

Using Machine Learning to Classify Stutter Events using fNIRS and EEG data

UCL, MSc Psychological Sciences 2018-2019

Candidate number: BNBM3

Supervisor: Professor Peter Howell

Word count: 11,762

Acknowledgements

This dissertation is dedicated to Buzzy Sadek. Thanks for being so kind, patient and supportive. And thanks for making me laugh.

I'd like to say a huge thank you to Professor Pete Howell who supervised this project. Peter, you've been so generous with your time. Thanks for all the guidance and advice, and for helping me initiate this project. I've really enjoyed it!

I'd also like to thank Claudia Contadini-Wright, Shannon McCann and Antonia Toneva for painstakingly collecting the data on which this project was built. And, also to Naheem Bashir for helping me navigate the data, and for being so open to helping from start to finish. I would have been really stuck without you!

Lastly, thanks to my amazing parents Myra and Neil for rearing me. I'm really glad you did that.

Contents

Glossary

Abstract

1. <u>Introduction</u>	10
1.1 <u>Stuttering and its causes</u>	10
1.2 <u>Neural regions associated with stuttering and analysis methods used</u>	11
1.3 <u>The gap in knowledge</u>	13
1.4 <u>Machine learning versus other statistical methods</u>	16
1.5 <u>fNIRS & EEG: The advantages and drawbacks</u>	23
1.6 <u>Hypotheses</u>	28
2. <u>Methods</u>	29
2.1 <u>Sample</u>	29
2.2 <u>Experiment procedure</u>	30
2.3 <u>Data pre-processing</u>	36
2.3.1 <u>fNIRS pre-processing</u>	36
2.3.2 <u>EEG pre-processing</u>	42
2.3.3 <u>Time-syncing and labelling the data</u>	49
2.4 <u>Analysis</u>	53
2.4.1 <u>Data combinations</u>	53
2.4.2 <u>Data formats</u>	54
2.4.3 <u>Machine learning classification</u>	55
3. <u>Results</u>	55

Classifying Stutter Events using fNIRS and EEG data

3.1 Four participants combined analysis (fNIRS and EEG)	55
3.2 IC versus channel data (EEG)	57
4. <u>Discussion</u>	58
<u>References</u>	69
<u>Appendices</u>	84
1. Neural regions most associated with stuttering - expansion.....	84
2. Neuronavigation data example.....	86
3. EEG electrodes and related neural regions.....	86
4. Experiment procedure - full.....	87
5. Time points and the time-syncing procedure.....	90
6. Python script for labelling the data.....	92
7. Python script for machine learning classification.....	94
8. Variable importance measures (VIMs)	107

Glossary

Area under the curve (AUC): A metric by which to assess the performance of a machine learning classifier across all classes (i.e., how well it correctly predicts both positive and negative outcomes).

Feature: The name for predictor variables in machine learning.

Feature selection/pruning: The process of reducing the number of predictor variables to reduce dimensionality and improve accuracy.

Hold-out method: The process of manually splitting the data into a test set (usually around 25%) and a training set (usually around 75%) for the machine learning classifier.

Independent component analysis (ICA): A method of transforming EEG channel data into maximally independent components for analysis. Produces independent components (ICs).

Imputation: The process of replacing missing data with substitute values.

K-fold cross validation: A method whereby the training set and test set in the data is rotated to artificially create a bigger data set from which the machine learning classifier can learn.

K-Nearest Neighbours (KNN): A machine learning classification algorithm. The idea is to search for closest match of the test data in feature space (Goel and Mahajan, 2017). Generally, KNN is used for multi-class problems, but it also can be used on binary problems.

Logistic Regression: A parametric machine learning classification algorithm which assumed independence between predictor variables.

Non-parametric classifiers: Classifiers which do not make assumptions about the data such as independence between predictor variables.

Overfitting: a common problem in machine learning, where a model performs well on training data but does not generalize well to unseen data (test data).

Random forest (RF): A machine learning classification algorithm which combines multiple decision trees.

Sensitivity: A metric by which to assess how well the classifier correctly predicts positive outcomes. A measure of the true positive rate.

Specificity: A metric by which to assess how well the classifier correctly predicts negative outcomes. A measure of the true negative rate.

Support Vector Machine (SVM): A machine learning classification algorithm. It works by creating a line or a hyperplane which separates the data into classes.

Variable Importance Measure: A measure of how much each variable contributes to a classification.

Classifying Stutter Events using fNIRS and EEG data

Abstract

Objective: The aims of this study were 1) to determine whether supervised machine learning classifiers can be used to classify stutter events from neural data, and 2) to investigate best practices for doing so in terms of optimal machine learning classifiers (for example random forest, support vector machine), data structures (high volumes of data versus balanced data) and data types (for example independent EEG components (ICs) versus channel data as predictor variables). In being able to classify stutters that have already occurred, it is hoped science can move towards real-time prediction of stutters (in advance of their occurrence), and as such, prediction-based interventions (using stimulation such as tDCS).

Sample: The baseline data set comprised of concurrent fNIRS, EEG and audio data from four male participants who displayed overt audible symptoms of childhood onset fluency disorder (COFD). Each participant carried out two verbal tasks, generating a total of 8 separate data sets (each of around fourteen thousand 0.1 second samples).

Method: Four machine learning algorithms (K-Nearest Neighbour, Support Vector Machine, Logistic Regression and Random Forest) were applied to several different compositions of the data including large volumes of data (8 data sets combined) with unbalanced classes (around 10% stutter events and 90% non-stutter events), and smaller sets of balanced data (50% stutter and 50% non-stutter). For EEG, data sets were also compared in which the machine learning features were either independent components generated through ICA (independent component analysis), or EEG channels (the pre-processed but un-transformed readings taken from the electrodes).

Findings: The balanced data (50% stutter, 50% non-stutter) performed better across both classes and so was used throughout. The best performing machine learning algorithm was random forest on the fNIRS data which correctly classified stutters versus non-stutters with an accuracy (measured by AUC) of 0.90. The third joint highest performers were the KNN and the SVM on the EEG IC data and the SVM on the full EEG set of EEG channel data, all with an AUC 0.67. This suggests machine learning can classify stutters through fNIRS data which is a critical step towards prediction-based intervention. Stutters could potentially be classified through EEG data with some adjustments. These are explained in the discussion section.

1. Introduction

1.1 Stuttering and its causes

Childhood onset fluency disorder (COFD), more commonly known as “stuttering”, is a neurodevelopmental communication disorder that disturbs the normal fluency and time patterning of speech (DSM-5 American Psychiatric Association, 2013; Carlson, 2013).

COFD is characterised as audible repetition or prolongations of sounds, syllables or words (e.g., “I-I-I-I see him”) or by frequent hesitations, pauses or silent blocks that disrupt the rhythmic flow and timing of speech production (Van Riper, 1982; Guitar, 2006).

Up to one in ten children are likely to stutter (Reilly et al., 2013), and 80-90% of these children start stuttering by the age of 6 years old, with onset ranging from 2-7 years (DSM-5 American Psychiatric Association, 2013; Carlson, 2013). Proposed causes of childhood stuttering include genetics (Howie, 1981; Andrews, Morris-Yates, Howie & Martin, 1991; Felsenfeld et al., 2000) and environmental factors which affect neurological development, such as stressful or traumatic experiences and family communication style (Ambrose, Cox & Yairi, 1997; Ward, 2006; Guitar, 2006; Bloodstein & Ratner, 2008; Yairi & Ambrose, 2013).

Most children grow out of stuttering by adolescence (Brown, Ingham, Ingham, Laird & Fox, 2005). However, around 1% of adults stutter (Andrews, 1964; Bloodstein, 1995; Felsenfeld, 2002; Brown et al, 2005).

Classifying Stutter Events using fNIRS and EEG data

Several neural processes are involved in speech production starting from auditory processing to motor planning and motor execution (Guenther & Gosh, 2003; Tourville & Guenther, 2011). If any of these neural regions and the connections between them is compromised, there is potential for stuttering (Beal, Gracco, Lafaille & Nil, 2007; Watkins, Smith, Davis & Howell, 2008).

1.2 Neural regions associated with stuttering and analysis methods used

Many studies have been carried out showing neurological differences between people who stutter (PWS) and fluent speakers (FS). The neural regions where these differences have been seen is summarised in Table 1 below.

Table 1

A summary of studies investigating the neural regions associated with stuttering.

Study	Data type	Statistical Method Used	Brain Region	Findings
Bashir (2019)	fNIRS	T-test, independent sample comparing differences between PWS and FS groups.	LIFG (Broca) RIFG	Showed that PWS had significantly greater neural activity in the LIFG compared to FS in the Social Conversation condition.
Wu et al (1995)	PET	T-tests, one and two tailed between PWS and FS groups	Wernicke LIFG (Broca) Left and right superior frontal lobes Right cerebellum Left deep frontal orbital Bilateral posterior	Decreased activity (decreased uptake of glucose) during a stuttering condition* in the LIFG, left Wernicke's area, left and right superior frontal lobes, right cerebellum, left deep frontal orbital, bilateral posterior cingulate cortex. (No areas showed <i>greater</i> activity during stuttering). *The “stuttering condition” also included non-stuttering events so is limited in relation to detecting biomarkers of stutters specifically.

Classifying Stutter Events using fNIRS and EEG data

			cingulate cortex	
Neumann, Gudenberg, Lanfermann & Preibisch (2003)	fMRI	T-test comparing differences between PWS and FS groups	Right frontal operculum (RFO)	RFO showed significantly higher activation than FS at every assessment time
Walsh et al (2017)	fNIRS	ANOVA comparing F and P-values between PWS and FS	LIFG PmC	PWS demonstrated deactivation over left dorsal IFG and left PmC. FS showed clear activation over left dorsal IFG and left pre-motor cortex (PmC), characterized by increases in Oxy-Hb and decreases in Deoxy-Hb. Non-significant differences were found for STG.
Sommer, Koch, Paulus, Weiller & Büchel (2002)	Voxel-based morphology	Two- sample t test to compare diffusion characteristics between FS and PWS groups.	Left sensorimotor cortex	Diffusion characteristics between FS and PWS differed significantly immediately below the laryngeal and tongue representation in the left sensorimotor cortex.
Brown et al (2005)	Various	<p>Meta-analysis using ALE to identify which areas were more or less active during stuttered versus fluent speech.</p> <p>Voxel-based meta-analysis.</p> <hr/> <p>Studies comparing stuttered speech to fluent speech in stutterers.</p>	<p>Right frontal operculum/anterior insula</p> <p>Auditory areas bilaterally</p> <p>Vermal region of lobule III of the cerebellum</p> <p>Primary motor cortex</p> <p>Supplementary motor area</p> <p>Cingulate motor area</p> <p>Cerebellar vermis</p>	<p>Found three neural signatures that seemed more or less specific to the stuttering group: (1) overactivation in the right frontal operculum/anterior insula; (2) absence of activation in auditory areas bilaterally; and (3) overactivation in the vermal region of lobule III of the cerebellum.</p> <p>The analysis showed that similar brain areas are involved in stuttered speech as in fluent speech, but with some important differences. Motor areas were over-activated in stuttering, including primary motor cortex, supplementary motor area, cingulate motor area, and cerebellar vermis</p>

Classifying Stutter Events using fNIRS and EEG data

Budde, Baron &Fox (2014)	fMRI & PET	Meta-analysis using ALE.	Same as the ALE by Brown et al (2005) plus: SMA	The previously reported neural signatures of PWS (Brown et al., 2005, ALE) were robustly confirmed, extended to previously unidentified regions including SMA.
Neef, Anwander, Paulus & Sommer (2016)	fMRI	Z significance testing between PWS and FS	Posteriordorsal area 44	Evidence of reduced activation of left posteriordorsal area 44. Functional coupling between left posterior area 44 and left inferior parietal lobule was deficient in stuttering.

