Locating and Identifying Blink and Muscle Artefacts in Magnetoencephalography Recordings

Ethan (James) Jolly, BSc (Hons)
Lancaster University

A dissertation submitted for the degree of
MSc Data Science

September, 2022

Declaration

I declare that the work presented in this dissertation is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work. I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Ethan (James) Jolly

Locating and Identifying Blink and Muscle Artefacts in Magnetoencephalography Recordings

Ethan (James) Jolly, BSc (Hons).
Lancaster University

A dissertation submitted for the degree of *MSc Data Science*. September, 2022

Abstract

Magnetoencephalography is a functional neuroimaging modality which is used to analyse the brain health of subjects who receive the scans. These scans are recorded by measuring the magnetic field activity coming from different regions of the brain. Artefacts can appear in these recordings in ways which distort the information, which can lead to incorrect analysis later in the pipeline, which then itself can lead to misdiagnosis of the subject. In this project, we investigate and propose procedures which we developed to locate and identify two types of common artefacts, blink and muscle artefacts. The methods which we'll show and test start from thresholding methods where a threshold for deciding if a point is an artefact is optimised. Then we move from attempting to locate the artefacts point-wise to a more temporal region-based windowing of the recording, using distinct feature extraction methods for windows depending on the artefact type, then testing using classifiers. Finally, we evaluate parametrisations of the Long short-term memory (LSTM) deep learning classifier. We show that with our procedures we achieve 87% accuracy for the internal validation data and 98% for the external validation for classifying blink artefact locations, then 99% and 98% accuracy for internal and external respectively for classifying muscle artefact locations, where the external data are recordings which are from full-length subject recordings from subjects which we did not use in any of the training data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Functional neuroimaging modalities are tools which are used to assist in analysing the brain activity of different subjects. One of the most common neuroimaging modalities which is used for this analysis is Electroencephalogram (EEG) which is used to analyse brain health and can be used to help diagnosis of different illnesses, such as epilepsy as shown by V. Srinivasan, Eswaran, and Sriraam (2007). A key limitation of EEG is that, due to its use of electrical signals to record activity, spatial quality is limited as electrical signals struggle to pass through the subject's skull, R. Srinivasan (1999). In this project, we will investigate the neuroimaging modality Magnetoencephalography (MEG), which solves these limitations since MEG uses magnetic signals to record the activity, which has much less trouble passing through the skull, providing much higher spatial and temporal resolution to the recordings. So the key benefit of MEG is higher quality recordings, which could allow for more detailed analysis, it is of great interest to keep the recorded signals clean, where the only data which would be analysed would be actual brain activity. Artefacts, which are caused by a source from outside the subject's brain, are what can cause the data to no longer be as clean as would be ideal, and therefore less useful for its applications as artefacts tend to appear in high-frequency bands which is where analysis associated with seizures is done Van Dellen et al. (2014), so if an artefact is left in the recording and inflates the level of the signal at high frequencies, this could lead subjects to provide incorrect information to medical professionals and even lead to misdiagnosis. Artefacts which can occur naturally, from any subject and will have an impact on the recording are blink and muscle artefacts, where blink artefacts come from natural eye blinks and muscle artefacts can come from any muscle tension in muscle groups around the head e.g jaw clenches or neck tension. In this paper, we choose to locate and identify these different artefacts in the recordings to provide more information on the structure of these artefacts, how often they are likely to occur and what characteristics they present, with an end option being to remove the identified locations of artefacts for a

cleaner signal.

1.1 Magnetoencephalography

Magnetoencephalography (MEG) is a type of functional neuroimaging which records the activity of the brain. This is different from other modes of neuroimaging, for example, MRI, as they record the structure of the brain, not the activity. The activity is measured based on the strength of the magnetic field detected from the brain. These magnetic fields are recorded using superconducting quantum interference devices (SQUIDs), these are a type of magnetic sensor which are used to build up the sensor arrays which are used in the majority of MEG scanners, Clarke (1994). SQUIDs are very sensitive to magnetic field activity, and because of this as the name suggests SQUIDs require super-cooling from lots of liquid helium, this requirement causes the scanners to be immobile. In the sensor array of SQUIDs, they are used as both magnetometers and gradiometers, where magnetometers are sensors which measure the magnetic field and gradiometers are sensors which measure the gradient of the magnetic field which has been detected, Wheless et al. (2004). The way MYndspan use this MEG data is by analysing the levels of activity in certain frequency ranges to determine a level of brain health, they use this analysis to build up information over multiple different scans to give a longitudinal insight into any changes to the health of the brain of a subject. The way that we interact with the data that these sensors record is via the form of multi-channel time series, where each channel is corresponding to a spatial sensor in the sensor array, as opposed to a frequency channel.

We can see from Figure 1.1, an example of what the time series data which we will be using in this project looks like in waveforms. Looking at the figure, on the left-hand side, we can see names, such as MEG0111, MEG0121 etc. These correspond to the names of different individual sensors, each recording the magnetic field from different spatial locations around the brain. Looking on the right-hand side of the figure we can see a vertical scroll bar, this would be used to scroll down and view the rest of the sensors in the sensor array, as from this view we are showing the first 20 channels when the full sensor array for this recording is 102. Then for the last part of the figure, the horizontal scrolling bar at the bottom of the figure is used to be able to scroll across the time of the recording, as in this image we are showing the data for the first 10 seconds, but the whole recording is for 600 seconds, which would not be possible to show reasonably in a single image.

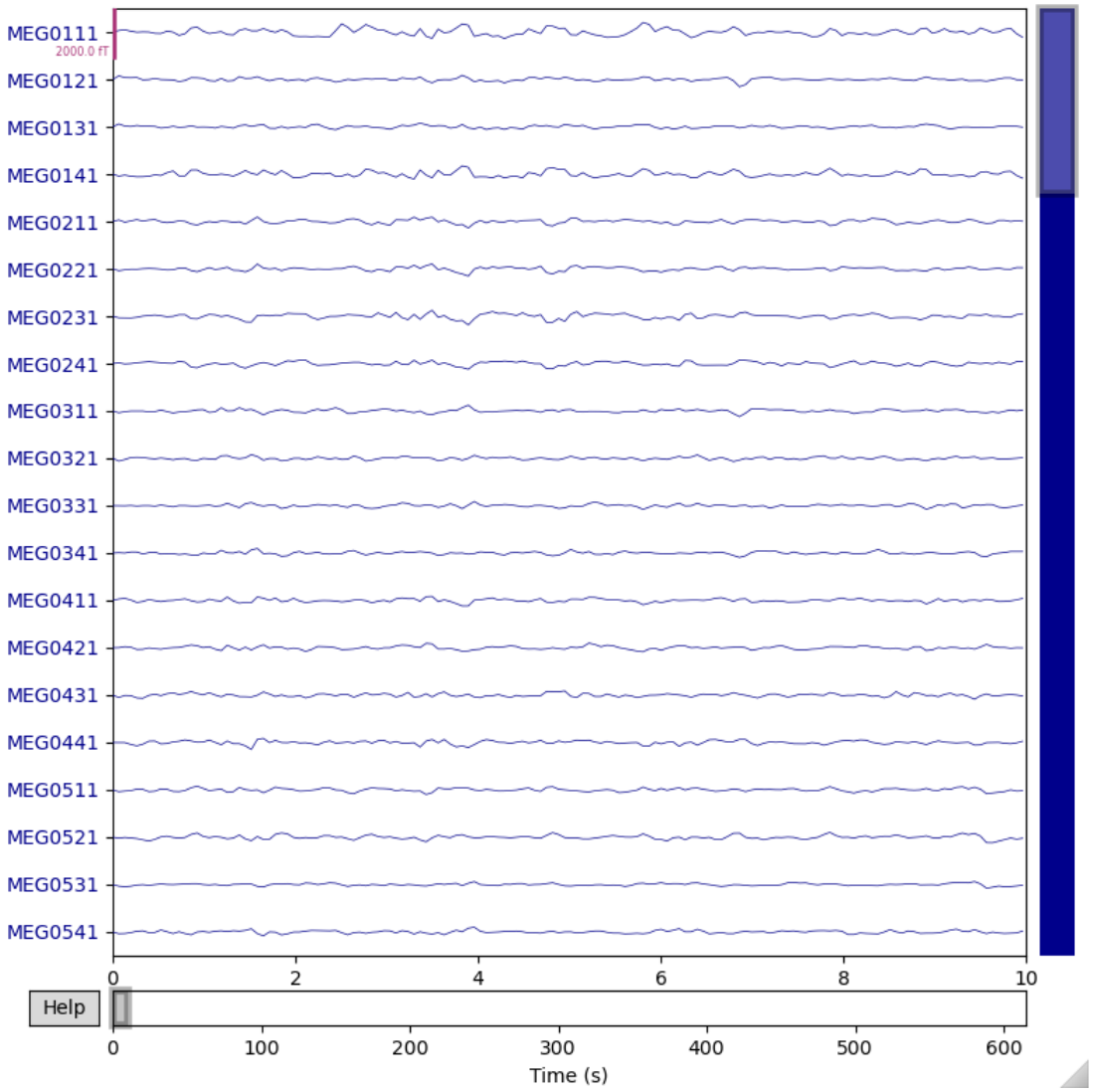


Figure 1.1: An example of the time series data from a MEG recording.

1.2 Project Aim

This project aims to create a robust well-evaluated procedure for locating and identifying both blink and muscle artefacts from the time series of subject recordings from MEG scans. In the pursuit of creating this procedure, we will look at related work from the field of neuroimaging to understand which methods are commonly applied to this area, and what methods to handle artefacts have been proposed before. We will create a variety of different methods, going from intuitive and more basic methods to more sophisticated approaches. We will do this to better understand the level

of approach which is required to solve the problem. For these methods, we will evaluate their performance using standard classification accuracy metrics, and using these compare against alternatives as well as possible changes in parameters for further optimisation. Then a further additional aim is to look at classifying individual source components as containing artefacts, as this is quite a popular technique used in the literature to handle artefacts for removal.

1.3 Objectives

1.3.1 Locate and Identify Blink and Muscle Artefacts

Create and evolve multiple methods of locating blink and muscle artefacts temporally in the time series. Do this by starting with the most intuitive and explainable methods, such as methods which utilise thresholding of the time series to label above the threshold as an artefact as well as methods which use correlation across sensors to see if there is a link during artefacts. Then moving on to using standard linear classifiers as a more robust approach, which should generalise across different subjects better than the starting methods. Also looking at ways to condense the input data, via feature extraction, for these methods to allow for easier interpretation from the models. Then moving on to more sophisticated approaches via deep learning from long short-term memory, which we would expect to be able to generalise even more than the standard machine learning classifiers, as they are well known to be very good at extracting training parameters from time series data.

1.3.2 Evaluate and Optimise the Best Approach

The other key objective of the project is to make sure we correctly evaluate and optimise the best approach, from the different methods and versions of methods which we will be creating. Since the methods which we will be creating will be methods of binary classification (artefact or not, given data), we will be using metrics to get results for these different approaches and use them as a comparison to optimise and find the best method, with one of the key optimisations that we will be interested in being for the long short term memory parameters. We will also evaluate the possible trade-offs we will trade to make, from using the results themselves then also via visualisation of the results for further perspective.

1.3.3 Classify Artefact Sources

The final objective we wish to look at is using a method of blind source separation to create independent sources from the recording, from which we can identify sources

which contain artefacts. We seek this as an additional objective as although we would not be locating the presence of artefacts temporally or spatially, being able to identify sources which contain artefacts allows for a method of removal which will leave more of the recording which can be considered clean than if you were to remove the located artefacts via the other methods. So even though the main aim of this project is not to find the best removal method we still think it is worth looking at this as it is a pretty useful method. We will be evaluating the performance by comparing two different types of deep learning classifiers on the inputs.

1.4 Current Approach

The current approach used by MYndspan for the aims of the project is that they currently do not have anything which locates artefacts in the same way which we aim to do. Current methods which they use, which are somewhat related are the use of a beamformer. For more information on how the beamformer works, see Litvak et al. (2010), but the general idea of beamforming is that it is used to spatially locate the sources of different signals in the brain. So in theory, if a signal is localised to a spatial region which is outside of the brain it can be set as an artefact and removed. The downsides to this method are that it works very much as a 'black box' approach as it is a very complicated approach and can be difficult to know if it is working correctly, this mainly is because the method itself is not designed for artefact removal or detection rather only source space calculation.

1.5 Motivation

The motivation for the research conducted for this project and the aims for which we have set out are first for the fact that providing a procedure for locating blink and muscle artefacts will provide a way of being able to better understand these artefacts, how frequent they are, how they can change from person to person and how they can appear temporally, spatially and spectrally. Another motivating idea is that providing these procedures will allow for if the decision is made to remove located blink and muscle artefacts, more knowledge will be available about what exactly is being removed, and how that will affect the resulting recording. This is in contrast to the current methods as described in the current approach section which provides little insight into what is being removed.

1.6 Report Overview

The rest of this report is organised as follows. In Chapter 2, we will explore and demonstrate our findings from the literature and works done in the related fields to this project. Following this, we will move into Chapter 3, which is where we will cover many of the definitions which will be used to help provide an understanding of the later explained methodologies and procedures, which will be taken through in Chapter 4. Where we will take an in-depth look at the different methods and procedures which we looked at during the project, and those which we will be taking a closer look at in testing. The results of testing these methods will be shown in the following chapter, Chapter 5, where we will present a qualitative and quantitative analysis of the findings. In Chapter 6, we will be discussing these results, what they mean and what implications they have for the aims we set out. Then finally in Chapter 7, we will provide our conclusions from what we set out to do and look at what could be done in future work.

Chapter 2

Related Work

For our look into the related work of this dissertation, we start with looking at related works which use more rule-based approaches to identify artefacts, whether this is using cross-correlation methods or thresholding approaches. We also consider approaches which utilise statistical approaches via wavelets or empirical mode decomposition, we will not be using these methods ourselves within our methodology as they are more focused on the removal of artefacts rather than the location or identification. The next grouping of related works that will be looked at will be approaches which attempt to create structured and well-reasoned feature vectors, usually comprising of statistical features from the recordings. Then using these feature vectors, linear-based classifiers are used as prediction models for classifying the artefacts. Moving on, we finally look at approaches which look at using deep learning models to learn their features, based on representations of the recording itself. Where we will be looking at convolutional Neural networks and long-short term memory recurrent neural network approaches. A common theme to make note of which appears in many of the related works is the use of independent component analysis (ICA) or some other blind source separation, this is because much of the related work in this field around artefacts focuses on artefact removal, which ICA is very good for, however, this is not a key aim of this dissertation so we will not be focusing on it for our methods. Also, something to note is that throughout this look at related work we will be looking at work from the application of both Magnetoencephalography (MEG) and electroencephalogram (EEG), this is due to the similarities in the way these applications work the data which is recorded is similar to the point that many of the same approaches can be transferred between them.

2.1 Rule-Based or Statistical Methods of identification

The first paper which we will look at for this section from Liu et al. (2010), has the aim to use methods of correlation, and coherence analysis to characterise the power modulations in subjects for different regions of the brain during different scenarios, such as eyes open, eyes closed and light sleep. Their pre-processing methodology for removing artefacts involved first band-pass filtering between 0.02 and 70Hz, then a notch filter to remove the power line frequency. The independent component analysis was used to decompose the time series into multiple components, then they used a positive peak autocorrelation rule-based method to detect artefacts using a threshold coefficient and time lag coefficient, so if an independent component has peaks above the threshold in line with the time lag, then it was labelled as an artefact component. After detection, a further manual visual inspection check was done to confirm the labelling. The methodology for analysis of spectral correlation, makes use of splitting the time series into multiple frequency band groups, then uses Pearson's linear correlation to create correlation matrices for each frequency band group, and then also uses a coherence metric to create coherence matrices. The results from these correlation and coherence matrices show that for the sub-gamma frequency bands spontaneous power modulations synchronize over spatial, regions with the note that for the gamma band the strength of the synchronization is still apparent but much weaker. These conclusions seem reasonable based on the resultant figures.

The aim of this next paper from Okada, Jung, and Kobayashi (2007) was to propose a method which can be used to identify and remove eye-blink artefacts from MEG signal data, as they are considering eye blinks as being inevitable and they can appear as large magnetic deflections in the time series. Their methodology for identifying the eye blinks starts with them applying principal component analysis (PCA) to decompose the channels of the scan into orthogonal components, this is an appropriate step before applying ICA next as it is known to help prevent estimation error. So independent components from the 80% explainable principal components are used. For the next step of identifying the component containing blinks, they define a cost function which is used as essentially a similarity measure to compare what the signal looks like at points where an EOG identifies a blink and each independent component, to see which component is most similar to the EOG signal, where EOG is a separate sensor which specifically is used to record when in time blinks occur. The next step for them is to validate the chosen components before removal, this process is normally done manually and can be very time-consuming, so they used an automatic rule-based correlation method, where they calculate the correlation between the EOG waveform and the identified component, if the correlation is greater than 0.7 then it

is removed. This threshold method of deciding to remove components seems like it could fall into issues when tested on multiple different subjects. For the results, their method does show that comparing the independent components to an EOG signal can identify components containing the blinks.

For this next paper, from Ferrante et al. (2022), the aim was to propose a more standardised pipeline for analysing MEG signal data, where the focus is on quantifying and localising brain activity but can be used for more multivariate brain analysis. The methodologies which they describe as being a part of the pipeline are, for when data is collected from the scans, they recommend that EOG and ECG sensors are used in parallel to capture blink and heartbeat artefacts easily, to later allow for identification and removal via the use of independent component analysis. An additional step to provide additional confirmation of eye artefacts that they suggest is to use a fast eye tracker, which would be able to detect blinks as well as saccadic movements, this can be used to confirm the results of the EOG. For the actual pre-processing annotation of artefacts, they suggest using a high-frequency band range of 110 – 140 Hz for muscle artefacts, and 1 – 10 Hz for eye artefacts. For identification in these ranges, they suggest the use of a threshold-based rejection method quantified on the z-score of the signal, so when the z-score reaches a threshold annotate a possible artefact. For further removal, they recommend the use of finding ICA components, and then manual inspecting them visually.

