiment procedure (see section 4.2.1), participants were instructed to maintain the MI movements for at least four seconds when the data is being collected. Ideally, it is assumed that the temporal structure is similar and periodic over the four seconds (hence as discussed thoroughly throughout techniques like Fast Fourier Transform (FFT) and CWT for EEG are the traditional techniques used for the analysis and feature extraction). A sample shuffling is suggested in this study, for simplicity, the E^j will be omitted

from the Eqs. since the following operation is applied over all electrodes at E^j in the same way. Given the time samples T^j of trial $X^j \in \mathcal{R}^{ET}$ and a crop $t^j[s,e] = \{t^j : T_s^j < t^j < T_e^j\} \subseteq T^j$ where $s\ and\ e$ denote the location of the starting and ending sample respectively. Samples t^j are swapped with non-overlapping samples $\bar{t}^{\,j}$ generating a new training example $\hat{X} \in \mathcal{R}^{ET}$. Finally, a crop may or may not be also swapped across trials i.e. swapping $\bar{t}^{\,j=10}$ with $t^{j=200}$. In this study, just one configuration of augmentation was tested, where the number of samples to be swapped was the equivalent of two seconds: $t^j = [s = 0, e = 500] \wedge \bar{t}^{\,j} = [\mathrm{s} = 500, \mathrm{e} = 1000]$ and further shuffling between trials was random and set to a maximum of 20 trials crossover shuffles.

4.2.6 Visualisation and analysis

Motor imagery BCI has been analysed thoroughly using the standard methods such as spectrograms, FFT, CWT and DWT. Researchers are presently working to understand and interpret what the artificial neural networks are learning and 'look into' the black box. Mainly looking into CNNs in computer vision problems. In Schirrmeister et al. (2017), the authors applied a technique wherein they extracted filters weights and applied a frequency correlation between the classes, showing an improvement of the increase and decrease (ERD and ERS) at deeper levels. A similar procedure was performed in the Lawhern et al. (2018), but the authors showed examples of the P300 component.

In this study, the visualisation and interpretation were focused on the receptive fields immediately following the spatial-filter. The technique used is known as Guided Gradient-weighted Class Activation Mapping (Guided Grad CAM) as described in Selvaraju et al., (2017) and discussed in detail in Chapter 5, where a trained model is forward and back-propagated to emphasise features or nodes that contribute the most to maximising a class. Put simply, the model is back-propagated to compute the gradient with respect to the provided class. Furthermore, a constraint was set where any weights < 0 would be set to be 0. Secondly, since signals are more difficult to visualise, the weights at the selected layer for visualisation are added to the original signal. Finally, a scalogram of the obtained signal maximising a class is calculated, as shown in Figs. 4.8 to 4.10.

4.3 Results

4.3.1 Performance of baseline networks

Baseline evaluation for Shallow net and EEGNet achieved similar results, with a decrease in kappa values for BCI IV 2b. This is believed to be a consequence of how it was tested (the difference will be discussed in the next section). The implementation of the original models was replicated locally and accuracies are compared with those reported for BCI IV 2b. Moreover, the tested period of the trial for the baseline was from $0.5 - 2.5s$ after cue onset for consistency with the reported accuracies.

Table 4.1: Classification accuracies and Kappa for 10×10 fold. Sessions 1, 2 and 3 only. The error is measured as SD across folds and all subjects. The error rate for accuracy is measured as SD across folds and all subjects and as the mean SE for kappa.

Network	Accuracy	Kappa	No. of Parameters
ShallowFBCSP	0.78 ± 0.13	0.51 ± 0.05	6482
EEGNet	**0.81** ± 0.10	**0.62** ± 0.04	1618
ShallowFBCSP-CWT	0.72 ± 0.13	0.45 ± 0.05	37482
EEGNet-CWT	0.73 ± 0.15	0.46 ± 0.05	17490

Table 4.2: Classification accuracies, Kappa and F-Beta using the unseen sessions 4 and 5 for testing.

Network	Accuracy	Kappa	F-Beta
ShallowFBCSP	0.78	0.57	{0: 0.78, 1: 0.79}
EEGNet	0.83	0.65	{0: 0.82, 1: 0.83}
ShallowFBCSP -CWT	0.76	0.51	{0: 0.76, 1: 0.75}
EEGNet-CWT	0.76	0.53	{0: 0.76, 1: 0.77}

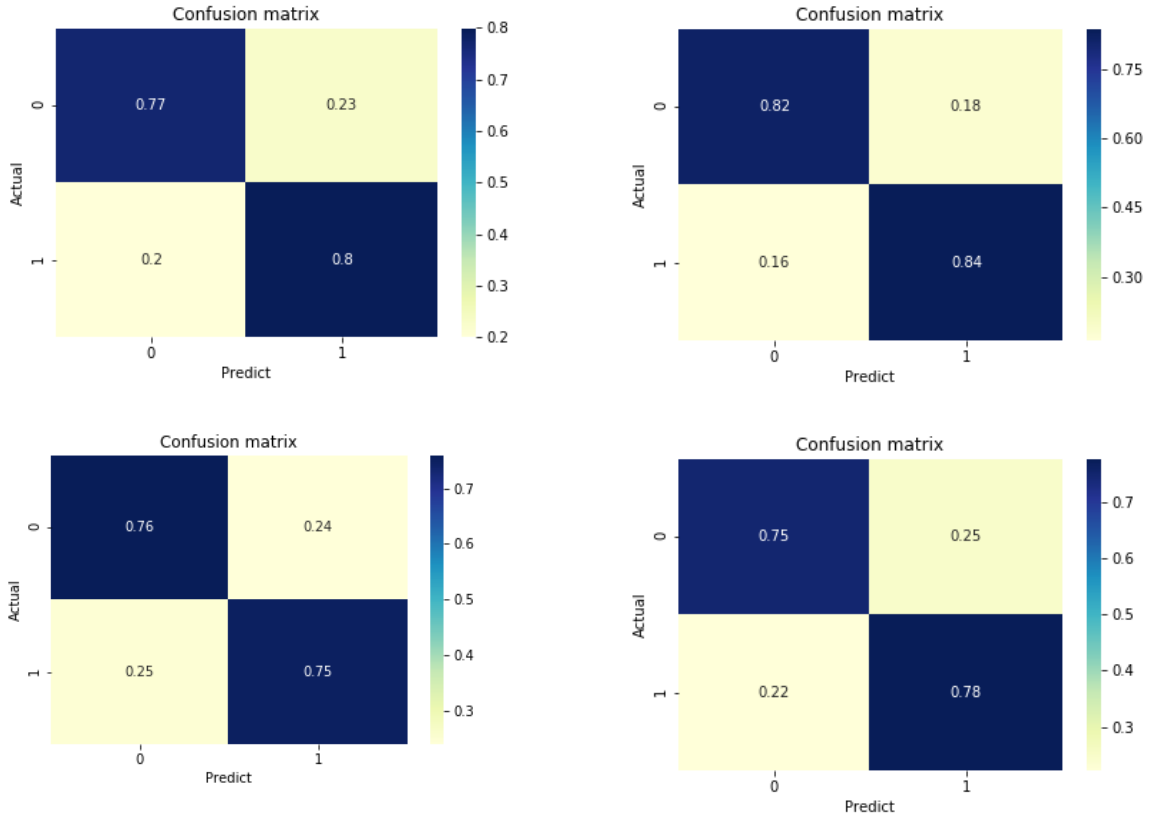


Figure 4.3: The normalised confusion matrix of the mean accuracies of the within-participants with the unseen sessions 4 and 5. Top Left is the Shallow net confusion matrix and the Top right is the EEGNet confusion matrix. Bottom Left is the Shallow net-CWT confusion matrix and the Bottom right is the EEGNet-CWT confusion matrix.

4.3.2 Replacing the first Temporal-Conv layer with the CWT features (32 scales).

Table 4.3: Classification accuracies and Kappa for 10 × 10 Fold. Sessions 1, 2 and 3 only. The error is measured as SD across folds and all subjects. The error rate for accuracy is measured as SD across folds and all subjects and as the mean Standard Error for Kappa.

Network	Accuracy	Kappa	No. of Parameters
Shallow net_CWT	0.74 ± 0.13	0.49± 0.05	7122
EEGNet_CWT	0.76 ± 0.13	0.54± 0.05	4762

Table 4.4: Classification accuracies, Kappa and F-Beta using the unseen sessions 4 and 5 for testing.

Network	Accuracy	Kappa	F-Beta
Shallow net_CWT	0.75	0.50	{0: 0.75, 1: 0.76}
EEGNet_CWT	0.77	0.55	{0: 0.77, 1: 0.77}

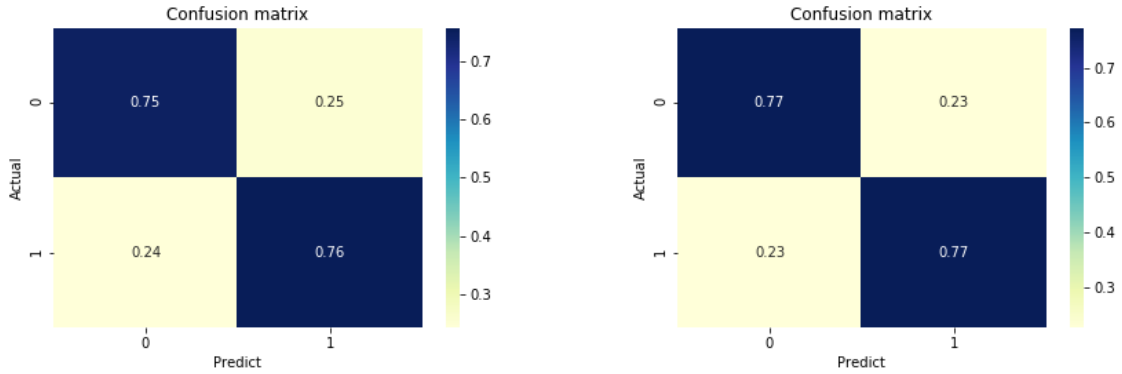


Figure 4.4: The normalised confusion matrix of the mean accuracies using CWT as input of the within-participants with the unseen sessions 4 and 5. Left is the Shallow net-CWT confusion matrix and the right is the EEGNet-CWT confusion matrix.

4.3.3 The performance of the PWSCN.

Table 4.5: Classification accuracies, Kappa and F-Beta using the unseen sessions 4 and 5 for testing.

Network	Accuracy	Kappa	F-B	Parameters
PWSCN	0.80	0.59	{0:0.79, 1: 0.80}	7760

Confusion matrix:

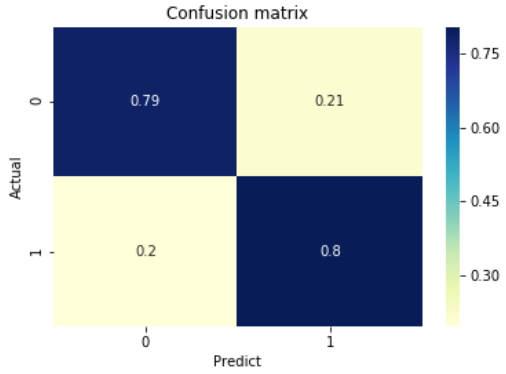
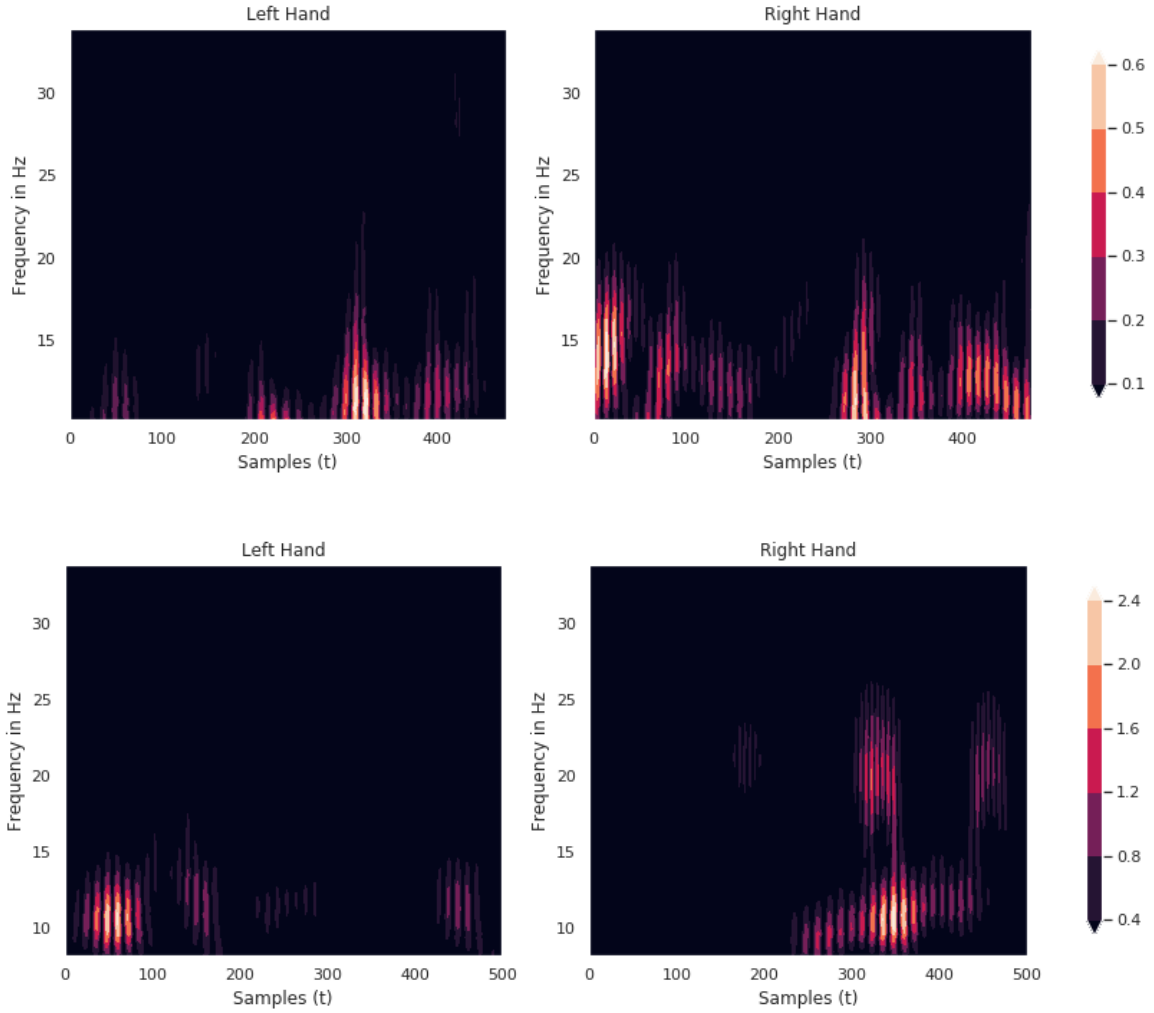
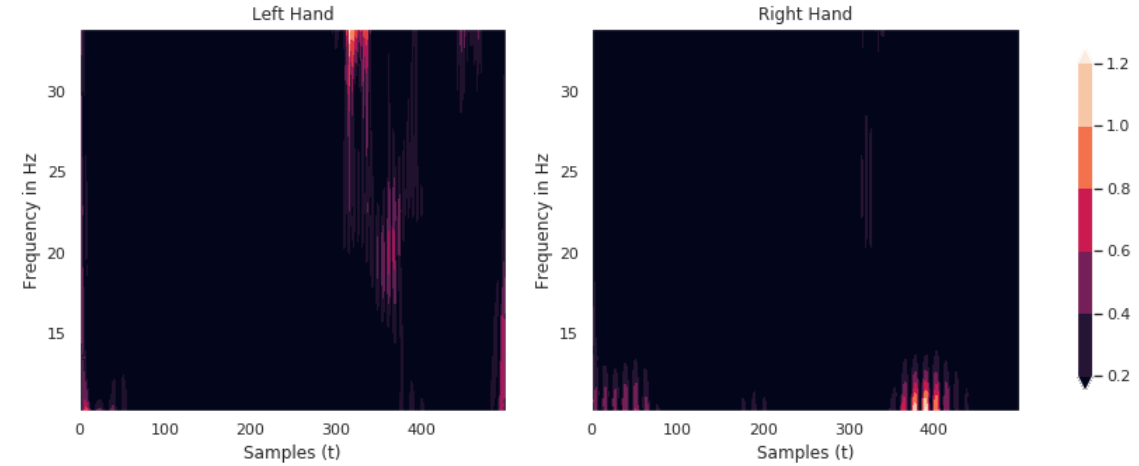


Figure 4.5: Normalised confusion matrix for the PWCN.

Result 4: The visualisation for some participants.




Left Hand
Right Hand
Frequency in Hz
Samples (t)
30
25
20
15
0
100
200
300
400
1.2
1.0
0.8
0.6
0.4
0.2

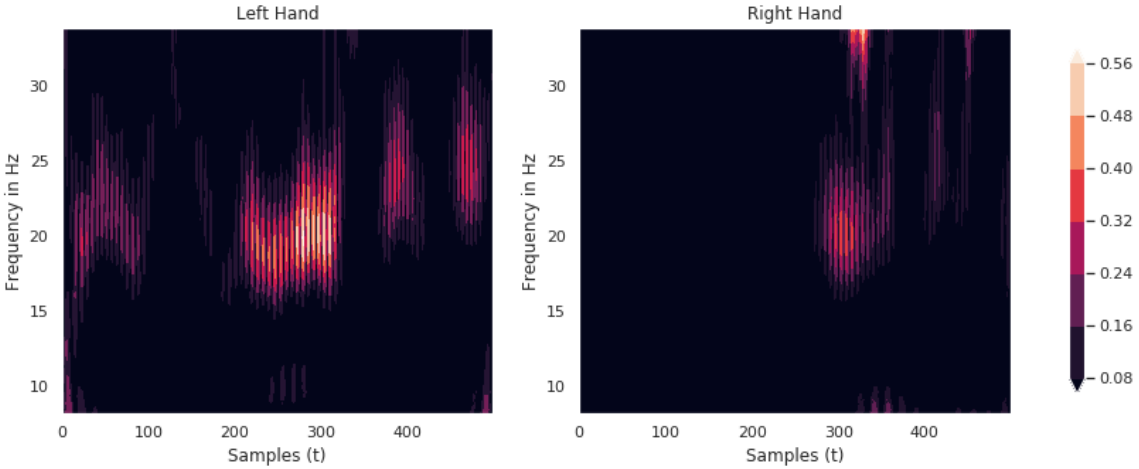
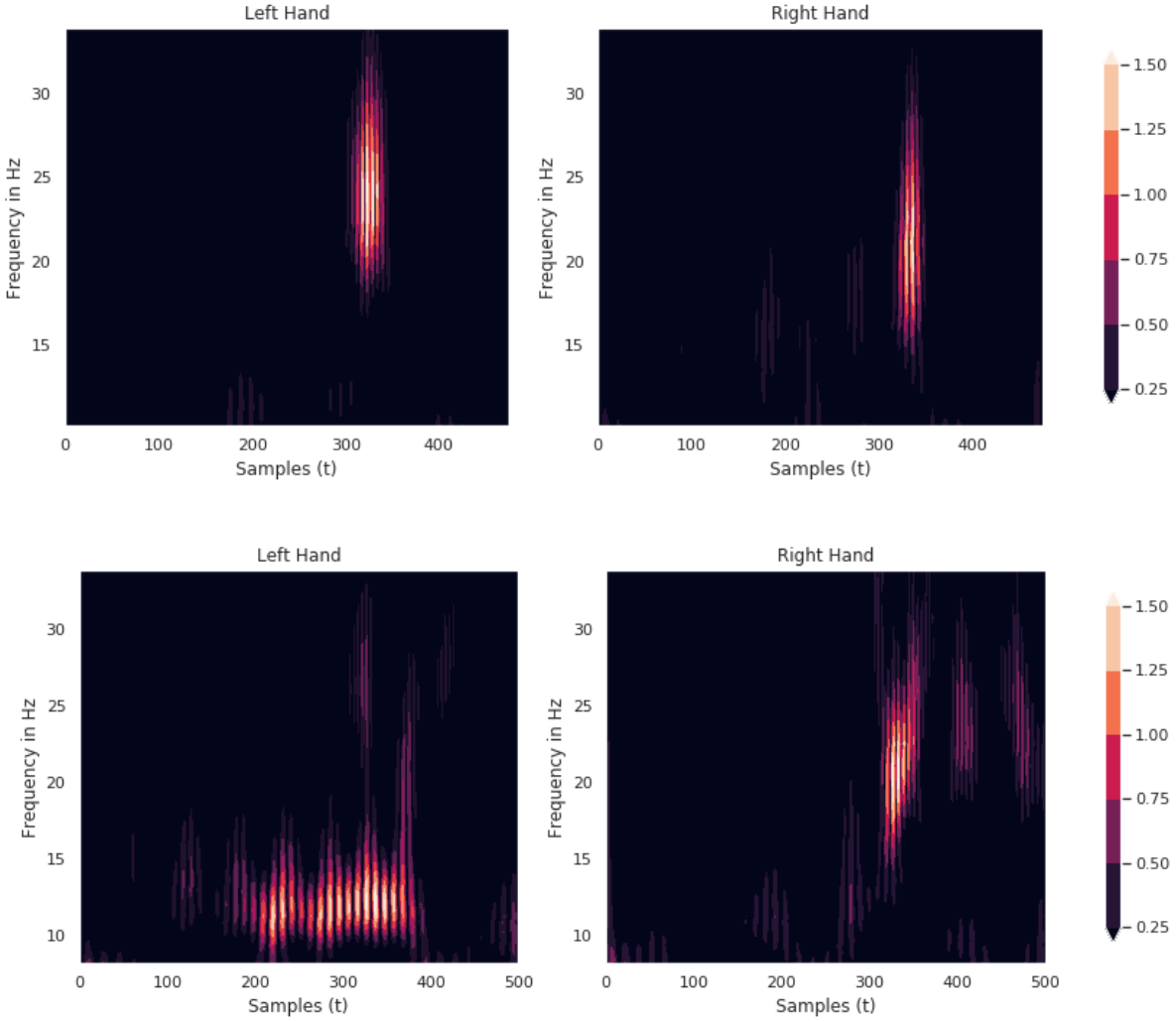

Left Hand
Right Hand
Frequency in Hz
Samples (t)
30
25
20
15
10
0
100
200
300
400
0.56
0.48
0.40
0.32
0.24
0.16
0.08

Figure 4.6: Scalogram after applying guided CAM. Applied on the spatiotemporal layer for participant 4. First row is the Shallow net, second row is the EEGNet, third row Shallow net_CWT and fourth row is EEGNet-CWT.