Overall, it would seem that over-activation of motor areas and under-activation of the LIFG are most strongly related to stuttering. Neural regions associated with stuttering are discussed in more detail in appendix 1.

1.3 The gap in knowledge

As seen in Table 1, most studies to date compare differences between groups of PWS and FS, usually using T-tests, and as such, there is a lot of evidence showing how brain region activity varies between PWS and FS. There have also been studies examining differences within groups of PWS, such as stutter rates in response to different conditions (for example Mock, Foundas, & Golob, 2016). However, very little has been done to find the real-time neural biomarkers of a stutter itself (these studies will be reviewed shortly). In other words, not just looking at which parts of the brain are involved in stuttering, but also looking at how activity differs in those parts, in real-time, between a stutter and fluent speech. There is ample evidence to suggest that brain activity differs in PWS to FS when they are talking, but within PWS, does it vary significantly when they are stuttering versus when they are not?

Classifying Stutter Events using fNIRS and EEG data

A biomarker is a measurable indicator of some biological state or condition. Biomarkers can indicate a disease (Boksa, 2013), for example, lower superior frontal gyrus grey matter volumes have been found as potential biomarkers of schizophrenia (Kumra et al. 2011). However, biomarkers can also represent a more temporary state. For example, increased heart rate is known to be real-time a biomarker of stress (Schiweck, Piette, Berckmans, Claes & Vrieze, 2019).

If we can determine exactly which biomarkers (in this case neural conditions) are in place when a stutter occurs to the extent that we can classify it by analysing neural data, we can also move towards *predicting* when a stutter is about to occur by analysing the seconds before the stutter started. Being able to predict stutters in advance of their occurrence could play a significant role in interventions. For example, transcranial direct current simulation (tDCS) has been shown to improve speech in people who stutter (Bashir, 2019; Watkins & Möttönen, 2017). In theory if a tDCS device (or any other appropriate cranial stimulation device) could be fed information from a neural monitoring technique that can predict when a stutter is about to occur (such as fNIRS or EEG), it could be programmed to stimulate brain regions just in time to prevent or reduce the severity of a stutter. Additionally, because the stimulation would only occur just before a stutter, it could allow the use of a wider variety of interventions, for example types of stimulation that are only safe in small quantities but not if they are applied continuously.

In the context of current technology, this type of prediction-based intervention would only be possible in a lab environment, or in a social situation where the PWS wears highly obvious head gear. However, if strong evidence can be found to suggest prediction-based stutter intervention is scientifically possible, hardware manufacturers may be more incentivised to

Classifying Stutter Events using fNIRS and EEG data

develop more discrete and portable devices that could, for example, fit inside a baseball cap. The company Shmidzu already sell comparatively discrete and portable fNIRS devices where the probes are connected to the fNIRS machine in a backpack (Shimadzu.com), and many types of tDCS are portable (for example, The Brain Driver, 2019).

To be able to do any of this however, clear evidence is needed showing functional neurological differences between a stutter and fluent speech in PWS. The most relevant study in this area is by Myers, Irani, Golob, Mock, & Robbins (2018) who developed a stepwise Linear Discriminant Analysis (sLDA) algorithm (a machine learning classifier) that could predict stutters in 81% of trials in two subjects (N=2). The data input was independent components (ICs) from EEG data taken just before the start of an utterance (using channels as features wasn't discussed). Variance was associated with the LIFG, RIFG, right pre-motor cortex and left auditory cortex. A limitation of this study was that it only reported overall classifier accuracy (the percentage of correct predictions overall) and did not report on the accuracy of correctly predicting positive classes (known as "sensitivity") versus the accuracy of correctly predicting negative classes (known as "specificity"). This means the results could have been skewed by an unbalanced data set (for full explanation of this problem, see "Machine learning classifiers" section later in the introduction and "Discussion"). The percentage of stuttered versus fluent trials was not reported in this study. As such, this study provides limited evidence to support the theory that stutters can be predicted by applying machine learning to EEG data.

It would appear that no other research has been done using machine learning on fNIRS, EEG, or any other neural data, to identify or predict a stutter versus fluent speech. However, machine learning has been used on neural data to find biomarkers of other aspects of

Classifying Stutter Events using fNIRS and EEG data

stuttering (as well as many other areas of neurology such as epilepsy (Subasi & Ercelebi, 2014)). Hosseini, Walsh & Shouyi (2016) for example, used machine learning on fNIRS data to detect biomarkers of stuttering *persistence*, i.e., to neurologically determine whether a child's stuttering will persist into adulthood. Several algorithms were compared including K-nearest neighbour (KNN), support vector machine (SVM) and decision trees. The decision tree performed best on the raw data (before feature pruning had been carried out) with an accuracy of 77.5%. However, after using mutual information sparse feature selection (MISS), the SVM was capable of differentiating neural activation patterns between children who do and do not stutter with an accuracy of 87.5% (based on a five-fold cross-validation procedure). The meaning of these accuracy figures was substantiated by sensitivity and specificity figures showing that the accuracy worked on both positive and negative cases.

Jiang et al (2012) used a support vector machine (SVM) on fMRI data to identify the biomarkers of stutters that were more typical (MT) to PWS, versus less typical (LT) stutters which are common to both PWS and FS. The SVM was able to correctly predict which of the two types of stutters had occurred with an average sensitivity of 0.91, specificity of 1 and generalization rates for classifying the types of stuttering symptoms of 0.97. As can be seen, the accuracy figures were also substantiated by corresponding sensitivity and specificity figures.

1.4 Machine learning versus other statistical methods

In this section, it is argued that the most appropriate statistical methods for identifying and predicting stutters are models based on machine learning. One of the key reasons for this is

Classifying Stutter Events using fNIRS and EEG data

the limitations posed by the assumptions of general linear models (GLM) such as logistic regression (which would be the most appropriate GLM model for a classification problem (Dunteman & Moon-Ho, 2006) such as predicting stutters versus non-stutters). The key problem assumption of GLMs is that of *no multicollinearity between predictor variables*. Multicollinearity occurs when the predictor variables in a data set are not independent from each other. Because nearly all activity recorded from scalp electrodes is the volume conducted sum of activities originating within a number of cortical domains (Delorme & Makeig, 2004), signal data recorded from systems such as EEG (and fNIRS) are highly correlated (Makeig et al, 1996).

To deal with multicollinearity in GLM, highly correlated variables are often removed. Given that we know many of the variables are likely to be correlated in neural data, this could lead to the removal of a high number of variables from the model, many of which could be useful. Therefore, a model that allows for multicollinearity, or is altogether non-parametric (meaning they do not assume anything about the data) is needed.

Another method of dealing with multicollinearity in EEG or fNIRS data, is Independent Component Analysis (ICA), (Makeig et al, 1996). ICA is a signal processing technique in which observed random data taken from the individual scalp channels are transformed into components (predictor variables) that are statistically maximally independent from each other (Oveis, 2009). ICA is also very useful in separating artefacts (such as eye movements and heart beats) from the data as these are usually independent of each other and so form independent components (ICs) that can be removed.

Classifying Stutter Events using fNIRS and EEG data

A downside of ICA in the present study, however, is that it can only be run per data set and produces a different number and order of components for each data set, making it difficult to concatenate. In this study, there are 8 separate data sets (the data will be fully explained shortly), for which ICA would produce a different number and type of components, the columns of which would not align if we were to concatenate them in a csv file for input into a prediction model. This means ICA could only be used on 1/8 of the data available and as such limits the amount of data we can use in prediction. If we want to combine all the data available for this classification problem, channel data needs used, and as such, models where no multicollinearity is assumed are needed. The name for models that do not make assumptions about the data is *non-parametric*.

Additionally, whilst ICs make better predictor variables for GLM analysis where predictor variables need to be independent, more research need done to determine whether they are also better predictor variables in models in which multicollinearity is not a problem. This study will aim to address that.

There are many types of supervised learning algorithms in machine learning which meet the criterium needed for this data set. The algorithms ran and contrasted in this study were K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Logistic Regression (Log Reg) and Random Forest (RF). Below are explanations on why each was chosen:

- 1) **K-Nearest Neighbour (KNN).** KNN is a *non-parametric* supervised machine learning classification algorithm and is one of the simplest available for supervised learning. A drawback of KNN is that it is especially prone to overfitting (Raschka & Morjalili, 2017).

- 2) **Support Vector Machine (SVM).** The kernel SVM (referred to now as SVM) is a non-parametric, supervised and increasingly popular and promising approach among classification studies. It has been used in a variety of biomedical applications; for example, to detect patterns in gene sequences (Hosseini et al, 2018). Moreover, SVMs have been the most widely used classifier in psychology in recent years (Arbabshirani et al. 2017, Kambeitz et al. 2015, Orru et al. 2012; Dwyer et al, 2018), which is in large part due its high accuracy rates. SVMs often come out on top in classifier comparison studies (Hosseini et al, 2018; Dwyer, 2018; Fernandez-Delgado, Cernadas, Barro, S. & Amorim, 2014). SVM functionality is also independent of the dimensionality of the feature space, making it reasonably resistant to overfitting (however as seen in Myers et al, 2018, it can still be improved through dimensionality reduction processes such as feature selection). SVM is particularly suited to binary class problems (Goel & Mahajan, 2017). A disadvantage is that it is more computationally expensive than some of the other algorithms such as KNN (but not artificial neural networks).
- 3) **Logistic Regression (Log Reg):** Logistic regression is a supervised parametric classifier, in which variables must be independent of one another. It is being used in this study as control to test whether non-parametric classifiers (KNN, SVM, RF) perform better than parametric classifiers (Log Reg) on multicollinear data. We would expect this to provide poor results on the fNIRS and EEG channel data, and better results on the IC data.
- 4) **Random Forest (RF):** RF is a supervised non-parametric “ensemble learning” technique consisting of the aggregation of a large number of decision trees, resulting in a reduction of variance compared to the single decision trees (Couronne et al, 2019). RF also has the advantage of being able to rank the variable importance measures (VIMs) of the features, with respect to their relevance in predicting the outcome (Couronne et al, 2019). This is useful for examining which channels or neural regions have contributed most to the

Classifying Stutter Events using fNIRS and EEG data

prediction of stutter versus non-stutter. VIMs can also therefore inform *feature selection* whereby low impact variables are removed to reduce dimensionality (if needed) and increase accuracy of the model. In a study comparing 179 machine learning classifiers across 121 data sets, RF was found as the joint top performer along with SVM (Fernandez-Delgado et al, 2014).