This next paper, from Kumar et al. (2008), is the first which we will include which uses a statistical method, the way they use this method is to use it to identify and remove eye artefacts from EEG recordings with the use of wavelet transforms and without the use of a separate EOG reference channel. The methodology that they propose starts with identifying the spikes in the recording, this is done by decomposing the EEG signal with a Symlet wavelet up to eight levels, which provides coefficients for the spikes. Then using these spike zone coefficients, a coefficient of variation is calculated, and the sections of the recording with the largest coefficient of variation are selected as containing eye artefacts. For removal of the identified artefacts, they use a threshold function method to determine what wavelet coefficient value to use in the wavelet reconstruction. For the results of this method, they test using multiple data sets, and they compare against other methods. They do show that their method can identify and remove blink artefacts from their data better than the compared methods. They also make sure to make clear that this removal procedure is only performed on the identified blink regions and does not affect low-frequency components and preserves the shape of the signal, to maintain clinical viability.

The aim of this next paper from Molla et al. (2012) is to propose a method

using adaptive time filtering via empirical mode decomposition (EMD) to identify and suppress artefacts in EEG recordings. EMD is applied to produce several band-limited signals called intrinsic mode functions (IMFs), where the energy of the IMFs is combined with a threshold for suppressing artefacts. The EMD algorithm used for the method works by decomposing the signal into IMFs where each IMF must satisfy that the number of extrema is the same as the number of zero crossings or differ at most by one, and the mean value of the envelope is defined by maxima and minima is zero. For an input signal, all local minima and maxima are calculated and connected, upper and lower envelopes are calculated, the mean is calculated, and then it is decided to be an IMF if it fits the stated conditions. Once all IMF components are found, the artefacts can be reflected and then removed. In the results for this paper, they compare their EMD method against a wavelet transform method and tested using both synthetic and real recorded signals, the results show that this method can identify and suppress components of a signal which contain artefact-induced noise.

2.2 Feature based Classifier Identification

The first paper using a calculated feature vector for a classifier which will be looking at is from Shoker, Sanei, and Chambers (2005) where they propose a combined method of using features from independent components a blind source separation algorithm to train a support vector machine (SVM) classifier to identify artefacts in EEG recordings. The methodology they use for the blind source separation is to use ICA for the components, with these components, they calculate four features to train the classifier. These are ratio statistics between the peak amplitude and variance of the signal, the absolute value of the skewness of the signal, as a signal containing artefacts, tends to be more skewed. The correlation between the component of interest and a different component which is known to contain an artefact. Then finally the statistical difference between the pdf of the component of interest and the component known to contain an artefact, for this they used the Kullback-Leibler distance metric. As mentioned, they use an SVM as the classifier, they chose SVM due to its generalisation performance, and they test using RBF, cubic and linear SVMs with the linear SVM providing the highest accuracy. Through the results of testing, via four-fold cross-validation on the data sets, they do clearly show that their method can successfully identify artefacts in EEG signals.

In this next paper from Yuan, Nathan, and Jafari (2016), they aim to propose a method for automatic identification of artefacts in EEG recordings, by isolating independent components of the signal via ICA, then calculating features based on these components to cluster the data using hierarchical clustering. The first step for

the methodology we to band-pass filter the recordings to between 0.5 – 50 Hz, then perform ICA to decompose the recording into independent components. Using these components, they extracted features which they believe will be able to differentiate eye and muscle movements from raw brain activity. The first of these features is to calculate the kurtosis of the components, as components with artefacts will have high kurtosis. The next is a spatial feature, which is the median column of the topographical weight matrix calculated for the component. The next feature is a spectral feature which is the product of the average power from each frequency band used for analysis i.e delta, theta, alpha, beta, and gamma. Then the final feature is a similarity over epochs of the component, as since artefacts can appear randomly, it is expected that a component containing artefacts will have a lower similarity across the epochs. As mentioned to distinguish between components of interest and the rest, hierarchical clustering is used, the reasoning they give is that the produced dendrogram not only groups for clusters, but it also provides information within cluster separation. From the results, they do show that the method can identify components containing artefacts, and they test using accuracy comparison against alternate methods, with statistical t-tests to measure the significance of their method.

For the next paper, we will look at from Winkler, Haufe, and Tangermann (2011), they aim to propose a method of automatic classification of independent components from EEG recordings containing artefacts. The methodology that they use starts with PCA to reduce the dimensionality of the data from 121 channels down to 30 channels, then they use ICA to get 30 independent components from the PCA components. For the training of the classifier, 23 EEG recordings were used, leading to 690 labelled components extracted from the recordings. From each component, they calculated 38 different features to describe the component for the classifier. Some of these features were the variance of the component, the maximum amplitude, the range of the amplitude, the max first derivative, the kurtosis, Shannon entropy, deterministic entropy, local variance of selected time intervals, mean local variance and skewness, the spatial distance between extrema. For the classifier, they choose to use a Linear programming machine (LPM), which like all binary linear classifiers finds a linear separating hyperplane. They train the classifier using cross-validation and compare the results against an SVM with a Gaussian kernel and an LDA, the metric they used was mean squared error, and the results were that their method showed an 8.9% error with the other methods having a 9.5% error, so it does show that their method can identify artefact components with a slightly better error than comparative methods.

2.3 Feature-less Classifier Identification

2.3.1 CNN Approaches

The first paper which we will be looking at which uses a method of a feature-less classifier is from Garg et al. (n.d.), where they propose a method for automatic artefact detection in MEG data using a convolutional neural network (CNN), without the use of additional EOG or ECG sensors to provide additional artefact time information. The methodology starts with pre-processing the data by bandpass filtering to between 1 – 100 Hz decomposing the recording into 20 independent components using ICA, then these components were each labelled by an expert with domain experience. For the CNN they decided to use a one-dimensional input size, which represents 40 seconds of the time series of the component, the CNN was then trained using a leave-one-out cross-validation method on 19 subjects, with 30 subjects left for testing. The structure of the CNN was to have 5 convolutional layers each followed by a max pooling layer with a single fully connected layer at the end. The results of this methodology in the tests they conducted gave 96% sensitivity and 99% specificity for a model trained to only recognise heartbeat artefacts, then 85% sensitivity and 97% specificity. These results do seem to confirm their conclusions of proposing a method which can detect artefacts without additional sensor help.

The next paper we will look at is from Treacher et al. (2021) where they propose a method to automatically detect and classify independent components containing artefacts, using a combination of two trained convolutional neural networks. The methodology starts with pre-processing the recording first with a band pass filter at between 1 – 100 Hz and a notch filter at 60 Hz to suppress power line noise. Then ICA is performed to decompose the recording into 20 independent components, these components from all the data sets were all hand labelled by a group of four experts with domain experience. The two different CNNs are trained on the independent components, however, one is trained using temporal information for the time course itself, then the other CNN is trained using a spatial topographic transformation of the independent components. The results of these two CNNs are then passed into a single fully connected neural network to classify the component. For the results of the tests run on the methodology, they use a metric of Fleiss' kappa, which measures the degree of agreement between the network and the experts who labelled the data. The method achieves above 90% classification on all artefacts which were being looked at, they also compare against the spatial and temporal networks independently to show that the combined results in better performance. The results do confirm the conclusions of being able to automatically detect artefacts in MEG data and achieve expert-level performance.

For this next paper we look at Croce et al. (2019) where they propose an improved method of classification for the standard independent component analysis to identify components with artefacts via visual inspection from experts for both MEG and EEG recordings, their improved method would be to use a bank of a few thousand independent components from recordings, train a Convolutional Neural Network (CNN) on these independent components to provide an automatic classification of components containing artefacts. Before independent component analysis pre-processing methods are applied to both EEG and MEG signals of filtering between respective frequency ranges as well as notch filtering out the power line frequency, also the topographic representations of components were standardised by z-core computation before being input to the CNN. For the architecture of the CNN, they used 3 convolutional layers with respective pooling layers, followed by a fully connected layer, in the convolutional layers they used 4,8 and 16 filters of size 5 and stride 1. For the results, the accuracy and cross-entropy metrics to measure the performance of the CNN over the epochs. For comparing the results, they used LDA, SVM and a single hidden layer standard ANN as other classifiers trained on independent components, where the CNN outperformed the other models on both EEG and MEG. The conclusions in this paper tell us that this CNN performance provides beyond state-of-the-art performance at automatically identifying components containing artefacts, which seems reasonable based on the results and metrics used.

This next paper is from Feng et al. (2021) and they propose a method of identifying blink artefacts in MEG signal data by using sensor information transformed into a topographical plot of the head, showing magnetic field strength, then feeding these images of sensor information into a CNN model to be able to classify when a plot looks like it contains blink artefacts. For this method, they use 8 different data sets to train, and 4 more data sets to test, each with a few thousand labelled blink artefacts. The labelling process was done by using an EOG sensor which marks clear spikes in time when a blink occurs during the scan. A processing step for the data is to go from the single-dimensional vector of the size of the number of sensors, and then transform this into a topographical image representation. For their CNN, they use transfer learning of the GoogLeNet model, for their model which will provide stronger performance without as much training. For the results, they use a single slightly modified accuracy metric, due to the fact of not being able to have true negative predictions, on the four different test data sets their model achieves between 80% - 100% accuracy on all of them, however, they provide no comparison used other metrics or alternate methods. From the results, it does seem like they do reach their conclusion of being able to successfully recognise blink artefacts.

2.3.2 LSTM Approaches

The first paper for this section which we will look at is from Tortora et al. (2020) and they propose a method to detect and classify artefacts caused in EEG recordings from subjects walking, and to be able to classify the difference in the gait patterns of the different subjects. In this paper they propose an implementation of an LSTM recurrent neural network, to be used as this classifier. Some pre-processing steps which they take before attempting to classify the gait through the LSTM are low and high band passes, to make sure the frequency range is correct, remove line noise via a notch filter for a cleaner signal and remove any bad channels. The actual architecture of the LSTM network uses an input which is equal to the number of channels of the recording, then the input is fed into two LSTM layers sequentially following each other, then into a fully connected layer followed by an output layer. For the evaluation of their approach, they tested many hyper-parameters to be confident in the ones that they chose and used two versions of linear discriminant analysis (LDA) implementation as comparison methods. The results of these were that the proposed LSTM approach achieved between 80% - 90% accuracy for classifying the gaits of the subjects, and this was higher than the methods which were used in the comparison.

The next paper we will look at is from Dash, Ferrari, and J. Wang (2020), where they aim to propose a method to be able to predict speech from an evoked phrase, by using jaw motion detected in the form of muscle artefacts in MEG recordings. In this paper for this aim, they propose an LSTM implementation, which uses regression to predict which type of jaw movement is used in the input. Pre-processing steps which were taken were to only use gradiometers, remove channels which, from visual inspection had an abnormal amount of noise then also remove line notch via notch filters. For the actual architecture of the LSTM network they use an input of windows of length 1ms time steps, with information across 196 channels in the recording, after the input they use two sequential LSTM layers, followed by a fully connected layer and then an output which is the same length as the input, but single dimensional and contains regression predictions of the inputs. For the evaluation they perform many tests of hyper parameterisation for the LSTM, changing unit size, learning rate and the number of epochs. Then for the results they compare the performance across 4 different subjects, where the task of the model is to identify 5 different phrases, they show this through correlation scores between subjects for the different phrases, they also check if different frequency ranges perform better. Overall, the predictions were highly correlated and do seem to confirm the aims of predicting speech via jaw motion.

Chapter 3

Background

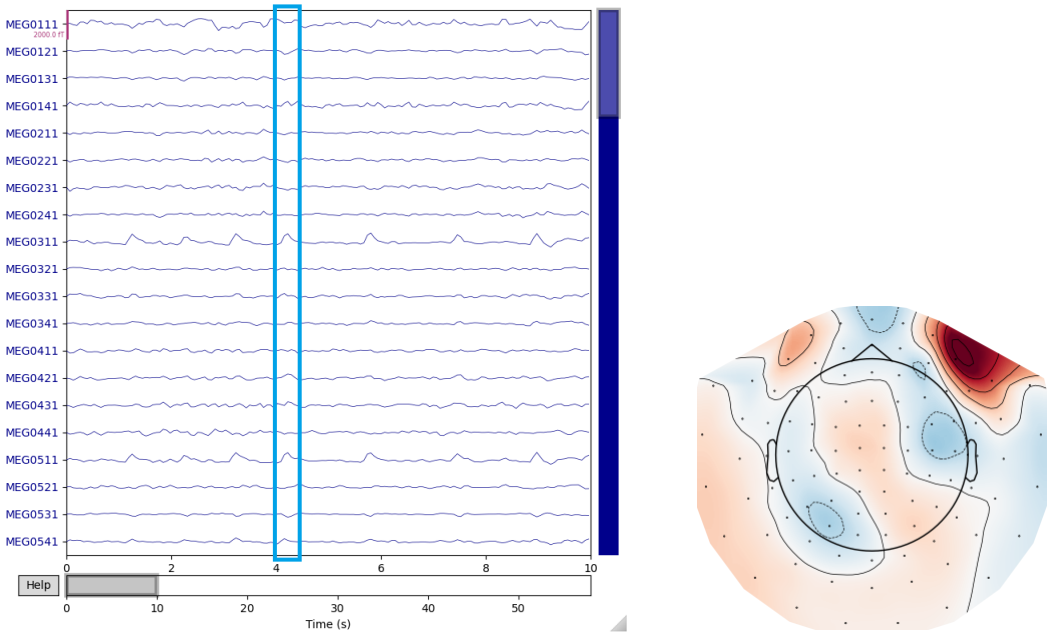
3.1 Topographic plots

Throughout the domain of MEG as well as EEG analysis, topographic plots of multichannel data are used as a useful visualisation tool, to be able to better identify regional activity within the brain in a way that is more visually interpretable than multichannel waveforms. For this dissertation, we will be using the topographic plots from the MNE-Python library. In this library, the topographic plots work by taking a single point in the time series, across all channels as input. This 1-dimensional vector is then passed through a cubic interpolation method, in this case, the Clough Tocher 2D interpolator, Amidror (2002), to get the 2D coordinate positions for the topographic plot. Then to get the colour gradients for the plots, as well as the contours on the plots, which are used to provide information about regions of higher intensity, the original points are extrapolated beyond their coordinates. Also, an additional note is that since the data we are looking at is detecting magnetic field strength it can be positive and negative, this is accounted for in the plots by scaling between red and blue for positive and negative respectively.
We will also be using these topographic plots in our methods which use independent component analysis, this works by taking the column vectors of the mixing matrix which is the length of the number of channels and passing these to be interpolated into the topographic plots. The idea behind this is that since the mixing matrix columns represent the weights for each channel of a given independent source, they can provide useful topographic information about the source.

3.2 Artefacts

For the artefacts that we will be locating and identifying in this dissertation, we will separate them into two distinct groups, those being blink artefacts and muscle artefacts. We chose to do this because these different artefacts have fundamental differences in how they appear in the time series wave-forms, how they are represented topographically, the length in time that they take, the peak amplitudes that they reach as well as the frequency ranges at which they are most present. So muscle artefacts tend to have higher peak amplitudes and last for a slightly longer amount of time, also spatially muscle artefacts can appear in any of the regions of the sensor array, however, blink artefacts should mainly be present in the frontal sensors, since they are the closest sensors to the blink origin. Separating the definition like this allows us to better tackle the problem of locating these artefacts, as we can create methods which are more tailored to the type of artefact.

We can see examples of blink and muscle artefacts in Figures 3.1 and 3.2, key

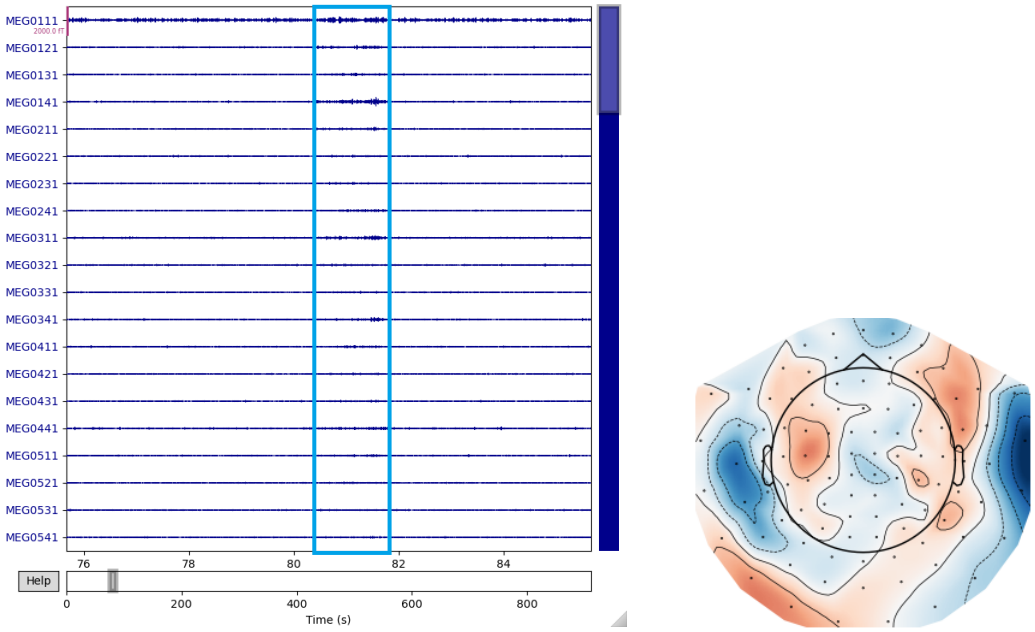


(a) Time Series Plot with the Blink Artefact Highlighted

(b) Topographic Plot of the Muscle Artefact

Figure 3.1: These two figures show the blink artefact from the two perspectives of the waveform and the topographic plot.

attributes to note which we can see from these figures are that between the blink and muscle artefacts in the time series the blink artefact lasts for a much shorter amount of time and blink has a much more of a single gradual peak compared to the more



(a) Time Series Plot with the Muscle Artefact Highlighted

(b) Topographic Plot of the Muscle Artefact

Figure 3.2: These two figures show the muscle artefact from the two perspectives of the waveform and the topographic plot.

turbulent muscle artefact. Then for the topographical plots, the blink artefact shows very high activity in the right frontal region, we would expect a similar level of activity from both the left and right however that's not what we have, but this is likely because some people may have blinks which a single side is stronger. For the muscle artefact topographic plot, we can see high activity on the right side of the head, then still high but not as high on the left-hand side of the head. This is standard for a muscle artefact, however, muscle artefacts can also have high activity at any of the exterior regions of the sensor array. A key extra note is that the difference between red and blue in the topographic plots does not matter for our use case as it is only showing which dipole the magnetic field is being recorded at and we are only interested in the strength.