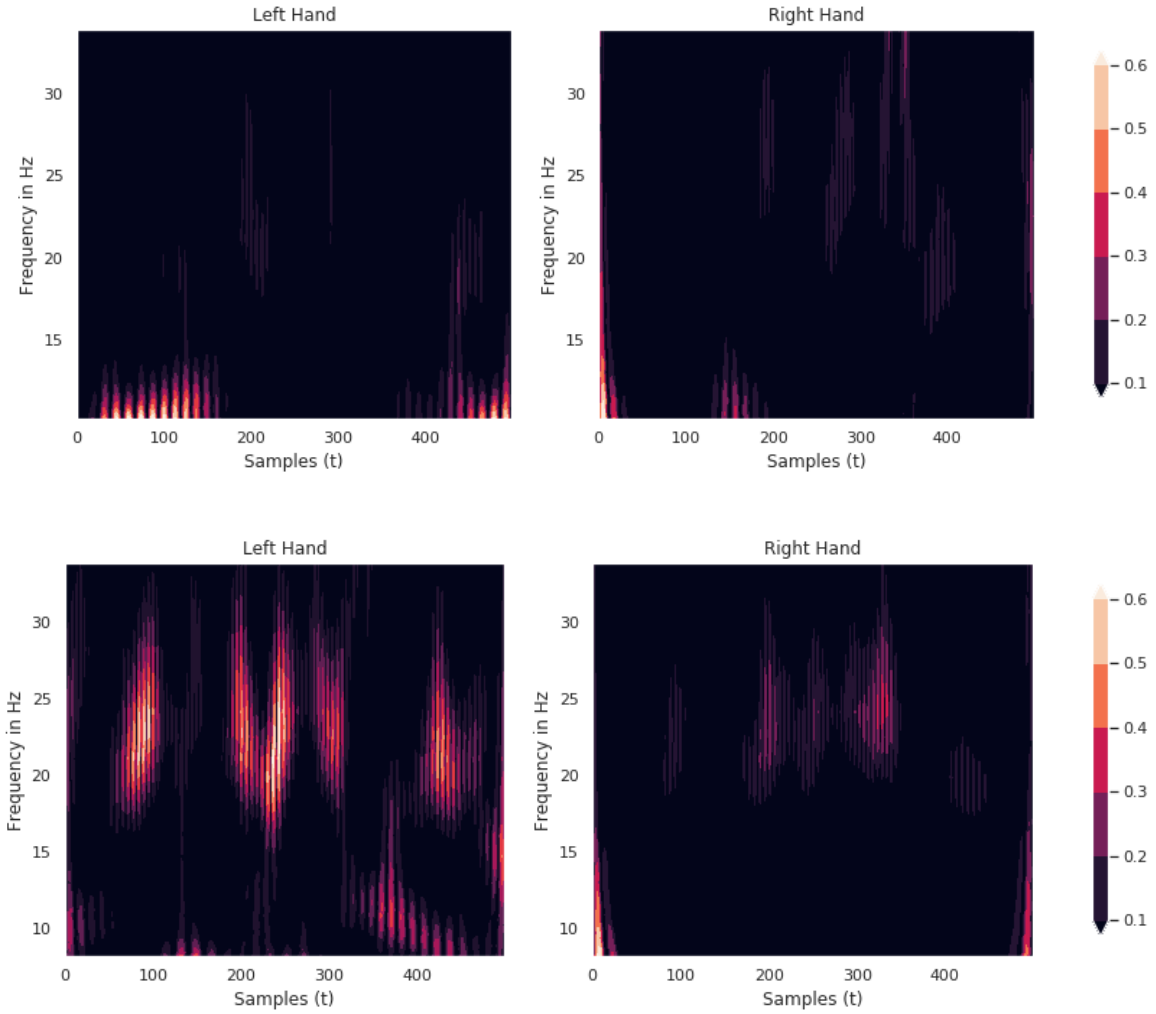
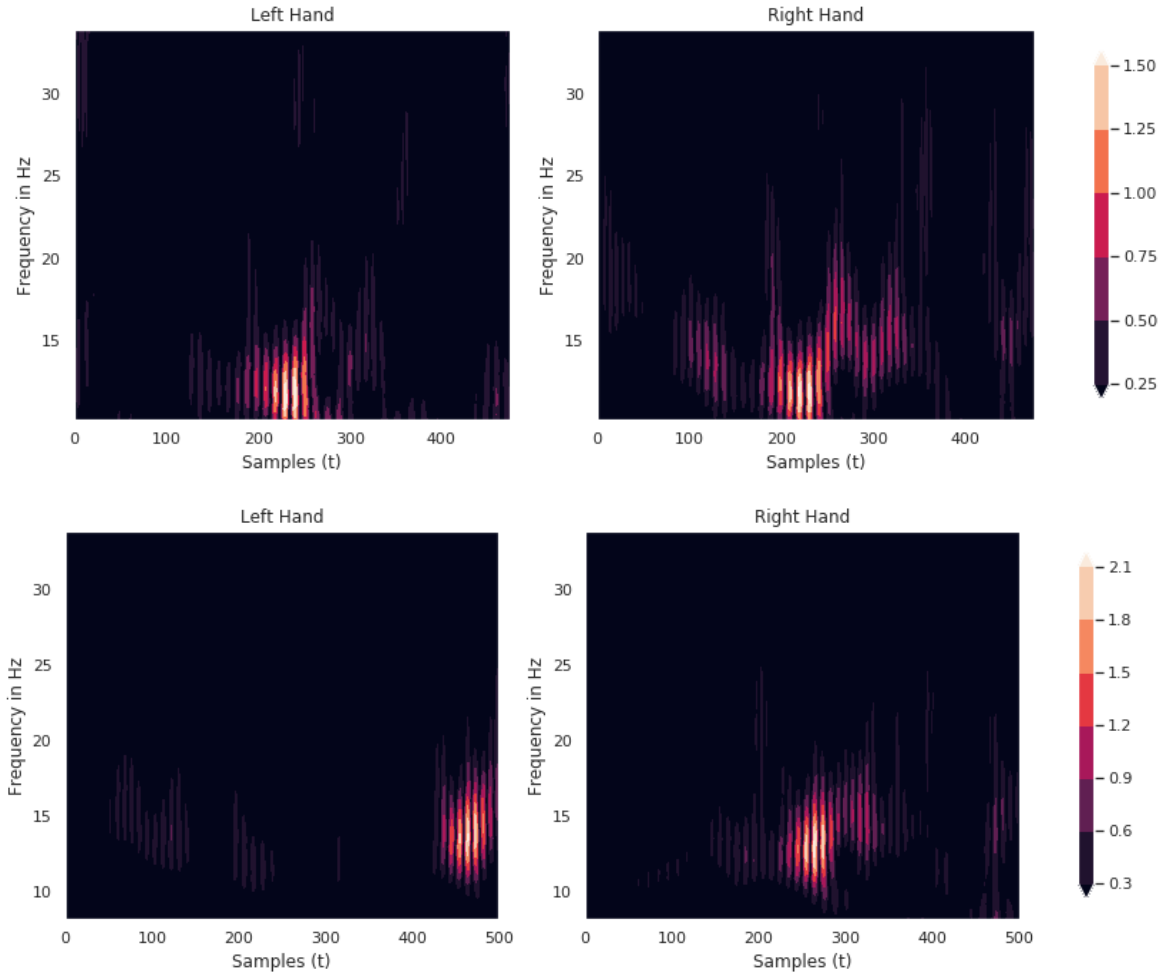


Figure 4.7: Scalogram after applying guided CAM. Applied on the spatiotemporal layer for participant 8. First row is the Shallownet, second row is the EEGNet, third row Shallownet_CWT and fourth row is EEGNet-CWT.


Left Hand
Right Hand
Frequency in Hz
Samples (t)
30
25
20
15
0
100
200
300
400
1.50
1.25
1.00
0.75
0.50
0.25
Left Hand
Right Hand
Frequency in Hz
Samples (t)
30
25
20
15
10
0
100
200
300
400
500
2.1
1.8
1.5
1.2
0.9
0.6
0.3

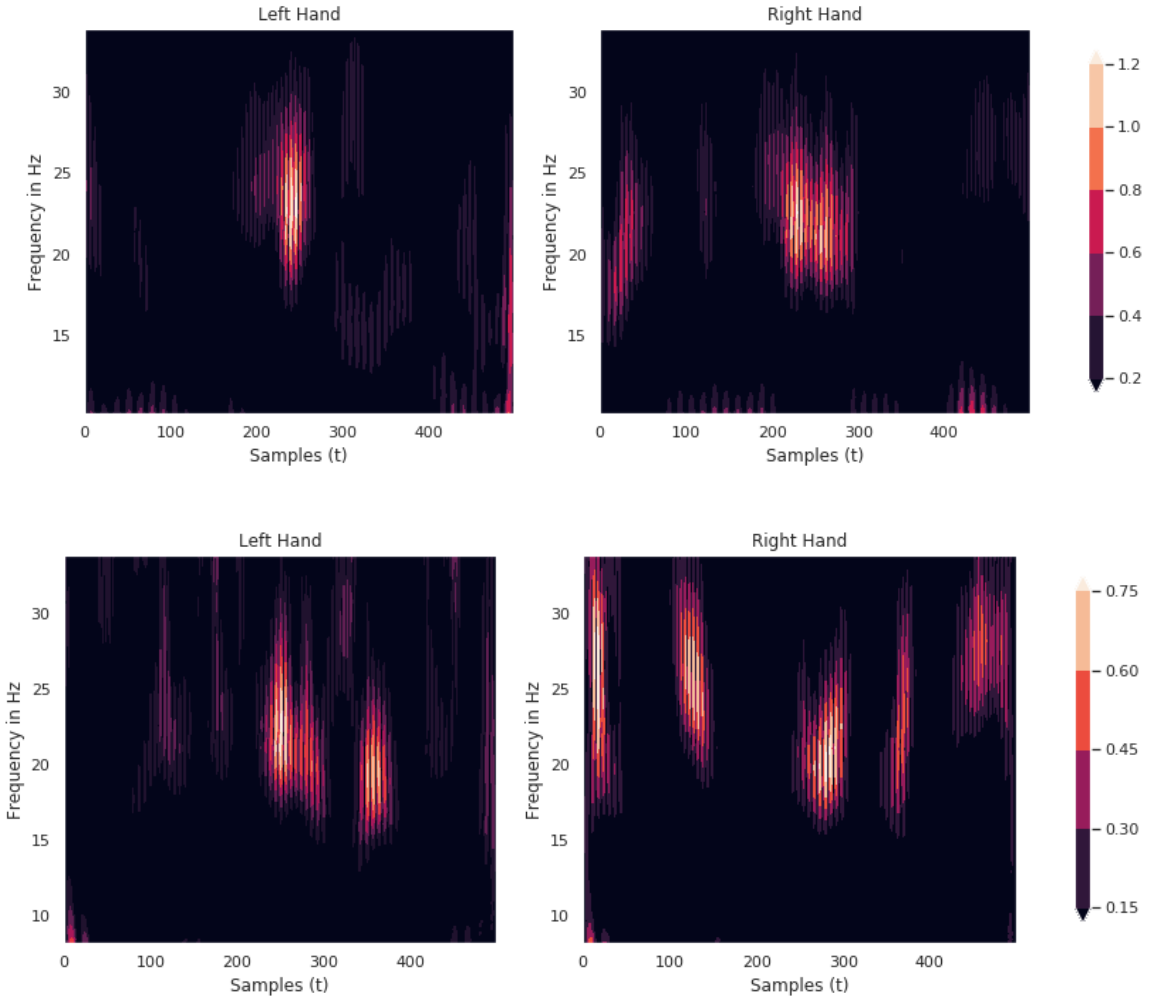
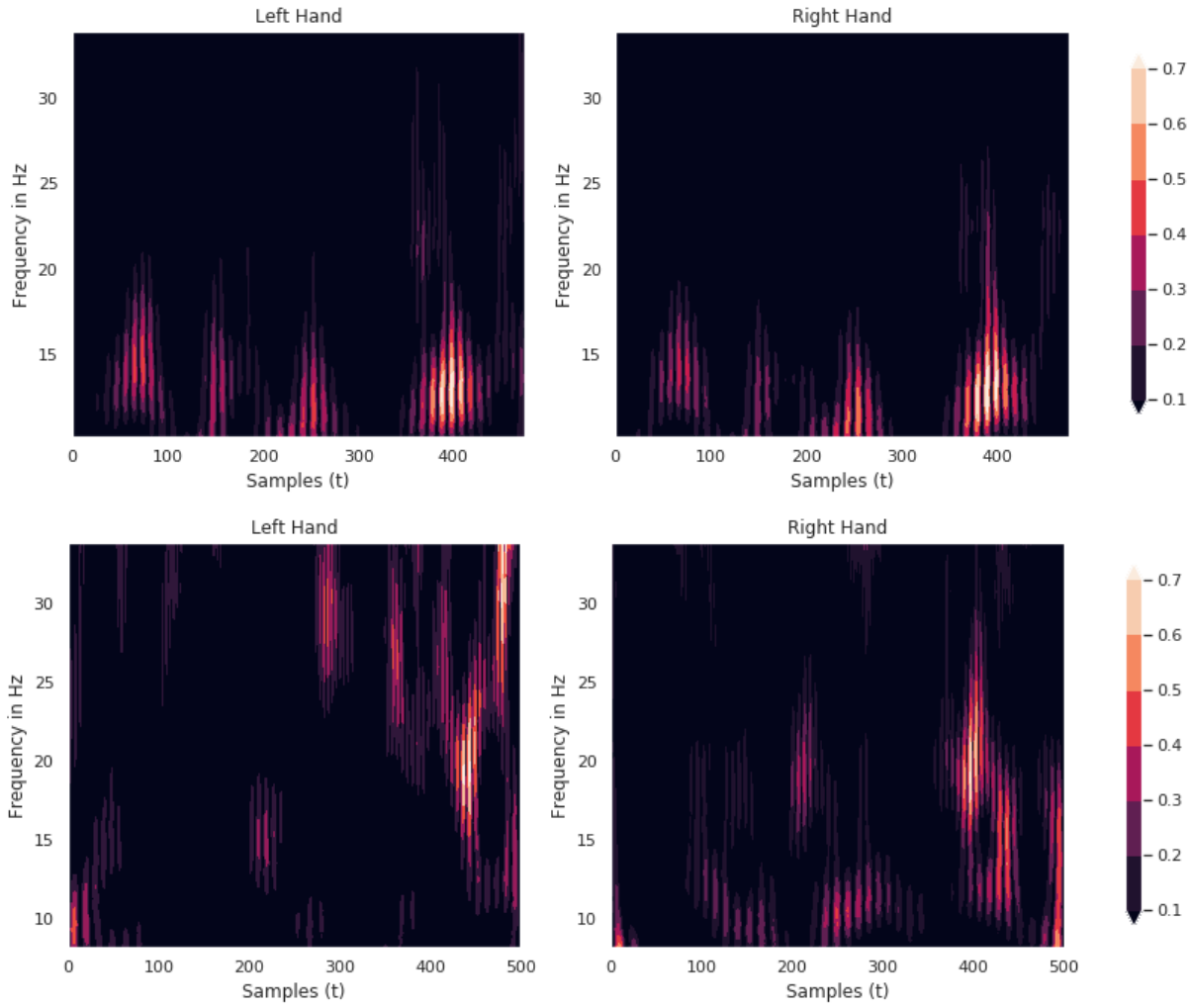


Figure 4.8: Scalogram after applying guided CAM. Applied on the spatiotemporal layer for participant 7. First row is the Shallownet, second row is the EEGNet, third row Shallownet_CWT and fourth row is EEGNet-CWT.


Left Hand
Right Hand
Frequency in Hz
Samples (t)
0
100
200
300
400
500
10
15
20
25
30
0.1
0.2
0.3
0.4
0.5
0.6
0.7

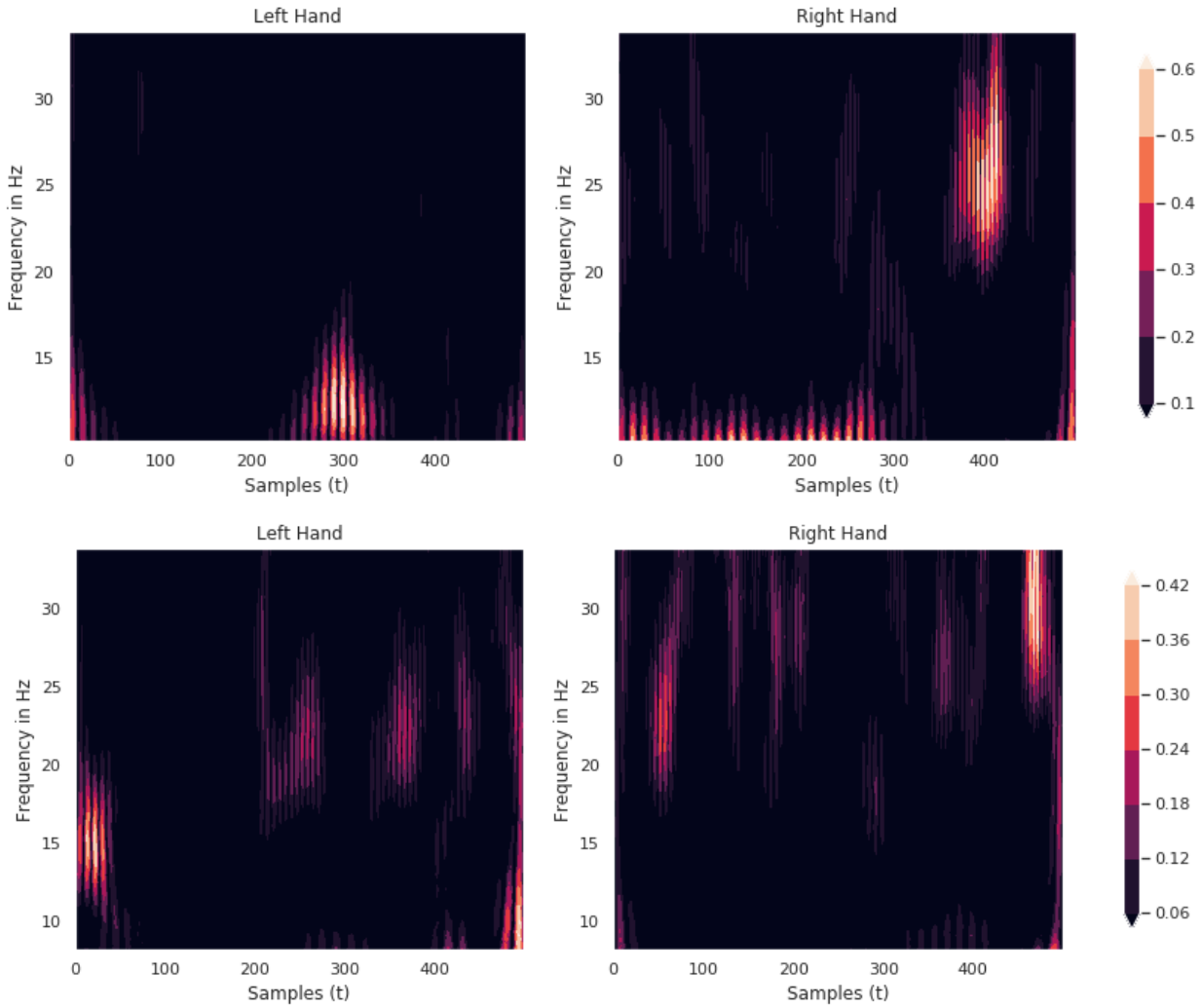


Figure 4.9: Scalogram after applying guided CAM. Applied on the spatiotemporal layer for participant 1. First row is the Shallownet, second row is the EEGNet, third row Shallownet_CWT and fourth row is EEGNet-CWT.

4.3.4 The effect of training using shuffled augmentation for BCI IV 2a.

In Shallow net and EEGNet for BCI IV 2b, the mean accuracy and kappa differences were almost the exact same. On the other hand, for BCI IV 2a, there was a slight improvement of 2% for EEGNet. However, the learned filters of the networks showed higher saturation, less scattered and better time localisation in the prominent features corresponding to each network as shown below.

Table 4.5: A summary of classification accuracies for BCI IV 2a dataset.

BCI IV 2a	Original	Augmented
SHALLOWFBCSP	0.70	0.70
EEGNet	0.63	0.65

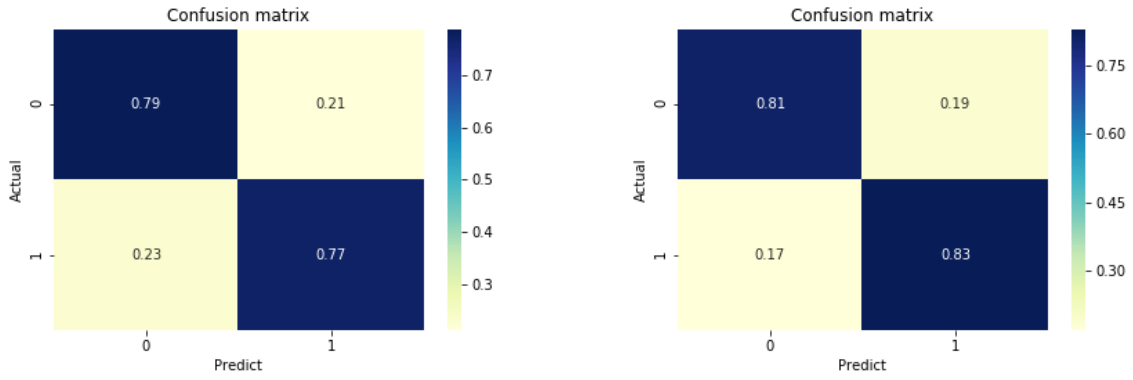


Figure 4.10: Normalised confusion matrix of Shallow net and EEGNet using augmented training.