The downside of non-parametric algorithms is that they require more data and have a higher risk of overfitting. The most common solution to overfitting is the resampling technique k-fold cross validation (see below for explanation). Using high volumes of data can also reduce overfitting, as can reducing the number of features (Raschka & Morjalili, 2017). More features mean the data has more parameters to learn from, allowing it to potentially create specific sets of conditions for each case in the training set, making it less likely to work in the real world. In this study, 10-fold cross validation will be used to artificially increase the data set and minimize overfitting, and all four participants data will be combined to maximize data volume.

Simple artificial neural networks (ANN) and stepwise linear discriminant analysis (sLDA) were excluded on the basis they are parametric and therefore not suitable for channel data which will likely contain multicollinearity (Myers et al (2018) applied ICA to the EEG data making it suitable for the sLDA classifier). This problem can be overcome in ANNs by adding in hidden layers, but in doing so the data demands increase considerably, often requiring data sets in the hundreds of thousands or millions, so it will not be prioritised as an algorithm in this study.

Classifying Stutter Events using fNIRS and EEG data

In supervised machine learning, a large data set (samples in the thousands at the least, ideally in tens or hundreds of thousands) is split into a training set (usually around 75% of the data) and a test set (around 25% of the data). The data in both sets are labelled, that is, the actual value of the dependent variable we are trying to predict (in this case a stutter or non-stutter) is included in each sample. The machine learning algorithm then uses this data to learn how to predict the outcome, which is then tested on the other 25% of data. This 25% of data is firstly fed in without the value of the dependent variable, then checked against the value of the dependent variable, in order to establish the performance accuracy of the model. This method of data splitting is known as the “holdout” method (Raschka & Mirjalili, 2017).

By training the classifiers in this way, they can produce a number of performance statistics which tell us how successful the classifier is in predicting the outcome. These metrics are explained below.

(1) **Accuracy using holdout method:** Accuracy is defined as the ratio of correctly classified test subjects to the total number of subjects (Hosseini et al, 2018). In other words, the fraction of outcomes the classifier predicted correctly. An accuracy of 0.86 would mean the outcomes were correctly classified 86% of the time. The *holdout method* is where the data is randomly split into a training set (usually around 75% of the data) and a test set (around 25% of the data). The drawback of the holdout method is that it makes inefficient use of data, since typically a relatively large proportion of the instances is used for testing (Diamantidis et al, 2000).

(2) **Accuracy with K-fold cross validation:** As a means of minimizing overfitting and validating the accuracy statistic, a process called *k-fold cross validation* can be implemented, in which the classifier is run k times, in each of which, the 25% of data

Classifying Stutter Events using fNIRS and EEG data

which makes the test set is rotated. This means we end up with k different test and training sets from which the model can learn. This artificially creates a bigger data set which helps reduce the risk of overfitting (where the model learns how to predict outcomes very well in the data with which the model was trained but is poor at predicting outcomes on new data).

(3) **Sensitivity:** Also known as “recall”, this tells us the true positive rate, in this case, how well the model predicts stutters. It tells us the fraction of true events that were correctly detected (Goodfellow et al, 2016). If the model correctly predicts 0 stutters, it would have a sensitivity of 0. If it correctly predicted 100 stutters, but only 50 were actually stutters, the sensitivity figure would be 0.5.

(4) **Specificity:** This tell us the same as sensitivity but for negative events (non-stutters). It tells us the fraction of negative events that were correctly detected.

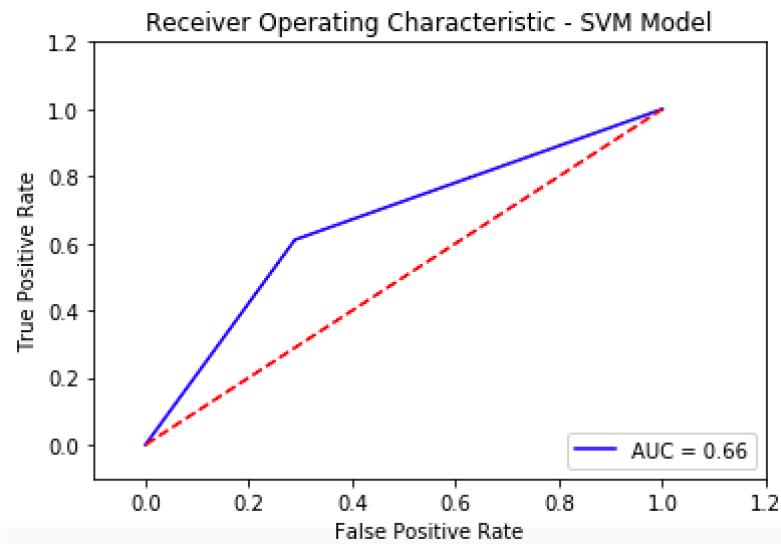
(5) **AUC (Area under ROC curve):** Whist the accuracy, sensitivity and specificity tell us useful information about the performance, and can help us identify problems with the model, it is argued by the author that the most informative figure is the area under the ROC curve (receiver operating characteristic). This is because the AUC tells us the combined ability of the model to correctly predict both positive (stutters) and negative results (non-stutters), (Florkowski, 2008). This is known as *discrimination*. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0 (Google Developers, ROC Curve). A 0.5 AUC means the answers are correct 50% of the time, meaning it is no better than chance in a binary classification problem. Several authors have also recommended AUC as the most useful metric to use specifically in psychology and psychiatry (Rice & Harris, 2004; Mossman, 1994; Swets, Dawes, & Monahan, 2000). As such, AUC will be the primary measure of accuracy in this study. The benchmark of what constitutes a good or acceptable AUC

Classifying Stutter Events using fNIRS and EEG data

varies between disciplines. In applied psychology, there is no formal agreement, however minimum AUC values of .70 would be expected (below that is considered unacceptable) and AUC of 0.8 and above would be considered very good (Howard, Helmus & Babchishin, 2017). In medical diagnosis, very high AUCs (.95 or higher) are sought (Rice & Harris, 2004).

Figure 1

An example of a ROC curve. The area under the ROC tells us the combined true positive rate and false positive rate of the model, telling us the combined ability of the model in correctly predicting both positive (stutters) and negative results (non-stutters) (Florkowski, 2018). (Image source is a python output from this study).



1.5 fNIRS & EEG: The advantages and drawbacks

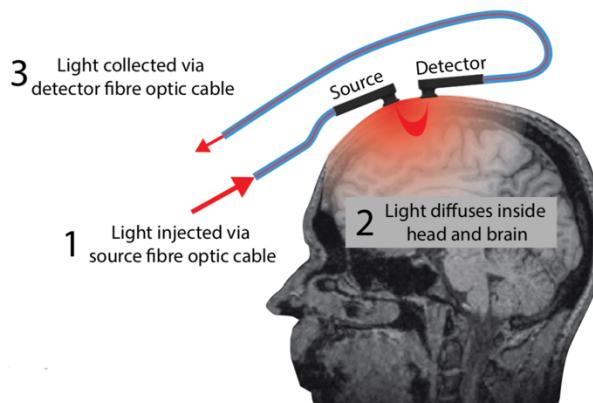
Functional near-infrared spectroscopy (fNIRS) is a non-invasive diffuse optical-imaging technique that can measure local metabolic demand in the surface of the cortex due to differential absorption of light by oxygenated and deoxygenated blood (Aslin, 2012). It is a neurovascular coupling (measuring neural activity through changes in blood flow) technique that works by injecting light into the brain via optodes, some of which is absorbed by the

Classifying Stutter Events using fNIRS and EEG data

brain. One of the materials that absorbs light is haemoglobin in the blood within the brain. The amount of oxygenated and de-oxygenated haemoglobin in the blood will influence the amount of light absorbed, meaning changes in the amount of oxygenated and deoxygenated haemoglobin in the blood will lead to very small changes of light absorbed within the brain (Brainsight NIRS Manual, 2018). By applying mathematical algorithms to the amount of light that enters via the source optodes and the amount that leaves via the detector optodes, the fNIRS system measures the amount of light absorbed. These changes can be associated in changes in brain function (as the brain works, it consumes oxygen in the blood).

Figure 2

fNIRS illustration taken from the Brainsight NIRS Manual (2018). Light enters the brain through the source optodes and leaves through the detectors. The amount of light absorbed inside the brain can then be measured and used as a representation of brain activity.



Multiple sources and detectors are placed on the scalp, usually about 3cm apart. Each source and detector relationship is known as a “channel”, and the light from each source will usually be picked up by more than one detector. This means that by having 4 sources and 8 detectors, we could end up with 12 channels (exact channels for this study are shown in the methods section). Each channel then produces a measure of oxygenated haemoglobin (HbO),

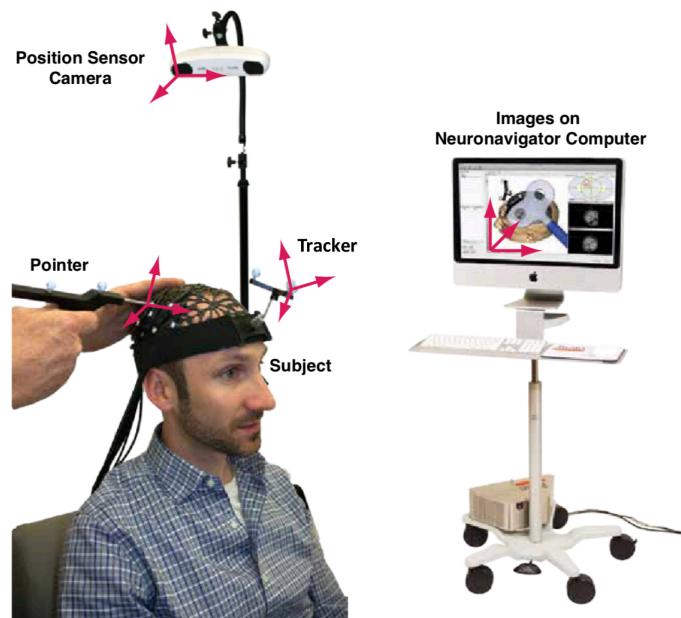
Classifying Stutter Events using fNIRS and EEG data

deoxygenated haemoglobin (HbR) and total haemoglobin concentrations ($HbT = HbO + HbR$) which can be used for analysis.