3.3 Principal Component Analysis

Principal component analysis (PCA) is a method of dimensionality reduction which means reducing the feature size of the data set. The general idea of PCA is to separate the data set into orthogonal features, ordered by the amount of variance the feature

describes for the data set. This can have the effect of reducing the dimensionality of the data as in the original data multiple features were likely describing the same variance effect in the data. The approach for PCA which we will be using is PCA via singular value decomposition (SVD), Wall, Rechtsteiner, and Rocha (2003). From the Equation 3.1 below we can see how it's defined:

$$\tilde{X} = USV^T \tag{3.1}$$

The left term of this equation $\tilde{X}$, is the mean-centred version of the data. The data is centred to ensure that the covariance matrix is equal to $\frac{X^T X}{(n-1)}$. For the right side of the equation, U represents a unitary matrix, which means it has the property that $U^T U = I$, where I is the identity matrix. S is a rectangular diagonal matrix of singular values, and then V is the matrix of basis vectors. This matrix V is what we use to retrieve the principal components, as the columns of V are the principal directions and since $XV = US$, the columns of US are the principal components. The reason why we use PCA in this dissertation is for a whitening step before passing the data to independent component analysis, whitening the data means to set the co-variances to zero, which is part of the process of PCA.

3.4 Independent Component Analysis

Independent component analysis (ICA) is a method of blind source separation, which is designed to pull out latent sources from multivariate data. So, a key assumption of ICA is that we observe a linear mixture of data which can be separated into independent sources. The other key assumption of ICA is that the independent sources are themselves non-Gaussian. This assumption is important as it is used as a key part of the source separation. As independent sources are ordered by the maximal non-Gaussianity, Vigário et al. (2000). From the Equation 3.2 below we can see:

$$X(t) = AS(t) \tag{3.2}$$

Where on the left-hand side $X(t)$, is the data matrix at time t, then on the right-hand side A, is defined as the mixing matrix and $S(t)$ are the source signals at time t. This equation tells us that the original data can instead be represented as a mixing matrix and a set of independent signals, so estimating this matrix A, is the key task of ICA algorithms. This is done by attempting to estimate the unmixing matrix, which is the inverse of A, and can be seen in Equation 3.3 here:

$$S(t) = W^T X(t) \tag{3.3}$$

Where W^T is the unmixing matrix, parameters of this unmixing matrix are estimated by maximising the non-Gaussianity of the created sources, which can be calculated

using metrics such as kurtosis or negentropy. The benefit of ICA when it comes to signal processing is that it allows for easy removal of independent source signals, as all you have to do is remove the independent source which you want to be removed then the mixing matrix will calculate what the data would be if that source was not present.

3.5 Low and High pass filtering

We will be using low and high pass filtering in the procedures and methods which we will be testing in this dissertation, to limit the frequency range which will be used for the methods. The reason why we want to limit the frequency range of our signals is that, when looking for artefacts, we want to limit the presence of signals which are not of interest as well as maximise the presence of the artefacts so that we can better identify them. Low pass filtering works by attenuating all frequencies above a threshold frequency, and high pass works oppositely, so attenuating all frequencies below a threshold. This creates a tighter range for the final signal, eliminating information which is not of interest.

3.6 Z-Scoring

For many of the methods and procedures that we will be testing in this dissertation, we will be using z-scoring to normalise the distribution of the data. The first reason why we do this is so that results from different recordings are more comparable as they will now be a part of the same distribution. Also, by default the values for the recordings are very small femto Tesla values, which do not tend to work very well with machine learning models, so using z-scoring alleviates this issue. We can see in Equation 3.4 below, the formula for z-scoring:

$$Z = \frac{x - \mu}{\sigma} \tag{3.4}$$

Where x is the current point in the signal which we wish to normalise, μ is the mean value in the signal, and σ is the standard deviation of the signal.

3.7 Metrics

3.7.1 Sensitivity and Specificity

Sensitivity and specificity are useful metrics which we will be using in our analysis of the different methods. Sensitivity is a measure of the true positive rate from the

classification, and specificity is a measure of the true negative rate from classification, we can see this from Equations 3.5 and 3.6.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.6}$$

3.7.2 Receiver Operator Characteristic (ROC) curve

A receiver operator characteristic (ROC) curve, is a way of plotting the results from a binary classification system at different thresholds for deciding the binary classification. The x-axis uses the false positive rate ($1 - Specificity$) and y-axis uses the true positive rate ($Sensitivity$), Jones and Athanasiou (2005). Viewing these plots allows for intuitive decision-making for choosing an optimal threshold, based on the threshold which maximises the y-axis and minimises the x-axis.

3.7.3 F1-Score

F1-Score is a measure of accuracy which we will be using in this dissertation. F1-Score is calculated using precision and sensitivity, where precision is defined as the number of true positive predictions divided by all positive predictions. The reason we choose to use F1-Score in this dissertation is that it does not take into account true negative values, which is useful in our case. We can see how its defined in Equation 3.7:

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.7}$$

3.7.4 Cosine similarity

Cosine similarity is used as a measure of similarity between two different vectors. This measure uses the cosine angle between the two vectors as the similarity, this can be seen in the Equation 3.8 below:

$$cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|} \tag{3.8}$$

where A and B are the two vectors which you are looking to calculate the similarity, and proportional vectors have a value of 1, orthogonal vectors have a value of 0 and opposite vectors have a value of -1.

3.8 Classifiers

3.8.1 Support Vector Machines (SVM)

Support vector machines (SVMs) are a method of classification that is widely used for classification tasks, Cervantes et al. (2020), and it is known for being able to handle high dimensional data sets pretty well. SVMs at a base level work by linearly separating the data with an optimal hyperplane such that the hyperplane which is chosen, is separating the different classes of the data with a maximised margin between the support vectors of the classes, where the support vectors are the points closest to the hyperplane, Gunn et al. (1998). For cases in which the data you are looking at is not linearly separable (i.e. the classes cannot be separated by a linear hyperplane), we use what is called a kernel trick, Schölkopf (2000). This is where the nonlinear data is transformed into a different space, where the data becomes linearly separable, using that you can transform back into the original space, and you have a custom hyperplane which separates non-linearly. The kernel which we will be using in this dissertation for SVMs will be the radial basis function (RBF), we chose this purely because it is a very generalized form of kernelization and is very widely used, as well as being able to be parameterized quite easily.

3.8.2 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptrons (MLP) are fully connected feed-forward artificial neural networks, these can be used for classification as well as regression tasks, however, for this dissertation, we will only be looking at the classification set-up, Gutierrez-Osuna (2002). An MLP works by taking an input vector and then using weights on the input layer and a hidden layer, which are gradually changed and configured over the training process to then determine an output, which in our case for this project would be a binary class output of 0 or 1 depending on if there is an artefact or not. These weights are configured by weight optimization which is done by a method called back-propagation, which is a process that optimises the gradient of weights back to front, with respect to minimizing some loss function, Gardner and Dorling (1998). The benefit of using MLP networks as classifiers is that they do not require a feature vector which has been specifically designed with extracted features, as the main idea of MLP is that it learns its own features via the weights from the input.

3.8.3 K Nearest Neighbour (KNN)

K Nearest Neighbour (KNN) is a supervised learning classification method. KNN works by taking plurality votes amongst the K nearest neighbours to the point you are wishing to classify, with the most common vote from the neighbours being the

value given to the new point, Zhang and Zhou (2007). Since KNN uses distance measures to find the nearest neighbours, all points will need to be normalised to make sure the scale of the data does not affect the results. The main benefit of KNN would be that since the method is rather simple, the results can be more explainable than many other methods. So if explainability is a key factor, then KNN is usually a good option.

3.8.4 Convolutional Neural Network (CNN)

Convolutional neural networks (CNN) are a form of neural networks which are specialised for and well known for their performance with image classification, Rawat and Z. Wang (2017). The distinct features of the CNN that separate it from other deep learning architectures are its use of its convolutional layers as well as pooling layers. The convolutional layers work by first, being given a parameter for the number of filters for the layer, where a filter is a window which slides across the image, where the information within the window is combined into a single value, Girshick (2015). Hyper-parameters for the size of the window as well as the stride of the window (how many pixels across and down the window moves with each iteration) are also important to the performance of a CNN. The pooling layer which most often comes directly after the convolutional layers in the architecture, works to create a summary of the convolutional output, by also using a sliding window, to either find the max, average or some other summary of the window. This is done to reduce the spatial representation of the input and reduce computation time.

3.8.5 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) neural networks are a form of a recurrent neural network (RNN), Mikolov et al. (2010), where RNN are a type of neural network, similar to a regular MLP, except it uses a memory concept for the hidden layers to feed back into the network. LSTM adds an additional concept where it can learn long-term patterns in the data. This becomes very useful in times-series tasks, as quite often there are longer form patterns which can help in predicting. The LSTM architecture adds three different layers which combine to provide this long-term memory, this can be seen from Figure 3.3, which shows the distinct layers of the LSTM. For the first part of this architecture, the horizontal line which runs across the top of the figure shows the cell state, this is what stores the long-term state of the network, Hochreiter and Schmidhuber (1997). The bottom half of the diagram shows how inputs combine with the hidden state (short-term memory), to change the long-term cell state if needed and decide on an output. The first connection between the hidden state and the cell state comes on the far left with a sigmoid layer, this is known as the forgetting layer,

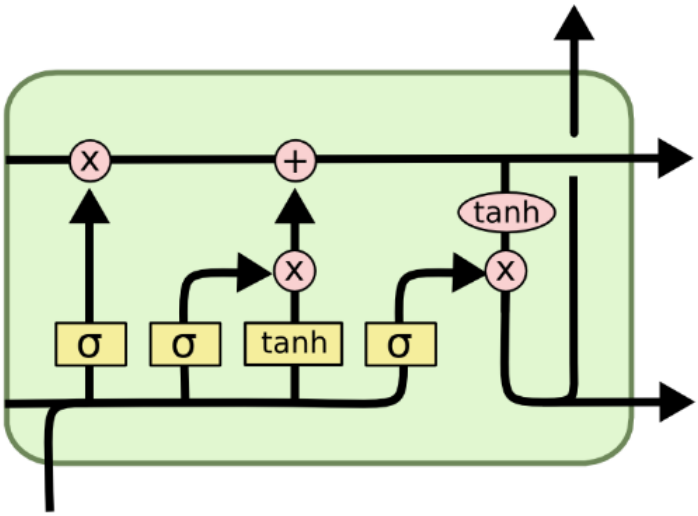


Figure 3.3: Diagram of the LSTM architecture. (Taken from Colah (2015))

and it decides if there is current information which needs to be forgotten. The next two connections come in the form of a sigmoid layer and tanh layer, this is known as the input layer and is where decisions are made on what values of the cell state need to be updated. Then the final connection comes with a tanh layer from the cell state and a sigmoid from the hidden state, which combines to create the new hidden state.

Chapter 4

Methodology

4.1 Artefact Identification

4.1.1 Pre Processing

The main pre-processing which needs to be undertaken for these methods mainly has to do with the distinction between blink and muscle artefacts. For locating these artefacts we want to limit the frequency regions which we are using to those which capture the information given from the artefacts the best, as well as not including as much information from other sources. So for muscle artefacts, we use low and high pass filtering to limit the frequencies between the ranges of 110 - 140Hz, this range is chosen because it is high enough to not be able to capture any brain activity, but still low enough to capture the muscle activity, Muthukumaraswamy (2013). Blink artefacts on the other hand are not as present in the higher frequencies, so we choose the frequency range 1 - 5Hz, as at this range there is still brain activity which can interfere with location, however, the amplitude of blink artefacts is usually stronger than the brain activity in this range. Other pre-processing notes are that for the sensor channels we choose to use magnetometers, as based on visually looking at the results they seem to be able to capture blink artefacts, especially in the waveforms clearer. Then also we use a notch filter, which is a filter which attenuates certain frequencies from the recording. We use this to remove power line noise from the recordings and ensure that this would not affect our approaches.

4.1.2 Exploratory methods

The first methods which we will be looking at are exploratory methods which we utilised to provide us with a better understanding of the problem at hand, and which way to approach it. These methods utilise the multi-channel nature of the data and

use similarity measures to compare the different channels to decide whether or not the part of the signal is an artefact or not. To do this we used a sliding window approach to create subsections of the recording, then comparing the similarity measurement across the windows from the different channels is what is used to decide. A part of this method is to decide on an optimal threshold for the similarity measure, as that is the deciding factor for artefact identification. To decide on the similarity between channels we tested using the covariance between channels as well as the correlation between channels.
The key idea for this method is that when an artefact occurs, the sensors spatially closest to the origin should all have similar readings, so the covariance and correlation should be notably higher than points in time with no artefacts, which would provide a way to identify the points of interest.

4.1.3 Thresholding Method

When first approaching this problem of identifying points in a time series which are of high amplitude and are of note, the first and perhaps most intuitive response is to start with a thresholding method, where if a point in the time series reaches above a set threshold in amplitude then those points are marked as artefacts of interest. For this method, for each channel in the recording, we first need to z-score the channel, however in this case since this thresholding method is particularly affected by bad channels which perhaps have a much higher variance in the readings than is standard, which could mean that there was a fault in the sensor when the recording was done. This affects this method since it is a static threshold, for bad channels it will over classify as most of the bad channel is not of interest. To tackle this issue we first calculate the standard deviation across each channel in the recording, then take the mean of all these standard deviations, we can call this σ_m. Then when we calculate the z-score of each channel, we use this value of standard deviation, as you can see in Equation 4.1:

$$Z = \frac{x - \mu}{\sigma_m} \tag{4.1}$$

We do this with the idea that this should make the results more robust to bad channels, as it is no longer using its own higher than normal standard deviation value to z-score. In the next steps of this method after taking the threshold on the channels, to get the points above, we need to remove duplicate points, since it is very likely that some artefacts will appear in multiple channels. Another issue which comes with this method is that since this threshold is static, especially for muscle artefacts it is likely that a single artefact in time may have points which jump above and below the threshold line, causing the classified points to be quite jagged. So to tackle this issue we attempt to smooth the points into a region of artefact activity based on how close

in time they are. We do this by connecting points which are 0.1 seconds in the range of each other and creating a range between the earliest and latest of the chained together points. The reason why we use a 0.1 second value for a range between points is that from visual exploratory analysis and discussions with a domain expert we decided that if points are that close to each other it should be reasonable to assume that they originate from the same artefact. Another point to note which is similar is that for blink artefacts, the labels provided to us by the domain expert are single-point labels at the max deflection of the blink, however, this does not provide us with information on how long the blink lasts, so we add a 0.1 *second* radius around each labelled point as that seems like a good minimum range to capture most of the blink. To determine the threshold value which we will use to assign artefact labels, we will create a ROC curve, as described in section 3.7.2, where we will test a range of thresholds on a set of training recordings, calculating the sensitivity and specificity, then choosing the threshold which best optimises the sensitivity and specificity for all of the training recordings, then see how that threshold performs on the test and external validation sets, doing this testing for both blink and muscle artefacts separately.

4.1.4 Sliding Window Methods

The next set of methods we created and will be showing are variations on the same idea of sliding window methods, Chu (1995). These are methods that when given a window of a set size, move across the recording, which then collects features based on the information in the window. For the different types of artefacts that we will be looking at (Blink and Muscle), it is better to use different size windows when looking for these different artefacts, as the windows should ideally be close to the size of the length of a standard artefact that is being searched for. So for blink and muscle artefacts, we chose to use window sizes of 200 and 500-time steps respectfully because from visual exploratory analysis and discussions with a domain expert these values do seem to represent well the standard length of the artefacts we are looking for. We will also test variations on the window size to see what effect that has on the method. Additionally, for the sliding windows, we chose the windows such that there is no overlap between any windows, the main benefit that would come from this would be to allow for further analysis to better identify edges of artefacts, however, this is not a key task of this project, and we are more concerned with the locations of artefacts which can be found from distinct windows.
The variations on the methods which we will be going over next are variations which were made for either blink or muscle artefact identification, based on the relevant characteristics of the artefacts.

4.1.4.1 ICA Window Comparison

The first sliding window procedure which was considered was to move from the idea of using cross-channel information as a measure to identify points of interest, which was described in 4.1.2, to this method where we use an independent component as a comparison to identify points. So for this, we thought to first calculate PCA on the window, which is done as a pre-processing step to whiten the input data ICA on the window itself, take the most non-Gaussian identifying source component then compare it to a single standard induced artefact component which was identified previously as a part of the induced artefact recording. For more details on PCA and ICA see sections 3.3 and 3.4. The idea behind this method is that the independent component for the induced artefact should have similar features to the naturally occurring versions of the same artefact, with the main difference being that induced artefacts do tend to have more power in the signal, which is just from the natural human response of when you are told to blink, for example, you will blink with more force than if you were doing it subconsciously. So, the sliding window moves across the recording and finds the most non-Gaussian independent component from the window, where we then decided to use cosine similarity as a comparison measure, to compare the component with the induced comparison component. Then if the similarity is greater than some threshold, decide if the window is an artefact or not. The issue we found when originally creating and testing this was that using a single induced artefact as a comparison measure was not consistent between different subjects, as different people when asked to induce a blink, for example, the strength of the blink will vary for person to person, which will change the shape of the induced blink to the point where sometimes it will not be a good representative of naturally occurring blinks.

4.1.4.2 Window Max Point Deflection

This next method was created using what we learned from the ICA window comparison method, in this method, we take the idea of comparing information from the window and we try to generalise it. We generalise by training classifiers on labelled artefact points and non-artefact points, which have been identified in the recordings by a domain expert. It is worth noting for this method, that since we are training on single points from artefacts, this works better for blink artefacts as blink artefacts most typically have a single peak, which can be used as the point of max deflection to define the artefact by, this does not work as well for muscle artefacts as they tend to have multiple peaks, so the peak deflection may not define the artefact as clearly. So, as mentioned the classifiers are trained on a 1-D vector of size N, where N is the number of channels in the recording. Before data is fed into the classifiers, as channels are z-scored to normalise the distribution, then we absolute the data since we wish to treat positive and negative peaks equally. The classifiers which we test for this

procedure are an SVM approach, an MLP and KNN. For further information on how these methods work see sections 3.8.1, 3.8.2 and 3.8.3.
For actually testing the classifier on new points, we use the sliding window approach, and since we are looking at blinks for this procedure, we use a window size of 200-time steps. Then for each window, again because we are looking to identify blinks, we chose to look for the peak deflection within the window from only the frontal sensors. The reason we do this is that, if we took information from all sensors to find the max deflection, it is quite possible that at the same time as a blink there may also have been high levels of activity present in a different region of the sensor array, in which case we would find a point of max deflection which would not be representative of a blink, and therefore we would miss the blink within that window. So once the max deflection from the frontal sensors is found, then the 1-D vector of length total number of sensors at this point is fed through the classifier to get an output.