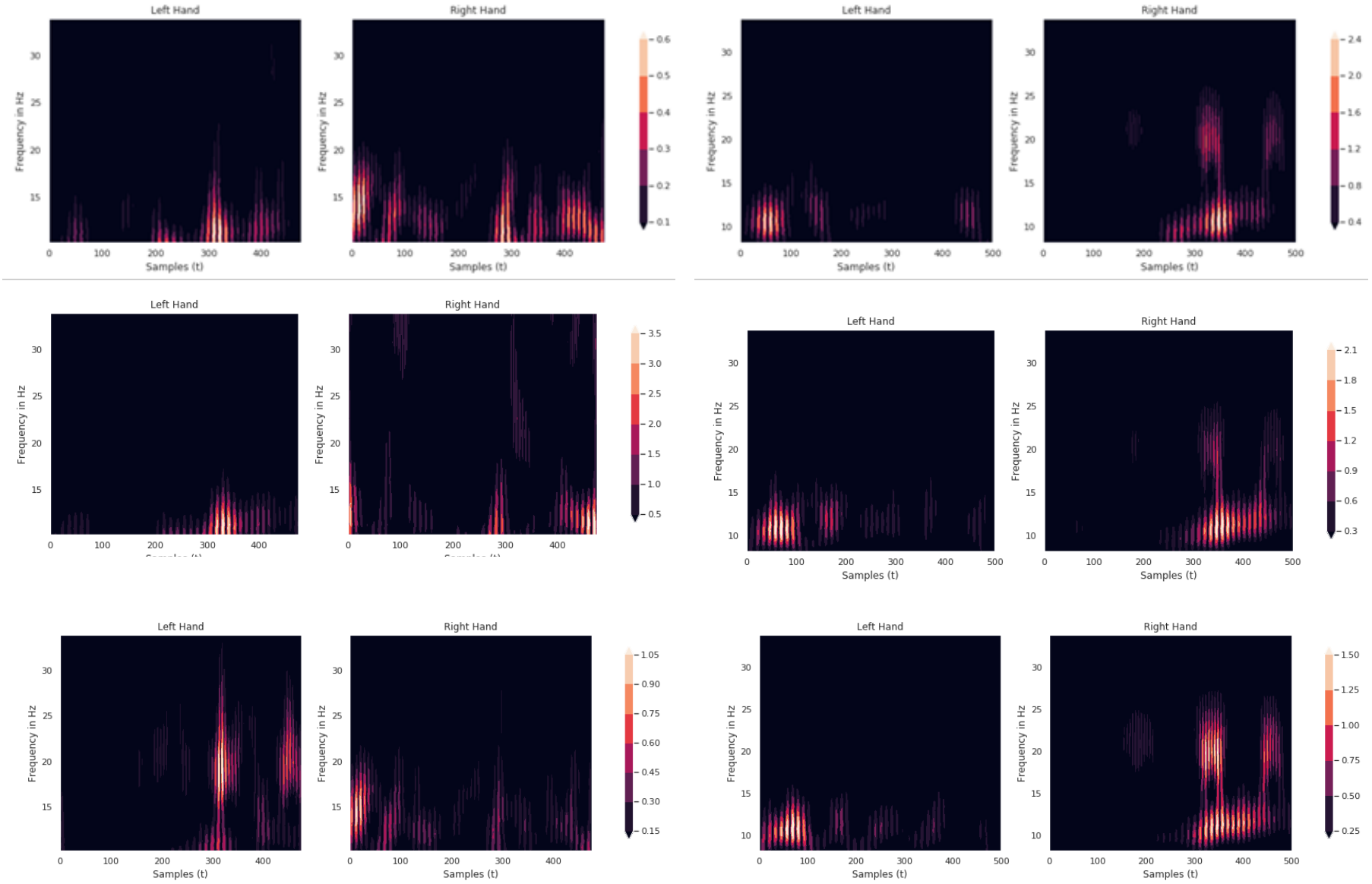


Figure 4.11: Participant 4. Layer (spatiotemporal). First row the scalogram obtained with no augmentation, the second row is the combined with shuffling between trials and shuffling within trials and the third row is shuffling between trials only.

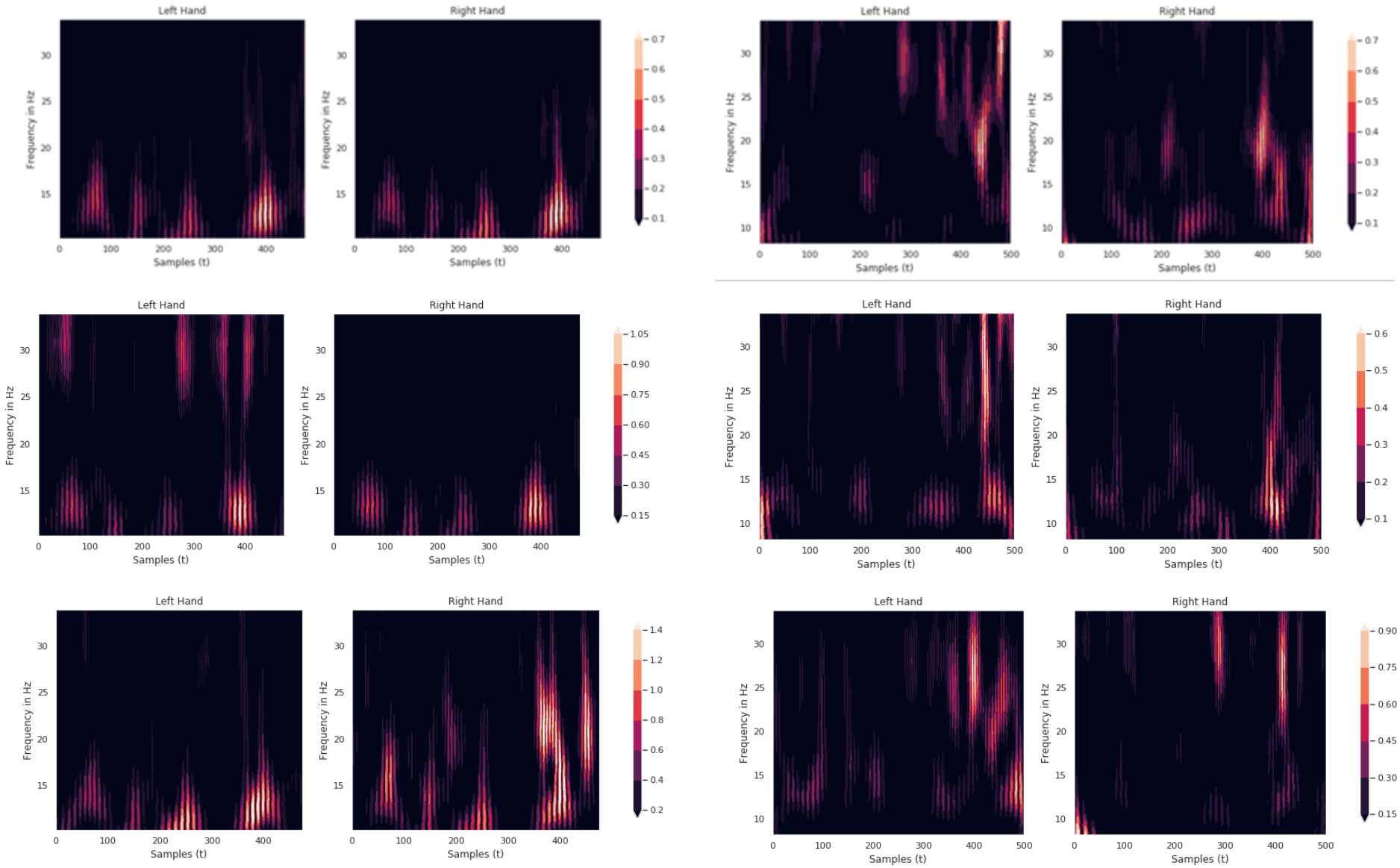


Figure 4.12: Participant 1. Layer (spatiotemporal). First row the scalogram obtained with no augmentation, the second row is the combined with shuffling between trials and shuffling within trials and the third row is shuffling between trials only.

4.4 Discussion

For the baseline performance, the replicated methods achieved similar accuracies with the reported accuracies of the original papers for both the Shallow net and EEGNet. The decline in performance (replicated method decline) could be a result of the testing method, i.e. it wasn't clear if the reported testing accuracy was on the unseen evaluation sets provided by the competition or if it was a held out set after combining both the training sets and the unseen evaluation set. In addition, it is not clear if it's the reported testing accuracy during the training epochs or was evaluated after the training phase in a cross-fold.

Directly providing the CWT to the existing architecture lead to a decrease of 6% and 8% for Shallow net and EEGNet, respectively. The decrease in the classification accuracy could be attributed to the fact that the first convolutional layer of the two networks performs a transformation on the signal that is equivalent to finding the band-power at different frequencies and by providing the CWT transformed signals to the network another transformation is performed decreasing the resolution of the signal and reducing the accuracy. To test this, the first convolutional layer in the two architectures has been removed, providing the CWT channels as the band-power filters instead of the learned filters. An increase of 2% and 4% in the classification accuracy for Shallow net-CWT and EEGNet-CWT is observed, noting that it is still lower than the original (Shallow net and EEGNet using raw EEG signals) architectures by 4% and 5%. The improvement in performance can be attributed to the same reason, where in this case the CWT transformed signals didn't lose as much resolution by not applying a transformation at the first layer, and finding the relationship between the channels with higher resolution at the second level. However, the performance compared with the original architecture suggests that there is a loss in resolution by fundamentally applying a CWT transformation to the original signals.

Furthermore, looking at the confusion matrix for the baselines (Fig. 3.3), the majority of the votes contributing to the accuracy are at class 1 (right hand) for three of the 4 architectures (Shallow net, EEGNet, EEGNet_CWT), same behaviour is witnessed for the preceding architectures Fig. 3.4 and Fig. 3.5. The difference between classes contribution isn't statistically significant, and no further analysis was performed.

Furthermore, looking at the confusion matrix for the baselines (Fig. 3.3), the majority of the votes contributing to the accuracy are at class 1 (right hand) for three of the 4 architectures (Shallow net, EEGNet, EEGNet_CWT), same behaviour is witnessed for the preceding architectures Fig. 3.4 and Fig. 3.5. The difference between classes contribution isn't statistically significant, and no further analysis was performed.

The suggested architecture PWCN scored a mean of 80% for dataset BCI IV 2b, where the architecture is based on CWT features, the performance is 5% and 4% higher than Shallow net-CWT and EEGNet-CWT, 2% higher than the original Shallow net and 2% lower than the EEGNet. However, evaluating with BCI IV 2a dataset, a significant decrease in performance was observed, reaching almost chance levels. The decline in performance can be partly attributed to the fact that the number of channels in the second dataset is 22, as opposed to two channels. This possibility can be tested by dropping the channels that are further from the sensorimotor cortex. For this experiment, the time and the computing capabilities didn't allow further investigation. Furthermore, given the poor performance on the BCIIV2a dataset, the analysis of the architecture was not pursued any further.

Investigating the prominent features using the guided Grad-CAM, in the context of this experiment, the most prominent features are the representation of the most contributing samples of the signal. A couple of observed behaviours are noteworthy. Firstly, the Shallow net and EEGNet seems to attribute higher weights to different samples across almost all participants, translating to different periods of the signal, for example for participant 4, the Shallow net seems to find the prominent features between the samples 200 to 400 for the left hand MI, while EEGNet seems to find the prominent features between 0 to 200 for the left hand MI (Figs. 4.6 to 4.9). Also worth mentioning is that, for the same participant, the prominent features between the two classes are also found in different periods. Secondly, the CWT based architectures seem to find the prominent features at different frequencies, looking at the most contributing filters shows faster frequencies. The findings might suggest that the two networks are learning different transformation functions, which could be due to the kernel size and the depth of the network. Combining that with the evaluation of both networks on BCI IV 2a where Shallow net performs better than EEGNet (Table 4.4) by 7%, which is also the same findings in Lawhern et al. (2018), and looking at the results of the individual participants, each of the networks seems to perform better or similarly for a group of participants and not

the rest. Implying that the transformation functions learned are different and more suited for some participants over others.

The authors of Shallow net, EEGNet, and most of the recent state-of-art methods relying on deep learning for EEG classification have adopted the cropped training as means of data augmentation. In this study, the new augmentation technique suggested *shuffled-crossover crops*, had no significant effect on performance, with only a small increase (of around 2%) witnessed using an EEGNet. The visualisation of the prominent features showed more saturated weights over the prominent features found with less scattered weights as opposed to training with the original data with no augmentation for both Shallow net and EEGNet. Noting that in Shallow net the authors found no significance improvement in using cropped training over the original dataset in the shallow model, it was found to increase performance in their suggested deep model. In this study, only the Shallow model has been investigated and EEGNet is two levels deeper than Shallow net that might suggest that the augmentation technique would lead to improvements for deeper models, which will be investigated in the next chapter (Figs. 4.11 and 4.12).

4.5 Summary of Contributions:

- A novel architecture was introduced using CWT that improved the performance over using the naïve methods discussed in Chapter 3;
- A novel data augmentation technique is introduced to address the small number of samples of the EEG datasets discussed in Chapter 1;
- Modified existing state-of-art models to operate of time-frequency features.
- Compared the performance of existing state-of-art models using raw EEG signals with the modified models using time-frequency features.

4.6. Limitations

- Limited to the number of scales evaluated due to limited computational power;
- Only tested with CWT (could have tested with other methods like spectrograms);
- Limited testing on BCI IV 2a dataset due to limited computational power.

Chapter 5:

A convolutional recurrent neural network with double attention using guided Grad-CAM for interpretability

5.1 Introduction

The use of Deep Neural Networks in the domain of electroencephalography (EEG) has shown great promise for EEG classification tasks. Some of the best performing models such as the EEG net, Shallow net and Deep net (as discussed in Chapter 4) are employed as an end-to-end solution with minimum or no pre-processing in comparison with the traditional methods such Common Spatial Pattern (CSP), Filter Bank Common Spatial Patterns (FBCSP) and Continuous Wavelet transformation (CWT) which entail a number of steps such as spatial and temporal filtering and removing artefacts (discussed in more detail the literature review, Chapter 1 and 3) aiming to increase the Signal-to-noise ratio (SNR). On the other hand, the traditional methods are based on linear methods that provides a higher level of interpretation of the decisions of the employed models. For example, CSP results in a neurophysiological interpretation of the brain spatial correlations and CWT and FFT are able to provide the spectral interpretation of such events, a combination of these methods are able to provide a sophisticated analysis of the EEG signals and provide researchers with an insight of brain activity associated with an event. Finally, in Chapter 4, the suggested model using CWT with data augmentation achieved a slight improvement in the classification performance on dataset BCI IV 2b (two classes) and slightly higher than random classification accuracy on the BVI IV 2a (four classes). Furthermore, using CWT features didn't show a significant improvement over raw EEG signals using CNN based models.

Deep Neural Networks interpretability is an emerging field and researchers are now able to acquire interpretations of the network decisions through different techniques. Two techniques in particular: Gradient-weighted Class Activation Mapping (grad-CAM) (Selvaraju et al., 2017) and Attention mechanisms which have proved to be very useful and showed great results in Machine Vision and Natural language processing (Bahdanau, Cho, & Bengio, 2015; Vaswani et al., 2017). At the time of writing, the grad-CAM has not been used in the interpretation or analysis of deep learning in the domain of motor-imagery and has only been used in few papers in the EGG domain using wavelet transformation features (Andreotti, Phan, & De Vos, 2018). On the other hand, only a handful of studies employed attention mechanisms in the motor imagery classification domain. In particular, the study conducted by Zhang, Yao, Chen, & Monaghan, (2019) in models addressing temporal data and employing attention

mechanisms achieved relatively high accuracies compared with state-of-art methods where the attention mechanism was added on the so like temporal features.

The authors in Lawhern et al., (2018) used three methods to explain the features with the highest contribution to the model decision and add a level of interpretability to their suggested model. Firstly, they visualized the activation maps as the averaged activations at the second level of their model arguing that since they are using separable convolution layers, visualizing the activation maps for different channels (depths) would provide the spatial correlation for the different narrow bands assuming that each filter is learning a narrow band frequency. Even though the authors showed consistent outcomes with the literature, and claimed a newly found theta-beta relationship using their suggested model. It is hard to argue that the resulting transformation at a layer is still a time-domain representation of the signal and applying CWT to the resulting activations could be misleading. The finding discussed in Chapter 4 where CWT is applied to the activations obtained with grad-CAM for the different architectures were inconsistent such that the different network architectures were not found to be focusing on the same samples (features) of the signal while the authors of the discussed architectures argued that they can infer a spectral analysis from their network activations at the top layers. However, scalograms generated in Chapter 4 don't support these claims. In other words, the networks don't completely agree on a set of samples that are maximising the performance. Secondly, the investigated the weights of the learned filters utilising the fact that separable convolution has direct mapping to the output channels. Visualising the weights would provide a low-resolution approximation of which features are more important than others due to the nature of the EEG signals where the events are not time-locked but phase-locked.

In the previous chapters, the main focus has been on temporal aspect of the EEG signal where in Chapter 3, the performance of the basic traditional classifiers with different windows and starting points has been studied to identify the best combination and obtain a baseline. In Chapter 3, some of the state-of-art neural network models have been further investigated and explored with few adjustments and manipulation such as using CWT features as the input to the networks instead of raw EEG signals. A basic form of grad-CAM has also been used with CWT to identify the most prominent features and analyse the decision of the networks.

Therefore, in this chapter, it is hypothesized that RNN exploiting temporal signal characteristics will improve the performance.
In this chapter, the aims are to:

1. Utilise the methods to extract and reduce the intervals used without comprising the performance in the classification task.
2. Obtain a higher resolution interpretation of the features.

To achieve the aims of this study, the following objectives are investigated:

1. To establish whether the outcome of the model can be fully interpretable in terms of spatial, temporal and spectral analysis.
2. Identify if adding a spatial attention mechanism will provide insights on the spatial correlation.
3. Identify if a temporal attention mechanism with provide better analysis for the signal intervals with higher contribution to the performance.
4. Establish whether a grad-CAM can be utilised for extracting and interpreting the prominent EEG features.
5. Identify what is the shortest signal period that be used while maintaining the state-of-art classification accuracy.
6. Establish whether data augmentation will improve the performance of the models.

And the following methods have been used to achieve the objectives:

1. An attention layer is fit at the top of the model to learn linear correlations between the channels aiming to obtain an easily interpreted spatial maps of the electrodes and highlight the most prominent samples;
2. A RNN layer using Gate Recurrent Units (GRU) to address the temporal nature of the signals;
3. An attention mechanism that learns a weighted attention of the prominent samples in time;
4. Different windows and slices are used to extract the shortest period while maintaining the state-of-art classification;
5. Utilising the grad-CAM for EEG interpretation.

The augmentation technique discussed in Chapter 4 is used in the training.

5.2 Methods

5.2.1 Dataset

The BCI Competition IV 2a was used for this experiment. The datasets are discussed in detail in section 2.1.

5.2.2 Apparatus

A NVIDIA 650 G-force GPU card was used for the calculations of the model parameters (Gradient descent and model updating), with CUDA 8 and Pytorch (Paszke et al., 2017) for the implementation of the proposed models and the Neural Networks based models evaluated (EEGNet, Shallow net, ConvLSTM) on a Linux based machine with 4 Quad-cores.

5.2.3 GRU Double Attention Conv-RNN

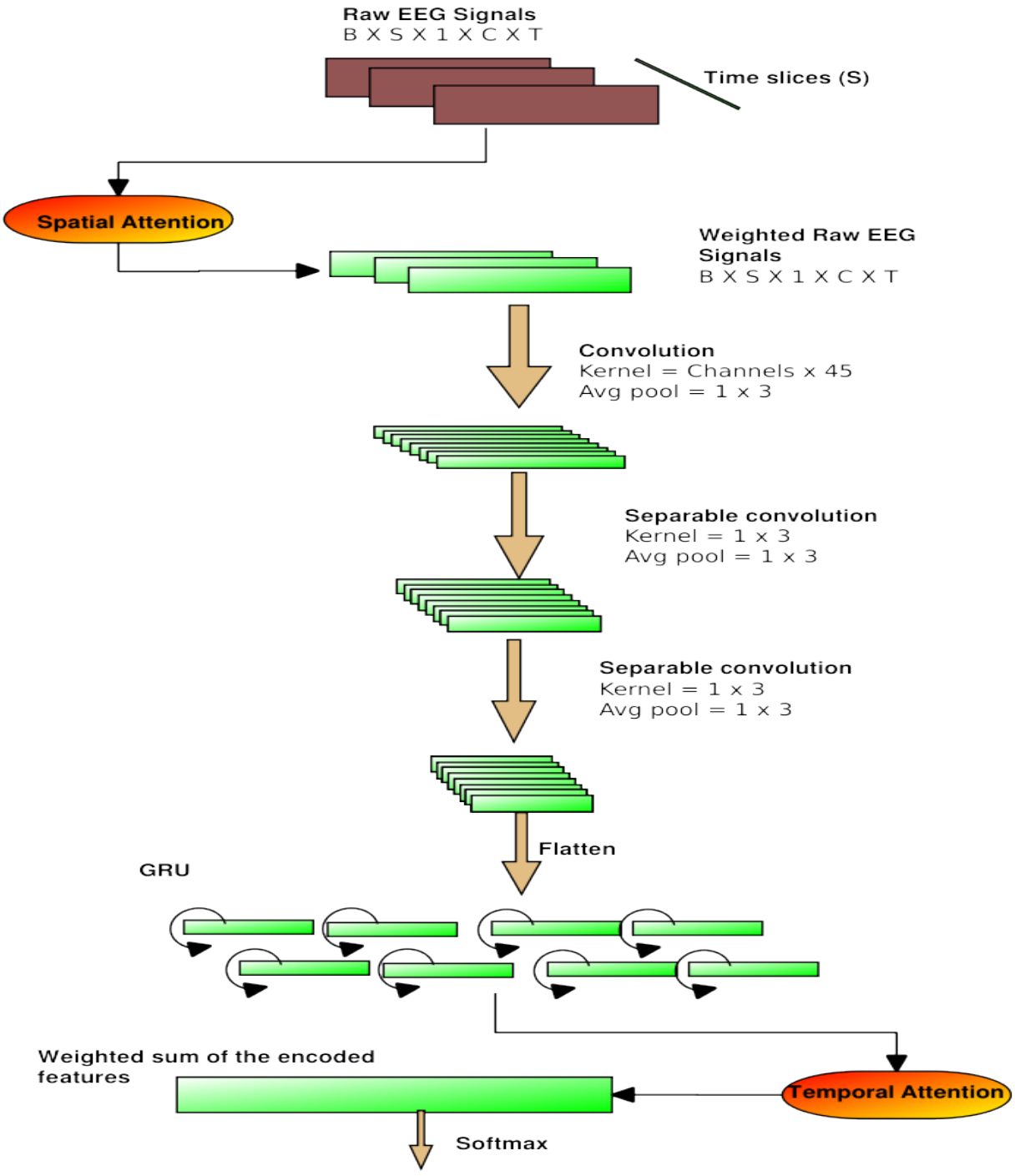


Figure 5.1**:** The proposed architecture. The dark brown boxes represent the raw EEG signals. The orange boxes show the local attention mechanisms (spatial attention and temporal attention). The green boxes represent neural network layers.