In order to associate the data with specific brain regions, neuronavigation measurements are taken, often using a neuronavigation system built into the fNIRS device. A position sensor (usually an optical camera) is positioned over the head area, upon which a tracker device is placed. A pointer is used to measure the head in relation to the trackers so that the positions of the optodes in relation to the brain can be recorded. See Figure 2.

Figure 2

A position sensor takes readings from the trackers placed on the head via a head band. *Brainsight Manual* (2018).



fNIRS mainly measures surface cortex with a spatial depth of about 10mm past the surface of the cortex and an excellent temporal resolution in the order of deciseconds (Pinti, 2018), (by comparison, EEG measures in the order of milliseconds; fMRI and PET in the order of

Classifying Stutter Events using fNIRS and EEG data

seconds). Because stutters can occur in durations of less than a second, neuro-monitoring with a temporal resolution of at least 10hz (deciseconds) is needed. fNIRS has a good spatial resolution of ~2–3 cm versus EEG ~5–9 cm (Pinti, 2018).

In recent years, fNIRS has been used to assess the regional activation, timing, and lateralization of cortical activation for a diverse number of perceptual, language, motor, and cognitive investigations (Hosseini, 2018), often as a complement to EEG (electroencephalography) and MEG (magnetoencephalography). In addition to its good temporal resolution, portability and cost (around a tenth of fMRI), one of its advantages is that it doesn't require rigid stabilization of the participant (Pinti et al, 2019).

This high temporal resolution of fNIRS is ideal for a study in which events can be in the duration of deciseconds, like stutters which can be less than a second in duration. fMRI or PET would not provide sufficiently time-accurate readings for this kind of study. The low spatial depth of fNIRS is not a problem given the main ROIs in this study are all cortical (LIFG, motor areas, auditory areas).

There are limitations of fNIRS data in relation to traditional statistical modelling such as GLM, however. As demonstrated by Barker et al (2013) physiological noise in fNIRS data results in serially correlated noise, which violates the assumptions of an uncorrelated, independent and identically distributed data set in GLM. In addition, fNIRS errors often exhibit heteroscedasticity due to motion artefacts (Huppert, 2016). Heteroscedasticity in this case, refers to noise that does not arise from a single uniform distribution (Barker et al, 2016). This violates the assumption of homoscedasticity in GLM, which stipulates that variance of the errors should be consistent for all observations. This means that fNIRS data requires a

Classifying Stutter Events using fNIRS and EEG data

great deal of processing and transformation to be used in traditional statistical models such as linear regression. Alternatively, models that do not require homoscedastic data are required to draw meaning from the data.

Electroencephalogram (EEG) provides a direct measure of cortical activity with millisecond temporal resolution (Subasi & Ercelebi, 2005). EEG works by recording and measuring the subtle electrical impulses which occur throughout the cortex when large groups of neurons fire in the brain. EEG electrodes (also known as channels) are used to measure the sum of this electrical activity from the scalp surface. As the voltage fluctuations measured at the electrodes are very small, the recorded data is digitized and sent to an amplifier. The amplified data can then be displayed as a sequence of voltage values known as delta (<4Hz), theta (4-7Hz), alpha (7-12Hz), beta (12-30Hz) and gamma (30-50Hz). Delta is associated with sleep (Eharmoney, 2013); theta is associated with a wide range of cognitive processing such as memory encoding and retrieval as well as cognitive workload (Klimesch, 1999); alpha waves are linked to a relaxed state of wakefulness, inhibition and attention (Klimesch, 2012); and beta waves are most associated with motor planning (Takahashi et al, 2007). Gamma is said to reflect attentive focusing and serves as a carrier frequency to facilitate data exchange between brain regions and increases with sensory drive, and with a broad range of cognitive phenomena, including perceptual grouping, and attention (Xiaoxuan, 2011). Gamma is generally not well detected by EEG as it is low amplitude and high frequency and therefore is noisy (low signal to noise ratio) when measured on the scalp (Xiaoxuan, 2011). For this reason, voltage above 30Hz is usually filtered out in pre-processing.

Like fNIRS, signals produced in EEG are highly correlated, and for many statistical models, require transformation. As discussed earlier, a common method for this is ICA, however it is

Classifying Stutter Events using fNIRS and EEG data

unclear whether this is necessary for machine learning. This will be investigated in this study (see hypotheses).

1.6 Hypotheses

The goals of this study were 1) to determine whether supervised machine learning classifiers can be used to classify stutter events from neural data, and 2) to investigate best practices for doing so in terms of optimal machine learning classifiers (for example random forest, support vector machine), data structures (high volumes of data versus balanced data) and data types (for example independent EEG components versus channel data as predictor variables).

Hypothesis 1: When trained with fNIRS and EEG data which are labelled with stutter and non-stutter events, machine learning algorithms should be able to identify when a stutter occurred at an accuracy of at least 70% (measured by AUC).

Hypothesis 2: SVM and random forest will show the highest performance in predicting stutter versus non stutter events.

Hypothesis 3: On equally sized data sets, ICA features from the EEG data should have a higher prediction rate than channel features.

Predictor variables: Oxygenated haemoglobin (HbO), deoxygenated haemoglobin (HbR) and total haemoglobin concentrations ($HbT = HbO + HbR$) measured by fNIRS. Channel data and independent components measured by EEG.

Dependent variable: Binary output. Whether speech was fluent (0) or stuttered (1).

Conditions: Stuttered speech and fluent speech.

2. Methods

2.1 Sample

In the original data collection, concurrent EEG, fNIRS and audio data were collected from 11 participants who stutter. Out of these participants, 4 exhibited clearly audible stutters which could be marked on the audio files and were therefore used in this study.

Table 2. List of participants for this study. Mean age = 28.25, SD = 6.18.

Participant	Sex	Age
006	Male	25
009	Male	23
017	Male	37
018	Male	28

Only male participants were recruited since COPD affects a higher proportion of males than females, with the ratio of male to female PWS changing from 2:1 in early childhood to 5:1 in adulthood (Yairi & Ambrose, 1999). PWS represented a wide variety of stuttering severities. PWS were recruited from the UCL Stuttering Self-Help Group, focus groups of the British Stammering Association, and the supervisor's personal contacts.

Inclusion criteria for PWS were right handedness, no sensory impairment criteria and having no other independent communication disorder. All participants also experienced childhood-

Classifying Stutter Events using fNIRS and EEG data

onset fluency disorder (rather than adult-onset). Adherence to aforementioned criteria was self-reported. A strong preference was placed on recruiting participants with short hair to facilitate signal clarity of the fNIRS and EEG recordings. Recruitment was also conducted with a preference for fine, lighter hair on account of a clearer fNIRS signal, however, variations in texture and colour were not criteria for exclusion.

Participants were paid £30 for their participation and had any travel expenses covered with approval from the UCL Division of Psychology and Language Sciences. All participants provided full consent, and all procedures were approved by the UCL Graduate School Ethics Committee

2.2 Experiment procedure

The data used in this study were collected by a team of three undergraduate students; Claudia Contadini-Wright, Shannon McCann and Antonia Toneva, who kindly provided the data used in the current analyses. The design adopted is detailed below. Everything except the “Experiment procedure” (described in this section) was conducted by the present author, including design of the analysis procedure, pre-processing, labelling and analysis of the data.

PWS performed verbal tasks (detailed below) whilst being concurrently monitored for fNIRS hemodynamic responses and EEG, and whilst being audio-recorded.

For each participant, fNIRS responses were recorded with a Rogue Resolutions continuous wave system (a type of radar system where a known stable frequency continuous wave radio energy is transmitted and then received from any reflecting objects), Brainsight fNIRS

Classifying Stutter Events using fNIRS and EEG data

(Rogue Resolutions, Cardiff, UK). The equipment used 15mW near-infrared lasers at 685nm (nanometers) and 830nm as light sources (the onset of near-infrared is typically at around 700nm-1mm (Wikipedia, Infrared, 2019)), and avalanche photodiodes (APDs) as detectors for measuring intensity changes in the refracted light sampled at 10-Hz (deciseconds). Lower sampling rates result in higher signal quality (Brainsight Manual, 2018) so 10Hz was considered the optimal temporal resolution versus signal quality. Millisecond (100Hz) sampling was not considered necessary in light of that fact stutters can be easily captured in deciseconds.

Once the participant was set up, a trial data acquisition was started. Detector sensitivity was adjusted up and down until a good signal, with low saturation of the detectors was evident (clearly indicated by red lines beneath the live data on the screen), which was at an average of 135V (voltage determines the gain of the detectors). All gains were adjusted per-channel (as opposed to as a group) until a clean signal quality was seen on the screen, combined with low saturation in the detectors (a detailed explanation of how this is carried out can be seen in the Brainsight Manual, 2018).

Twelve optodes were fitted into an optode cap positioned over the LIFG and RIFG (4 detectors and 2 sources on each hemisphere), with around 3cm between each source and detector pair. This allowed the fNIRS system to acquire signals from 12 channels (6 over the LIFG and 6 over the RIFG), which can be seen in Table 2. A head strap with POLARIS localisation measurement trackers (as referred to in the introduction) was then placed on the participant's head so that optode positions in relation to brain regions could be picked up during neuronavigation. Once participants had been positioned optimally in front of the camera, crucial anatomical landmarks (nasion/top of nose, left and right pre-auricular/front of

Classifying Stutter Events using fNIRS and EEG data

upper ears) were mapped with the help of stickers placed on the participant. The rest of the head was then digitised with the aid of a pointer (explained in introduction) and foot pedal to give an overall map of the scalp surface. The accuracy of landmark mapping was checked against previous marks done on the nasion, LPA and RPA and re-done if inaccurate. The position of pre-set electrodes was then mapped, including the four sources and eight detectors. The position measurements can be seen in the neuronavigation data (see example in appendix 2). See Figure 4 for an indication of optode placements for one individual.

Figure 4

The pink dots on the scalp represent sources and the blue dots represent detectors. We can see on the right image that the right source is linked to 4 detectors (the red lines), giving us 4 channels. The left source is linked to two detectors (the white lines), giving us 2 channels.