4.1.4.3 Window Standard Deviation Vector

This next method is somewhat similar to the previous method in that, we are again using distinct windows to segment the recording. The reason we have this method is for it to be used for muscle artefacts. This is because, as mentioned, a characteristic of muscle artefacts is that using peak points deflection would not work since muscle artefacts tend to have multiple peaks, so for this method, we still use the idea of creating 1-D vectors to train classifiers, the difference being that we use the standard deviation of the points within each window as a value for the feature vector, so the vector is of size N, where N is the number of channels. The idea behind using this information as a feature vector is that since muscle artefacts tend to have multiple peaks across their time span, they will likely have a higher than the normal standard deviation in the window. Another key idea for this method is that since we are using this for muscle artefacts, we will be using all recording sensors as information for the models, which is different to the previous method which identified based on frontal sensor information, we use all sensors here because muscle artefacts are less centralised and are more diversely created by a range of different muscles around the head. As was for the previous method, before segmenting into windows, all of the channels are z-scored independently, and then the absolute value of the channels is used. When the windows are segmented we use a window size of 500-time steps. The classifiers that we will use to test this approach will also be SVM, MLP and KNN.

4.1.4.4 Window Spectral Information

An additional method that we would looked at using is a method based on using the spectral frequency information of a window to classify artefacts. Based on the idea that artefacts likely have increased activity for different frequencies compared

to regular brain activity in the recording. So the method would be set out where, we would calculate the power spectral density using Welch's method, Solomon Jr (1991), on each window separately. This would provide a feature vector based on the activity of the different frequencies in the range. A key pre-processing step which should be mentioned is that the activity values are passed through a log base 10 function, to remove the scaling effect. This feature vector would then be tested on classifier methods such as SVM. A key issue that we ran into with this approach is that the feature vector which we would have to use is very small because we need to use relatively short window sizes, which means the number of frequencies which can be sampled for the vector is very small, meaning usually is it not enough information for the classifiers to perform well.

4.1.5 Long Short-Term Memory (LSTM) Methods

4.1.5.1 LSTM as a Classifier

The next methods we will look at creating are LSTM-based methods. We look at these as the next step past the more traditional classifiers of SVM, for example, because LSTM approaches are well known to perform well for applications of time-series classification, Lipton, Berkowitz, and Elkan (2015), which is what the main aim of this project is, they also are usually expected to generalise better than most methods of classifiers in this domain, due to the structure of the method.
The architecture that we will be using to train our LSTM models will be the same for both types of artefacts and has precedence for time-series classification tasks. This is a bidirectional LSTM layer with 100 units, followed by a fully connected layer with 50 nodes, and a single node output layer. For further details on how the LSTM layer specifically works see section 3.8.5. A key idea which we add to seek improvement above standard LSTM implementations is to make the LSTM layer bidirectional. This bidirectionality means that instead of only running through the data from past to future, it is also run from future to past, which when compared to a standard unidirectional LSTM which can only preserve information about the past, a bidirectional is able to preserve information about the past and the future, which allows the model to better understand the context of the data, Yildirim (2018).
For the inputs into the LSTM, we will be testing two different approaches, the first being to input the raw recording as windows into the LSTM, where the windows will be size 500 when training for muscle artefacts and 200 when training for blink artefacts (in order to be consistent with the other methods), also where the number of features for each window will be N, where N is the number of channels in the recording, to make sure all the information is being captured. The other approach will be to use the sliding window feature vectors as described in Sections 4.1.4.2 and 4.1.4.3, for looking at muscle and blink artefacts respectfully. The idea behind testing this second

method is to compare the results from different representations of the data, seeing if the condensed representation allows for the model to train better. Then for the initial training of the LSTM models, we will use default parameters for learning rate and train for 50 epochs with early stopping criteria where the validation loss does not improve after 10 epochs then stop the training and take the weights from the epoch with the best validation loss.

4.1.5.2 LSTM layer outputs as feature vector

A second approach we take following the training of the LSTM models is to instead of using the resultant model as a classifier in its own right, take the LSTM layer outputs from the model, and input those 1-D vectors into more standard classifiers which are built to classify data in this form. This data is collected by passing through each labelled window through the trained model, taking the outputs from just the LSTM layer and using those feature vectors of size 200, as the training and testing data. The size 200 comes from the fact that we use LSTM layers with 100 units, then making them bidirectional doubles the number of units. This approach purely comes from the idea that LSTM is learning its own features from the data, then a more standard feature-based classifier can use those features to classify, which is a more current approach in the literature, Khan et al. (2021), Kiruthika and Thailambal (2022).

4.2 Artefact Source Classification

A second way of looking at the task at hand for this project of identifying artefacts in the recordings is instead of purely locating the temporal points which have artefact properties, would be to classify the source signals of the recording which contain the blinks. We look at this method as an option even though it does not provide as much statistical information about when and where the artefacts are located the benefits come in the fact that you can rebuild a recording by removing the artefact-infected sources, and possibly have a cleaner final result without having to crop out sections. So this method should be seen more as a method of classifying for removal, rather than locating. This idea is featured in much of the literature for artefact identification in MEG, so we feel like it would be remiss if we were not to include it.

4.2.1 ICA Source Separation

The procedure which we will be taking to do this source for artefact classification is ICA. ICA is a method of blind source separation which finds independent sources based on how non-Gaussian they are, for more information see Section 3.4. The first step of this approach is to first perform PCA on the recording, this whitens the

data, removing the covariance. The next step is to choose the number of independent components which we wish to isolate. For this dissertation, we choose to isolate 30 components, this value has been decided based on reviewing literature and seeing that most uses of this approach used between 20 - 30 components. Using these components, the next step is to train a classifier to be able to differentiate between whether the sources contain artefacts or not, also the labelling of the sources is done by a domain expert. For the training of the classifiers, we will be using the weights from the column vectors of the mixing matrix as described in the second paragraph of section 3.1, to represent the source signals in a way that represents the effect from the source on the different channels. We will be using two versions of inputs to test the approach, the first will be to use this version of the feature vector and input into an MLP, SVM and KNN to provide us with a good comparison for this input. The parameters we will be using are relatively default which we found via a parameter search. The second will be to take the weight vectors create the topographic plots from them, and use them to train a CNN. The idea behind this is to test if viewing the information from a topographic point of view can generalise the results better. For the CNN method, we first have to pre-process the images, this process involves normalising the pixels, by dividing them all by 255, then we grayscale to images to reduce the input size, we also need to crop the images so that just the topographic plot is in the image. The architecture of the CNN which we will use is a pretty standard image classification architecture, which has three convolutional layers, each with a max pooling layer following it. For these convolutional layers we use 16, 32 and 64 filters respectfully and filter sizes of 8, 4 and 2, all with a stride of 1. After passing through all convolutional and pooling layers, the data is flattened and passed through a fully connected layer with 128 nodes, then to a single output node.

4.3 Testing Approach

For testing these methods, the first distinction which needs to be made is that we will be separating the testing between methods being testing on locating blink artefacts and methods being testing on locating muscle artefacts. This is purely because the artefacts are very different and have different characteristics therefore the performance should be validated separately. Also for the training and test sets which will be used for the different methods, blink methods will be trained and tested using the shorter length artefact recordings, and then we will use a single full-length actual analysis recording as external validation. Where for muscle artefact methods we will be using full-length recordings for training, testing and external validation. The reason for this difference is because, for blinks, the artefact recording should be sufficient because even though they contain induced blinks, they also contain naturally occurring blinks due to the subconscious nature of blinks, also the other reason is that the artefacts

had to be manually labelled, and doing this for the full subject length recordings for blinks (which occur at a much higher frequency than muscle artefacts) would take an unreasonable amount of time for the domain expert, within the constraints of the project. The reason why we use full length for muscle artefacts is first that the artefact recordings only contain a single muscle artefact which is induced and does not reflect the same characteristics as the more natural, accidental artefacts in the full length, also the other reason is that muscle artefacts are less common so manually labelling them is not as time intensive. Then another point to note is that to ensure that the external validation data is as external as possible the training and internal testing recordings are taken from the same two subjects just from different days, whereas the external validation recordings are taken from entirely different subjects on different days.

For the actual recordings data sets which we will be using to test the various

Recordings	No. Positive labels	No. Negative labels	Percentage Split
Blink Artefact training set 1	37	327	10.2%/89.2%
Blink Artefact training set 2	40	204	16.4%/83.6%
Blink Artefact training set 3	34	270	11.2%/88.8%
Blink Artefact training set 4	61	366	14.3%/85.7%
Blink Artefact training set 5	59	252	19%/81%
Blink Artefact training set 6	35	368	8.7%/91.3%
Blink Artefact training set 7	39	289	11.9%/88.1%
Blink Artefact training set 8	72	355	16.9%/83.1%
Blink Artefact test set 1	63	226	21.8%/78.2%
Blink full length external set 1	50	3024	1.6%/98.4%

Table 4.1: A table showing the distribution of blink artefacts in the training, test and validation data used for the blink methods.

methods you can see from Table 4.1 and 4.2, the distribution of positively and negatively labelled blinks and muscle artefacts respectfully. We can see that for the blinks, the distribution is around a $10\% - 15\%$ positive label split, whereas the muscle distribution is about a $1\% - 2\%$ positive label split. This difference is what we expect, as muscle artefacts are not as naturally occurring as muscle artefacts so we expect to see them less. Something else to note is that all of the data sets are somewhat weighted to negative labels, we choose to keep the sets like this, instead of say reduced down to a 50/50 split because it is expected for this kind of data to be distributed in this way since we expect the models to be slightly biased to negative labels anyway.

Looking now at the actual generic testing method which we will keep consistent across

Recordings	No. Positive labels	No. Negative labels	Percentage Split
Muscle full length training set 1	41	836	4.7%/95.3%
Muscle full length training set 2	8	1209	0.7%/99.3%
Muscle full length training set 3	7	1228	0.6%/99.4%
Muscle full length training set 4	17	835	2%/98%
Muscle full length training set 5	34	1226	2.7%/97.3%
Muscle full length test set 6	112	1709	6.2%/93.8%
Muscle full length test set 7	13	1221	1%/99%
Muscle full length validation set 8	112	1170	8.7%/91.3%
Muscle full length validation set 1	13	1216	1%/99%

Table 4.2: A table showing the distribution of muscle artefacts in the training, test and validation data used for the muscle methods.

the different methods, for blink and muscle artefacts separately. This approach is to keep consistent the training, and test validation sets across methods to make sure results are comparable, for the recordings themselves we will feed as predictions the same number of windowed sections into the methods, again to make the results comparable. For all classifiers we will be calculating ROC curves (which are described in section 3.7.2), where we will optimise the probability of classification threshold to improve the labelling, based on the Youden metric, which is a standard metric for optimising ROC and is defined in Equation 4.2 as:

$$J = Sensitivity + Specificity - 1 \tag{4.2}$$

where J is Youden's metric. We choose this also as it provides an optimising balance between sensitivity and specificity, as described in section 3.7.1. This is something we decided we wanted as in the case of this project, having to choose between sensitivity or specificity, provides no single answer. As a lower sensitivity means, higher false negatives, which means we are classifying artefacts as clean data, which is bad. However, lower specificity means higher false positives, meaning we are classifying clean data as artefacts which would mean that when the artefacts are removed clean brain activity would be removed. When it comes to the metrics we will use to measure the optimised predictions, we will use accuracy as a standard measure, however, it should be noted that due to the very highly weighted negative labels, accuracy values will be somewhat inflated due to the true negative predictions. So we will also use F1-Score as a measure, which is good for this case as it does not take into account true negative predictions and is described in further detail in section 3.7.3.
The testing environment for which all of these tests have been conducted are through MYndspans server which has 64 cores and 64GB of RAM, it runs on Ubuntu 20.04

and is hosted by oracle cloud, the key libraries which we use in testing are MNE-Python Gramfort et al. (2013), Keras Chollet et al. (2015), sci-kit learn Pedregosa et al. (2011) and NumPy Harris et al. (2020).

Chapter 5

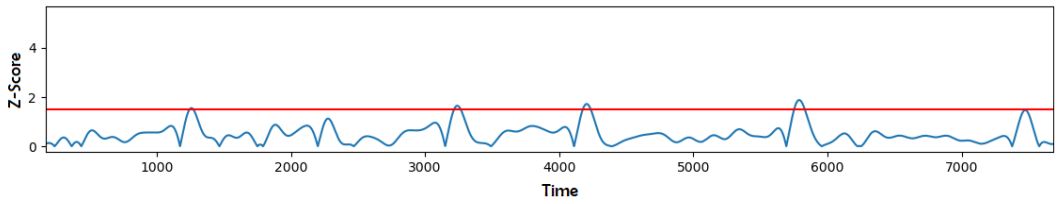
Results

For the results of this dissertation, we will be separating into sections to first show the results and decisions made from optimization and comparison of different methods. Where we will look at the optimization of the threshold value for the threshold approaches, as well as a comparison of classifiers for the sliding window methods and a comparison of inputs and parameters for the LSTM methods. Then we will do a more in-depth comparison of the results for both blink and muscle artefact method results separately. As well as looking at the results for our testing of classifying source components of blink artefacts.

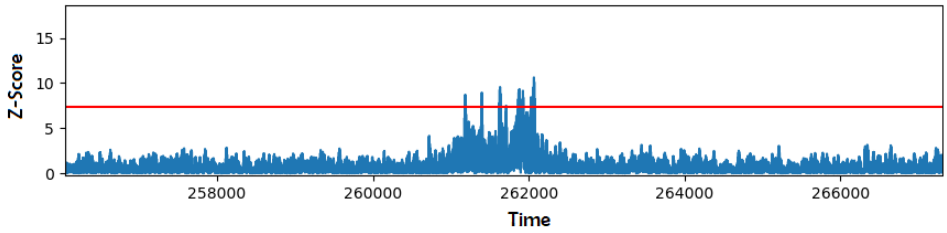
5.1 Threshold Method Optimization

For the threshold method optimization, as discussed in section 4.1.3 we will be testing multiple values of threshold, recording the sensitivity and specificity of each threshold across the multiple training sets then using these recordings to create a ROC curve plot and decide on the threshold which optimizes the Youden index, which takes in to account both specificity and sensitivity. We will be testing this method for both blink artefacts and muscle artefacts separately, finding different thresholds for the different types of artefacts. These thresholds can be seen visually in Figure 5.1, where the blink data is in a) and the muscle data is in b).

Looking first at the results for testing for blink artefacts, we can see the ROC curve calculated from Figure 5.2 and the results of using the optimally selected threshold from Table 5.1. Looking first at the graph in Figure 5.2, we can see that ideally for a ROC we would want it to peak into the top-left hand corner as close to it as possible, so for this graph, it does not seem to reach very far to the top left, as it seems to struggle to get high values of sensitivity, which suggests that for trying to recognise blink artefacts there are quite a lot of false positives. For the metric results from the table, we can see that this threshold for the test set does reasonably well with



(a) Blink artefact threshold shown for a single channel, zoomed in to some of the blink artefacts.



(b) Muscle artefact threshold shown for a single channel, zoomed in to one of the muscle artefacts.

Figure 5.1: This figure shows two graphs, each showing how the threshold works on a single channel of the recording.

Blink Artefacts				
Data Sets	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	0.76	0.84	0.71	0.72
External Validation	0.16	1	0.14	0.07

Table 5.1: A table showing the results of using the optimized threshold value from the training sets to the test and validation sets for blink artefacts.

good values between 0.7 - 0.86 for the accuracy, sensitivity and F1-score, however with a somewhat low specificity at 0.71. For the external validation, we can see that this method does terribly, with very low accuracy and F1-score. The fact that the sensitivity is a perfect 1, tells us that this chosen threshold must be completely over classifying points as being positive. The reason why this is is because of how we have had to structure the testing for the blink artefacts. Since for blink artefacts we chose to train and test using shorter recordings induced artefact recordings, and then a full-length recording as validation, this means that in this case when we z-score the channels, the length of the recording will affect the z-scoring, meaning that the optimized threshold is not optimized for recordings which are full length. The

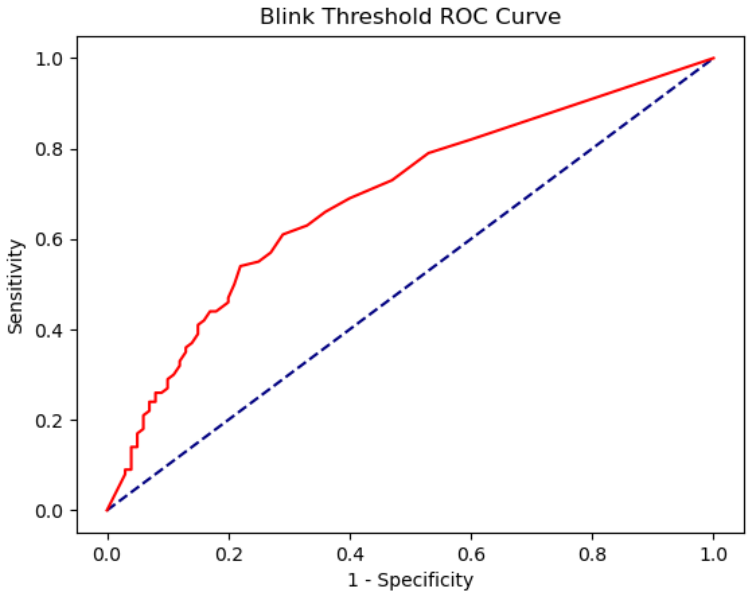


Figure 5.2: Graph showing the ROC curve calculated for the thresholding method, for blink artefacts.

only way to fix this issue would be to train using blink-labelled full-length recordings, however as mentioned in section 4.3, this would take an unreasonable amount of time for our domain expert.