5.2.3.1 Spatial-Attention

The spatial attention in this experiment is used to refer to attention on the electrodes. The purpose of the attention mechanisms in the proposed model is to improve the interpretability of the model by obtaining attention vectors signifying the electrodes contribution to the classification. Two attention mechanisms have been implemented operating as self-attention (references) on local features.

The first attention mechanism was inspired from the Squeeze and Excitation used in Hu (2018) where the Squeeze and Excitation operate on the filters learned by the model emphasizing the filters that maximize the performance. In the proposed attention mechanism, the operation is applied on the each of the electrodes E^i representing the rows of the matrix (the height). The time samples T^i representing the columns (width) of the matrix are averaged to obtain a global representation for each electrode E^i (as shown in Eq. 5.1).

$$z^i = \frac{1}{T} \sum_j T_j^i \qquad \text{Eq. 5.1}$$

The number of electrodes is further reduced by a rate r, and non-linear ReLU activation function is applied to select the channels with higher contribution such that:

$$\mathrm{Red}(z^i) = \mathrm{ReLU}(z^i w_1 + b)$$
$$ReLU(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \qquad \text{Eq. 5.2}$$

where w_1 are the weights and b is the bias term, w_1 is a matrix of order $\mathbb{R}^{\mathrm{E}\,\mathrm{x}\,\frac{\mathrm{E}}{r}}$. Last but not least, the weights of the attention is then calculated as:

$$\text{attn} = sigmoid\left(\text{Red}(z^i)\,\text{w}_2 + \text{b}_2\right)$$
$$sigmoid(z) = \frac{1}{1 + e^{-z}} \qquad \text{Eq. 5.3}$$

where w_2 is a matrix of order $\mathbb{R}^{\frac{E}{r}\text{ x }E}$ to remap the E to its original size and the sigmoid activation function to learn the non-linear correlations. Finally, the learned Attention vector is point-wise multiplied by the input x

$$\bar{X} = \text{Attn} \,.\, \text{x} \qquad \text{Eq. 5.4}$$

As shown in Fig. 5.1.

1. The second attention mechanism was inspired from Zubarev, Zetter, Halme, & Parkkonen, (2019) where a linear spatial distribution is assumed between the electrodes. In Zubarev, Zetter, Halme, & Parkkonen, (2019), the matrix is reduced across the electrodes dimension which is similar to convolving across the electrodes E for each time sample T. The authors were then able to interpret the spatial distribution by applying CSP to the learned weights after the training and manually choose the best filters explaining the distribution. However, following this procedure is not highly reliable since the features minimizing the error could be a combination of the learned filters, even though one filter might have the best spatial distribution, it is still one approximation that doesn't provide the information of those combinations that improved the performance. Accordingly, the same method was used for learning only one set of weights (one filter) and those set of weights are then used as an attention vector which is multiplied by the input channels as opposed to dot product reduction then fed to the convolutional layer.

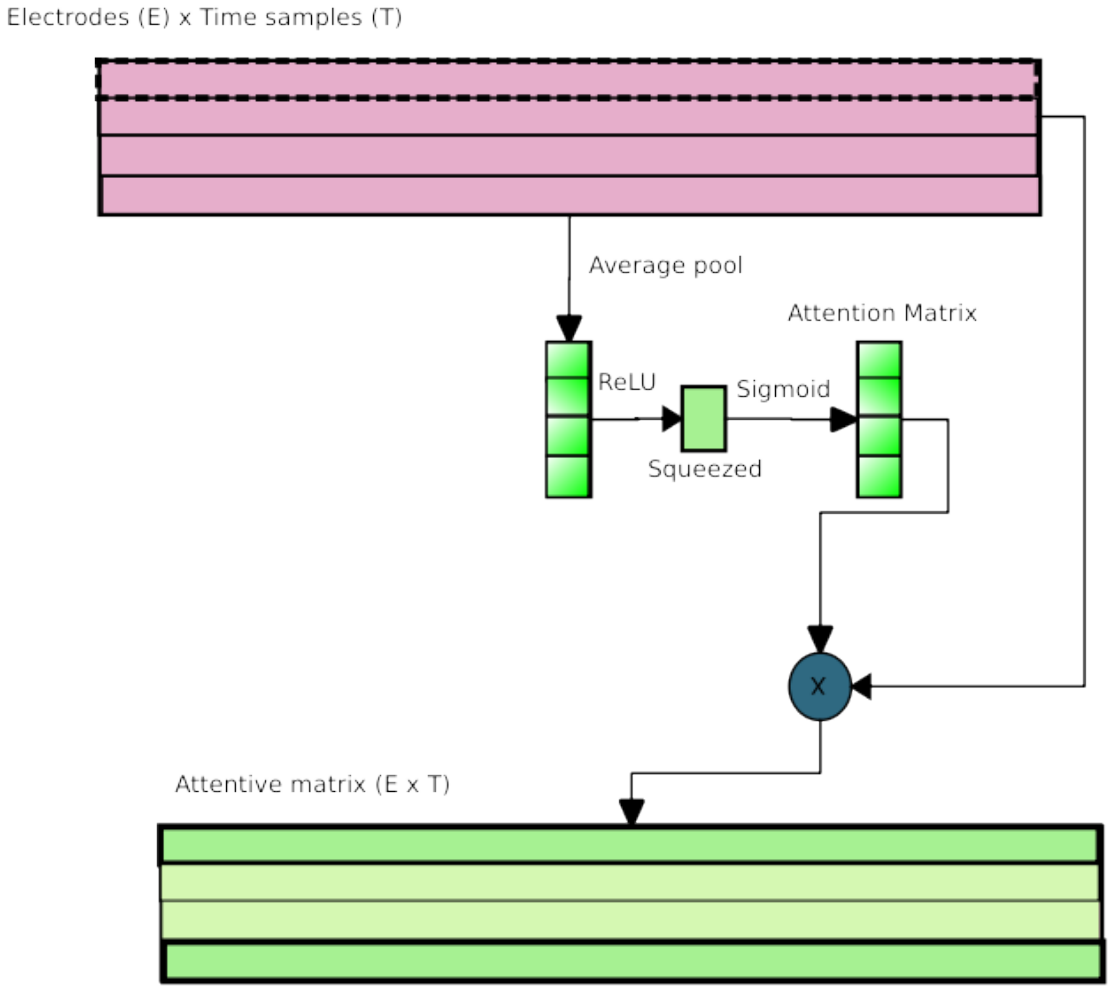


Figure 5.2: Illustration of the global attention mechanism. An average pool is applied on each row vector representing an electrode. A non-linear activation function (ReLU) is applied for a squeezed representation. Followed by an excitation (expanding to the original size) and a Sigmoid to transform the values to a probability between 0 and 1 and obtaining the attention matrix. Finally, pointwise multiplication of the attention matrix with identity matrix generating the attentive matrix.

5.2.3.2 Conv Blocks

Block 1_Layer 1: This layer was not used in the baseline measurement. The first Convolutional layer is a normal convolutional layer operating on the time courses T. The purpose is to add complexity to the signal to enhance the performance of the spatial attention layer described above.

Block_1_Layer 2: The layer uses normal convolution operating on the Electrodes. The purpose of this layer is to find the spatial correlation between the electrodes and add more complexity to the signal. The kernel size is equal to $E \times 1$ where E is the number of electrodes.

Block_1: A batch normalisation is used in both layers after the convolutional layers where the momentum was set to 0.993. A Leaky-ReLU is used as the non-linear activation function for both Blocks, the Leaky-ReLU is a modified function of the ReLU that handles the negative values such that Leaky-$ReLU(z) = \begin{cases} z, & z > 0 \\ \alpha z, & z \leq 0 \end{cases}$, where α is a constant gradient and was left with default value (0.001).

Finally, an average pool with kernel size (1×3) and stride size (1×3) after the second layer to reduce the feature dimensionality. The Spatial Attention mechanism lies between the block_1_layer 1 and block_1_layer 2 as shown in Fig. 4.1.

Block_2: Blocks of the second type were added to deepen the model. However, in these mid blocks, separable convolutional layers have been used since they reduce the number of trainable parameters and showed great performance in the state-of-art models such as EEGNet as discussed in Chapter 4. Two blocks are added with Batch Normalisation and Leaky-ReLU. In addition, to an average pool with the same size of kernels (1×3) and stride (1×3) and finally, a dropout layer in the two blocks with $p = 0.5$ where p is the dropout probability.

5.2.3.3 RNN GRU Block

The vanilla RNN as widely known and accept face a vanishing gradient problem. Where Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRU) have been developed to address these problems. In this research the latter was used, since it showed promising results and was easier to implement since it has one less gate. The literature showed similar results using GRU and LSTM (Cai, Wei, Tang, Xue, & Chang, 2018; Ma et al., 2018; Wang et al., 2018).

Two stacked GRU layers are used after the Conv Blocks, operating on time slices such that the recurrent operation is applied between the seconds on the encoded features obtained from the Conv blocks. The output of the GRU layers contains the output of each time slice and the final hidden state. To have a better interpretability of the model and increase the accuracy, another attention mechanism was implemented.

5.2.3.4 Temporal Attention

Since the participant is supposed to maintain the motor imagery movement for about 4 seconds over many trials, the concentration of the participant varies across each trial. Hence, for each trial in the imagining period a relaxing period or a period where the participant loses focus is going to exist. Potentially having a feedback loop notifying the participant that they are losing focus or discarding bad trials according to the learned attention vector, could lead to the acquisition of better training data sets. To identify the relevant periods and reduce the contribution of the relaxed periods, two temporal attention mechanisms were implemented to address this problem.

The first temporal attention mechanism was the mechanism presented by Zhang et al. (2019). The encoded output of the recurrent layer, in this case the GRU layer is processed by the attention network where each slice transformed into a latent space in a non-linear manner, and an attention vector is obtained reflecting the importance of each slice. The softmax activation function (Eq. 5.1) is applied to the attention vector to constrain the weight values sum to be equal to 1. Furthermore, a weighted sum is applied to provide one slice representing all the time slices. Noting that the attention network has about 17,000 learnable parameters in the configuration of 64 hidden units and 256 neurons of the attention network.

The second attention mechanism is also based on the Squeeze and Excitation introduced in Hu, (2018) and described above (Eq. 5.1 to Eq. 5.4). However, the operation is opposite, it can be better described as Excitation and Squeeze with about 800 learnable parameters as opposed to the 17,000 learnable parameters in the first mechanism. Given that the latent features $F \in \mathbb{R}^{PH}$ where P are the number of slices and H are the hidden latent representation (encoded features). An average pooling is applied to obtain a global representation for each P^i such that:

$$z^i = \frac{1}{H} \sum_{j}^{H} F_j^i \quad \text{Eq. 5.5}$$

where z^i describes the aggregated information of every time slice P^i. To learn the non-linear importance of the time slices, z^i is further mapped to a latent space and a non-linear ReLU activation function is applied as shown in Eq. 5.6.

$$\acute{z}^i = ReLU(z^i w_1 + b_1) \quad \text{Eq. 5.6}$$

Where $w_1 and\ b_1$ are the weights and bias term respectively. Moreover, To acquire an attention vector for each of the time-slices and learn the non-linear dependencies between the time slices, the latent features $\acute{z}^i$ is mapped back with a sigmoid activation function:

$$a^i = Sigmoid\left(\acute{z}^i w_2 + b_2\right), \in \mathbb{R}^P \quad \text{Eq. 5.7}$$

Where $w_2\ and\ \ b_2$ are the weights and bias term respectively. The vector a^i is considered to be the attention vector and is then time slice-wise multiplied by the vector H^i to emphasise the importance of each time slice obtaining the rescaled hidden features $\bar{F}^i$as shown in Eq. 5.8.

$$\bar{F}^i = a^i . F^i \quad \text{Eq. 5.8}$$

5.2.3.5 Fully connected layer

Finally, a fully connected layer with the input neurons L connected to C output Neurons where is the number of classes. A softmax activation function (Eq. 5.9) is applied at the output neurons for the classification.

$$softmax(y_i) = \frac{e^{y_i}}{\sum e^{y_i}} \quad \text{Eq. 5.9}$$

5.2.3.6 Training/testing and validation configuration (loss, optimisation and number of epochs):

The cross-entropy function was used to calculate the loss of the model. In the training process, the model tries to minimise that loss and the model is penalised when the probability of the predicted class is diverged from the actual class. The binary Cross-entropy is defined in Eq. 5.10.

$$-ylog(p) + (1 - y)\log(1 - p) \quad \text{Eq. 5.10}$$

Where y is the correct label (in the binary either 0 or 1) and p is the predicted probability. For multi-class, the loss is calculated for each class c separately and the losses are summed to obtain a single loss value (Eq. 5.11).

$$-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c}) \quad \text{Eq. 5.11}$$

Adam (Kingman and Ba, 2014), for the optimisation and updating of the weights and a weight decay to employ L2 regularisation. A total of 800 epochs exist for each run. The reported accuracies are the mean of five runs similar to a five-cross-fold validation where the training samples are non-overlapped randomly selected for each run while the testing set is completely unseen in the training and the same test set is used for all the runs.

5.2.3.6 Guided Grad-CAM

One of the earliest methods for visualising what convolutional networks are learning in the work, presented by Zeiler & Fergus (2014), focused on de-convolving the CNN by reversing the flow from the activations back to the input image. The approach consisted of a number of

additions to the network itself and another set of instructions for the visualisation task such as setting activations to zero while others to non-zero, and the most discriminative pixels are in turn those who were highly activated by the non-zero neurons. Extending that work to improve the interpretability of CNNS, Zhou, Khosla, Lapedriza, Oliva, & Torralba, 2016 suggested the Class Activation Mapping (CAM) to incorporate the fully connected layers (where the network usually employ the class classification) to describe the relative importance between the most discriminative pixels and a specific class. The CAM method replaced the flattening of the feature maps at the higher-level of the network with a global averaging pool and used them as features for a fully connected layer enabling localised information to be mapped back to the input pixels (hence a better localised visualisation of the most prominent pixels). However, the suggested model was still changing the architecture of the network and trading off complexity for interpretability. The suggested model in this study has a RNN layer before the softmax as discussed above which in turn leads to the infeasibility of using CAM approach for the interpretation of the model.

Fortunately, the work presented by Selvaraju et al. (2017) combined the two aforementioned approaches to generated guided Gradient Class Activation Maps (guided Grad-CAM) providing a generalisation of CAM that is feasible for any CNN-based architecture. Although, these methods were mainly designed for visualising images and were optimised to obtain fine-grained detail (high resolution) visual explanation, while with raw EEG signals, a high-resolution image won't be interpretable by visual inspection and not all the rules apply. In addition, in this study, the focus is two-fold, one is approximately localising the signal intervals that are highly contributing to the classification of a specific class where it can be used as a feedback loop when the model is being trained and two identifying the spatial correlation (the relationship between the electrodes) for research purposes in aim to minimize the number of electrodes used in real life applications. Hence, a few adjustments are introduced.

In the CAM approach, let $A^k \in \mathbb{R}^{uv}$ be the feature maps where u is the width and v, as described earlier, a Global Average Pooling (GAP) is applied to A^k, followed by a linear transformation by w_k^c to obtain the score y^c where c is the class as in Eq. 5.12.

$$y^c = \sum_k w_k^c \frac{1}{Z} \sum_i \sum_j A_{ij}^k \quad \text{Eq. 5.12}$$

$$y^c = \sum_k w_k^c \frac{1}{Z} \sum_i A_i^k \quad \text{Eq. 5.13}$$

To keep the electrodes information which corresponds to the height will refer to it as v to maintain the notation consistency, instead of using a GAP, an average pool is applied on the width u and $A^k \in \mathbb{R}^v$ such that the CAM is now expressed as the following:

To then make the CAM weighted by the gradients instead of the final layer learned weights as described in the main study, the gradient of y^c is calculated with respected to the feature maps A_i^k, as shown in Eq. 5.14.

$$\alpha_k^c = \frac{1}{Z} \sum_i \frac{\partial y^c}{\partial A_i^k} \quad \text{Eq. 5.14}$$

Where the partial linearisation α_k^c is obtained by average pooling the gradients acquired through back-propagating from the feature maps of the target class c. Finally, in the case of the electrodes relevant heatmaps, the heatmaps are defined as the combination of the feature maps and are also followed by a ReLU to acquire only the features that are contributing positively (the more important Electrodes, such that:

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \quad \text{Eq. 5.15}$$

Where $L_{Grad-CAM}^c$ is the final generated heatmap that will have the same size as the feature map i.e. (22×400 at the second layer in the suggested model). Noting that, after applying the convolution operation at the spatial layer (where the kernel is 22×1), the following feature maps are going to be one-dimensional (1×400) and thus the adjusted same procedure (Eq. 5.2 to 5.4) is applied to the rest of the features maps. In addition, in one of the variations of the suggested models, there is only one feature map where $k = 1$.

5.2.3.7 Dataset Formal representation and pre-processing

The datasets used in this experiment are defined as $D^i = \{(X^1, y^1),\ldots,(X^{Ni}, y^{Ni})\}$ such that D is the dataset of subject i, *and* N^i denotes the number of trials for subject i. X denotes the number of trials which belongs to class y and $X^j \in \mathcal{R}^{ET}$ where j is the trial number or example in the context of neural networks and $1 < j < N_i$, E *a*s the number of electrodes recorded and T as the total number of time samples per trial. Concretely, the models are evaluated on the BCI IV 2a competition dataset with the four classes: left hand, right hand, foot, or tongue, such that a trial j has a corresponding class $y^j \in \{l_1 =$ Hand left, $l_2 =$ Hand right, $l_3 =$ Foot, $l_4 =$ Tongue$\}$.

5.2.3.8 Augmentation by Shuffling crops and Sliding windows

Furthermore, to simulate real-time signals, the trials were cropped and overlapped with a window size $W = \{200,400\}$ with a step $S = 50$ over an interval $I = \left[T_j\ ,\ T_{j+S}\right]$ forming a new set of trials $\bar{X}$ and the resulting crops are added to a new dimension P representing the number of slices/patches to acquire the new input matrix $\bar{X}^j \in \mathcal{R}^{ETP}$.

Data augmentation is a common technique used in training neural networks, which it aims to reduce model overfitting using existing information in the training generating new data samples. The widely generic practices entail cropping, flipping an image, rotating the image and colour changes or in other words geometric augmentation (Cireşan, Meier, Gambardella, & Schmidhuber, 2010; Yang, Zhao, Chan, & Yi, 2016). As discussed in the experiment procedure (Section 3.2.1), the participants are instructed to maintain the MI movements for at least four seconds when the data is being collected. Ideally, it is assumed that the temporal structure is similar and periodic over the four seconds (hence as discussed thoroughly throughout techniques like Fast Fourier Transform and Wavelet decomposition for EEG are the traditional techniques used for the analysis and feature extraction). A sample shuffling is suggested in this study, for simplicity, the E^j will be omitted from equations since the following operation is applied over all electrodes at E^j in the same way. Given the time samples T^j of trial $X^j \in \mathcal{R}^{ET}$ and a crop $t^j[s,e] = \left\{t^j : T_s^j < t^j < T_e^j\right\} \subseteq T^j$ where $s\ and\ e$ denote the location of the starting and ending sample respectively. Samples t^j are swapped with non-

overlapping samples $\bar{t}^{\,j}$ generating a new training example $\hat{X} \in \mathcal{R}^{ET}$. Finally, a crop may or may not be also swapped across trials i.e. swapping $\bar{t}^{\,j=10}$ with $t^{j=200}$. In this study, just one configuration of augmentation was tested, where the number of samples to be swapped was the equivalent of two seconds: $t^j = [s = 0, e = 500]$ and $\bar{t}^{\,j} = [\mathrm{s} = 500, \mathrm{e} = 1000]$ and further shuffling between trials was random and set to maximum of 20 trials crossover shuffles.

5.3 Results

5.3.1 Classification accuracy for the suggested model without the top Block1_1 and without the top Spatial Attention mechanism over the four group variations.

Table 5.1: Independent Subjects classification accuracy over various windows and periods.

	BCI IV 2a
W/S/I	**Mean Accuracy**
400/50/4	0.68 +- 0.007
200/50/4	0.65 +- 0.013
400/50/2	0.65 +- 0.012
200/50/2	0.65 +- 0.012

There was no significance found between the three groups shown in $F(3,241) = 0.92, p > 0.01$. Post-hoc pairwise tests between individual window sizes using Tukey HSD method, suggest no significant differences between 100 and 200, 100 and 400, 200 and 400 where p = {0.9,0.40,0.61} respectively. The four seconds interval with 400 (1.6s) window yielded an average accuracy of 68%, which is 3% higher than the rest of the groups.

5.3.2 The classification accuracies between: with the additional top Block1_1, the two suggested Spatial attention mechanisms and the De-mixing layer as implemented in Zubarev, Zetter, Halme, & Parkkonen, (2019).

There was no statistically significant difference was found between the 12 groups (Block1_1 No attention, De-mixing Layer, Squeeze and Excitation Attention and Attention De-mix for the four configurations 400/50/4, 400/50/2, 200/50/4, 200/50/2) where $F(12,) = 0.15, p > 0.05$.