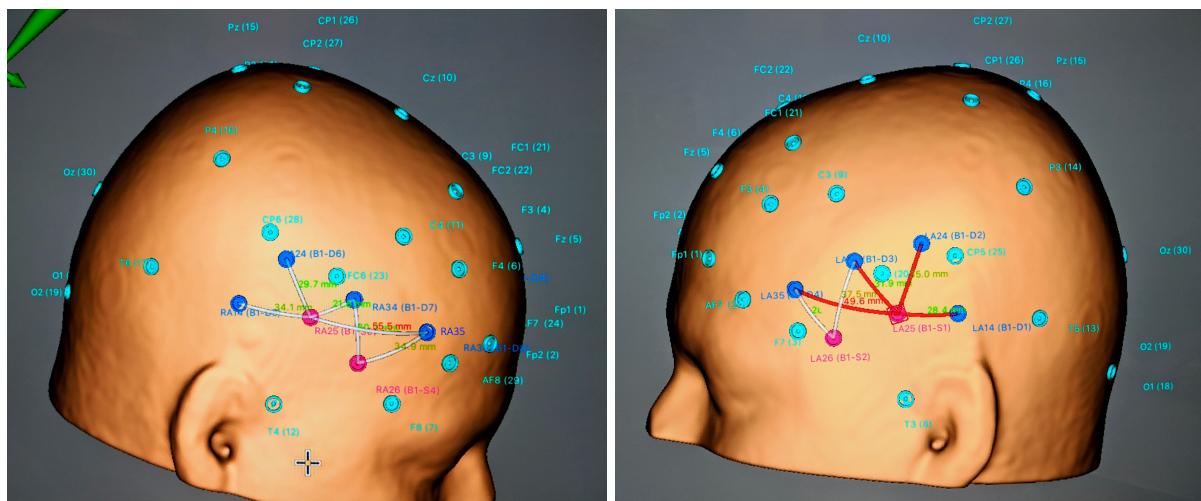


Table 2

From the 4 sources (S) and 8 detectors (D), the following 12 channels were recorded for this study.

Channels	
LIFG	RIFG
S1-D1	S3-D5
S1-D2	S3-D6
S1-D3	S3-D7
S1-D4	S3-D8

Classifying Stutter Events using fNIRS and EEG data

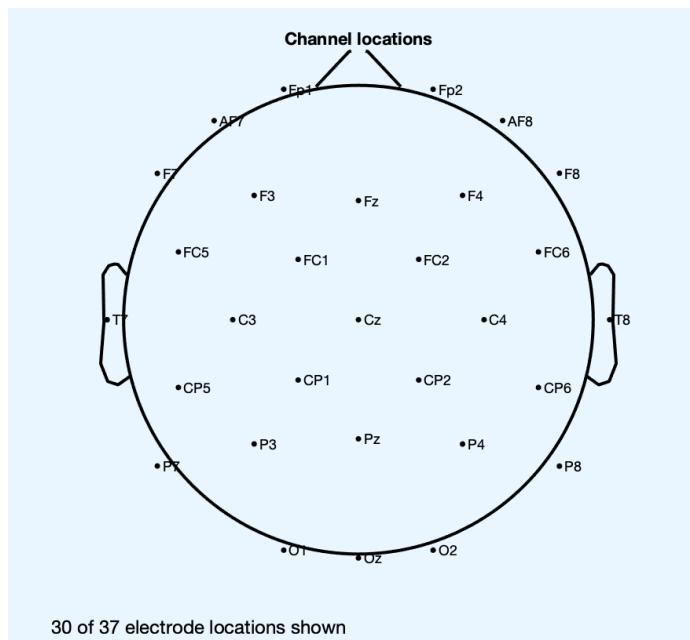
S2-D3	S4-D7
S2-D4	S4-D8

Once neuronavigation measurements had been taken, the POLARIS strap was replaced by a basic headband to limit the leaking of laser light from fNIRS source optodes, and the signal was quality checked.

For EEG, 29-channels were measured concurrently with fNIRS using TruScan EEG systems' Deymed TruScan 32 EEG amplifier (DEYMED Diagnostic, Hronov, Czech Republic). Electrodes were laid out according to the International 10-20 system on the cap that also accommodated fNIRS optodes (see Figure 5).

The 29 electrodes and related neural regions can be seen in appendix 3.

Figure 5. International 10-20 system. Showing 29 electrode positions (Oz is a location measurement). Taken from EEGLAB for this study.



Classifying Stutter Events using fNIRS and EEG data

Electroconductive gel was applied under each electrode with syringes, and impedance was monitored throughout the experiment to ensure it was in the 0-5kΩ range. Straps were applied on either end of the cap and clipped into a larger strap placed around the participant's chest to limit head movement. The cap was attached to the EEG machine, and EEG signals were acquired at a 500Hz sampling rate (these were later down sampled to 10Hz for analysis). For a full explanation of this process, see the Truscan Neurofeedback User Guide (DEYMED Diagnostic, 2013).

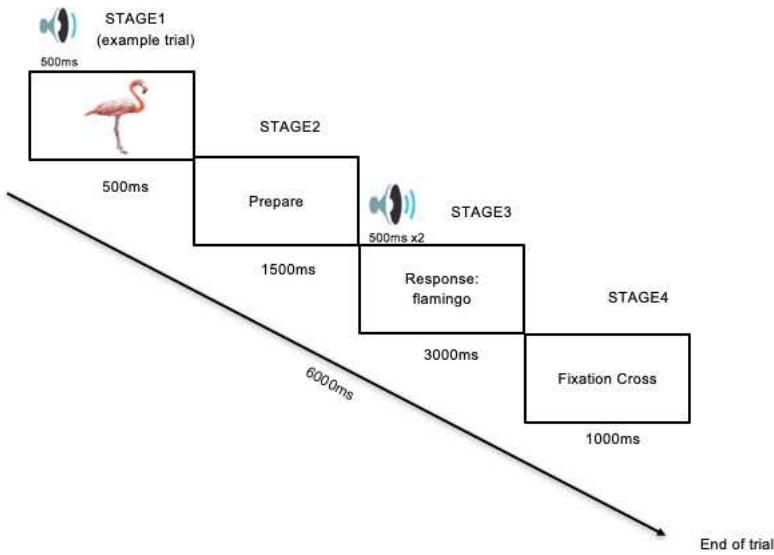
After checks were completed for both EEG and fNIRS, participants were asked to carry out two tasks; "naming" and "social".

In the naming task, participants were shown images which they were asked to name out loud. Each picture naming block was made of five trials each showing a picture cue to be subsequently named by the participant (see Figure 6). Because stuttering is often absent during oral reading (DSM-5 American Psychiatric Association, 2013; Carlson, 2013), images rather than written words were shown as prompts. A researcher was present at the back of the room, but out of eye contact of the participant. The naming task lasted around 23 minutes, after which the researchers came into the room and explained what was required in the next (social) task.

Figure 6

Naming task visual.

Classifying Stutter Events using fNIRS and EEG data



In the social task participants saw a single visual prompt that required five corresponding items to be named (eg. Cue = Global leaders. Response = Angela Merkel, Donald Trump, Emanuel Macron, Theresa May, Vladimir Putin). See Figure 7. During the social task participants maintained eye-contact with one of the researchers to simulate a social interaction. The social task also lasted around 23 minutes

Figure 7

Social task visual. Participants were shown a category cue (in this case global leaders) and asked to name five items from that category.

Classifying Stutter Events using fNIRS and EEG data

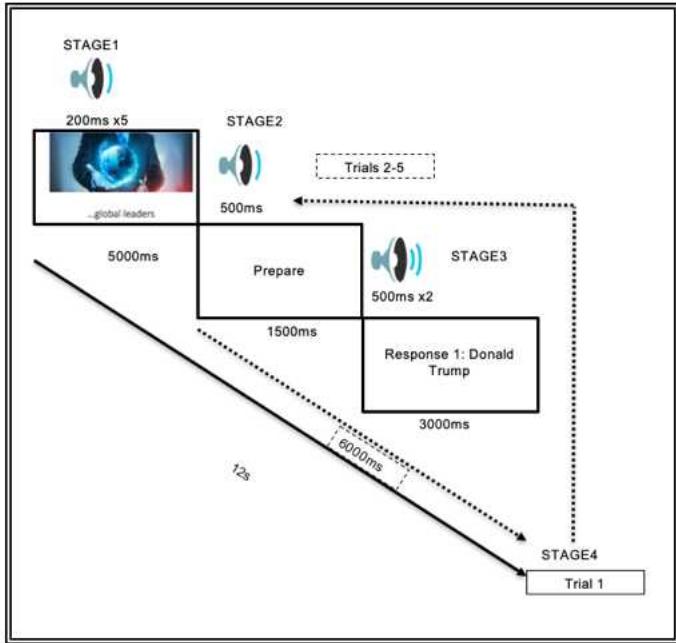


Figure 2. The Social Task contained 20 blocks, each beginning with the presentation of image and text cue. The cue prompts five separate responses, given while participants maintain eye-contact with the experimenter. Multiple beeps prompted the participant to break the eye-contact and fixate on the screen to see the cue. Single beep prompted the speech preparation for first response while double beep prompted the speech response itself.

The experiment itself was built in PsychoPy (Pierce, 2007) version 1.8 and run on a laptop. A Yeti USB microphone was used to record speech in Audacity for fluency analysis.

In the original study for which this data was collected, the experiment was run in blocks, as separate trials. These trials are not relevant to the present study which looks only at stutter events versus non-stutters, so has not been explained here. Details can be seen in appendix 4.

All work from here was carried out by the present author.

2.3 Data Pre-Processing

2.3.1 fNIRS pre-processing

Classifying Stutter Events using fNIRS and EEG data

The recorded fNIRS data comprised oxygenated and deoxygenated readings on each of the 12 channels (6 on the LIFG and 6 on the RIFG). The data were exported from Brainsight in .nirs format and loaded into Homer2 (Huppert, Diamond, Franceschini & Boas, 2009) using MatLab (The Mathworks Inc, v2018b). Once the data were in Homer2, the processing steps in Figure 8 were undertaken. The steps were as follows.

Figure 8

Processing steps for the fNIRS data this project.



There is currently a lack of international standards on fNIRS processing (Pinti et al, 2019), however in recent years, more work has been done to establish best practices (Pinti, Scholkmann, Hamilton, Burgess & Tachtsidis, 2019; Pinti et al, 2018; Hocke et al, 2018). This research has been used to inform the processing parameters which can be seen in detail below.

Channel-pruning (Step 1)

Channel-pruning is necessary as fNIRS signal quality can be jeopardized by a number of factors including instrument and environment noise, poor coupling of optodes to the head, and optical interference by coarse or heavily pigmented hair (Orihuela-Espina et al, 2010). Channels can also be of poor quality if the gap between the source and detector is substantially over the recommended distance of 3cm. A commonly used method for channel

Classifying Stutter Events using fNIRS and EEG data

pruning is the visual inspection of the ~830–850 nm wavelength raw signal (Hocke et al, 2018) where flat or low signal (close to 0) channels are removed.