Looking next at the results for the testing of muscle artefacts, we look at Figure 5.3

Muscle Artefacts				
Data Sets	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	0.99	0.86	0.997	0.82
External Validation	0.99	1	0.997	0.87

Table 5.2: A table showing the results of using the optimized threshold value from the training sets to the test and validation sets for muscle artefacts.

and Table 5.2. Looking first at the graph from Figure 5.3, we can see that the ROC curve which was calculated for optimizing the threshold, as we can see thresholding for muscle artefacts has done significantly better than blink artefacts for the same method, as it must have been able to reach much higher sensitivity's and specificity's, and therefore can reach much further into the top left of the graph. For the actual

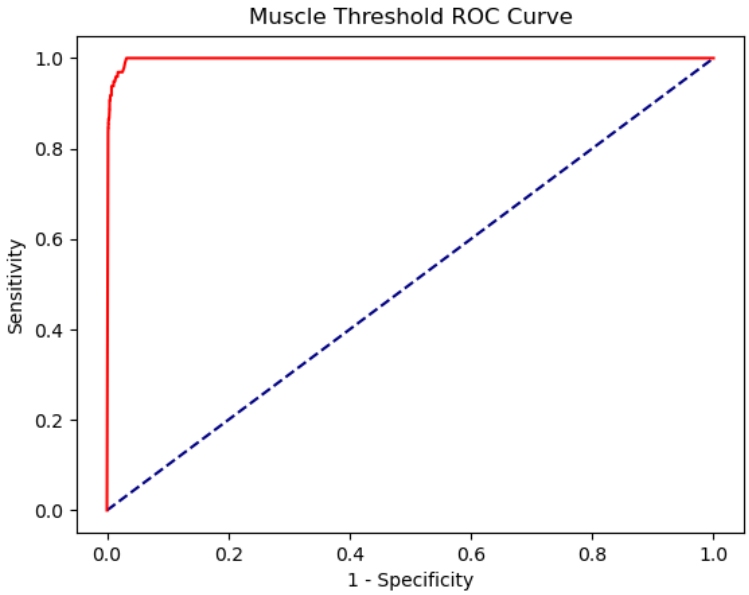


Figure 5.3: Graph showing the ROC curve calculated for the thresholding method, for muscle artefacts.

results, we look at Table 5.2, where we can see that for both the test set and external validation this optimised threshold does very well, achieving high accuracy, very high Specificity and high F1-scores at above 0.8. In comparison to the blink artefacts, this thresholding method appears to perform much better on the muscle artefacts, this is what we expect as muscle artefacts tend to reach much higher amplitudes than anything else in the same frequency range that we search for them, so a thresholding method is less likely to pick up something else as a muscle artefact, when compared to the blinks which in their frequency range are not as prominent.

5.2 Sliding Window Methods Comparison

For the testing of sliding window methods, we will be using the methods described in Sections 4.1.4.2 and 4.1.4.3, for blink and muscle artefacts separately. For each of these methods, the comparisons which we will be looking at are to train these methods on a support vector machine classifier (SVM), a K nearest neighbour classifier (KNN) and a multi-layer perceptron classifier (MLP), to see which method provides the best test and external validation results. For all of these classifiers, we will be using reasonably default hyper-parameters, for example for SVM with RBF kernel and gamma set to $\frac{1}{N}$ where N is the number of features, for KNN we set $K = 5$ and for MLP we use a

single hidden layer with 300 nodes and a standard learning rate of 1×10^{-3}.

The results for the two versions of the sliding window methods for blink and muscle

Max Point Deflection					
Data Sets	Classifier	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	SVM	0.82	0.84	0.81	0.78
	KNN	0.75	0.54	0.89	0.62
	MLP	0.86	0.76	0.92	0.81
External Validation	SVM	0.95	0.91	0.94	0.51
	KNN	0.93	0.73	0.94	0.39
	MLP	0.98	0.85	0.98	0.70

Table 5.3: A table showing the results of training multiple different classifiers on the method of using a slice at the max deflection point as training for blink artefacts.

Standard Deviation Vector					
Data Sets	Classifier	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	SVM	0.998	1	0.998	0.91
	KNN	0.995	1	0.995	0.78
	MLP	0.9	0.96	0.99	0.58
External Validation	SVM	0.998	1	0.998	0.93
	KNN	0.99	0.88	0.99	0.69
	MLP	0.96	1	0.96	0.34

Table 5.4: A table showing the results of training multiple different classifiers on the method of using standard deviation vector of a window for muscle artefacts.

artefacts can be seen in Tables 5.3 and 5.4. Looking first at Table 5.3, which shows the max deflection point method for blink artefacts, we can see that the results for the test set across the three different classifiers are reasonably strong with SVM and MLP outperforming KNN. But then MLP marginally outperforms SVM in accuracy, specificity and f1-score, but not in sensitivity, which suggests that MLP has more of a trade-off for shifting to more false negatives than false positives. Then for the external validation data, we can see that SVM and MLP still outperform KNN, especially for F1-score. However for comparison between SVM and MLP, MLP does do better than SVM by a significant 0.21 margin in F1-score, showing that for this method of input calculation the MLP performs better.

Looking next at Table 5.4, which shows the results of using the standard deviation vector for the three different classifiers. From this we can see that all three classifiers perform very well and leave little room for improvement, for both the test set and

external validation set the same pattern appears where for f1-score MLP is the worst at 0.58 and 0.34, followed by KNN at 0.78 and 0.69, then SVM at 0.91 and 0.93. This is very good, especially for SVM, when considering the fact that in these sets the number of true positives compared to true negatives is very low, which means an f1 above 0.90 has very few false positives or negatives.

5.2.1 Further Optimising Max Point Deflection

As we can see from the results collected for the Max point Deflection method for blinks and standard deviation vector for muscles, the muscle artefact leaves very little room for improvement in comparison to the blink artefact results which achieve at best high 0.8 accuracy. So in this section, we take the best performing method for max point deflection, which was MLP, then change the learning rate and the number of nodes in the hidden layer to see if these hyperparameter changes can improve the performance of the method.

The results for these hyperparameter tests can be seen in Tables 5.5 and 5.6.

Changing Learning Rate					
Data Sets	Learning Rate	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	1×10^{-3}	0.86	0.76	0.92	0.81
	1×10^{-4}	0.86	0.84	0.88	0.82
	1×10^{-5}	0.84	0.71	0.92	0.77
	1×10^{-6}	0.71	0.38	0.91	0.50
External Validation	1×10^{-3}	0.98	0.85	0.98	0.70
	1×10^{-4}	0.98	0.85	0.98	0.70
	1×10^{-5}	0.98	0.85	0.98	0.69
	1×10^{-6}	0.92	0.73	0.92	0.36

Table 5.5: A table showing the results of changing the learning rate for the max point deflection sliding window method.

Looking first at table 5.5, we have varied the learning rate between 1×10^{-3} and 1×10^{-6} while keeping the number of nodes stable at 300. We can see that for the test set the results for different learning rates is pretty similar, with only 1×10^{-6} performing notably worse than the others, which is likely because it may not have fully trained to the data by the time it had stopped. Then between the rest of the learning rates 1×10^{-3} and 1×10^{-4} perform marginally better than 1×10^{-5}, with 1×10^{-4} improving the sensitivity for a slight trade-off with specificity. For the external validation, it is again a similar story in that 1×10^{-6} performs worse than the others, and the others all perform practically the same across all metrics, which suggests that changing the learning rate has little effect on the improvement of this

Changing Number of Hidden Nodes					
Data Sets	No. Nodes	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	50	0.84	0.84	0.85	0.80
	150	0.87	0.77	0.92	0.81
	300	0.86	0.76	0.92	0.81
	500	0.86	0.81	0.88	0.81
External Validation	50	0.97	0.86	0.97	0.64
	150	0.96	0.85	0.96	0.57
	300	0.98	0.85	0.98	0.70
	500	0.96	0.88	0.96	0.57

Table 5.6: A table showing the results of changing the number of hidden layer nodes for the max point deflection sliding window method.

method. Looking next at table 5.6, which shows the results for changing the number of nodes between 50 and 500, with the default which we have been using being 300. From these results we can see a similar story with these as was the case with changing the learning rate, there is little change to the effectiveness of the model when changing the number of nodes for both test sets and external validation. As we can see the only changes which seem to occur for the test set results are trade-offs between that sensitivity and specificity. Then for external validation, f1-score is the only metric with notable changes and 300 nodes performs the best at 0.7. So overall we have found that even though there seems to be room to improve, it appears that with this method we may have reached the maximum.

5.3 LSTM Methods Optimization

As discussed in the LSTM methodology Section 4.1.5.1, we will be doing some additional testing of inputs as well as further optimization for the LSTM methods. This will first be testing the input of windows of the raw recording across all channels compared with a calculated feature vector for each window, whereas for blink artefacts it takes the point of max deflection from the frontal channels, and then uses data from all channels at that point as the feature vector. Then for muscle artefacts, the vector is a standard deviation calculation of the window for each channel in the recording. Once we find which input works better we will also run some optimization by configuring the learning rate to different values, and seeing if this has an impact on training performance.

Starting off as we can see from the results Tables 5.7 and 5.8, which show the results for changing the input type for blink and muscle artefacts respectfully, on

Blink Artefacts					
Data Sets	Input Type	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	Raw Recording	0.76	0.78	0.74	0.71
	Calculated Vector	0.80	0.74	0.84	0.74
External Validation	Raw Recording	0.85	0.78	0.86	0.25
	Calculated Vector	0.90	0.84	0.90	0.35

Table 5.7: A table showing the results of changing the input the LSTM classifier method on Blink artefact data.

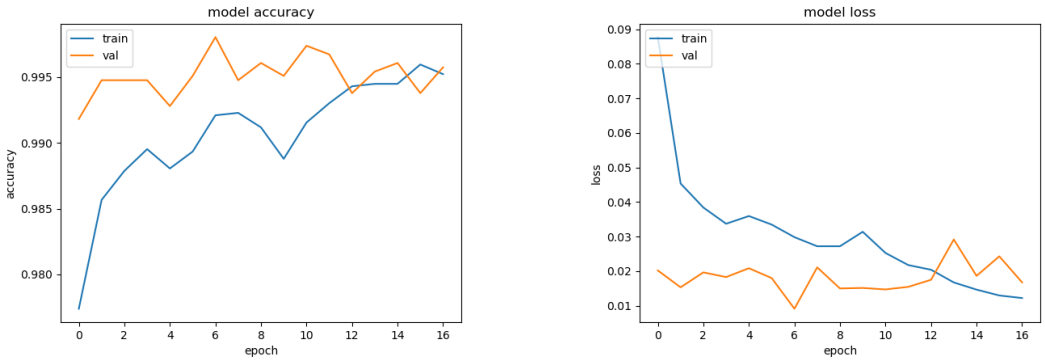
Muscle Artefacts					
Data Sets	Input Type	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	Raw Recording	0.98	0.96	0.98	0.51
	Calculated Vector	0.98	0.96	0.98	0.48
External Validation	Raw Recording	0.99	0.96	0.99	0.62
	Calculated Vector	0.99	1	0.995	0.80

Table 5.8: A table showing the results of changing the input the LSTM classifier method on Muscle artefact data.

both the test set and the external validation set. Looking first at Table 5.7, we can see that for the test set both the raw recording and calculated vector achieve similar values of accuracy, sensitivity and F1-score, but the calculated vector has notably low specificity (true negative rate), then for the external validation, we can see that the calculated vector outperforms the raw recording for every metric. However, something to note is that the F1-scores on the external data are much lower than the test sets. This is because the external validation recordings are much longer than the test set recordings, this means that with the same or slightly lower false positive rate, there will be a much greater total number of false positives in the external set, causing a lower value in the F1-score. Overall for blink artefacts, the calculated vector performs better as an input.

Looking next at Table 5.8, we can see that this table shows the results for training for muscle artefacts for both the raw recording and a calculated vector as input, on both the test and validation sets. We can see that for the test set that both the raw recording and calculated vector perform very similar across all four metrics, with the raw recording doing marginally better on F1-score however, this is likely not statistically significant. For the external validation, we can see that for accuracy, sensitivity and specificity, both inputs perform similarly however the F1-score for the calculated vector is notably higher than the raw recording. So overall for muscle artefacts we go forward with the calculated vector as the input of choice.

As mentioned we will be looking at using different learning rates and comparing the



(a) Accuracy Graph (b) Loss Graph

Figure 5.4: These two figures show the training and validation accuracy and loss for the muscle artefacts training, using a learning rate of $1 \times 10^{-}3$.

results for training at these different rates. The reason we chose to do this can be seen from Figure 5.4, as in both of these graphs we can see that for the validation results the values seem to be high instantly and then fluctuate around a stagnant value. This indicates that the learning rate is too high and that the model could be bouncing around a local minima. So testing at different learning rates provides the opportunity for the model to learn slower and perhaps learn further.

Looking now at the results which we have for varying the learning rate for

Blink Artefacts					
Data Sets	Learning Rate	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	1×10^{-3}	0.8	0.74	0.84	0.74
	1×10^{-4}	0.82	0.75	0.87	0.76
	1×10^{-5}	0.81	0.68	0.88	0.73
	1×10^{-6}	0.82	0.73	0.88	0.76
External Validation	1×10^{-3}	0.90	0.84	0.90	0.35
	1×10^{-4}	0.75	0.47	0.83	0.14
	1×10^{-5}	0.88	0.21	0.90	0.10
	1×10^{-6}	0.70	0.91	0.69	0.16

Table 5.9: A table showing the results of changing the learning rates for the training of the LSTM between values of 1×10^{-3} and 1×10^{-6}, for blink artefacts.

the training for the LSTM approach, from Tables 5.9 and 5.10. Looking first at the

Muscle Artefacts					
Data Sets	Learning Rate	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	1×10^{-3}	0.98	0.96	0.98	0.48
	1×10^{-4}	0.98	1	0.98	0.50
	1×10^{-5}	0.99	1	0.996	0.79
	1×10^{-6}	0.99	1	0.996	0.79
External Validation	1×10^{-3}	0.99	1	0.995	0.8
	1×10^{-4}	0.96	1	0.97	0.37
	1×10^{-5}	0.99	1	0.998	0.91
	1×10^{-6}	0.99	1	0.997	0.83

Table 5.10: A table showing the results of changing the learning rates for the training of the LSTM between values of 1×10^{-3} and 1×10^{-6}, for muscle artefacts.

results from Table 5.9, we can see that for changing the learning rate the effect for training results for the test set is very minimal and any change which has occurred is likely not statistically significant. The perhaps only notable change is that there is a higher Specificity for learning rates lower than 1×10^{-3}. Looking at the results for the external validation, we can see that when changing the learning rates from lower than 1×10^{-3}, the results get worse across all metrics. The F1-Score decreases by over half from 0.35 to 0.15. The only measure which seemed to increase was the sensitivity at 1×10^{-6} which increased to 0.91, however, this is at the cost of the specificity, which decreases to 0.69, meaning many more false positives. So we have found that in the case of blinks for this method, the best performing learning rate is to keep 1×10^{-3}. Looking next at the results from 5.10, we can see that for changing the learning rates, the first three metrics, accuracy, sensitivity and specificity, all only change very marginally, so instead we focus on the F1-Score. So for the test set the F1-score increases as we change the learning rate, from 0.48 at the default to 0.79 at 1×10^{-5} and 1×10^{-6}. Then for the external validation set, we can see a similar story with the metrics where only really the F1-score is the metric which is changing and from that, we can see that the learning rate at 1×10^{-5} gives the best results at a 0.91 F1-score, which is good. So overall for the muscle artefacts, changing the learning rate did improve the performance of the method, and going forward we choose to use 1×10^{-5}.

The last modification which we will be looking at for the application of the LSTM approach is to test what is described in section 4.1.5.2, where we take the trained LSTM classifiers, but instead of using the output from the LSTM layer to go through a fully connected section to receive a classification, use the outputs from the LSTM layer as feature vectors to input into a linear classifier such as an SVM. For the testing of this it should be noted that we will be using the chosen input type and learning

rate for blink and muscle artefacts respectfully, so the crafted vector for both, then 1×10^{-3} and 1×10^{-5}.

We can see from Tables 5.11 and 5.12 the results from this approach. Looking first

Blink Artefacts				
Data Sets	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	0.82	0.67	0.91	0.74
External Validation	0.92	0.83	0.93	0.41

Table 5.11: A table showing the results of using the outputs from the LSTM layer as input vectors to a SVM for blink artefacts.

Muscle Artefacts				
Data Sets	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	0.99	1	0.997	0.83
External Validation	0.99	1	0.998	0.91

Table 5.12: A table showing the results of using the outputs from the LSTM layer as input vectors to a SVM for muscle artefacts.

at the table for blink artefacts from table 5.11, we can see that the results we have for the test set are similar or the same scores for the accuracy and F1-score, however, there seems to have been a trade for higher specificity by 0.07, for the sensitivity also lowering by 0.07, for the external validation set we see a marginal improvement across all metrics, most notably the F1-score which increased by 0.06. So for blink artefacts, using this extra approach does seem to be worth it for the benefits in the external validation. Looking next at table 5.12, we can see that the results are very good, with near-perfect accuracy, sensitivity and specificity, with the F1-score being the only place with improvement left, as there are still some false positives. For this approach there is a slight improvement for the F1-score in the test set, improving by 0.04, but no improvement in the external validation results, however, the results we already very good and any additional improvement would have been difficult. So for the muscle artefacts, since there is improvement in the test set results without losing anything in the external validation, we see this extra procedure as being worth it.

5.3.1 Further Improvements for Blink LSTM

As we can see from all the variations of results for the LSTM methods, the muscle artefacts perform much better than the blink artefacts. So with this room for improvement for the blink artefact results, we attempt to further optimise. The idea we will be testing is that since for blink artefacts we know that we can localise them spatially to the frontal region of the sensor array, however, we have been using information from all of the sensors for the tests. So if we restrict the input vector to only contain the information from the frontal sensors then we would be removing quite a lot of noise which may be interfering with the model's ability to predict. We will be testing this different input using the calculated max deflection vector and using a range of 4 different learning rates.

We can see the results for this test in Table 5.13. If we look first at the results for

Frontal Only Input for Blink Artefacts					
Data Sets	Classifier	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	1×10^{-3}	0.75	0.71	0.78	0.69
	1×10^{-4}	0.78	0.72	0.82	0.72
	1×10^{-5}	0.81	0.78	0.83	0.76
	1×10^{-6}	0.81	0.78	0.83	0.76
External Validation	1×10^{-3}	0.95	0.80	0.95	0.50
	1×10^{-4}	0.94	0.76	0.94	0.44
	1×10^{-5}	0.83	0.58	0.84	0.18
	1×10^{-6}	0.92	0.79	0.92	0.38

Table 5.13: A table showing the results for limiting the input to only the frontal sensors.

the test set we can see that this version performs worse for learning rates 1×10^{-3} and 1×10^{-4}, then about the same for 1×10^{-5} and 1×10^{-6}. This suggests that the learning rate to too high and the model fits too fast to a local minima. However then comparing to the external validation we can see that using the learning rates at 1×10^{-3} and 1×10^{-4} produce better results than previously seen, then learning rates 1×10^{-5} and 1×10^{-6} perform worse or about the same. This seems to suggest that a model which performs worse on the internal validation test set, does better for the external validation set, which could make sense as the internal validation contains, as mentioned, induced blinks which may be causing the models to generalise worse when they learn them. So overall using only the frontal sensors as information input does perform better for external but for the trade-off of internal performance.