The performance has not improved significantly over the baseline employing the de-mixing attention mechanism using any of the configurations and was almost identical to the baseline model. A 3% increase in the mean accuracy over the baseline is witnessed in the 400/50/2 having the Block1_1 with De-mix attention. A 3% increase over the baseline is witnessed with no attention and 4% increase employing the de-mix in the mean accuracy for 200/50/2. Employing the De-mixing layer (not with attention) performed the worst with a decrease of 2% for 400/50/4 over the baseline, 1% increase in 200/40/4 and identical for 400/50/2. In addition, the performance was almost identical between the Attention De-mix and No attention where a 1% increase was witnessed in 400/50/5 and 200/50/2 and the identical for 400/50/2. In addition, the Squeeze & Excitation attention was identical to the de-mix attention with 1% decrease in all the groups but the 200/50/4 which was identical. As show in Table 4.2.

Table 5.2**:** The classification accuracies of the suggested model with the suggested model comparing the accuracies between the different model configuration.

	Block1_1 No attention	**Demixing Layer**	**Spatial Attention two Demix**	**Spatial Attention Squeeze& excitation**
W/S/I	**Mean Acc.**	**Mean Acc.**	**Mean Acc.**	**Mean Acc.**
400/50/4	0.68 +- 0.007	0.66 +- 0.008	**0.69** +- 0.006	**0.68** +- 0.009
200/50/4	**0.67** +- 0.013	0.66 +- 0.011	0.66 +- 0.009	0.66 +- 0.030
400/50/2	**0.68** +- 0.012	0.65 +- 0.006	**0.68** +- 0.004	**0.67** +- 0.004
200/50/2	0.68 +- 0.009	0.66 +-0.020	**0.69** +- 0.008	**0.68** +- 0.015

5.3.3 Using all the four seconds of training as Two seconds intervals and the proposed Augmentation. The performance after using data Augmentation.

An increase in the classification accuracy of 4% is witnessed over the best performing in the above groups (400/50/4 with $0.69 \approx 69\%$) using the four second interval divided into two-two seconds and applying augmentation. Applying a one-way ANOVA between the two groups showed no statistical significance $F(2) = 0.78, p > 0.05$. An increase of 1% and 3% with the augmented training set over the not augmented in the two groups 400/50/2 and 200/50/2 respectively.

Table 5.3: The classification accuracy for the best performing configurations: Four seconds Intervals (I) divided into two-two seconds.

	Not augmented		Augmented		
W/S/I	**Block_1_1**	**Block_1_1+ Demix Attention**	**Block_1_1**	**Block_1_1 + Demix Attention**	**Excitation and Squeeze Temporal Attention. No Spatial Attention**
400/50/2	0.70 +- 0.03	0.68 +- 0.02	0.71 ± 0.02	0.70 ± 0.06	0.70± 0.12
200/50/2	0.70 +- 0.02	0.67 +- 0.01	0.73 ± 0.10	0.71 ± 0.14	0.73 ± 0.08

5.3.3 Final top performing model in comparison with top performing methods in the literature for dataset BCI IV 2a.

Table 5.4: The comparison between the best performing methods in the literature measured in Cohen Kappa and the suggested model with the double attention (CRNN-DA).

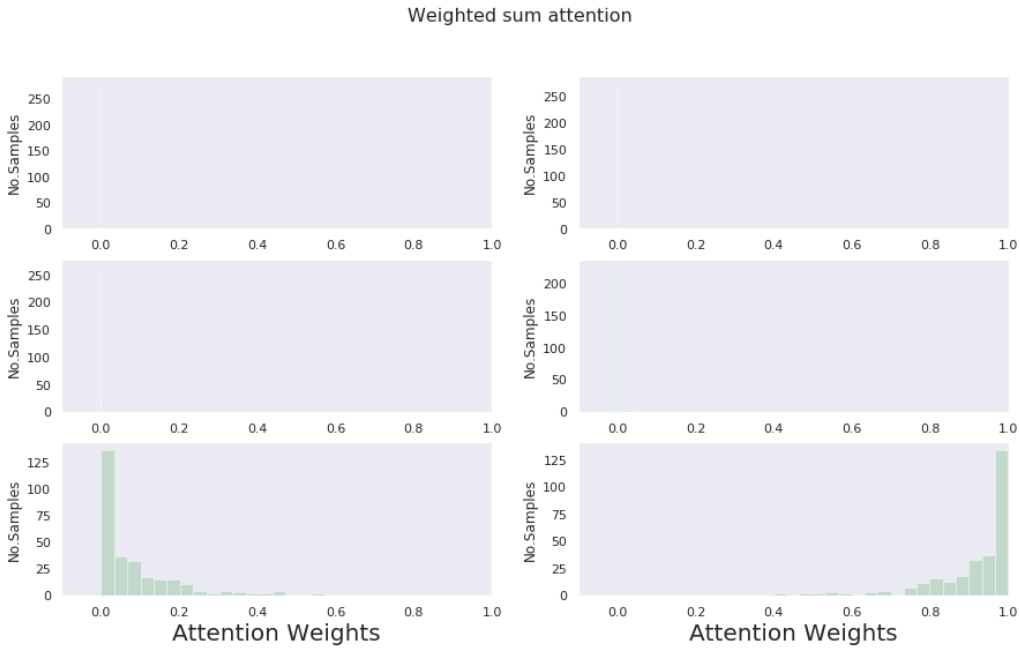
Participant	1st	2nd	3rd	RSTNN	CRNN-DA
01	0.68	0.69	0.38	0.69	**0.71**
02	0.42	0.34	0.18	0.29	**0.42**
03	0.75	0.71	0.48	0.68	**0.79**
04	0.48	0.44	0.33	0.34	**0.51**
05	0.40	0.16	0.07	0.09	**0.54**
06	0.27	0.21	0.14	0.30	**0.37**
07	**0.77**	0.66	0.29	0.57	0.75
08	**0.75**	0.73	0.49	0.49	0.72
09	0.61	0.69	0.44	0.56	**0.63**
Mean	**0.57**	**0.52**	**0.31**	**0.45**	**0.60**

The suggested model achieved the highest performance with a mean accuracy of 0.60 (60%) where the performance is measured as the Cohen Kappa. The suggested model achieved superior performance for 6 subjects.

5.3.4 Attention and Grad-CAM

The output of the first temporal attention mechanism (Weighted sum attention) shows that the most contributing features lie in the second half of the signal at about 2.5 seconds when the model is trained without the augmentation (i.e. Fig 5.4 row A and row B).

6 time slices (200/50/2). Weighted Sum attention.



2 time slices (400/50/2). Excitation and Squeeze.

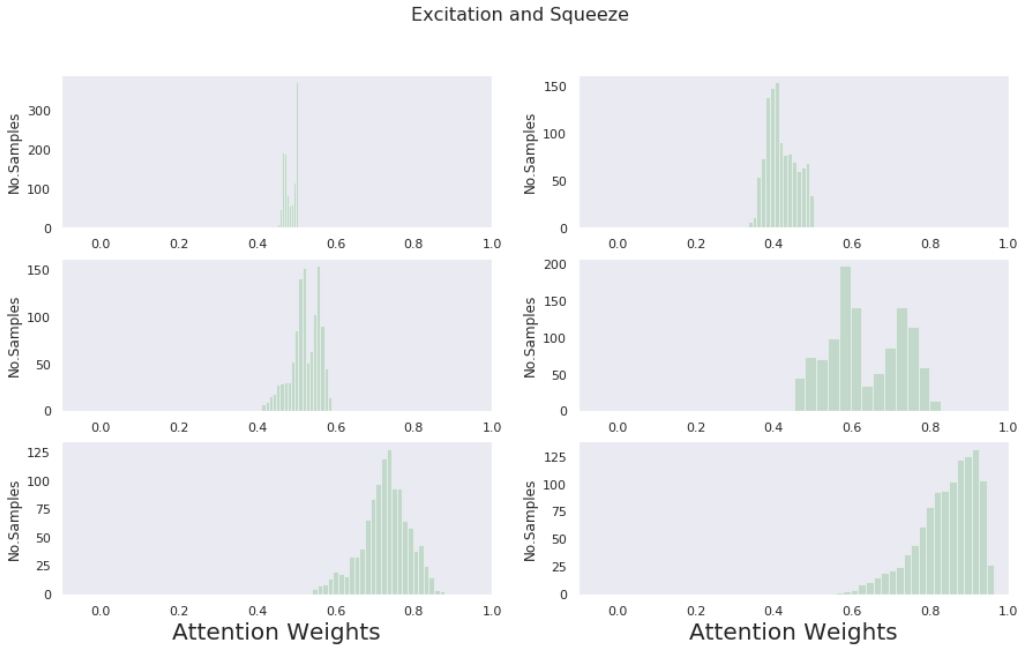
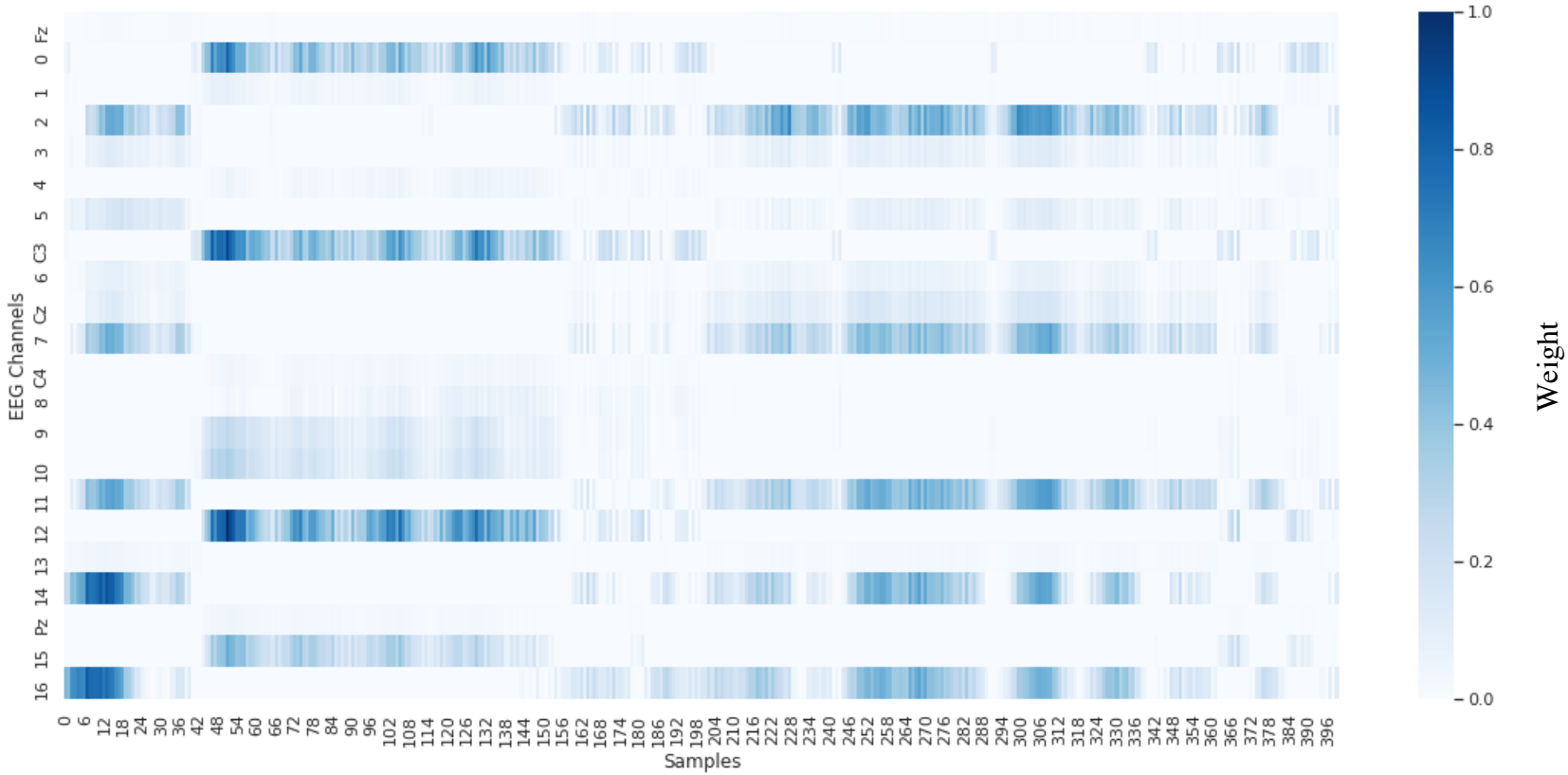
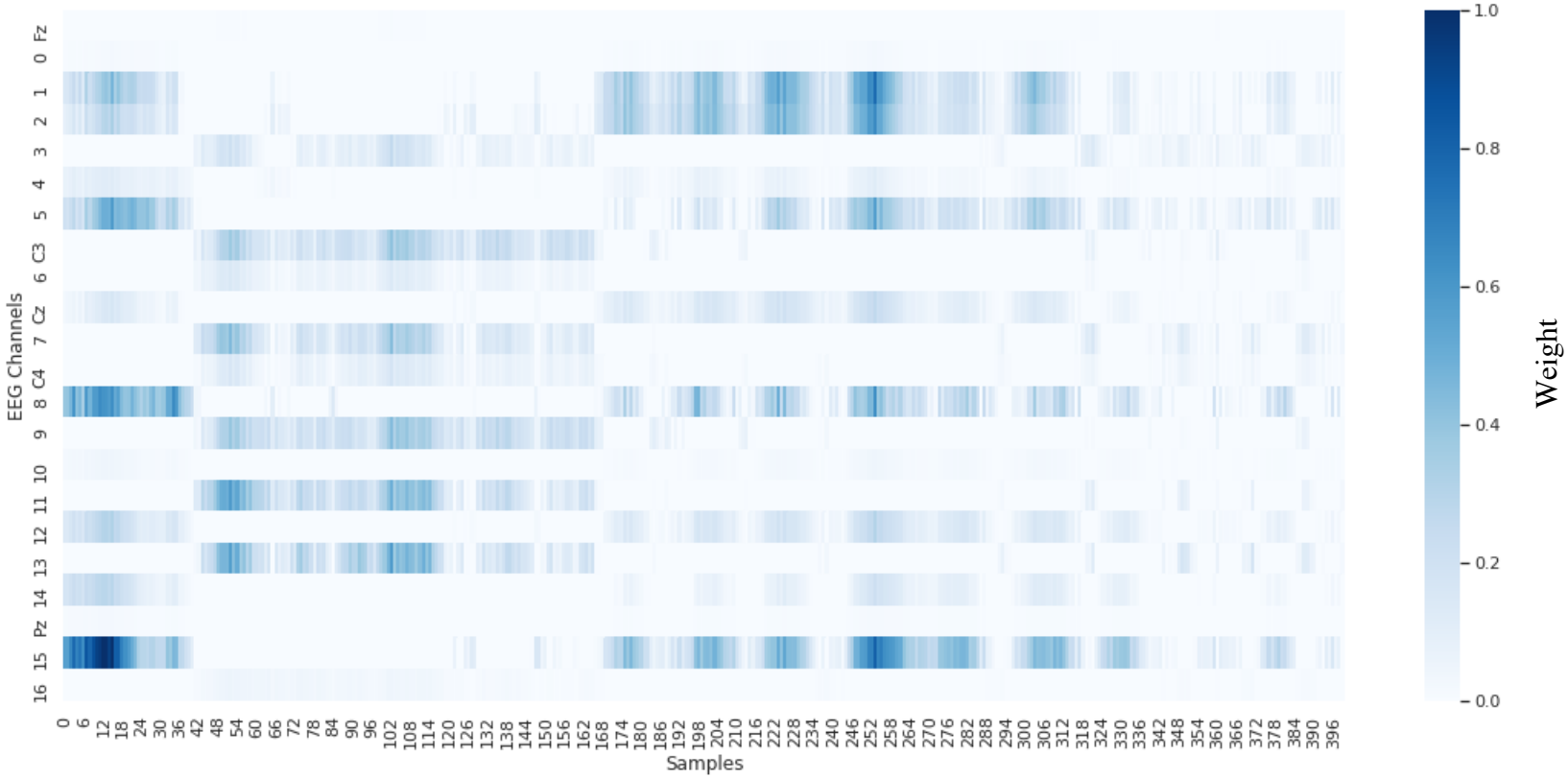


Figure 5.3: Top panel showing histograms of 6 time slices using attention mechanism one (the weighted sum attention). The bottom figure showing histograms of 6 time slices using attention mechanism two (Excitation and Squeeze). The x-axis represents attention weight and y-axis are the number of samples. The example is from participant 3.

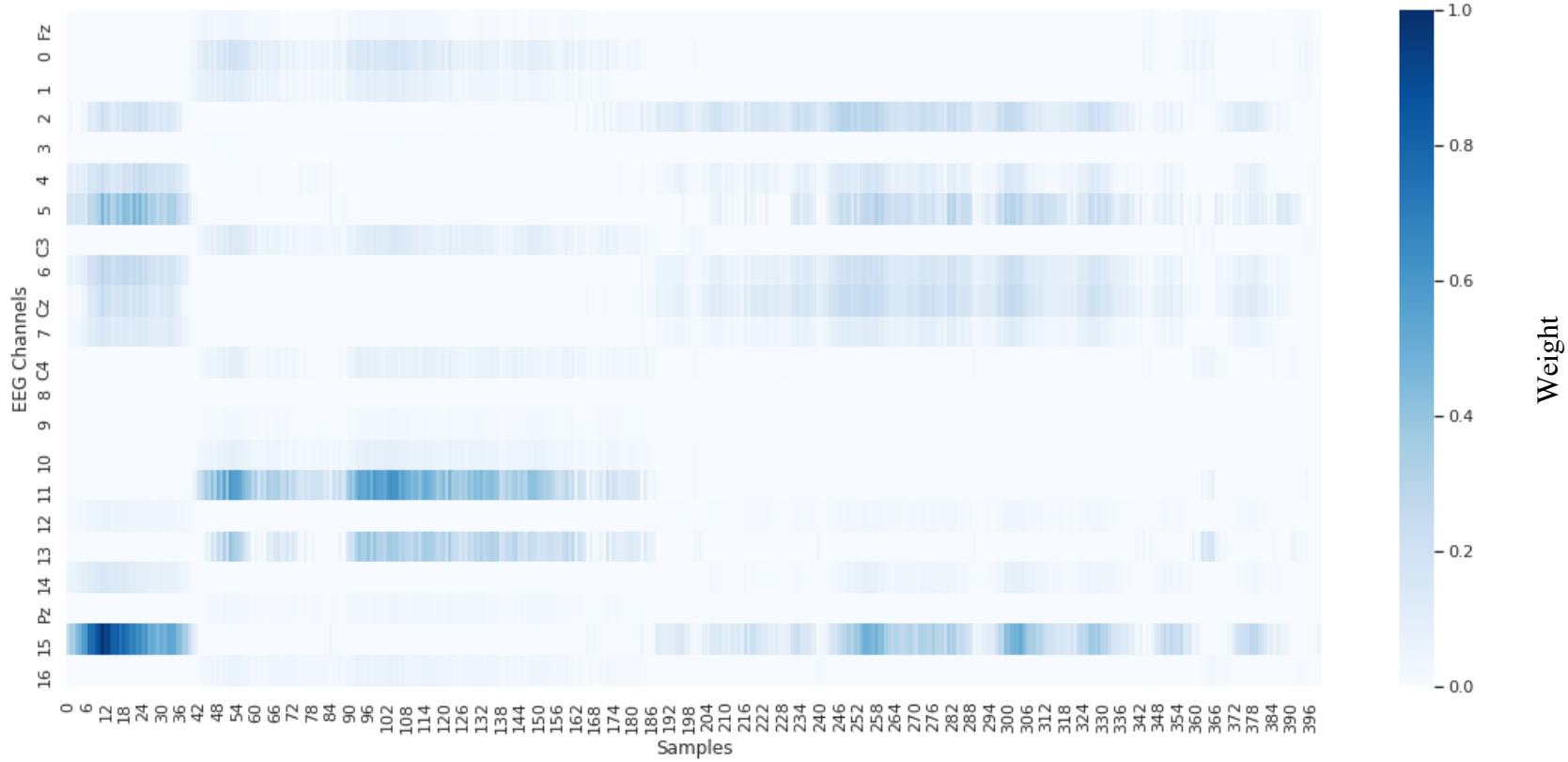
A


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
0
6
12
18
24
30
36
42
48
54
60
66
72
78
84
90
96
102
108
114
120
126
132
138
144
150
156
162
168
174
180
186
192
198
204
210
216
222
228
234
240
246
252
258
264
270
276
282
288
294
300
306
312
318
324
330
336
342
348
354
360
366
372
378
384
390
396
1.0
0.8
0.6
0.4
0.2
0.0
Weight

B


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
Weight
1.0
0.8
0.6
0.4
0.2
0.0

C


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
Weight
1.0
0.8
0.6
0.4
0.2
0.0

D

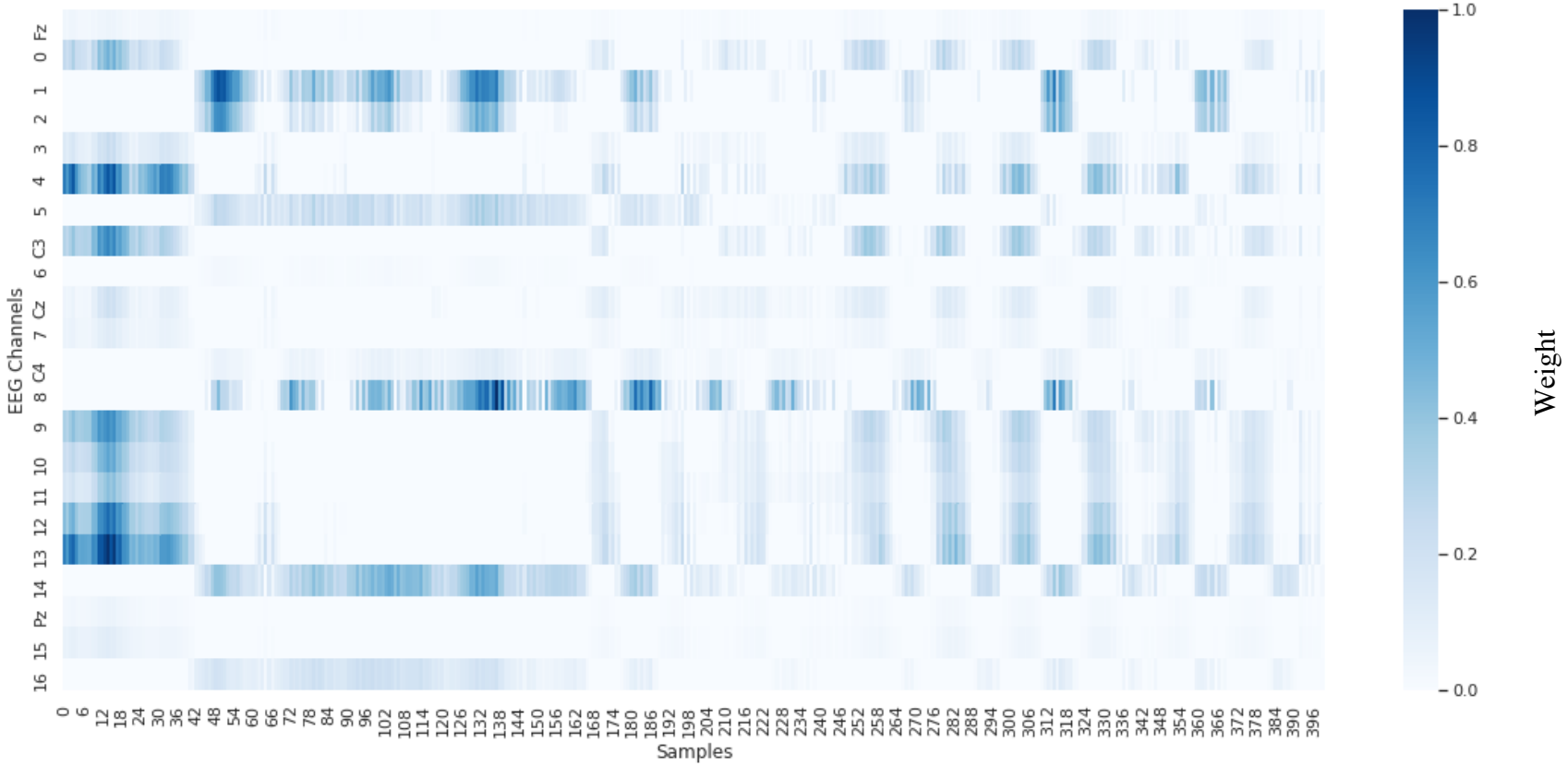


Figure 5.4: An example of the heatmaps of the guided Grad-CAM of participant 3. A. is for the left hand, B. right hand, C. feet and D. tongue.