Visual inspection of the ~830nm wavelength optical density signal was carried out in Homer2 (Huppert, Diamond, Franceschini & Boas, 2009) to ensure each channel represented a clear neural signal. Flat or noisy channels with a poor optical coupling were noted for removal later in csv. See Figure 9 and Table 3.

Figure 9

Example of channel exclusion based on flatness. The neural signals are pink, green and orange. The purple (far bottom) channel was removed as it is close to zero in amplitude and flat, and therefore doesn't provide us with a useful neural signal (Image is a screen shot of the Homer2 GUI from this study).

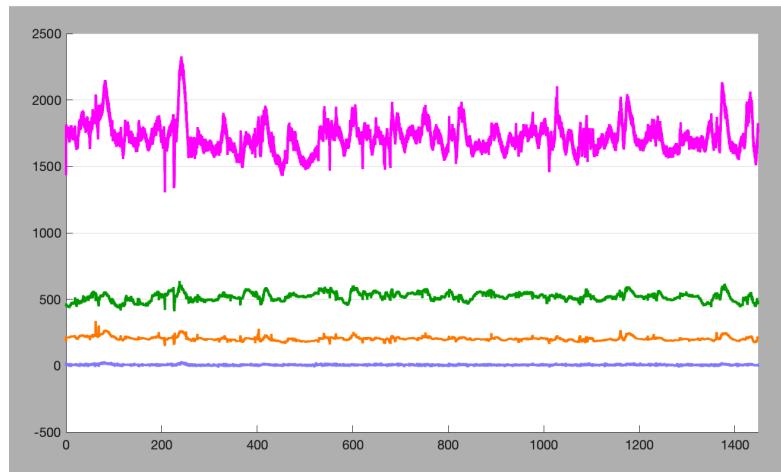


Table 4

The following fNIRS channels were removed.

Data set	Channels removed
006_Naming	A4, C8
006_Social	A4, C8
009_Naming	A4, C8
009_Social	A4, C8
017_Naming	A4, C8
017_Social	A4, C8

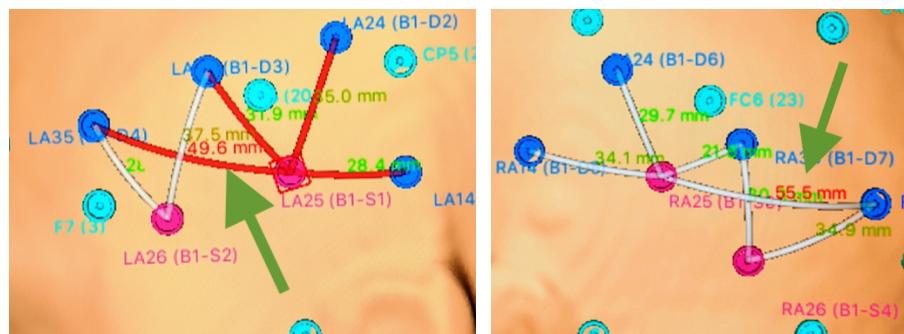
Classifying Stutter Events using fNIRS and EEG data

018_Naming	A4, C8
018_Social	A4, C8

Because the removed channels were the same for all participants, it is likely that this was due to poor optode coupling. If we zoom into the image of the source and optode pairings from Figure 10, we can see that appears to be the case.

Figure 10

The left image shows the left hemisphere and the right image, the right hemisphere. The green arrows point to pairings in which the distance between the source and optode significantly exceeds 3cm, at 4.9cm and 5.5cm respectively. These distances have also been highlighted in red by the Brainsight interface from which these images were taken.



Removing physiological artefacts (Step 2)

fNIRS data are usually contaminated by physiological noises not directly related to cortical brain activity that can deteriorate the signal-to-noise ratio (SNR), and mask and/or mimic the presence of brain hemodynamic responses (Tachtsidis & Scholkmann, 2016). Bandpass filters which remove specific frequency content from the measured signals, can be used to remove much of this noise. Low-pass filters are used to remove high-frequency instrument cardiac oscillations. High-pass filters, remove slow, low-frequency content, such as blood pressure oscillations.

Classifying Stutter Events using fNIRS and EEG data

In this study, bandpass filtering was applied in the processing stream at high pass = 0.01 Hz, low pass = 0.50 Hz. This allowed the filtering out of high frequency heart rate artefacts (usually seen at around 1Hz), respiration (0.2–0.6 Hz) and very low frequency vasomotion regulations (<0.1 Hz), (Pinti et al, 2018; Hocke et al, 2018).

Other filters such as Chebyshev, elliptic and zero-phase Fast Fourier Transform (FFT) filters (similar to wavelet) were rejected on the basis they have been found to offer no absolute advantage over band-pass filtering and their influence is less well-documented (Hocke et al 2018).

Removing movement artefacts (Step 3)

Although fNIRS is more robust to movement artefacts than other neurological measurement methods such as fMRI, EEG and MEG (Pinti et al, 2019), signals can be corrupted by movement. A majority of motion artefacts appear in the form of abrupt changes in the amplitude of the signal and can be removed by using automated filters and through visual inspection. In a systematic comparison of fNIRS motion correction techniques, wavelet filtering (Molavi et al, 2011) was recommended on the basis of its high contrast-to-noise ratio by Cooper et al, (2012) and again in a more recent review in 2018 by Pinti et al (2018).

Motion artefact correction was first conducted with a wavelet filtering technique applied using the HOMER2 *hmrmotionCorrectionWavelet* function. This function eliminates outlying wavelet coefficients which are assumed to be motion artefacts, by implementing a probability threshold to identify outliers. The coefficients which lay more than 1.5 times

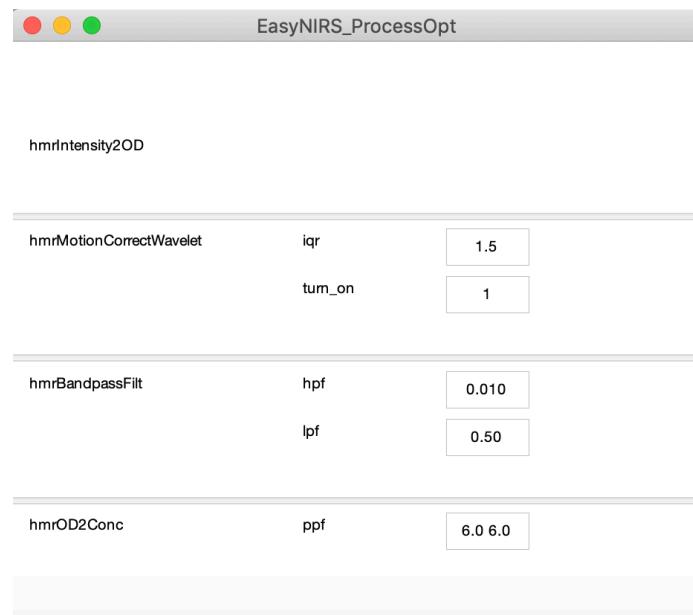
Classifying Stutter Events using fNIRS and EEG data

outside the interquartile range are considered outliers (Pinti et al, 2018) in the form of motion artefacts and are removed.

The automated filtering above (bandpass and wavelet) was done in the Homer2 processing stream as can be seen in Figure 11.

Figure 11

Homer2 processing stream. The data were first converted from raw to optical density data (“hmrIntensitytoOD”). Wavelet (1.5 iqr) filtering and bandpass filtering (0.01-0.5) and were then carried out. “hmrOD2conc” converts optical density data to haemoglobin the concentration data needed for analysis (Image is screen shot of the Homer2 GUI from this study).

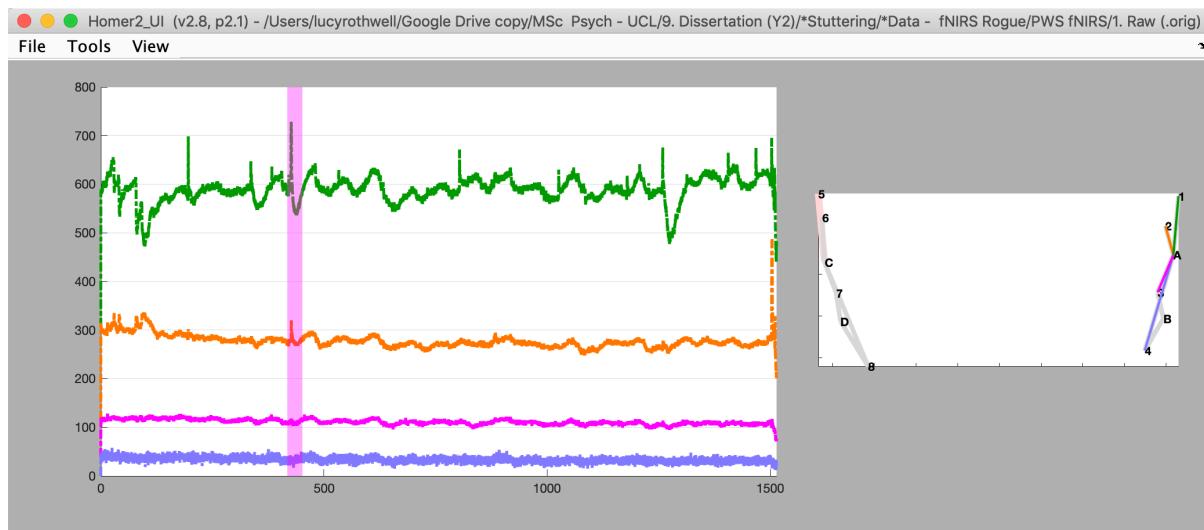


After processing, ~830nm data were also visually inspected to check for any other spikes that may not have been picked up by the automated filtering, and any obvious movement artefacts were manually labelled and removed. Movement artefacts can be seen by large spikes in the raw data as shown below (example shown in Figure 12)

Figure 12

Classifying Stutter Events using fNIRS and EEG data

The vertical purple area shows a manually labelled movement artefacts which were removed from the data (image is a screen shot of the Homer2 GUI from this study).



The concentration data were exported from MatLab using the following code:

```
load 'foo.nirs' -mat # loading pre-processed .nirs file  
Data=squeeze(procResult.dc); # pulling concentrated data  
Data( tIncMan==0, : )=NaN; # including manual time removal  
csvwrite('foo.csv', Data); # writing to csv
```

The channels identified for exclusion were then manually deleted from the exported csv. The values in the removed time points (*tIncMan*) appeared in the csv as “NaN” rows (NaN in every column across the row). Two options were compared for dealing with the missing values (both done later within the python script). The first was imputation (the process of replacing missing data with substituted values) using a statistical mean test (whereby the missing value is replaced by the mean of the column values). The second was removal of the rows altogether. The second method of removal was chosen on the basis it led to higher performance during classification.