5.4 Overall Results

In this section, we will be showcasing the overall results from the different final methods which we have looked at in detail to be able to give a final comparison and will be done. We will be showing the results from the different methods collected into a single table for both the test and external validation results. We will also be showcasing the actual labels and predictions for the windows for which we separated the recording with a visualisation of multiple binary graphs where it is one where an artefact has been labelled and zero for no artefact in the window. From these binary graphs, we will also take examples to be put through a topographic plot to provide spatial information on what the points that the different methods look like when they are identified as artefacts.

5.4.1 Blink Method Results

Overall Blink Results					
Data Sets	Method	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	Thresholding	0.76	0.84	0.71	0.72
	Max Point Deflection MLP	0.86	0.76	0.92	0.81
	LSTM as Classifier	0.8	0.74	0.84	0.74
	LSTM to SVM	0.82	0.67	0.91	0.74
External Validation	Thresholding	0.16	1	0.14	0.07
	Max Point Deflection MLP	0.98	0.85	0.98	0.70
	LSTM as Classifier	0.90	0.84	0.90	0.35
	LSTM to SVM	0.92	0.83	0.93	0.41

Table 5.14: A table showing the overall best results from the different final methods which we applied to locating blink artefacts.

We look first at the overall results for the blink methods, we can see these results from Table 5.14. From this table, when taking into account the metric which we are interested in the most, the F1-score, the Max point deflection MLP outperforms the other methods by a notable margin in both the internal test data and external validation data. The MLP also performs the best in specificity and second best in sensitivity, meaning that it provides low numbers for both false positives and false negatives. This result is a little surprising as we would expect LSTM to be able to generalise better or equally to the MLP approach. Looking next at the figures 5.5 and

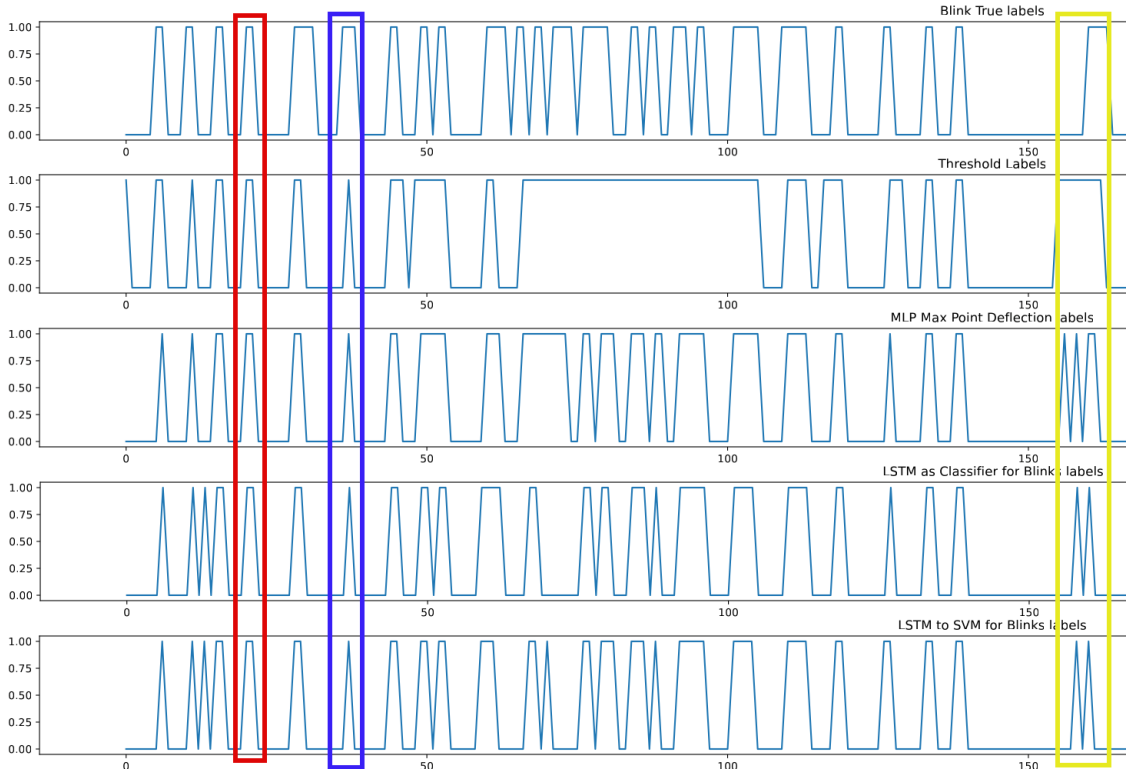


Figure 5.5: A graph showing the left-hand side of the visualisation of the predictions per window of the test recording for blink artefacts.

5.6, these show two halves of the same graphs. The top graph shows the binary steps of the actual labels which were used to compare and get the predictions, the following 4 graphs show the binary steps for the same 4 methods in the same order which is shown in table 5.14. One thing to note is that the x-axis represents the window number since each 200-step window is given the same classification. Looking first at an overall of the figure we can see that the number of blink artefact regions being identified is pretty high, then as you look across you can see that the different methods do seem to get the general locations correct for most of the labels and the downside being that the models seeming to struggle with either over or under classifying based on the true labels depending on the methods, as for the threshold method that seems to be over classifying regions quite a bit, which does make sense for the method but does indicate that maybe further threshold optimisation could improve it. Then on the other side, the LSTM models seem to under classifying many of the labels, this can be seen in the blue highlighted section. Another key overall point to make is that the classifications which look the worst in this recording are the ones between 50 - 100 on the x-axis, this is because this section contains the 4 or 5 induced blinks from

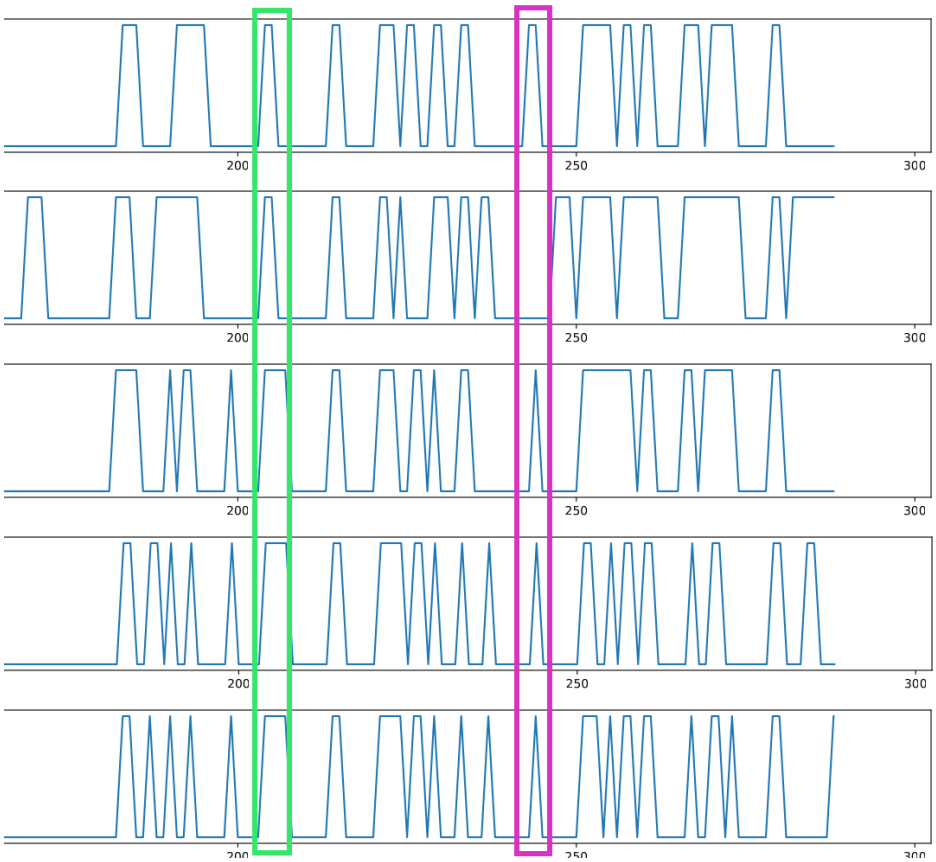
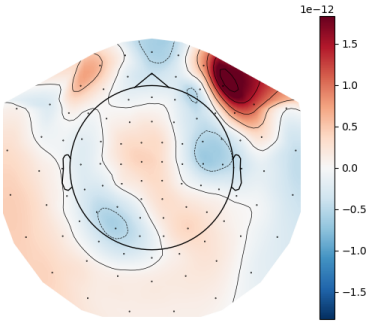
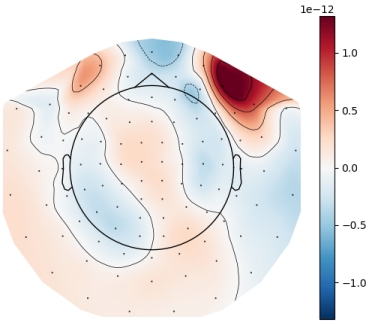


Figure 5.6: A graph showing the right-hand side of the visualisation of the predictions per window of the test recording for blink artefacts.

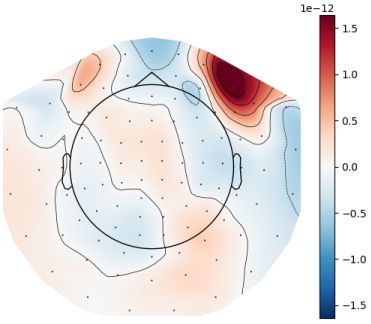
the subject, and since these can be so different in strength and time for which they last between subjects, we are less interested in how the methods do for these since they will not appear in external validation anyway.
Looking now at comparing the colour regions and the topographic plots in figure 5.7, we can see that immediately all 5 example blinks look similar, which is expected. The points to note from these are first that we see a much stronger intensity in the recording for the right frontal region, however, there is still some higher than average activity in the left frontal as well. This still shows that it is a blink, as this can happen in some subjects who are perhaps more right eye dominant and therefore have slightly more intense blink on that side. Another explanation is that before the scan anything metallic should be removed, however, if metallic mascara was not removed correctly from the right eye then this result could occur. Another point to note is that in figure 5.7 e), we can see that there are other darker regions other than only the top right, this seems to suggest that the blink artefact, in this case, is not strong in this example



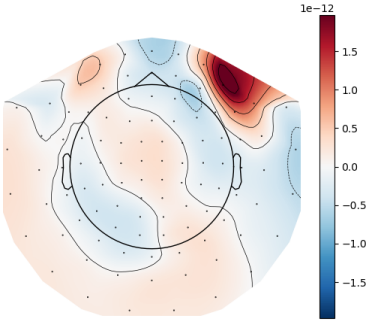
(a) Topographic plot for the Red highlighted block.



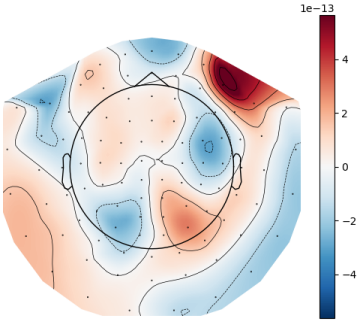
(b) Topographic plot for the Blue highlighted block.



(c) Topographic plot for the Yellow highlighted block.



(d) Topographic plot for the Green highlighted block.



(e) Topographic plot for the Pink highlighted block.

Figure 5.7: This figure shows multiple topographic plots of blink artefact predictions, from the colour highlighted sections in Figures 5.5 and 5.6.

when compared to this others, looking at the fact that this figure corresponds to the pink highlighted region in figure 5.6, we can see that the threshold method was not able to pick up this artefact suggesting that this blink was not strong enough to reach the threshold.

5.4.1.1 Blink Ground Truth Thresholding

In the previous section, section 5.4.1, we mention a key idea which we observed from figures 5.5 and 5.6 that the methods seem to get the general position of blinks but the labels appear to be longer than they should be due to how we added the ranging for the blink labels, which was to add a 100-time step radius around the max blink point which was identified by the expert. This process appears to have caused an additional window to be labelled in some cases, affecting our results.
So an additional step which we have tested is to have a post-processing procedure for the ground truth labels, where we decide on if the label should be changed based on, first, an amplitude thresholding per window using the threshold calculated in section 5.1, then second a new threshold for how much of the window will be above the first threshold to decide if the label should change.

Looking at Figure 5.8 we can see that as we increase the threshold for the number

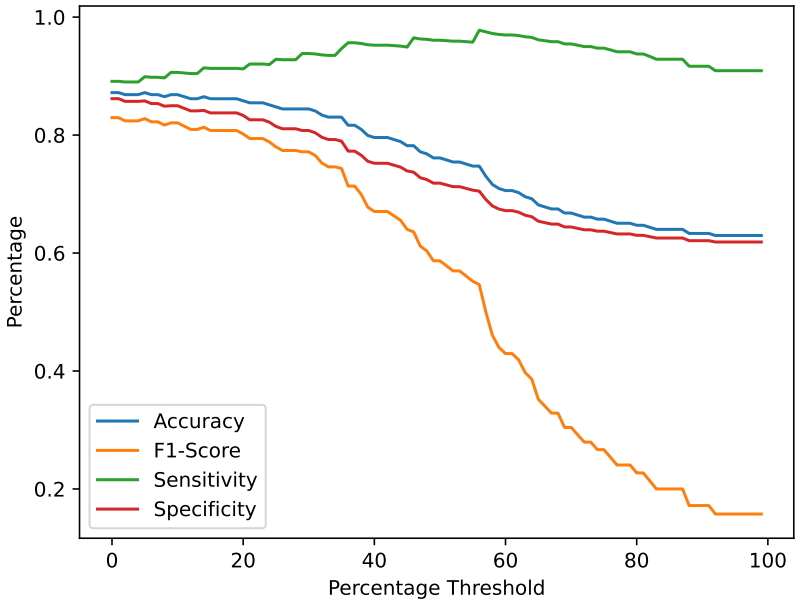


Figure 5.8: A graph showing the percentage accuracy, f1-score, sensitivity and specificity as the percentage threshold for deciding if a label is changed.

Ground Truth Threshold Results				
Data Sets	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	0.87	0.89	0.86	0.83
External Validation	0.98	0.85	0.98	0.70

Table 5.15: Showing the results for the MLP classifier after ground truth changes.

of points in the window to be above the threshold the score improves marginally, to begin with for the internal test data, as seen in Table 5.15, then the metric results get worse. So this approach seems to give slight improvement for checking labelled windows which don't have any points above the threshold but were labelled positive, so were not picked up by the model. The next step past this would be to retrain the models using the new ground truths, however since the increase in performance is marginal for changing the ground truth of the test set post-training, the best method for deciding a threshold for training data would still need to be found, which means we would need to optimise thresholds per training set, requiring retraining each for each test of a new threshold, which would take an unreasonable amount of time.

5.4.2 Muscle Methods Results

Overall Muscle Results					
Data Sets	Method	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	Thresholding	0.99	0.86	0.997	0.82
	Standard Deviation Vector SVM	0.998	1	0.998	0.91
	LSTM as Classifier	0.99	1	0.996	0.79
	LSTM to SVM	0.99	1	0.998	0.83
External Validation	Thresholding	0.99	1	0.997	0.87
	Standard Deviation Vector SVM	0.998	1	0.998	0.93
	LSTM as Classifier	0.99	1	0.998	0.91
	LSTM to SVM	0.99	1	0.998	0.91

Table 5.16: A table showing the overall best results from the different final methods which we applied to locating muscle artefacts.