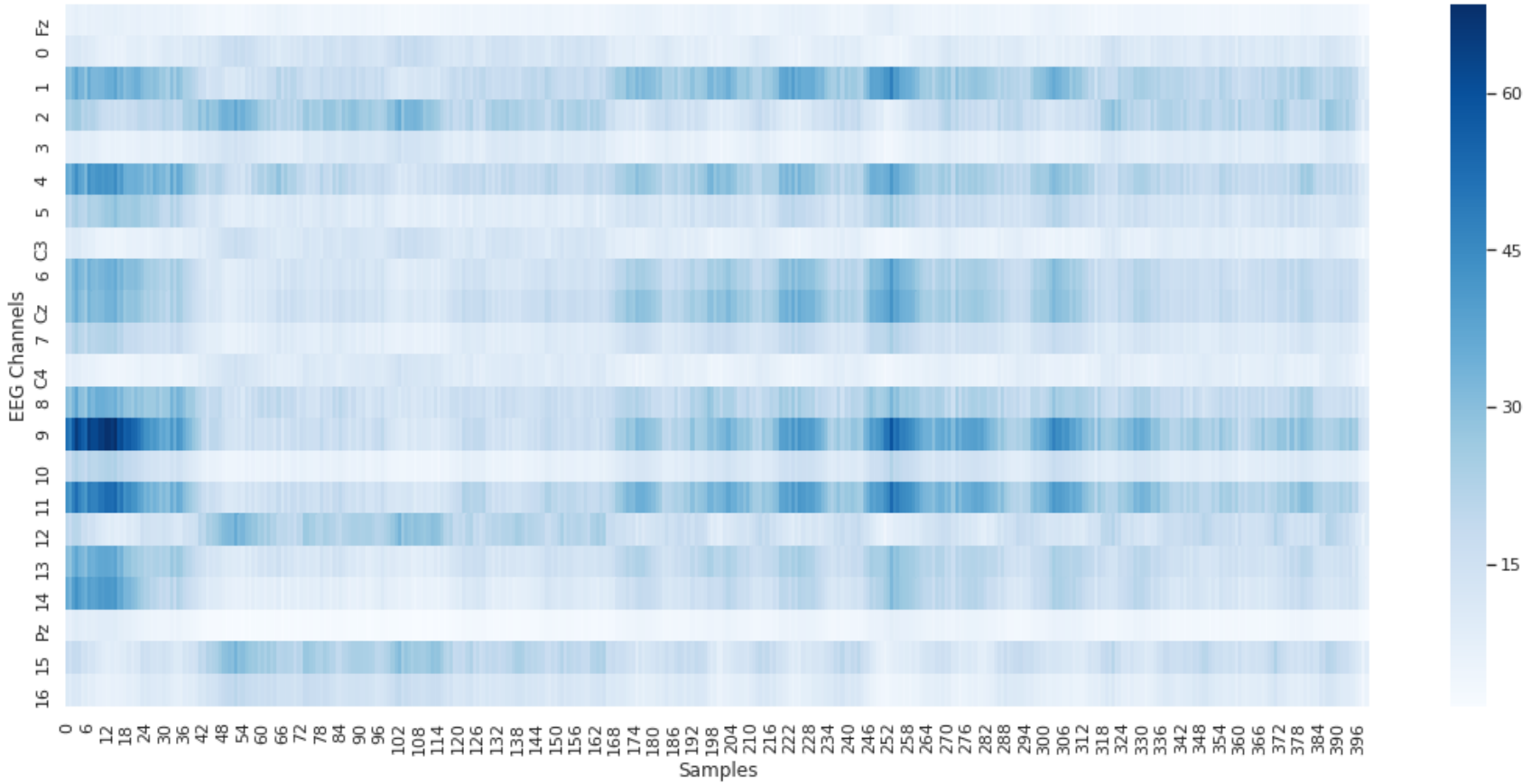


Figure 5.5: An example of the heatmaps of forward activation of participant 3 for the Right Hand class.

5.4 Discussion

The suggested model has achieved better performance with $mean\ kappa = 0.60$ than the state-of-art RSTNN ($k = 0.45$), and the winner of BCI IV 2a competition with ($kappa = 0.57$) at the time that this chapter was written as shown in Table 5.4. The additional layers and the spatial attention mechanism didn't improve the performance significantly. However, the additional Block1_1 layer did have a slight improvement in the accuracy and enabled a comparison between the feed-forward activation maps and the activation maps obtained through guided Grad-CAM. It also increased the accuracies for a number of participants Appendix A Table A.1.

In this study, the different variations ($1.6s$ and $0.8s$) windows over either four seconds interval or two seconds interval has been fed to the model for training and testing as shown in Tables 5.1 to 5.2. Noting that the results reported are for the unseen test set. The four seconds intervals with the 400 samples window in the baseline had a superior performance. Whilst, the additional configurations to the network (Block1_1) and the spatial attention lead to an increase in the classification accuracy of the shorter windows and intervals, such that the model was able to have identical performance between 400/50/4 and the 200/50/2, implying that the speed of the classification task is twice as faster (i.e., the 400 window over 4 seconds interval would require a delay of 4 seconds and an input of 1.6 seconds per one step of classification a two second delay with 0.8 seconds input per one step of classification) which is slightly better than the current real-time EEG based classification applications. Another advantage of shorter windows and intervals is that less memory is required to store the signals, thereby increasing the potential for real-time EEG applications.

The data augmentation where intervals are swapped for training the model (not applied to the testing set) and the four seconds divided into two second intervals with augmentation have lead to the highest performance (higher accuracies) where the mean accuracy is 71% for the 400/50/2 and 72% for the 200/50/2 groups implying that the suggested augmentation technique can increase the model performance further.

The output of the first temporal attention mechanism (Fig. 5.3, top row), shows that the most contributing features lie in the second half of the signal at about 2.5 seconds when the model is trained without the augmentation. Moreover, when training the network with the two-two seconds augmented intervals, it showed a similar distribution where the most contributing features seems to also lie in the second half of the signal. However, as explained in Chapter 4, the augmentation method used in this study, swaps between the first half and the second half of the signal, the augmentation should have showed a wider distribution or an even distribution between the two halves of the signal. On the other hand, the second attention mechanism (Excitation and Squeeze) had similar performance in the classification task and was able to show a better interpretation for the temporal slices as shown in Fig. 5.4. The attention weights seem to be better distributed over all the slices providing more informative interpretation or higher resolution interpretation (Fig. 5.4). Hence, the suggested Excitation & Squeeze temporal attention mechanism is suggested for architectures employing a recurrent layer since it is not computationally costly where only 800 parameters are learned, and the attention vectors are informative to be used as a feedback loop.

The Squeeze and Excitation Spatial attention failed to provide useful attention scores, where all the channels ended up with the same attention scores (0.004). Another interpretation would be each of the used channels contributed equally to the model decision, whilst this interpretation is valid, it disagrees with the linear spatial methods such as CSP and FBCSP where these methods base their classification on the spatial correlation (eigen vector filters) learned by CSP (refer to Chapter 3). On the other hand, the generated heatmaps obtained guided grad-cam shows higher contribution of selected channels between the different classes (Left-Hand, Right-Hand, Feet and Tongue) as shown in the example in Fig. 5.4. The resulted Grad-CAMs were multiplied by the original input tend to have a superior visual resolution as opposed to the heatmaps generated by feed forward as shown in Fig. 5.5 where it is difficult to identify the contribution of the features and tend to have a lower resolution for visualisation and hence lower interpretability.

Another finding worth mentioning is that when empirically testing the location of the spatial attention at different layers before employing the channels convolution (since the spatial attention operate on the electrodes, it can't be added after convolving the channels) a significant decline in performance was observed (the maximum mean accuracy achieved was below 60%).

This suggests that convolving the channels at the shallow levels of the model leads to better performance. These findings are consistent with the top performing models in the literature. Such that, in all the high performing models, the first few layers will usually contain the convolution operating on the channels at the first two or three layers. Moving the attention mechanism to the second layer, performed similarly to the one at the beginning.

5.5 Summary of Contributions

- Investigation of the classification performance of MI over different windows and intervals;
- A novel deep architecture based on GRU and Convolution units is introduced;
- A novel temporal and a novel spatial attention mechanism are introduced;
- A generalised guided Grad-Cam for EEG is introduced for higher interpretability;
- A novel EEG data augmentation technique is suggested.

5.6 Limitations

- The performance of the proposed methods was not measured using fewer channels (as discussed).
- The performance of the proposed methods was not measured using only the periods (time slices) that the attention mechanisms extracted as the prominent features.
- The models were not trained and evaluated combining all the participants data (where the model is trained for all participants to acquire a participant agnostic model.

Chapter 6:

General Discussion

The primary aim of this research was to obtain a deep neural network architecture with reliable performance for classifying MI movements recoded as EEG signals. This research was focused on retaining the state-of-art classification accuracy while reducing the intervals (number of samples) needed for high performing classifiers. To test the performance of the novel suggested methods, it was essential to acquire a baseline of the naïve traditional methods. Accordingly, In the first experiment (Section 3.2.3), the traditional and basic methods Common Spatial Patterns (CSP) and Fast Fourier Transform (FFT) were explored in correlation with the intervals and window sizes of the input. The results showed that using longer windows (two and four second windows) would results in significantly better performance (Section 3.3, Tables 3.1 and 3.2). Even though, in Chapter 3 these methods were not fully optimised and only the basic techniques were applied, the state-of-art classification accuracies are usually obtained with intervals > 2 seconds which are not ideal for real-time systems. The experiment was designed to act as the baseline for the following studies. In Chapter 4, the state-of-art Convolutional neural networks (CNNs) based architectures have been investigated and compared with CNNs trained with Continuous Wavelet Transform (CWT) features instead of raw EEG signals.

The novel architecture Point-Wise Convolutional Network (PWCN) was suggested which aims to reduce dimensionality of the input at the first layer of the network by convolving depth-wise the samples of the acquired scales (frequency bands) after applying CWT aiming to learn one feature map as a representation of all the scales, in other words, it can be thought of as reconstructing the signal back from the transformation with the difference that the parameters of the transformation are learned via gradient descent in relation to the output class (Section 4.2). The suggested model achieved 80% mean accuracy (Table 4.4) that is relatively close to the state-of-art for the BCI IV 2b dataset (2 classes and 3 Electrodes) but the performance on the BCI IV 2a dataset was almost at chance level classifying four classes with 22 Electrodes. Furthermore, applying guided-Grad-CAM to the suggested models using CWT features and to the state-of-art models (EEGNet and FBCSP) showed interesting results (Figs. 4.6 to 4.9) where each model was extracting the features from different samples in time and different intervals which suggested that if one model (architecture) is able to learn the correlations across different intervals, a significant improvement in the performance should be witnessed.

In Chapter 5, the novel architecture Convolutional Recurrent Neural Network with Double attention CRNN-DA (Section 5.2.3) was introduced addressing the shortcomings and building upon the findings of (Chapter 3 Section 3.4 and Chapter 4 Section 4.4). The suggested architecture also achieved state-of-art accuracies ($kappa = 0.60$ and 73% $mean\ accuracy$) using two second trials and 0.8 second windows (200 samples) as shown in Section 5.3 (Table 5.4). Suggesting that for real time systems, 0.8 seconds intervals are used for the classification.

The CNN based architectures were proven to extract and encode features from raw EEG signals but didn't account for the time aspect of the signals and limiting the interpretability of what the model learned. The CRNN-DA included a recurrent layer in the built of Gated Recurrent Units (GRUs) to learn the temporal correlations between time slices which are natural in the EEG signals. Furthermore, two spatial attention mechanisms (Demix and Squeeze & Excitation Section 5.2.3.1) and two temporal attention mechanisms (Weighted Sum Attention and Excitation & Squeeze Section 5.2.3.4) were added to the model to highlight the most contributing features in the form of attention vectors.

The spatial attention mechanisms didn't show any improvement and it wasn't clear from the analysis afterwards what they learned is useful or interpretable. The Demix attention mechanism calculated the attention vector as learned weights which is static and not ideal for the interpretation, it can provide global representation independent of the class. While the Squeeze and Excitation learned the same attention scale for all channels (was always around 0.004) which wasn't useful or as discussed and unlikely, the network learns from all the electrodes equally.

The temporal attentions were not compared with no temporal attention in terms of performance, but the attention vectors of the two suggested mechanisms were compared (Fig. 5.3) where the weighted sum attention always showed that the last time slice has much higher contribution to the classification, the Excitation and Squeeze attention showed a better distribution (while the last slices still showed higher contribution). Furthermore, the Excitation and Squeeze temporal attention has about only 800 learnable parameters as opposed to about 17,000 parameters for the weighted sum attention. Suggesting that the excitation and squeeze is a better attention mechanism and potentially can be applied to any recurrent layers. It can be also slightly modified to include the information from stacked layers. For further investigation, the Guided

Grad-CAM which is one of the best algorithms for visualising DNN models in relation with the specific classes. The Grad-CAM has been modified to extract useful interpretation provided EEG signals for input, since the method has been developed for inputs such as images and EEG signals is not highly interpretable in that form. Hence, the modified Guided Grad-CAM (Section 5.2.3.6) for EEG can be used for approximating the most contributing features in terms of which electrodes and time samples (as shown in Chapter 4 and Chapter 5). Additionally, the resulting signals or heatmaps from the Grad-CAM can be used for further analysis such as applying spectrograms or scalograms that are specific to the class being analysed (i.e., Right Hand, Left Hand).

Future work

As shown in the results throughout the thesis, the accuracies for each participant are different and some methods work better for a group of participants over other. Trying to obtain more datasets, preferably larger datasets to evaluate the proposed methods would provide a better insight of this behaviour that might lead to acquiring a more generalisable model based on the proposed methods with minimal tweaking.

The discussed methods have shown an improvement in the performance being applied to offline datasets. However, the main aim of this research is to improve the performance using short windows for real-time applications. Hence, testing the suggested method on real-time data and measuring its performance would be one of the first studies moving further from this presented work, Ideally, the performance shouldn't decrease significantly but in a real-time application, there are more variables that might not be accounted for.

Additionally, it would be interesting to have several experiments with the temporal attention as a feedback loop to the participants and comparing the quality of the datasets. An additional experiment needed to validate the temporal attention results would be filtering the trials removing the intervals with low attention weights and training different models, if an improvement in the performance is witnessed, the suggested model and procedure could be added to the pipeline for data cleaning and extracting the best trials. If the there was no change in the performance, that would validate the efficacy of the suggested methods implying that

the model is able to extract only the relevant information. If a decrease of performance is witnessed, further investigation would be required.

Last but not least, since the spatial attention didn't seem to provide useful insight. Measuring the performance with retrained models with different subsets where channels are randomly removed or set to 0 should provide an answer to whether the model actually learns from all the channels and they are all necessary or not.

Finally, the BCI headsets are becoming more mainstream now and they are bought for affordable prices outside of the academic environment. Potentially, this will lead to having more EEG data which will be very useful for future research. Essentially, deep neural networks achieve better performance with big datasets. A very interesting area of research would be having a unified participant agnostic model trained using all the participants (ideally millions of EEG data records) that generalises for any new unseen EEG Motor Imagery task. Such a system might be ambitious, but there are a number of advancements in that field including easy electrode implants and existing technologies such as the Electrocorticography (ECoG) and intracranial electroencephalography (iEEG) where the same methods discussed in this thesis could be applied to.

References

Abdalsalam, E., Yusoff, M.Z., Mahmoud, D., Malik, A.S. and Bahloul, M.R., 2018. Discrimination of four class simple limb motor imagery movements for brain–computer interface. *Biomedical Signal Processing and Control*, *44*, pp.181-190.

Al Rahhal, M.M., Bazi, Y., Al Zuair, M., Othman, E. and BenJdira, B., 2018. Convolutional neural networks for electrocardiogram classification. *Journal of Medical and Biological Engineering*, *38*(6), pp.1014-1025.

Al-Fahoum, A.S., Al-Fraihat, A. a, 2014. Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains. *ISRN Neuroscience 2014*, 730218. doi:10.1155/2014/730218

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A. and Bengio, Y., 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

Andreotti, F., Phan, H. and De Vos, M., 2018. Visualising convolutional neural network decisions in automatic sleep scoring. *CEUR Workshop Proceedings* (pp. 70-81).

Ang, K.K., Chin, Z.Y., Wang, C., Guan, C. and Zhang, H., 2012. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in Neuroscience*, *6*, p.39.

Ang, K.K., Chin, Z.Y., Zhang, H. and Guan, C., 2008. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (pp. 2390-2397), Hong Kong, 19-24 July.

Bahdanau, D., Cho, K. H., & Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. 3rd *International Conference on Learning Representations, ICLR 2015* - Conference Track Proceedings, 1–15, San Diego, California, United States of America 7-9 May.

Bashivan, P., Rish, I., Yeasin, M. and Codella, N., 2015. Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*.

Behri, M., Subasi, A. and Qaisar, S.M., Comparison of machine learning methods for two class motor imagery tasks using EEG in brain-computer interface. In *2018 Advances in Science and Engineering Technology International Conferences (ASET)* (pp. 1-5), Dubai, 6 February- 5 April.

Bishop, C., 2006. Pattern Recognition and Machine Learning, 1st ed, 1613-9011. Springer, New York.

Brigham, E.O., 1988. *The fast Fourier transform and its applications*. Prentice-Hall, Inc..

Bright, D., Nair, A., Salvekar, D. and Bhisikar, S., 2016, June. EEG-based brain controlled prosthetic arm. In *2016 Conference on Advances in Signal Processing (CASP)* (pp. 479-483). IEEE.

Brodu, N., Lotte, F. and Lécuyer, A., 2011, April. Comparative study of band-power extraction techniques for motor imagery classification. In *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)* (pp. 1-6), Singapore, April 16-19.

Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W. and Ramanan, D., 2015. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2956-2964), Santiago, Chile, 7-13 December.

Carreiras, C., Sanches, J.M., 2011. ERD / ERS Event Detection from Phase Desynchronization Measurements in BCI 1–2. *Portuguese Conference on Pattern Recognition*, Porto, Portugal, 28 October.

Chin, Z.Y., Ang, K.K., Wang, C., Guan, C. and Zhang, H., 2009, September. Multi-class filter bank common spatial pattern for four-class motor imagery BCI. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 571-574), Hilton Minneapolis, Minnesota, 2-6 September.

Cireşan, D.C., Meier, U., Gambardella, L.M. and Schmidhuber, J., 2010. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, *22*(12), pp.3207-3220.

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Das, P., Sadhu, A.K., Konar, A., Bhattacharya, B.S. and Nagar, A.K., 2015. Adaptive Parameterized AdaBoost Algorithm with application in EEG Motor Imagery

Classification. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). Killarney, Ireland, July 12–16.

Daubechies, I., 1990. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, *36*(5), pp.961-1005.

Devlaminck, D., Wyns, B., Grosse-Wentrup, M., Otte, G. and Santens, P., 2011. Multisubject learning for common spatial patterns in motor-imagery BCI. *Computational Intelligence and Neuroscience*, *2011*, p.8.

Elstob, D. & E. 2016. a low cost EEG Based BCI prosthetic using motor imagery. *International Journal of Information Technology Convergence and Services*. 6. 23-36. 10.5121/ijitcs.2016.6103.

Forney, E.M. and Anderson, C.W., 2011, July. Classification of EEG during imagined mental tasks by forecasting with Elman recurrent neural networks. In *The 2011 International Joint Conference on Neural Networks* (pp. 2749-2755). San Jose, California, USA,

Fukunaga, K. 2013. Introduction to statistical pattern recognition. Academic press.

Gareis, I.E., Vignolo, L.D., Spies, R.D. and Rufiner, H.L., 2017. Coherent averaging estimation autoencoders applied to evoked potentials processing. *Neurocomputing*, *240*, pp.47-58.