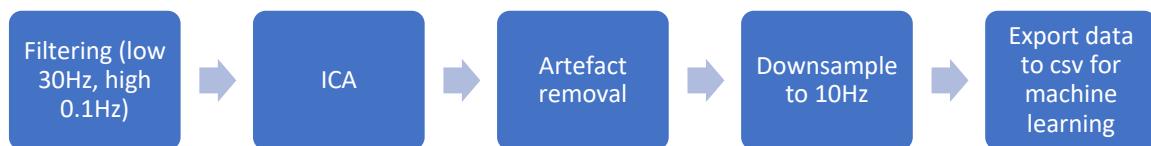
2.3.2 EEG pre-processing

Classifying Stutter Events using fNIRS and EEG data

Raw EEG data from the 29 channels were loaded into EEGLAB (Delorme & Makeig, 2004) using MatLab (Mathworks, Version 2018b). The pre-processing steps taken are summarized in Figure 13. For the channel data, all steps except *ICA* and *artefact removal* were performed.

Figure 13

Processing steps for the EEG data in this project.



Labelling channel locations (Pre-step)

Once the data were in EEGLAB, the channel locations were labelled using the EEGLAB automatic naming function. This allowed the identification of channels (e.g. Cz, FP1, FC3) when the data were exported to csv files later (where columns represented channels).

Filtering (Step 1)

The data were then filtered using the EEGLAB *Basic bandpass FIR Filter*. A low pass filter (30Hz) was applied to reduce gamma (as explained in the introduction) and high amplitude activity such as muscle artefacts. A high pass filter (0.1Hz) was applied to reduce waves that were unlikely to be related to cerebral activity such as changes in electrolyte due to sweating (Motamedi-Fakhr, Moshrefi-Torbat, Hill, Hill, & White, 2014). These filtering parameters have also been successfully used in other EEG-based stuttering studies including Vanhoutte et al (2016).

Classifying Stutter Events using fNIRS and EEG data

In EEG, voltages recorded at each electrode are computed with reference to another electrode or a set value. The reference value used in this study was the average (the average of activity at all electrodes), meaning after filtering, data needed re-referenced to the new average reference point. This was done using the EEGLAB *re-reference* tool.

Independent component analysis (Step 2)

ICA was then run to create independent components (ICs) for analysis, but also as a means of artefact removal. Because physiological artefacts such as muscle movements, heart rate and eye-movements produce unique scalp patterns that are independent of each other, they are picked up as independent components by the ICA (Onton & Makeig, 2012). The EEGLAB *runica* function was used to perform ICA on all data. The *fastica* and *jader* ICA algorithms were also considered but rejected on the basis they provide near equivalent output to *runica* for data sets with less than 100 channels (SCCN, 2019). The *runica* function was also used by Myers (2018) which is the most similar study to the current one. An average of 35 components was found.

Artefact removal (Step 3)

Removal of non-neural components was then carried out with the assistance of the function *ICLabel* (Pion-Tonachini et al, 2019), an artificial neural network (ANN) and convolutional neural network (CNN)-trained machine learning model for classifying components from ICA into either brain, eye, heart, muscle or “other” (a component is labelled “other” when it does not converge onto a meaningful signal because it consists of either a mixture of signals or noise (Pion-Tonachini et al, 2019)). The ICLabel method was chosen on the basis it was shown to equally perform or outperform all other publicly available automated ICA

Classifying Stutter Events using fNIRS and EEG data

component classification methods (Pion-Tonachini et al, 2019) at the time this study was written.

The components flagged by ICLabel as being non-neural, were then visually inspected according to industry guidelines (Onton & Makeig, 2006; SCCN, 2019; Raduntz et al, 2015)), which are briefly explained below in Figure 15. If visual inspection matched the ICLabel classification, the component was removed. Figure 14 shows 34 labelled components identified by ICLabel from a single participant (*006_Naming*). Figures 15-17 below, show how example cortical activity, eye and muscle artefacts were confirmed through visual inspection. On average, 11 brain components were left for machine learning analysis after artefact removal.

Figure 14

The output from ICLabels where components were identified as brain, muscle, eye, heart or “other” in the naming task from one participant. The components are ranked in terms of how much they contribute to variance, with component 1 being the highest contributor. Examples of the artefacts are examined in the following figures (Image source: screen shot taken from EEGLAB this study).

Classifying Stutter Events using fNIRS and EEG data

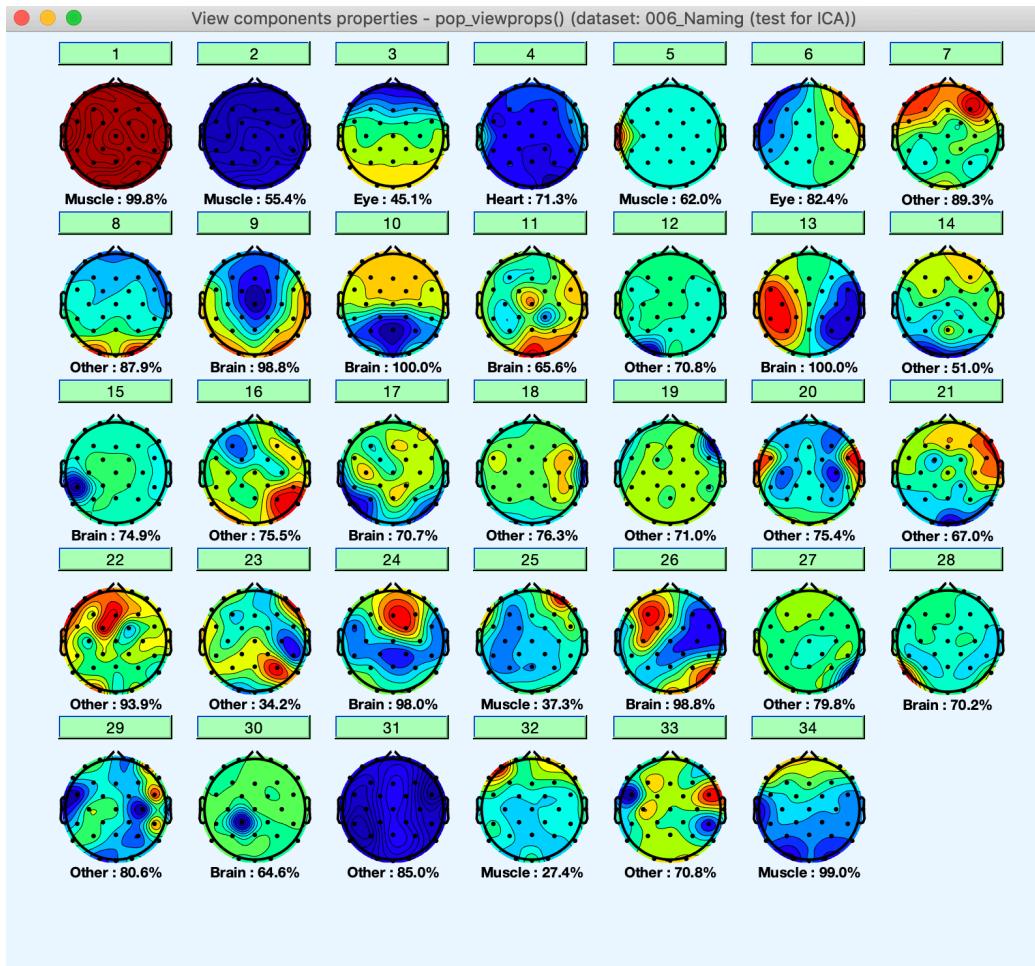


Figure 15

Cortical activity. The image below shows an example of cortical activity which can be identified by dipoles on the scalp map (top left) and spectral peaks in either the (8-12 Hz) alpha band (posterior Ics) or (4-7 Hz) theta band (bottom panel; see black arrow on image).

Classifying Stutter Events using fNIRS and EEG data

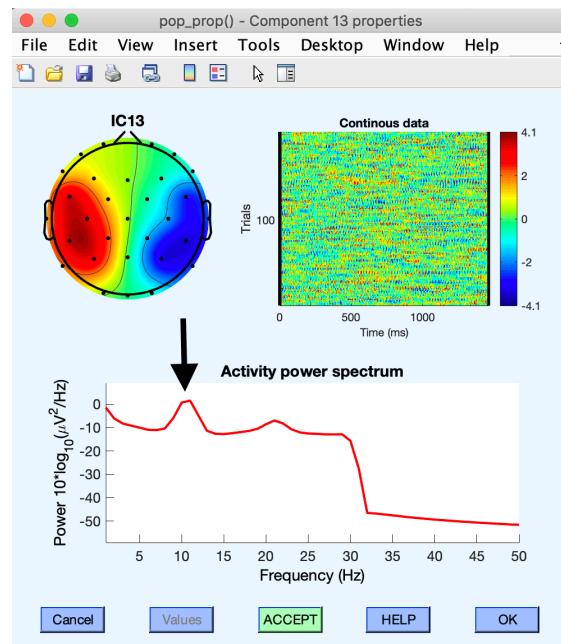


Figure 16

Eye movement artefact. The image below shows very clear examples of eye movement artefacts, which are indicated by strong far-frontal projection either horizontally or vertically. Eye artefacts are the easiest to identify and can be done by scalp map alone as they generally look like one of the two images below. Eye artefacts are removed.

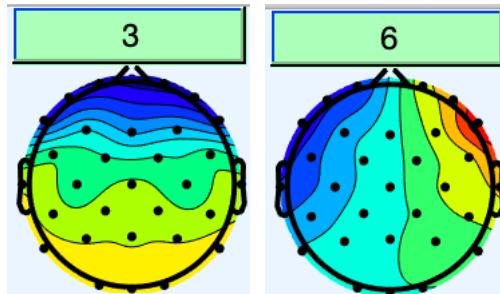


Figure 17

Muscle artefacts are spatially localized (concentrated in one area) on the scalp map. The power spectrum is also a strong indicator as muscle activity usually displays the highest power at frequencies of 20Hz and above. The left image below shows a muscle component identified mainly through the scalp map (which also shows frequency in the 20Hz range) and the right image shows a classic frequency range of 20-30Hz (which can be used to identify the artefact since the scalp map is atypical). Muscle artefacts are removed.