For this next section, we will look at the overall results of the muscle methods. We can see these results from Table 5.16. From looking at the table it is clear to see

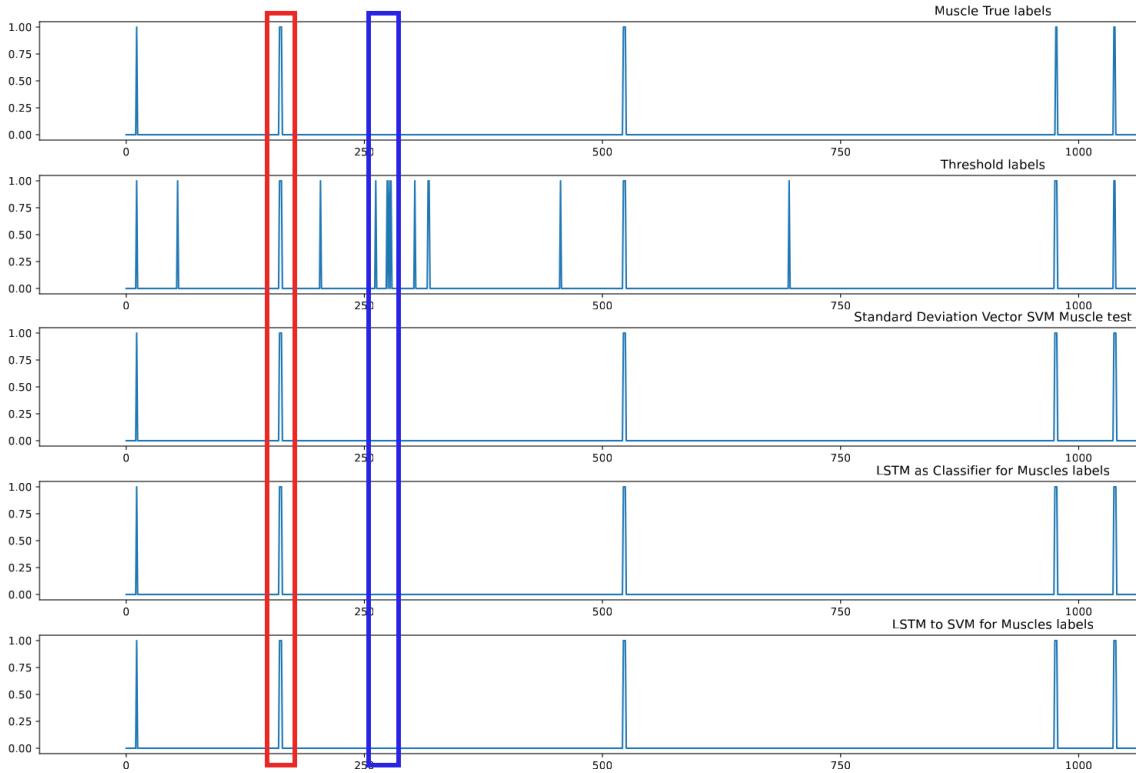


Figure 5.9: A graph showing the left-hand side of the visualisation of the predictions per window of the test recording for muscle artefacts.

that the only metric which shows any significant change across the different methods for both the test set and external validation set is the F1-score. Then using this we see that for the test set the standard deviation vector with SVM by itself performs the best with 0.91, then for the external validation, we see that the same method does the best again with 0.93, closely followed by both the LSTM versions. This result is perhaps slightly surprising as we would perhaps expect the LSTM methods to be able to generalise slightly better, however, the performance is likely more similar than it appears due to how low the number of true positives is, any single additional false positive or negative has a decent implication on the f1-score.
Moving now to comparing these results to the binary step visualisation graphs in Figures 5.9 and 5.10, as well as the topographical plots in Figure 5.11. Looking first at these binary step graphs a clear first point to note is that these show very well how less frequent the muscle artefacts are compared to the blink artefacts as shown in section 5.4.1. A clear pattern which can be immediately seen from these graphs is that the threshold method produces many more false positives than the other methods. This is due to the fact that if a single channel has a small spike, but that is the only

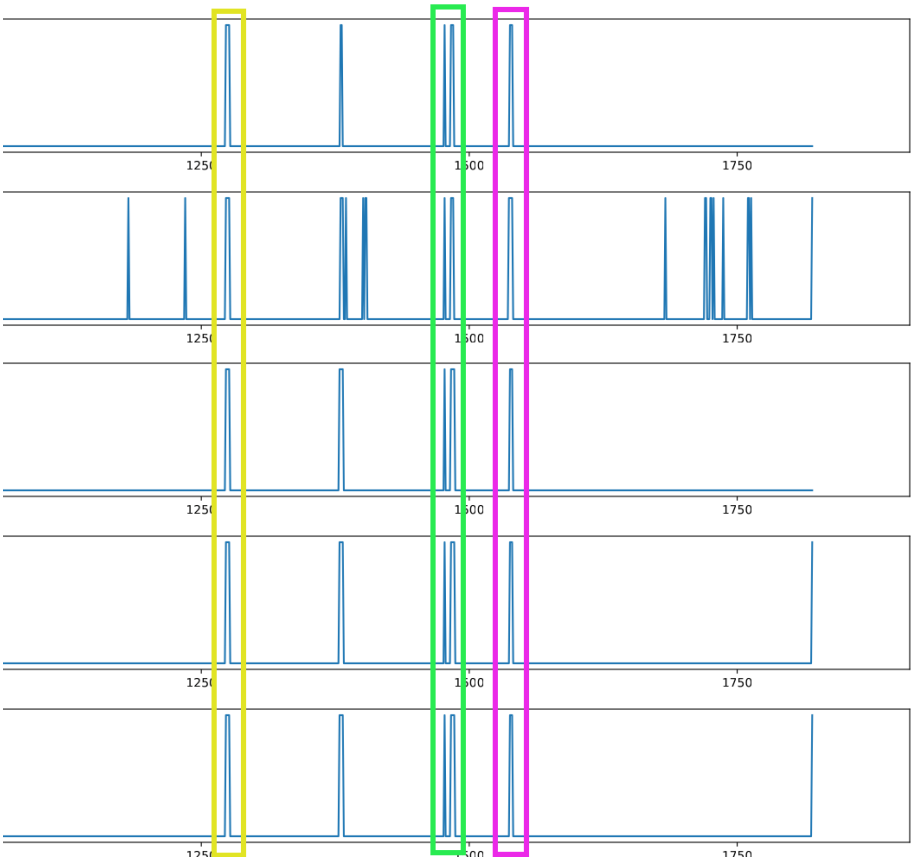
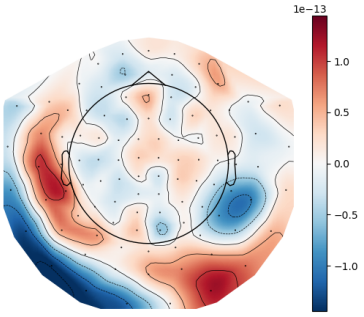
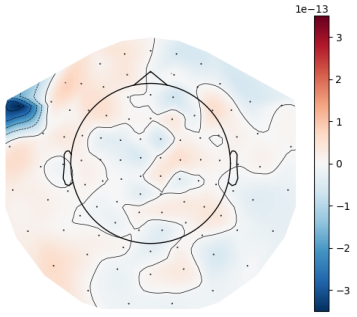


Figure 5.10: A graph showing the right-hand side of the visualisation of the predictions per window of the test recording for muscle artefacts.

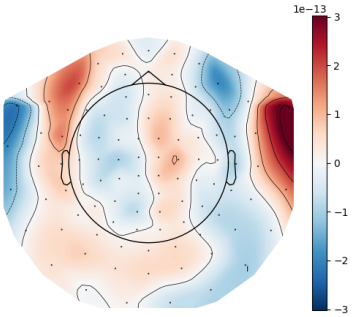
channel which picked up this spike, then it is likely an artefact of some sort but not a muscle artefact so it is not a part of our labelling, but since the threshold method has to look at every channel it will pick up these false positives. This can be seen well in the blue highlighted section, where the threshold method is the only one to have labelled this region as an artefact, then looking at the topographical plot in figure 5.11 b), we can see that there is only a peak in activity in a single channel in the left side of the sensor array. The other point to notice from these graphs is that both the SVM and LSTM methods perform practically identically from this perspective, with the only noticeable difference being a false positive at the end of the graph for the LSTM methods.
Looking at the other highlighted sections and their corresponding topographic plots, we can see that a) has high activity in the rear and left regions of the array, which suggests that it is a form of neck muscle artefact. Then with c) and d) there is higher than average activity on the far left and right of the sensor array, which suggests



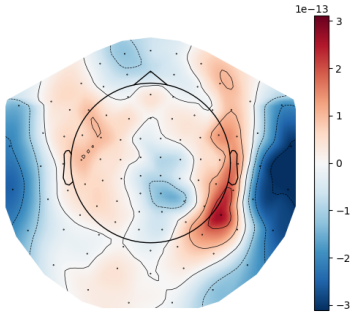
(a) Topographic plot for the Red highlighted block.



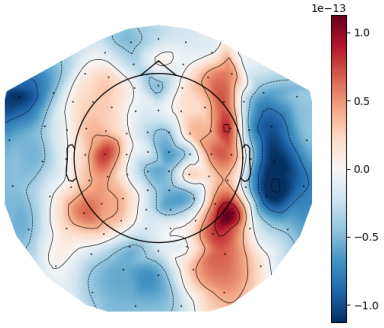
(b) Topographic plot for the Blue highlighted block.



(c) Topographic plot for the Yellow highlighted block.



(d) Topographic plot for the Green highlighted block.



(e) Topographic plot for the Pink highlighted block.

Figure 5.11: This figure shows multiple topographic plots of muscle artefact predictions, from the colour highlighted sections in Figures 5.9 and 5.10.

that the muscle artefact could be a jaw clench. The last topographic plot shown in e) shows a case where it is less clear what type of muscle artefact it is, as there is high activity on the far right, but then three or four more localised points of activity around the array, this all together suggests that the artefact is still a muscle artefact but that there may also be other points of activity which occurred at the same time as the artefact.

5.5 Separate Source Classification Results

For this section, we will be looking at the results which we have gathered from our tests on identifying the independent source components using different classifiers. The different classifiers which we will be using are MLP, SVM, KNN and CNN. Where, as described in section 4.2.1, the inputs for the first three of these methods will be the same, a 1-dimensional column vector from the mixing matrix of ICA used to represent the source, then for CNN the topographical plots for which these column vectors would create. A key point to note for the results of these methods is that since as mentioned in section 4.2.1, for each recording we are using 30 independent components which are labelled, and of these only between 3 - 8 are labelled as containing blink artefacts, this means that the effect of a single false negative or false positive will significantly affect the sensitivity and f1-score results due to the low number of true positives. However, this is the same for all methods so the results are still comparable.

We can see the results for these tests in Table 5.17. From these results, we can see

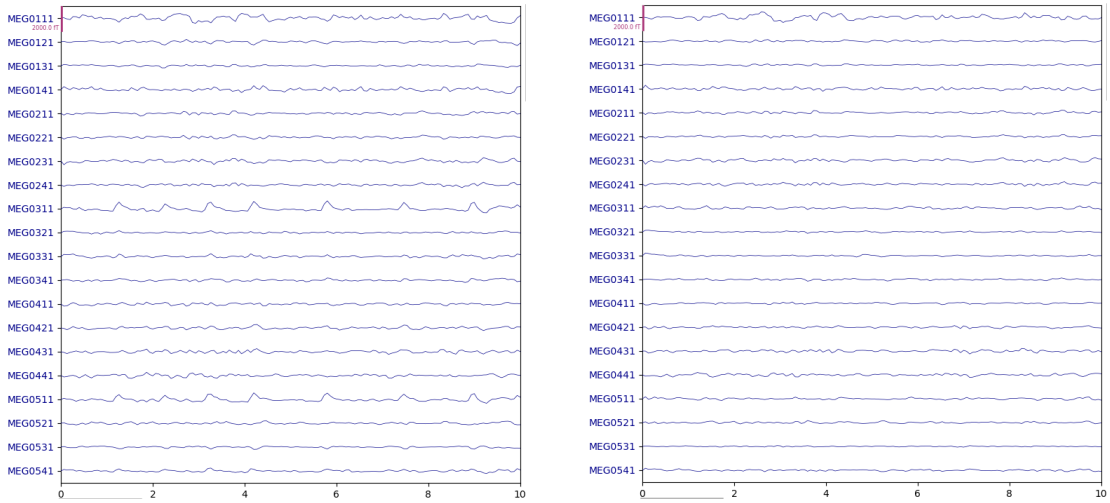
Source Separation Results					
Data Sets	Method	Accuracy	Sensitivity	Specificity	F1-Score
Test Set	MLP	0.87	1	0.82	0.78
	SVM	0.93	0.86	0.96	0.86
	KNN	0.87	0.71	0.91	0.71
	CNN	0.90	0.71	0.96	0.77
External Validation	MLP	0.97	0.75	1	0.86
	SVM	0.93	1	0.92	0.80
	KNN	0.97	1	0.96	0.89
	CNN	0.90	0.5	0.96	0.57

Table 5.17: A table which has the results for the tests for different classifiers for identifying the independent sources which contain the blink artefacts.

that overall all four of the approaches work rather well. Focusing first on the results of the test set we can see that SVM is the clear best performer with it reaching the best results in all four metrics, notably 0.93 accuracy and 0.86 f1-score. The next

best classifier is CNN, however, MLP does perform marginally better in the f1-score but this is likely not a significant difference. Then the worst performing classifier is the KNN. For the external validation set, the results switch, where the KNN classifier performs the best and highly across all four metrics, only being beaten by MLP in specificity which got a perfect 1, meaning no false positives. Then SVM performs slightly worse than these two, but CNN performed the worst with quite a low f1-score at 0.57. So overall we have a mixed level of performance across the two testing sets, we do value external validation higher than internal validation since that is what is being used for analysis, so perhaps KNN would be the best model, however as it performed the worst for the test sets, maybe the MLP model would be a better recommendation as it performed about second best for both internal and external validation.

For additional visualisation of how ICA can remove the blink artefacts via source



(a) Internal test recording channel plot before ICA Removal.

(b) Internal test recording channel plot after ICA Removal.

Figure 5.12: This figure shows the plots for all channels in the test recording before and after the source components were removed.

separation, in Figure 5.12, we show the before and after for each of the top channels (within limited space), before and after the source components which were identified by the MLP classifier were removed. We can see the main differences between these graphs in channel MEG0311 and MEG0511, as these are frontal sensors and you can see the blink artefacts, then graph b), you can see that they are gone.

Chapter 6

Discussion

The aims and objectives for this project were to go through, create and evolve our methodology for tackling the problem of locating and identifying blink and muscle artefacts, then evaluate the performance of these methods correctly to provide us with as much information as possible for us to make a well-informed decision. Through the results which we have provided these aims and objectives have been achieved, and through the following upcoming sections, we will discuss this.

6.1 Artefact Location and Method Evaluation

6.1.1 Locating Blinks

The analysis for methods created to locate blink artefacts was interesting as for the initial internal validation testing data, all of the methods gave good results with between 75% - 85% accuracy and 70% - 80% F1-score, with good results for sensitivity and specificity as well where different methods would usually trade some performance in one metric for doing better in the other. However when these same methods were tested on the external validation full recording testing set, the results became inconsistent as some methods struggled in this environment much more than others, in particular the thresholding method which we discussed the reason for in section 5.1, but to summarise the poor performance was because when z-scoring the length of the recording is going to affect the normalised values for the recording, meaning that since the value of the threshold was optimised for shorter recordings it does poorly on the full-length recordings. Going back to the external validation results, the only methods which seemed to adapt well to these tests were the classifiers which we used for evaluating the max point deflection vector, as both the MLP and SVM classifiers for this method achieved F1-scores of between 50% - 70%, which outperforms the

LSTM approaches and the attempts we made to improve the LSTM approaches. As with the LSTM methods they seem to max out at around 40% - 50%, with the best externally performing version of the method coming from using the frontal sensors as input to reduce noise for classification. These results are surprising and do go against what we would expect as we would expect that LSTM methods would be able to generalise the information provided better as the method is structured particularly well to understand the context of time series points and remember this context via the long and short form methods when it comes to predicting new points. From this, we think that perhaps a reason for an LSTM method doing worse is caused by a bottleneck from the labelling being done the way it is. As the labelling was done as if all N channels were 1, so if a window was labelled as containing an artefact, all N windows for that time frame would be labelled an artefact. We believe that if the labels were done per channel per window that this would have improved the performance of the LSTM methods as it would provide a greater detailed level of information as to what the artefact looks like anywhere in the recording, without the confusion of mislabelled channels for convenience sake. This addition would have also had the benefit of providing more detailed spatial information, as each channel corresponds to a spatial sensor, from beyond what we can achieve from the topographic plots alone. The reason why we did the labelling as it is was that it would not have been reasonable within the constraints of the project to label the required number of windows to train the models, for example, if we were to label each window per channel for the internal test set, that would require about $\sim 180,000$ labels. So overall, even though the LSTM methods did not perform as expected, the MLP max deflection method still shows good results on both internal and external data, showing the method is robust and appears to be able to generalise beyond the training data. Another point to note is that the LSTM methods do take much longer to train (approx 30 - 50 seconds per epoch depending on parameters) than the other methods which take a few seconds, but the testing time is practically negligible for all approach's so this is as a factor does not change any conclusions on the final method. However, it would be worth noting that an LSTM approach for testing new samples would be invariable to the size of the new samples, compared to a method like KNN which varies the testing time dependent on the amount of data.

6.1.2 Locating Muscle Artefacts

For the results of the methods used to locate muscle artefacts, these worked very well on both the internal and external validation sets as all the methods and improvement variations on those methods provides results, which in many cases, were near perfect compared to the same methods applied to the blink artefact data. As mentioned before

in the results many of the muscle artefact methods achieve accuracy in the high 90%'s, even though this is still good, it should be acknowledged the fact that the values are inflated largely by large values of true negative labels. For the threshold method using a threshold optimised using ROC via youden's index gives good F1-scores at 85% for both internal and external sets. For the sliding window methods, we tested using three different classifiers which again gave very good results which were consistent across both internal and external, with MLP perhaps actually underperforming compared to KNN and SVM. But SVM outperforms the others with between 91% - 93% F1-score for both internal and external validation, these values are very high and since we know that there are very few true positives compared to true negatives, for such a high f1-score, there must be very few errors, and since we have a max 100% sensitivity for both we know that there are no false negatives, so the only incorrect labels are false positives. Then we achieve similar levels of results through the LSTM methods after multiple iterations of optimisation, the best results we achieve come from the method of using LSTM outputs into an SVM, which gives 83% for the internal set and 91% for the external set. It should be noted that as seen in Figures 5.9 and 5.10, for the SVM and LSTM methods the only false positives come from some of the artefacts being predicted as lasting longer than what has been labelled, this is likely due to models struggling to find the boundaries of the artefacts, however still finding the general centre every time which is what our aim for these methods is. We did consider looking at ways to improve the boundary case of the artefacts however it was decided to not be a point of significant enough interest for the constraints of this project. So overall for the muscle artefact methods, the SVM with standard deviation vectors does perform the best, however, the margin between is quite small, which we can see visually, and as mentioned in the paragraph discussing the blink artefact analysis using per channel labels would likely improve performance of LSTMs here as well.

6.2 Classifying Artefact Sources

The additional objective which we set for this project was to look at methods for classifying source components, from using ICA as a blind source separation method to receive independent components. We tackled this aim by evaluating the approach using different classifiers as well as testing using a different type of input via CNN. Through this, we believe that we have demonstrated that you can accurately classify independent source components for removal. The testing itself between these different classifiers was interesting as some methods performed better on either the internal sets or the external sets. An example of this was the CNN method which achieved second best for the internal results with 77% F1-score but worst on the external sets

with 57%, suggesting that the use of the topographical plots as inputs for classifiable information is worse for the full recordings, this is interesting as it could suggest that using the topographic plots adds an extra layer of noise which interferes with the generalisability. Then the other example is that the KNN classifier is the best performer for the external validation with 89% F1-score but the worst performer for the internal test at 71%, this difference is interesting as if were to expect a difference you would usually expect it to be the other way around, a reason for this could purely be that since KNN is completely reliant on the K value, the default value we use may not be the best for the internal test set. Then for the last two classifiers both SVM and MLP perform interestingly again where MLP does well for the external and bad for the internal then SVM does the opposite. So overall we choose MLP as the best classifier for completing this objective of the project.