Gomez-Rodriguez, M., Grosse-Wentrup, M., Hill, J., Gharabaghi, A., Scholkopf, B. and Peters, J., 2011. Towards brain-robot interfaces in stroke rehabilitation. *2011 IEEE International Conference on Rehabilitation Robotics*. Zurich, Switzerland, 29 June - 1 July.

Graimann, B., Huggins, J.E., Levine, S.P. and Pfurtscheller, G., 2002. Visualization of significant ERD/ERS patterns in multichannel EEG and ECoG data. *Clinical Neurophysiology*, *113*(1), pp.43-47.

Güler, N.F., Übeyli, E.D. and Güler, I., 2005. Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications*, *29*(3), 506-514.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778). Las Vegas, Nevada, USA, 26 June- July 1.

Hema, C.R., Paulraj, M.P., Yaacob, S., Adom, A.H. and Nagarajan, R., 2007. Brain machine interface: Classification of mental tasks using short-time PCA and recurrent neural networks. In *2007 International Conference on Intelligent and Advanced Systems* (pp. 1153-1156). Kuala Lumpur, Malaysia, 25- 28 November.

Hinton, G.E. and Sejnowski, T.J., 1983. Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 448-453). Washington, D. C. June.

Hu, J., Shen, L. and Sun, G., 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7132-7141). Salt Lake City, Utah, USA, 18-22 June.

Hung, C.I., Lee, P.L., Wu, Y.T., Chen, L.F., Yeh, T.C. and Hsieh, J.C., 2005. Recognition of motor imagery electroencephalography using independent component analysis and machine classifiers. *Annals of Biomedical Engineering*, *33*(8), pp.1053-1070.

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning*, ICML 2015, 1, 448–456. Lille, France, 6-11 July.

Jahankhani, P., Kodogiannis, V. and Revett, K., 2006, October. EEG signal classification using wavelet feature extraction and neural networks. In *IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)* (pp. 120-124). Sofia, Bulgaria, 3-6 October.

Kalcher, J. and Pfurtscheller, G., 1995. Discrimination between phase-locked and non-phase-locked event-related EEG activity. *Electroencephalography and Clinical Neurophysiology*, *94*(5), pp.381-384.

Kevric, J. and Subasi, A., 2017. Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system. *Biomedical Signal Processing and Control*, *31*, pp.398-406.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. 3rd *International Conference on Learning Representations*, *ICLR 2015* - Conference Track Proceedings, 1–15. Banff, Canada, 14-16 April.

Ko, W., Yoon, J., Kang, E., Jun, E., Choi, J.S. and Suk, H.I., 2018, January. Deep recurrent spatio-temporal neural network for motor imagery based BCI. In *2018 6th International Conference on Brain-Computer Interface (BCI)* (pp. 1-3). High 1 Resort, Korea, 15-17 January.

Kobler, R.J. and Scherer, R., 2016, October. Restricted Boltzmann machines in sensory motor rhythm brain-computer interfacing: a study on inter-subject transfer and co-adaptation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 000469-000474). Budapest, Hungary, 9-12 October.

Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P. and Lance, B.J., 2018. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), p.056013.

Lee, H.K. and Choi, Y.S., 2018, January. A convolution neural networks scheme for classification of motor imagery EEG based on wavelet time-frequecy image. In *2018 International Conference on Information Networking (ICOIN)* (pp. 906-909). Chiang Mai, Thailand, 10-12 January.

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H. and Pfurtscheller, G., 2007. Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(4), pp.473-482.

Lemm, S., Schafer, C. and Curio, G., 2004. BCI competition 2003-data set III: probabilistic modeling of sensorimotor mu rhythms for classification of imaginary hand movements. *IEEE Transactions on Biomedical Engineering*, *51*(6), pp.1077-1080.

Li, J., Struzik, Z., Zhang, L. and Cichocki, A., 2015. Feature learning from incomplete EEG with denoising autoencoder. *Neurocomputing*, *165*, pp.23-31.

Lu, N. and Yin, T., 2015. Motor imagery classification via combinatory decomposition of ERP and ERSP using sparse nonnegative matrix factorization. *Journal of Neuroscience Methods*, *249*, pp.41-49.

Lu, N., Li, T., Ren, X. and Miao, H., 2016. A deep learning scheme for motor imagery classification based on restricted boltzmann machines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(6), pp.566-576.

Luu, T., Nakagome, S., He, Y. and Contreras-Vidal, J., 2017. Real-time EEG-based brain-computer interface to a virtual avatar enhances cortical involvement in human treadmill walking. *Scientific Reports*, 7(1).

Ma, X., Qiu, S., Du, C., Xing, J. and He, H., 2018, July. Improving EEG-Based Motor Imagery Classification via Spatial and Temporal Recurrent Neural Networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 1903-1906). Honolulu, Hawaii, 17-21 July.

Marchant, B.P., 2003. Time–frequency analysis for biosystems engineering. *Biosystems Engineering*, *85*(3), pp. 261-281.

McMahon, M. and Schukat, M., 2018. A low-cost, open-source, BCI-VR prototype for real-time signal processing of EEG to manipulate 3D VR objects as a form of neurofeedback. *2018 29th Irish Signals and Systems Conference (ISSC)*.

McMullen, D.P., Hotson, G., Katyal, K.D., Wester, B.A., Fifer, M.S., McGee, T.G., Harris, A., Johannes, M.S., Vogelstein, R.J., Ravitz, A.D. and Anderson, W.S., 2013. Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(4), pp. 784-796.

Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, *6*(4), pp. 525-533.

Müller-Gerking, J., Pfurtscheller, G. and Flyvbjerg, H., 1999. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, *110*(5), pp. 787-798.

Naeem, M., Brunner, C., Leeb, R., Graimann, B. and Pfurtscheller, G., 2006. Seperability of four-class motor imagery data using independent components analysis. *Journal of Neural Engineering*, *3*(3), p. 208.

Novi, Q., Guan, C., Dat, T.H. and Xue, P., 2007, May. Sub-band common spatial pattern (SBCSP) for brain-computer interface. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering* (pp. 204-207). Kohala Coast, Hawaii, 2-5 May.

Padfield, N., Zabalza, J., Zhao, H., Masero, V. and Ren, J., 2019. EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges. *Sensors*, 19(6), p.1423.

Pfurtscheller, G. and Aranibar, A., 1979. Evaluation of event-related desynchronization (ERD) preceding and following voluntary self-paced movement. *Electroencephalography and Clinical Neurophysiology*, *46*(2), pp. 138-146.

Pfurtscheller, G. and Da Silva, F.L., 1999. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, *110*(11), pp. 1842-1857.

Pfurtscheller, G. and Neuper, C., 1997. Motor imagery activates primary sensorimotor area in humans. *Neuroscience Letters*, *239*(2-3), pp. 65-68.

Pfurtscheller, G. and Neuper, C., 2001. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, *89*(7), pp. 1123-1134.

Pfurtscheller, G., Kalcher, J., Neuper, C., Flotzinger, D. and Pregenzer, M., 1996. On-line EEG classification during externally-paced hand movements using a neural network-based classifier. *Electroencephalography and Clinical Neurophysiology*, *99*(5), pp. 416-425.

Pfurtscheller, G., Neuper, C., Schlogl, A. and Lugger, K., 1998. Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE transactions on Rehabilitation Engineering*, *6*(3), pp.316-325.

Preece, J., 2002. *Human-Computer Interaction*. Harlow: Addison-Wesley.

Qiu, Y., Zhou, W., Yu, N. and Du, P., 2018. Denoising Sparse Autoencoder-Based Ictal EEG Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(9), pp.1717-1726.

Qiu, Z., Jin, J., Lam, H.K., Zhang, Y., Wang, X. and Cichocki, A., 2016. Improved SFFS method for channel selection in motor imagery based BCI. *Neurocomputing*, *207*, pp. 519-527.

Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W. and Ball, T., 2017. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*(11), pp.5391-5420.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 618-626). Venice, Italy, 22-29 October.

Shahid, S. and Prasad, G., 2011. Bispectrum-based feature extraction technique for devising a practical brain–computer interface. *Journal of Neural engineering*, *8*(2), p.025014.

Shiman, F., Irastorza-Landa, N., Sarasola-Sanz, A., Spüler, M., Birbaumer, N. and Ramos-Murguialday, A., 2015, August. Towards decoding of functional movements from the same limb using EEG. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 1922-1925). Milano, Italy 25-29 August.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–14*. Banff, Canada, 14-16 April.

Soman, S., 2015. High performance EEG signal classification using classifiability and the Twin SVM. *Applied Soft Computing*, *30*, pp.305-318.

Springenberg, J.T., Dosovitskiy, A., Brox, T. and Riedmiller, M., 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Stone, M., 1974. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2), pp. 111-133.

Subasi, A. and Gursoy, M.I., 2010. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert systems with applications*, *37*(12), pp. 8659-8666.

Subasi, A., 2007. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, *32*(4), pp. 1084-1093.

Sutskever, I. and Hinton, G., 2010. Temporal-kernel recurrent neural networks. *Neural Networks*, *23*(2), pp. 239-243.

Tang, X., Zhang, N., Zhou, J. and Liu, Q., 2017. Hidden-layer visible deep stacking network optimized by PSO for motor imagery EEG recognition. *Neurocomputing*, *234*, pp. 1-10.

Tang, Z., Li, C. and Sun, S., 2017. Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik-International Journal for Light and Electron Optics*, *130*, pp. 11-18.

Tavakolian, K., Nasrabadi, A.M. and Rezaei, S., 2004. Selecting better EEG channels for classification of mental tasks. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)* (Vol. 3, pp. III-537). Vancouver, BC, Canada, 23-26 May.

Übeyli, E.D., 2009. Analysis of EEG signals by implementing eigenvector methods/recurrent neural networks. *Digital Signal Processing*, *19*(1), pp.134-143.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, L., Pedregosa, F. and Mueller, A., 2015. Scikit-Learn. GetMobile: Mobile Computing and Communications.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008).

Vijayendra, A., Saksena, S.K., Vishwanath, R.M. and Omkar, S.N., 2018, January. A Performance Study of 14-Channel and 5-Channel EEG Systems for Real-Time Control of Unmanned Aerial Vehicles (UAVs). In *2018 Second IEEE International Conference on Robotic Computing (IRC)* (pp. 183-188). Laguna Hills, CA, USA, 31 Jan – 2 Feb.

Wang, H., Tang, Q. and Zheng, W., 2011. L1-norm-based common spatial patterns. *IEEE Transactions on Biomedical Engineering*, *59*(3), pp. 653-662.

Wang, P., Jiang, A., Liu, X., Shang, J. and Zhang, L., 2018. LSTM-based EEG classification in motor imagery tasks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(11), pp. 2086-2095.

Wang, Y., Zhang, Z., Li, Y., Gao, X., Gao, S. and Yang, F., 2004. BCI competition 2003-data set IV: an algorithm based on CSSD and FDA for classifying single-trial EEG. *IEEE Transactions on Biomedical Engineering*, *51*(6), pp. 1081-1086.

Woo, J.S., Müller, K.R. and Lee, S.W., 2015, January. Classifying directions in continuous arm movement from EEG signals. In *The 3rd International Winter Conference on Brain-Computer Interface* (pp. 1-2). Gangwon Province, Korea, 2-14 Jan.

Yang, B., Duan, K. and Zhang, T., 2016. Removal of EOG artifacts from EEG using a cascade of sparse autoencoder and recursive least squares adaptive filter. *Neurocomputing*, *214*, pp. 1053-1060.

Yang, J., Zhao, Y., Chan, J.C.W. and Yi, C., 2016, July. Hyperspectral image classification using two-channel deep convolutional neural network. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 5079-5082). Beijing, China, 10-15 July.

Yi, W., Qiu, S., Qi, H., Zhang, L., Wan, B. and Ming, D., 2013. EEG feature comparison and classification of simple and compound limb motor imagery. *Journal of Neuroengineering and Rehabilitation*, *10*(1), p. 106.

Yu, Y., Zhou, Z., Yin, E., Jiang, J., Tang, J., Liu, Y. and Hu, D., 2016. Toward brain-actuated car applications: Self-paced control with a motor imagery-based brain-computer interface. *Computers in Biology and Medicine*, *77*, pp. 148-155.

Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* (pp. 818-833). Springer, Cham.

Zhang, D., Yao, L., Chen, K. and Monaghan, J., 2019. A Convolutional Recurrent Attention Model for Subject-Independent EEG Signal Analysis. *IEEE Signal Processing Letters*, *26*(5), pp. 715-719.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A., 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2921-2929). Las Vegas, Nevada, USA, June 26-July 1.

Zubarev, I., Zetter, R., Halme, H.L. and Parkkonen, L., 2019. Adaptive neural network classifier for decoding MEG signals. *NeuroImage*, *197*, pp. 425-434.

Appendix A:

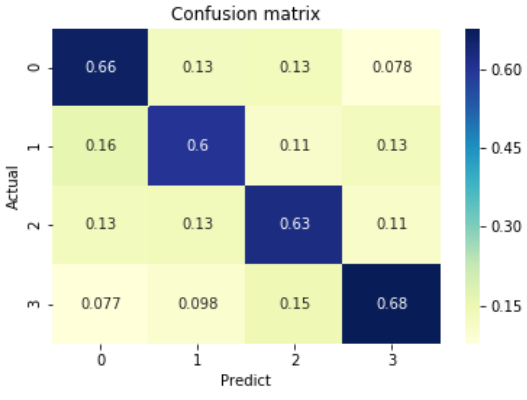
CRNN-DA and Grad-CAM

In Chapter 4, the CRNN-DA was trained with a subset as training (80%) and 20% validation, where the training set was divided into training set and a validation set. The procedure of dividing the dataset ensures that the model performance is consistent to a degree (where the standard deviation is reported). However, the testing set is completely unseen and using the full training set would provide the model with more examples (since EEG datasets are usually small in size as discussed in Chapter 3 and 4). In real life applications, the full training set can be used for training, and the suggested model and training procedure doesn't use an early stopping. Hence, the following is the classification performance using the full training dataset. The best performing architecture was trained with the full training set without a validation set.

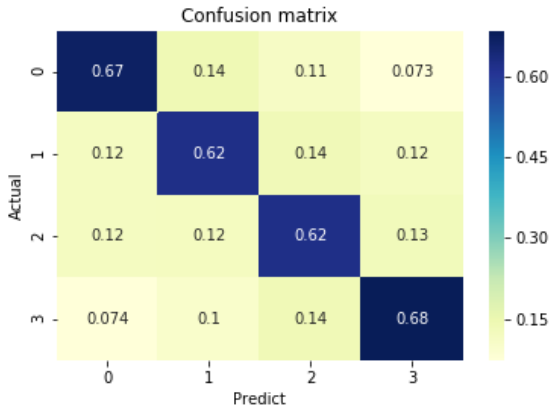
Table A.1: The mean classification accuracies training with the full training set.

	BCI IV 2a
W/S/I	Mean Acc.
400/50/4	**0.74**+- 0.007
200/50/4	**0.72**+- 0.013
400/50/2	0.71+- 0.012
200/50/2	**0.72**+- 0.012

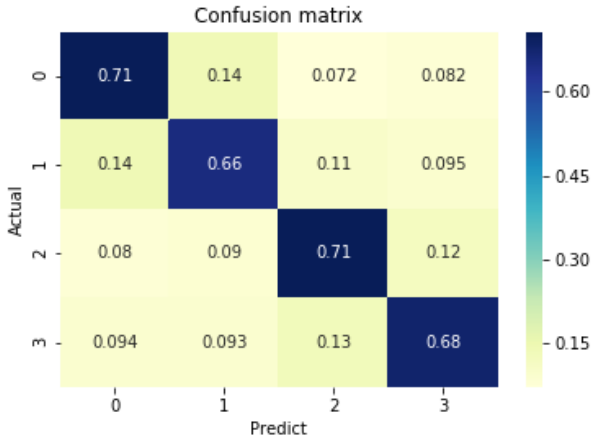
A



B



C



D

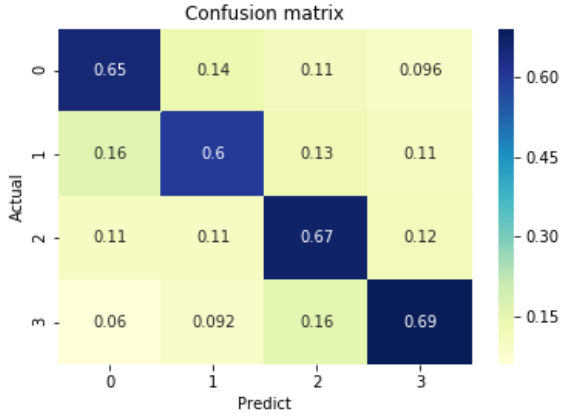
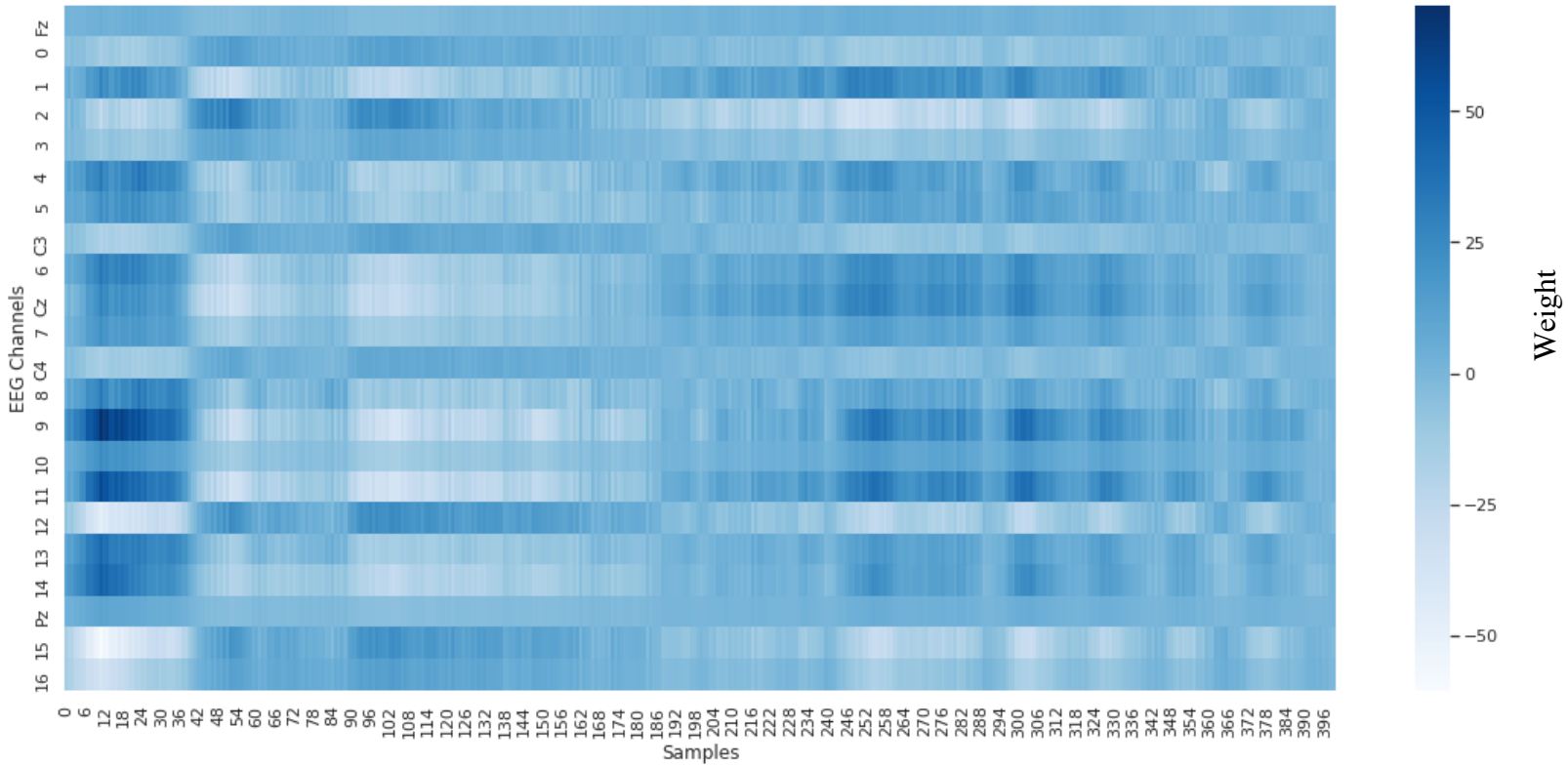


Figure A.1: Confusion matrices. A: 400/50/4 with No-Attention. B: 200/50/4 No-Attention. C: 200/50/4 No Attention- two-two. D: 200/50/4: Attention- two-two.

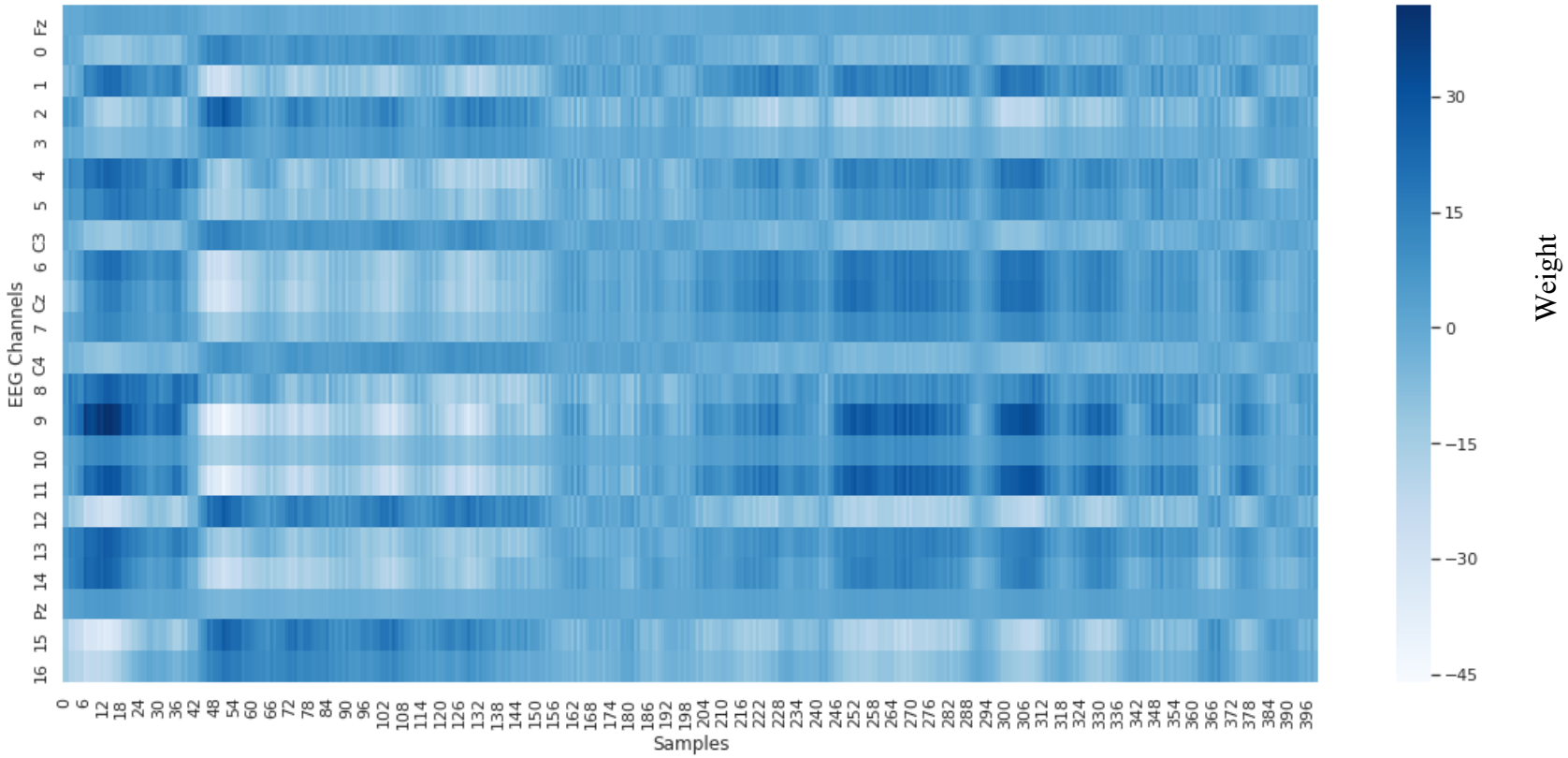
Table A.2: The best performing model with the two-two augmented trials results for each participant against the baseline.

Participant	1	2	3	4	5	6	7	8	9
Two-Two	**0.78**	0.51	**0.85**	**0.68**	0.64	0.57	**0.86**	**0.84**	0.75
Baseline 2s	0.73	0.50	0.75	0.60	0.68	**0.60**	0.69	0.77	0.73
Baseline 4s	0.74	**0.53**	0.82	0.58	**0.71**	0.55	0.71	0.77	**0.79**

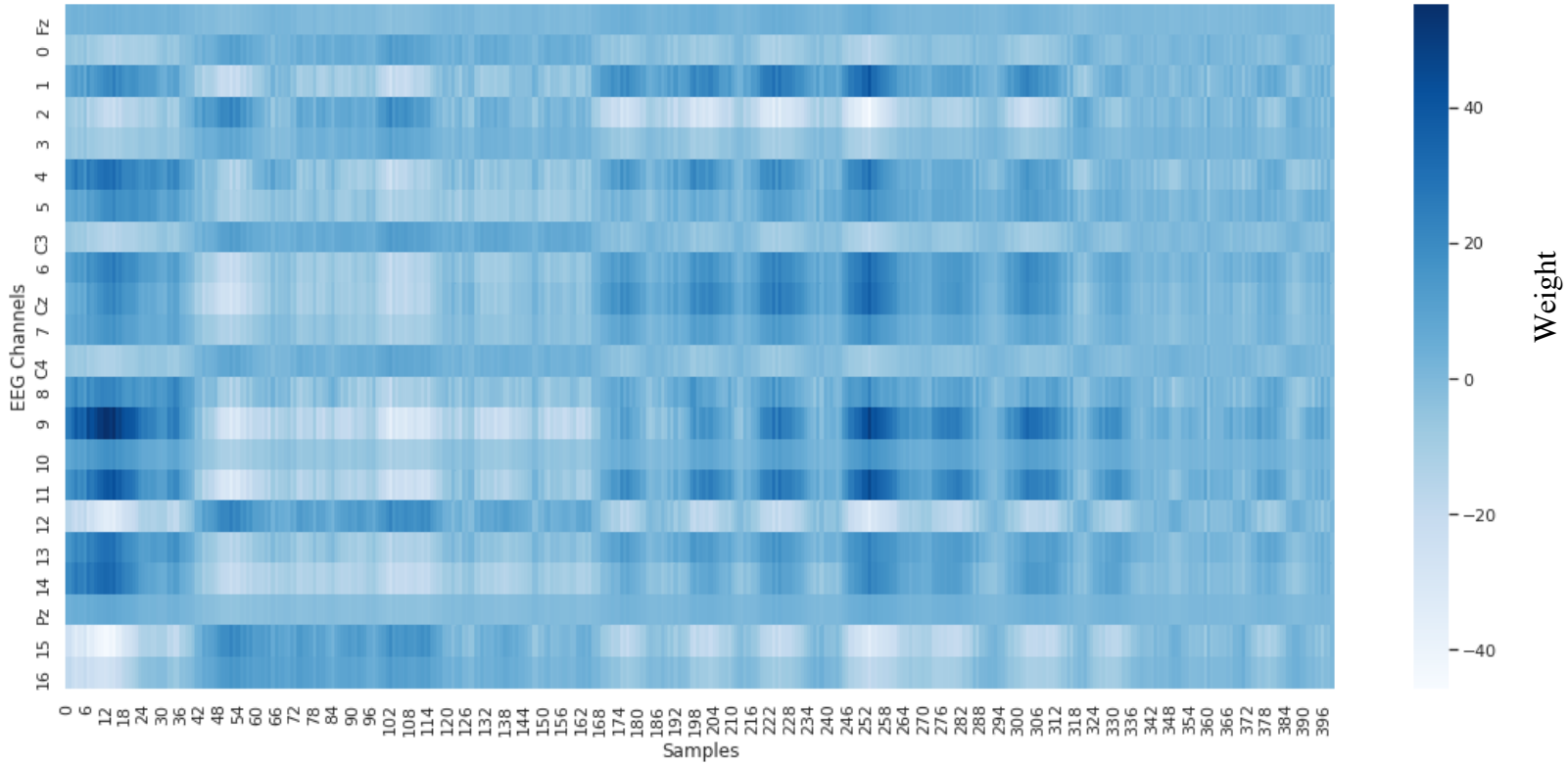
A


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
0
6
12
18
24
30
36
42
48
54
60
66
72
78
84
90
96
102
108
114
120
126
132
138
144
150
156
162
168
174
180
186
192
198
204
210
216
222
228
234
240
246
252
258
264
270
276
282
288
294
300
306
312
318
324
330
336
342
348
354
360
366
372
378
384
390
396
Weight
50
25
0
−25
−50

B


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
0 6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162 168 174 180 186 192 198 204 210 216 222 228 234 240 246 252 258 264 270 276 282 288 294 300 306 312 318 324 330 336 342 348 354 360 366 372 378 384 390 396
Weight
30
15
0
−15
−30
−45

C


EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
Weight
40
20
0
−20
−40

D

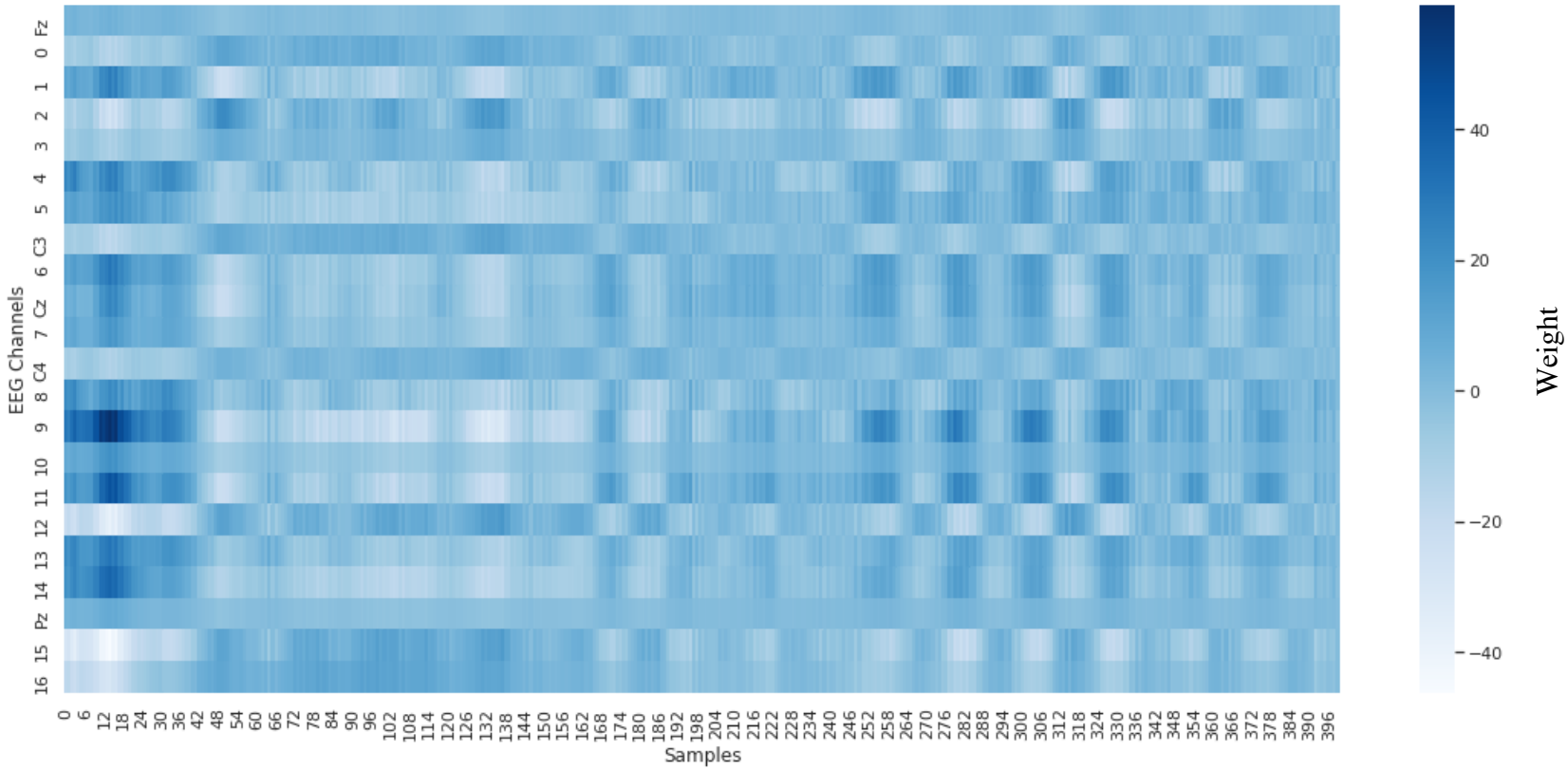

EEG Channels
Fz
0
1
2
3
4
5
C3
6
Cz
7
C4
8
9
10
11
12
13
14
Pz
15
16
Samples
0 6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162 168 174 180 186 192 198 204 210 216 222 228 234 240 246 252 258 264 270 276 282 288 294 300 306 312 318 324 330 336 342 348 354 360 366 372 378 384 390 396
Weight
40
20
0
−20
−40

Figure A.2 An example of the heatmaps of forward activations of participant 3 for the Right Hand class. A. is for the left hand, B. right hand, C. feet and D. tongue.

Appendix B:

Code Snippets

Spatial Attention:

```
from torch import nn



class SELayer(nn.Module): # Squeeze and excitation spatial attention
    def __init__(self, channel, reduction=2, depth=1):
        super(SELayer, self).__init__()

        # Average pooling over the time samples
        self.avg_pool = nn.AvgPool2d((1,200), stride=(1,200))

        self.fc = nn.Sequential(
# Squeezing the channels to channels / 2
            nn.Linear(22, 22//2, bias=False),

            nn.Sigmoid(),
# Exciting the channels back to obtain 22 weights for each channel

            nn.Linear(22 // 2, 22, bias = False),
        )
    # The forward method to pass the input to the layers defined above
    def forward(self, x):
        b, c, h, w = x.size()
        y = self.avg_pool(x)
```

```
        y = y.view(b, h)
        y = self.fc(y)
        y = y.view(b, 1, h, 1)

        return x * y, y
```

Temporal attention

```
import torch.autograd.function
from torch.nn.parameter import Parameter
import math
import torch.nn as nn
import torch
from torch.nn import functional as F
from torch.nn import init

class Attention(nn.Module):

    def __init__(self,hidden_size, attn_size, slices=6):

        super(Attention, self).__init__()
     # getting the spatial average over all the hidden states (time slices)
        self.avg_pool = nn.AdaptiveAvgPool1d((1))
        self.fc = nn.Sequential(
         # Exciting the time slices to an attn size (64 in the study)
```

```
            nn.Linear(slices, attn_size, bias=False),
        # Non linear activation  relue
            nn.ReLU(True),
        # squeezing back to  the number of time slices
            nn.Linear(attn_size, slices, bias = False),
        # Sigmoid function to adjust weights between 0 and 1
            nn.Sigmoid()

        )

    def forward(self, x):
        inp = x
        x = self.avg_pool(x)
        αs = self.fc(x.squeeze(-1))
        output = inp * αs.unsqueeze(-1)

        return output, αs
```

Separable convolution blocks:

```
import numpy as np
import torch
import torch.nn as nn
```

```
import torch.nn.functional as F
from ext_functions import _squeeze_final_output, safe_log, square, conc_augmented
class midBlock(nn.Module):

    def __init__(self, in_channels, output=20, kernel_size=(1,16) ,
                 stride = 1, groups = 1, dilation = 1, padding_type='SAME',pool_size=8, cuda=True):

        super(midBlock, self).__init__()
        self._cuda = cuda
        self.__dict__.update(locals())

        # A convolution block with groups equal to thee number of channels (Depth)
        self.Conv2D = nn.Conv2d(in_channels,in_channels,
                                kernel_size = self.kernel_size,groups=in_channels)
        # Combining the channels with a pointwise convolution
        self.pointwise=nn.Conv2d(self.in_channels,self.output,kernel_size=(1,1), groups=1)
        # Batch normalisation layer
        self.BN = nn.BatchNorm2d(self.output, momentum=0.01, affine=True )
        # Non-linaer activation function leaky-relu
        self.activation =nn.LeakyReLU()
        # The pooling layer (Average)
        self.pool = nn.AvgPool2d(kernel_size = (1,pool_size), stride=(1,pool_size))
        # Drop out with propability 0.5
        self.drop = nn.Dropout(p=0.5)
```

```
def forward(self, x):

    identity = x

    x = self.Conv2D(x)
    x = self.pointwise(x)
    x = self.BN(x)
    x = self.activation(x)
    x = self.pool(x)
    x = self.drop(x)

    return x

@staticmethod
def _get_padding(padding_type, kernel_size):
    #assert isinstance(kernel_size, int)
    assert padding_type in ['SAME', 'VALID']
    if padding_type == 'SAME':
        return (kernel_size - 1) // 2

    return 0
@staticmethod
def _calculate_output(H,padding,dilation, kernel_size, stride):

    numerator = (H + 2*padding-dilation * (kernel_size -1) - 1 )
```

```
        denominator = stride
        H_out = (numerator/denominator) + 1

        return H_out
    @staticmethod
    def _calculate_strided_padding(W, F, S):
         #  W= Input Size , F = filter size (kernel), S = stride,
        P = ((S-1)*W-S+F)//2

        return P
    @staticmethod
    def get_dilated_kernel(k,d):
        #kernel size, dilation
        new_k = k+(k-1)*(d-1)

        return new_k
```

Main Model:

```
import torch.autograd.function
from torch.nn.parameter import Parameter
import math
import torch.nn as nn
import torch
from torch.nn import functional as F
```

```
from torch.nn import init
from SelfAttn import Encoder, Attention, Classifier
from Temporal_Attn import Attention
from SqueezeAndExcitation import SELayer
from midBlock import midBlock
from CW_attn import CW_Attention

class convLayer(nn.Module): # A normal convolutional block

    def __init__(self, cin, cout, kernel_size, dense = False):
        super(convLayer,self).__init__()
        # A 2d convolutional layer
        self.conv = nn.Conv2d(cin,cout,kernel_size, bias = True, padding = (0, kernel_size[1]//2))
        # Batch normalisation layer
        self.bn = nn.BatchNorm2d(cout,momentum=0.993, eps=1e-5)
        # Non-linear activation function leaeky relue
        self.act = nn.LeakyReLU(True)

    def forward(self,x):
        identity = x
        x = self.conv(x)
        x = self.bn(x)
        x = self.act(x)
```

```
        return x

class convrnn(nn.Module): # The full model definition
# This definition doesnt include the spatial attentionself.
# The softmax function is applied in the loss function.
    def __init__(self, classes = 2, inchans = 3, input_size = None, attention=True):
        super(convrnn, self).__init__()
        kernel_1st      = (1,3)

        self.slices = input_size[1] # The number of time slices

        self.latent = 520 if input_size[-2] == 400 else 240
        self.attention = attention # If we want to use spatial attention

        self.conv_3 = convLayer(1,1,kernel_1st) # The block1_1

        self.conv1_1 = convLayer(1,40,(inchans,1)) # Normal conv block
        self.pool0 = nn.AvgPool2d((1,3),stride=(1,3)) # Averagy pooling
        # Seperable convolution blocks
        self.conv2 = midBlock(in_channels=40, output=60, kernel_size =(1,3),pool_size=3)

        self.conv3 = midBlock(in_channels=60, output=40, kernel_size =(1,3),pool_size=3)
        # The GRU layer with a dropout of 0.5
        self.rnn = nn.GRU(self.latent, 64, 2,bidirectional= False, batch_first = True, dropout= 0.5)
        # The temporal attention layer
```

```
        self.attn = Attention(64 ,64, slices = self.slices)

        self.drop1 = nn.Dropout(p=0.5)
        # Fully connectted layer
        self.Linear = nn.Linear(64 * self.slices ,classes)

    def forward(self,x):

        x = x.permute(0, 1,4, 2, 3)
        x = x.reshape(x.size(0)*self.slices,x.size(2),x.size(3),x.size(4))
        x = self.conv_3(x)
        x = self.conv1_1(x)
        x = self.pool0(x)
        x = self.conv2(x)
        x = self.conv3(x)
        x = x.reshape(-1,self.slices, x.size(1)*x.size(2)*x.size(3))
        output, states = self.rnn(x)
        attention, αs = self.attn(output)
        attention = self.drop1(attention)
        attention = attention.view(attention.size(0),-1)
        output = self.Linear(attention)
        return output, αs, 0
```

```
Guided-Grad CAM
```

```
"""
```

```
Created on Thu Oct 26 11:06:51 2017
@author: Ahmed Selim - github.com/w1ndsurf3r
"""
## INspire from https://github.com/jacobgil/pytorch-grad-cam/blob/master/grad-cam.py

import numpy as np
import torch

from torch.autograd import Variable
import torch.nn as nn
import torch.nn.functional as F

def class_subset(class_, X, y):
    return X[np.where(y== int(class_))[0]]

class Gradients(): # The function is to add hooks on feed forward to save the gradients

    def __init__(self, model,target_layers):

        self.model = model
        self.gradients = []
        self.target_layers= target_layers
        self.tuples = ['demix','gru','attn'] ## The name of the layer which return tuples of outputs
        self.outputs = []  # saving the outputs of each layer
```

```
def save_gradient(self,grad):

    self.gradients.append(grad)

    return grad

def __call__(self, x): # Just for the ease of calling the function later. Full forward on the model

    for name, module in self.model._modules.items():

        if name in self.tuples:
            if name == 'demix':
                x_attn = x.squeeze(1)
                _,spat_attn= module(x_attn)
                x = x * spat_attn.view(-1, 1, 22, 1)
            if name == 'gru':
                x = x.reshape(-1,2, x.size(1)*x.size(2)*x.size(3))
                x,_ = module(x)
            if name == 'attn':
                x,_ = module(x)
        elif name not in self.tuples:
            x = module(x)
        if name in self.target_layers: # If the layer specified save the gradients
            print('registered')
            x.register_hook(self.save_gradient)
```

```
                self.outputs += [x]

        return self.outputs, x

class Activations():## To be implemented. Only useful if your model has no classifier
    def __init__(self):
        print('To be implemented')

class GradCam():

    def __init__(self, model, target_layer_names, use_cuda=False):

        self.model = model
        self.model.eval() # Don't update the parameters and run in eval mode
        self.cuda = use_cuda
        if self.cuda:
            self.model = model.cuda() # Using the GPUs
        self.extractor = Gradients(self.model, target_layer_names)

    def forward(self, input):
        return self.model(input)

    def __call__(self, input, index = 0):

        features, output = self.extractor(input)
```

```
self.model.zero_grad()
one_hot_output = torch.FloatTensor(input.shape[0]//2, 4).zero_()
# Activates the class we are interested in (left hand,right hand, tongue, feet) at the last layer
one_hot_output[0][index] = 1
output.backward(gradient=one_hot_output, retain_graph=True)
gradients = self.extractor.gradients
outputs = self.extractor.outputs
grads_val = gradients[-1] # For ease, returning the last gradients from the list
outs = outputs[-1]
weights = torch.sum(grads_val, dim = (3))[0, 0,:] #Averaging the gradieents over the temporal dimension
target = outs # The output of the convolutional layers
target = target.sum(dim=0)[0, :]
cam = torch.zeros(target.shape, dtype = torch.float32)

for i, w in enumerate(weights):

    cam[i]= w * target[i, :] # Multiply the weights with the output of the layers for visualisation
# Applying relu to improve the resolution and emphasize the prominent features
cam = F.relu(cam).cpu().data.numpy()
outs = F.relu(outs)
#  Normalizing
cam = cam - np.min(cam)
cam = cam / np.max(cam)
#print(cam.shape)
return cam , outs.sum(dim=0)[0, :].cpu().data.numpy()#.sum(axis=1)
```