Chapter 7

Conclusions

7.1 Summary of Achievements

Taking all the work which we have shown in this report into account, we believe that the aims and objectives set out for this project have been met and that this project contributes to the research in this field of artefact detection and location for MEG. This project developed methods and ideas for the issue of locating artefacts, showing an iterative process through the development of initial exploratory ideas moving to more complex and involved methods, which is shown in our methodology as we moved from the cross channel correlation methods to thresholding and then using ideas of feature extraction for classifiers, then finally searching further improvement by exploring the deep learning approach in recurrent neural network LSTMs. As well as exploring the additional objective of identifying independent source components using multiple different types of classification and two different ways of using the input data. We also demonstrated significant evaluation of methods where for every method we made sure to test using both internal and external test sets to provide a more rounded view of the generalisability of the methods.

The overall results of these methods showed that the best methods are the feature extraction methods we showed, whether that's using the window max point deflection for blink artefacts, or the standard deviation of the window for muscle artefacts. Where for muscle artefacts we reached above 99% accuracy and above 90% F1-score for both internal and external validation, then for blink artefacts we reached 87% accuracy and 83% F1-score for internal and 98% accuracy and 70% F1-score for external. These overall results clearly show that the methods are able to locate these artefacts with significant accuracy as well as being significantly robust to external data. We have also discussed in our discussion some possible reasons why these classifiers outperformed the deep learning LSTM approach, and how the limitations

from the labelling of the recordings could have improved the methods, even though especially for the muscle artefacts the LSTM approaches did not underperform by a large margin. Then also the overall results for the method of identifying blind artefact sources gave results where the MLP classifier performed the most consistently across both internal and external sets giving, 87% accuracy and 78% F1-score for internal and 97% accuracy and 86% F1-score for external, meaning that choosing this option provides the most robust option between sets. Also, these results show that the method of using the mixing matrix column vectors as features can be used to identify the sources themselves.

Being able to locate and identify artefacts such as those discussed in this thesis in MEG recordings will provide ways to better understand the characteristics of these artefacts in different subjects, and how they influence the surrounding waveforms that are used to diagnose serious illnesses. Removing these artefacts will allow for less contaminated recordings using methods which are more explainable than those currently used in MYndspan, providing cleaner analysis and reducing the chance of misdiagnosis further down the pipeline.

7.2 Future Work

The main area of change which we would look at for future work to be done related to this project would be to use a more detailed labelling approach as, although it would take a significant amount of time to be able to label a reasonable number of recordings in order to train the models, the amount of additional information that would be provided for spatial information, as well as providing less noise in the number of sensors which do not show any artefact presence at the time point but are labelled as such due to another sensor at the same time having an artefact. Another point of interest which could be looked at in future work would be to focus more on isolating the boundaries of the artefacts, for a more accurate location. For efficiency, we limited the location to a window of set time steps. Overall this project and experience with MEG data and time series data, in general, required a lot of learning and understanding of previously unfamiliar topics, to where the results and methods which we came to feel rewarding.

References

Amidror, Isaac (2002). "Scattered data interpolation methods for electronic imaging systems: a survey". In: *Journal of electronic imaging* 11.2, pp. 157–176.

Cervantes, Jair et al. (2020). "A comprehensive survey on support vector machine classification: Applications, challenges and trends". In: *Neurocomputing* 408, pp. 189–215.

Chollet, Francois et al. (2015). *Keras*. URL: https://github.com/fchollet/keras.

Chu, Chia-Shang James (1995). "Time series segmentation: A sliding window approach". In: *Information Sciences* 85.1-3, pp. 147–173.

Clarke, John (1994). "SQUIDs". In: *Scientific American* 271.2, pp. 46–53.

Colah (2015). *Understanding LSTM Networks*. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 08/01/2022).

Croce, Pierpaolo et al. (2019). "Deep Convolutional Neural Networks for Feature-Less Automatic Classification of Independent Components in Multi-Channel Electrophysiological Brain Recordings". In: *IEEE Trans Biomed Eng* 66.8, pp. 2372–2380. ISSN: 0018-9294. DOI: 10.1109/TBME.2018.2889512.

Dash, Debadatta, Paul Ferrari, and Jun Wang (2020). "Decoding Speech Evoked Jaw Motion from Non-invasive Neuromagnetic Oscillations". In: *IJCNN*, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9207448.

Feng, Yulong et al. (2021). "An Automatic Identification Method for the Blink Artifacts in the Magnetoencephalography with Machine Learning". In: *Applied sciences* 11.5, p. 2415. ISSN: 2076-3417. DOI: 10.3390/app11052415.

Ferrante, Oscar et al. (2022). "FLUX: A pipeline for MEG analysis". In: *Neuroimage* 253, pp. 119047–119047. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2022.119047.

Gardner, Matt W and SR Dorling (1998). "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences". In: *Atmospheric environment* 32.14-15, pp. 2627–2636.

Garg, Prabhat et al. (n.d.). "Automatic 1D convolutional neural network-based detection of artifacts in MEG acquired without electrooculography or electrocardiography". In: *2017 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, pp. 1–4. ISBN: 1538631598.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Gramfort, Alexandre et al. (2013). "MEG and EEG data analysis with MNE-Python". In: *Frontiers in neuroscience*, p. 267.

Gunn, Steve R et al. (1998). "Support vector machines for classification and regression". In: *ISIS technical report* 14.1, pp. 5–16.

Gutierrez-Osuna, Ricardo (2002). "Pattern analysis for machine olfaction: A review". In: *IEEE Sensors journal* 2.3, pp. 189–202.

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Jones, Catherine M and Thanos Athanasiou (2005). "Summary receiver operating characteristic curve analysis techniques in the evaluation of diagnostic tests". In: *The Annals of thoracic surgery* 79.1, pp. 16–20.

Khan, Pritam et al. (2021). "Warehouse LSTM-SVM-Based ECG Data Classification With Mitigated Device Heterogeneity". In: *IEEE Transactions on Computational Social Systems*.

Kiruthika, NS and Dr G Thailambal (2022). "Dynamic Light Weight Recommendation System for Social Networking Analysis Using a Hybrid LSTM-SVM Classifier Algorithm". In: *Optical Memory and Neural Networks* 31.1, pp. 59–75.

Kumar, P Senthil et al. (2008). "Removal of ocular artifacts in the EEG through wavelet transform without using an EOG reference channel". In: *Int. J. Open Problems Compt. Math* 1.3, pp. 188–200.

Lipton, Zachary C, John Berkowitz, and Charles Elkan (2015). "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019*.

Litvak, Vladimir et al. (2010). "Optimized beamforming for simultaneous MEG and intracranial local field potential recordings in deep brain stimulation patients". In: *Neuroimage* 50.4, pp. 1578–1588.

Liu, Zhongming et al. (2010). "Large-scale spontaneous fluctuations and correlations in brain electrical activity observed with magnetoencephalography". In: *Neuroimage* 51.1, pp. 102–111. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2010.01.092.

Mikolov, Tomas et al. (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari, pp. 1045–1048.

Molla, Md Khademul Islam et al. (2012). "Artifact suppression from EEG signals using data adaptive time domain filtering". In: *Neurocomputing (Amsterdam)* 97, pp. 297–308. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2012.05.009.

Muthukumaraswamy, Suresh D (2013). "High-frequency brain activity and muscle artifacts in MEG/EEG: a review and recommendations". In: *Frontiers in human neuroscience* 7, p. 138.

Okada, Y., J. Jung, and T. Kobayashi (2007). "An automatic identification and removal method for eye-blink artifacts in event-related magnetoencephalographic measurements". In: *Physiol Meas* 28.12, pp. 1523–1532. ISSN: 0967-3334. DOI: 10.1088/0967-3334/28/12/006.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Rawat, Waseem and Zenghui Wang (2017). "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9, pp. 2352–2449.

Schölkopf, Bernhard (2000). "The kernel trick for distances". In: *Advances in neural information processing systems* 13.

Shoker, L., S. Sanei, and J. Chambers (2005). "Artifact removal from electroencephalograms using a hybrid BSS-SVM algorithm". In: *IEEE signal processing letters* 12.10, pp. 721–724. ISSN: 1070-9908. DOI: 10.1109/LSP.2005.855539.

Solomon Jr, Otis M (1991). *PSD computations using Welch's method.[Power Spectral Density (PSD)]*. Tech. rep. Sandia National Labs., Albuquerque, NM (United States).

Srinivasan, Ramesh (1999). "Methods to improve the spatial resolution of EEG". In: *International journal of bioelectromagnetism* 1.1, pp. 102–111.

Srinivasan, Vairavan, Chikkannan Eswaran, and Natarajan Sriraam (2007). "Approximate entropy-based epileptic EEG detection using artificial neural networks". In: *IEEE Transactions on information Technology in Biomedicine* 11.3, pp. 288–295.

Tortora, Stefano et al. (2020). "Deep learning-based BCI for gait decoding from EEG with LSTM recurrent neural network". In: *J. Neural Eng* 17.4, pp. 46011–046011. ISSN: 1741-2560 1741-2552. DOI: 10.1088/1741-2552/ab9842.

Treacher, Alex H et al. (2021). "MEGnet: automatic ICA-based artifact removal for MEG using spatiotemporal convolutional neural networks". In: *NeuroImage* 241, p. 118402. ISSN: 1053-8119.

Van Dellen, Edwin et al. (2014). "Epilepsy surgery outcome and functional network alterations in longitudinal MEG: a minimum spanning tree analysis". In: *Neuroimage* 86, pp. 354–363.

Vigário, Ricardo et al. (2000). "Independent component approach to the analysis of EEG and MEG recordings". In: *IEEE transactions on biomedical engineering* 47.5, pp. 589–593.

Wall, Michael E, Andreas Rechtsteiner, and Luis M Rocha (2003). "Singular value decomposition and principal component analysis". In: *A practical approach to microarray data analysis*. Springer, pp. 91–109.

Wheless, James W et al. (2004). "Magnetoencephalography (MEG) and magnetic source imaging (MSI)". In: *The neurologist* 10.3, pp. 138–153.

Winkler, Irene, Stefan Haufe, and Michael Tangermann (2011). "Automatic classification of artifactual ICA-components for artifact removal in EEG signals". In: *Behav Brain Funct* 7.1, pp. 30–30. ISSN: 1744-9081. DOI: 10.1186/1744-9081-7-30.

Yildirim, Özal (2018). "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification". In: *Computers in biology and medicine* 96, pp. 189–202.

Yuan, Zou, Viswam Nathan, and Roozbeh Jafari (2016). "Automatic Identification of Artifact-Related Independent Components for Artifact Removal in EEG Recordings". In: *IEEE J Biomed Health Inform* 20.1, pp. 73–81. ISSN: 2168-2194. DOI: 10.1109/JBHI.2014.2370646.

Zhang, Min-Ling and Zhi-Hua Zhou (2007). "ML-KNN: A lazy learning approach to multi-label learning". In: *Pattern recognition* 40.7, pp. 2038–2048.

Appendix A

Project Specification – Identification and Removal of Muscle Artefacts appearing in Magnetoencephalography signals

Ethan Jolly – Student ID: 35982495

Project hosted by MYndspan

15/06/2022

Background and Motivation

Magnetoencephalography (MEG) scanners uses functional neuroimaging techniques for mapping brain activity, instead of directly mapping the structure as, for example, is done in MRI scanners. MEG maps the activity by recording the magnetic fields which are naturally produced from electrical currents in the brain. To take this recording approximately between 100 – 275 individual sensors placed around the brain. These sensors are comprised of arrays of both magnetometers and gradiometers, where magnetometers are used to directly record the magnetic field and gradiometers are used in MEG to measure the gradient of the magnetic field detected.

These signals collected from MEG scans can be analysed across each time-series signal produced from the different sensors to provide insight for a variety of neurological disorders and illnesses, such as concussions and PTSD. This project would be involved in implementing a system in to MYndspans existing pipeline, which

adds the processing step of identifying the location of possible muscle artefacts in the signals, quantify the artefacts by using statistics to measure the artefact duration and intensity as well as comparing the artefact to the induced examples to see what was most likely the cause. Identifying and locating these artefacts will be the key challenge of this project, but if complete in reasonable time we could look at possibly removing muscle artefacts from the signal, which may not always be possible while retaining the signal integrity, so the best course of action could just be to crop out the identified, with the hope being that with a cleaner signal it should be able to provide better insights. To provide a clearer idea of the analysis done by MYndspan, after each scan the subject receives a short report providing a baseline brain health assessment as well as a more detailed power analysis, which provides activity insights based on different frequency ranges which are known to be attributed to different states of brain health. These frequency ranges which are looked at are Delta(1-3Hz), Theta(4-7Hz), Alpha(8-12Hz), Beta(13-25Hz) and Gamma(<25Hz).

Data

The data for this project will be subject data recordings, first being a recording where the subject is asked to induce certain muscle artefacts which are known to appear in the MEG recordings, these are eye blinks as well as jaw clenches. These initial induced recordings will be used to provide a baseline for this subject for what the artefacts may look like in their case. The next set of data will be the actual subject recordings, which can take anywhere between 5 – 15 minutes depending on what was specified beforehand, where the subject is staying as stationary as possible while keeping their focus on a small white cross on a black screen. All data which will be used for this project will be provided by MYndspan, which they have received consent from the subjects for the data to be processed and analysed. The data itself will be time-series data of the amplitude of the magnetic signal recorded, with each of the 200 sensors providing a different time series. Also before we can start looking for artefacts in the data, crucial pre-processing steps of the data which will require domain knowledge of the scanners operations will need to be carried out on each data set, one step would be to identify and remove the head localization spike in the data (which is used as a pre-scan procedure to make sure the subjects head is in a correct position), as well as either applying projection matrices to the data or gradient compensation matrices depending on if the manufacturer of the scanner is MEGIN or CTF. Another step which requires domain knowledge would be to remove artefacts which are caused by heating elements in the scanner, these cause the recorded levels of the magnetic field to be well above anything that would be physiologically possible. Performing these steps will help to make the whole process more generalizable to different scanners. This project will be implemented in Python, with most of the work done using the

MNE-Python library, but will also make use of other scientific python libraries such as numpy, scipy and matplotlib.

Main Project

The main outline for the project is to propose a system which takes in multi-channel time-series data, locates muscle artefacts, quantify the artefacts, decide on a removal strategy and finally output the signal. As already mentioned a successful implementation of this system would be a useful asset to the MYndspan pipeline as it is not something which has been directly implemented and would avoid any problems that the noise from artefacts currently cause.

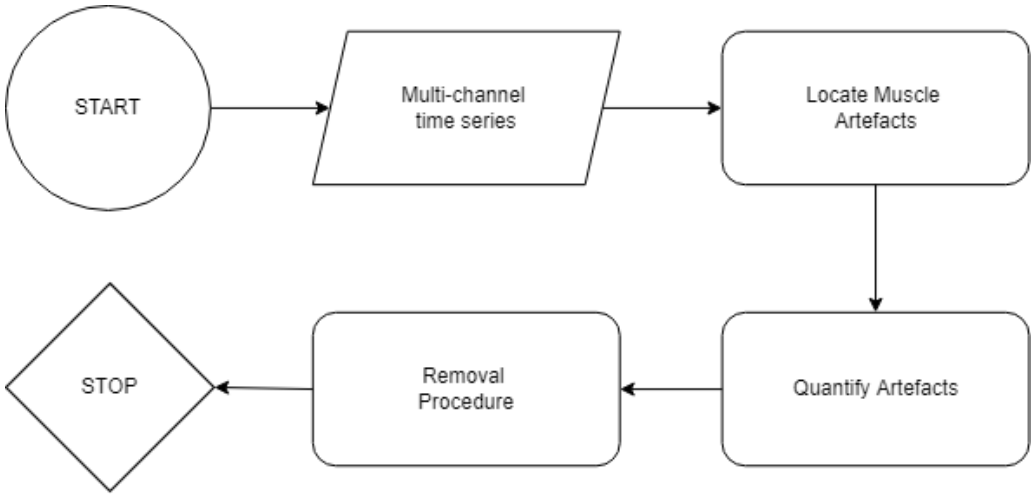


Figure A.1: Flow chart diagram showing the flow of data input, processing the artefacts and then outputting the results.

Locating and Quantifying Muscle Artefacts

The questions which we need to answer when identifying the artefact are: what is the spatial locale of the artefact?, what is the temporal locale? and what is the artefact? In order to successfully answer the question of locating the temporal locale we can use cross-channel correlation at given time periods and frequency ranges, using these correlation patterns we can then compare them against what we know as standard muscle artefact correlation patterns from the induced subject recordings, completing this comparison allows us to also understand the spatial locale by knowing which sensors are much more active than the others, this can also be done visually in topographical plots of the sensors around the head. Performing the

this cross-channel correlation, would likely require an implementation of a sliding window algorithm, where we would take a moving range on the time-series to look at comparison. However, this approach may have limitations as it could be heavily parameter dependent, for example the size of the window or level of correlation, so further approaches will be explored. Then for the question of what is the artefact, not only will we want to label the artefact to a known cause i.e. Jaw clench, eye blink. We want to quantify what the artefact is by creating statistical descriptions of the artefact, for example these could be average and max amplitudes of the artefacts, average frequency range of the artefact and total length of the artefact. These statistical details can be used to compare against other artefacts in order to provide a clearer picture and more evidence to identifying the artifact. Additionally we could look at using independent component analysis (ICA), we could do this by taking the components produced by ICA, and check if the cross-channel correlation for the component is similar enough to the induced muscle artefact recordings.

Removal Procedure

As mentioned already the standard removal procedure would be to crop out the time length that was determined to contain an artefact, as this would ensure that the remaining signal still maintains the integrity of the final signal, to only contain brain activity. An additional removal procedure to look in to could be to isolate the time range of the artefact, then perform ICA on this smaller section, and remove the components which contain the artefact, however this may not always be possible for the different types of artefacts. Depending on the success, and time scale of completing the identification task, changes may be made to the aims for the removal procedure, for looking in to more methods.

Timeline

June:

- **Initial Analysis:**
  - Develop enough understanding of the domain knowledge, required libraries and methods.
  - Develop understanding of the signal data and how to manipulate it.
- **Identify artefacts:**
  - Complete initial Literature review.
  - Be able to manually identify the artefacts in the induced recordings.

July:

  - Implement statistics for descriptions of the artefacts.
  - Look at alternate methods of identifying artefacts.
  - Implement identifying muscle artefacts in actual subject recordings.

August:

  - Completion of artefact identification and localisation methodology

- **Removal Procedure:**
  - Look in to best removal procedure based on scope remaining
  - Implement best procedure

Deliverables

- A recommended approach to identifying and localising artefacts spatially and temporally using MEG waveform data.
- Produce a dissertation that covers the entire approach of the project.
- Present the approach and findings to MYndspan.
- Create a poster outlining the project for the poster session.