Classifying Stutter Events using fNIRS and EEG data

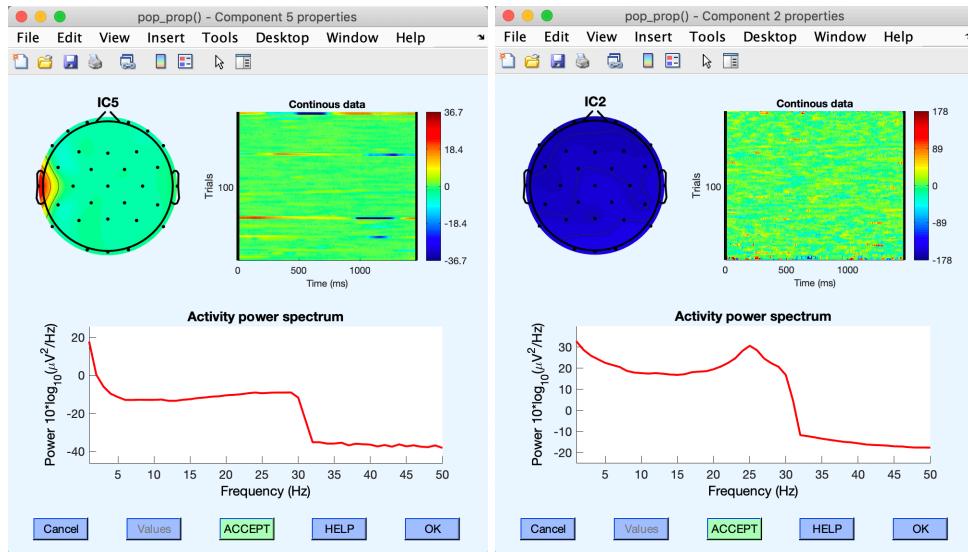


Table 5

Components removed from each EEG data set.

Data set	Components removed
006_Naming	1-7, 9, 10, 14-19, 23, 27-29, 31, 33
006_Social	1-8, 10, 13, 15, 17, 19, 21, 22, 24, 25, 27, 29, 32, 33
009_Naming	1-5, 8-15, 17, 18, 21-23, 26, 28, 29, 31, 32
009_Social	1-4, 7, 8-16, 18, 19, 21, 22, 23, 25, 28, 29, 31-33
017_Naming	1, 2, 4, 6-10, 12, 13, 15-17, 20, 21, 24, 25, 27, 29, 34
017_Social	1-3, 5-12, 14, 17-19, 22, 24-26, 28, 29, 32-34
018_Naming	1-8, 11-13, 17-23, 26-34
018_Social	1-7, 11, 13-18, 20-25, 28-30, 32-34

The EEG data were then down-sampled to 10Hz (0.1 second samples) and exported to two separate csv files; in one of which each column represented a channel (such as Cz, Fp1 etc), and in the other, where each column represented a brain component produced from ICA. By splitting the data sets in this way, we can compare the machine learning classifiers on data

Classifying Stutter Events using fNIRS and EEG data

sets where independent components (ICs) are features to data sets where the EEG channels are features, to test hypotheses 3.

The EEG data and the fNIRS data were not combined into the same csv to prevent overfitting which can occur when there are too many predictor variables (as discussed in the introduction).

2.3.3 Time-syncing and labelling the data

In the experiment, the fNIRS, EEG and audio started recording at different times. Therefore, all three needed time-synced. This was done by syncing everything to the “first preparation cue” which was audible on the audio files as the first beep. All audio before this point was cut out of the recording on Audacity (Audacity Team, 2019). See Figure 20 below.

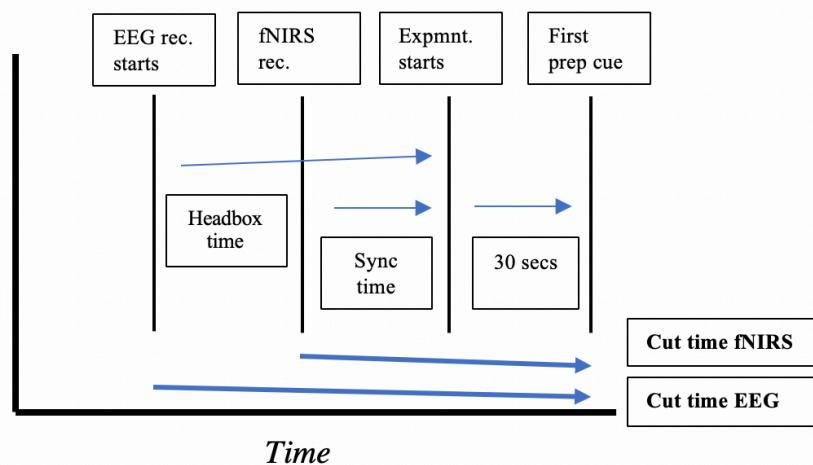
The fNIRS and EEG data were also synced to the “first preparation cue” once they had been exported to separate csv files. The “first preparation cue” was identified based on time points taken during the experiment. These time points indicated how many seconds before the first preparation cue, the fNIRS and EEG had started recording. That number of seconds was then multiplied by 10 (for deciseconds since this is what the EEG and fNIRS had been sampled in), and this number of rows was removed from the csv files. This is illustrated briefly in Figure 18 below. A detailed time-syncing procedure and the time points taken for all data can be seen in appendix 5.

Figure 18

Time syncing graphic. The “headbox” time showed the distance in seconds between the EEG recording start and the “experiment start”. The sync time showed the distance in seconds between the fNIRS recording start

Classifying Stutter Events using fNIRS and EEG data

and the “experiment start”. There was always 30 seconds between experiment start and first prep cue. The combined figure (headbox or sync + 30) gave us the “cut time” in seconds for EEG and for fNIRS, which were then multiplied by 10 to equal deciseconds. That number of rows was then deleted from the corresponding fNIRS or EEG csv exports.



Labelling

Once the fNIRS and EEG data were in csv format, the rows (each representing 0.1 second) needed labelled with “1” (stutter) or “0” (non-stutter). The aim was to create csv files as shown in Figure 19 below, for both EEG (one for independent component and one for channel data) and fNIRS, which could be fed into the machine learning classifiers for supervised learning.

Figure 19

Final “goal” data frame for one participant for one task. Columns = features (x variables). Rows = decisecond readings. The green rows represent stutter events (represented by “1” in the “Stutter” column).

Time	Stutter	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature N...
0.1 second	0	1.81	1.81	1.81	1.81	1.81	1.81

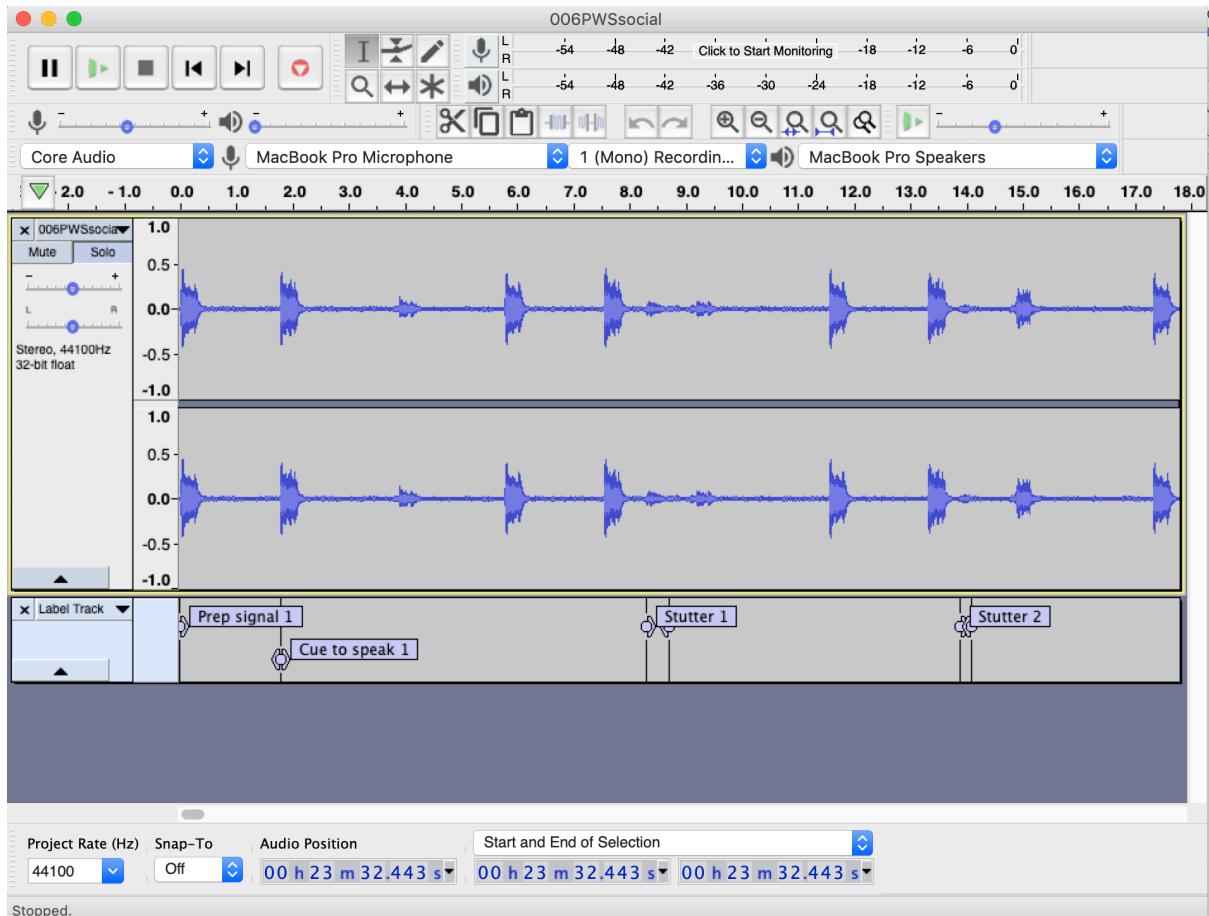
Classifying Stutter Events using fNIRS and EEG data

0.2 second	1	2.18	2.18	2.18	2.18	2.18	2.18
0.3 second	1	1.76	1.76	1.76	1.76	1.76	1.76
0.4 second	1	2.01	2.01	2.01	2.01	2.01	2.01
0.5 second	1	2.11	2.11	2.11	2.11	2.11	2.11
0.6 second	0	3.24	3.24	3.24	3.24	3.24	3.24
0.7 second	0	7.98	1.89	7.19	2.34	7.98	5.68
N second...	0	2.76	1.79	1.71	1.96	2.76	1.52

To label the fNIRS and EEG data, the audio recordings for all 8 data sets were listened to from start to finish, with the researcher labelling each stutter using Audacity free online software (Audacity Team, 2019). The start of the stutter was identified when the participant started making sound but did not produce the word in full (i.e., without repetition or a break). The stutter ended around 0.1 seconds before the word was actually produced as a whole. If several stutters occurred within very close succession (less than a second between them), they were labelled as one stutter.

Figure 20. A sample from the audio labelling the process. The audio file was cut up to “Prep cue 1”. The stutters were then labelled.

Classifying Stutter Events using fNIRS and EEG data



The labels showing the start and finish time of each stutter event were then exported from Audacity into a text file as seen in Figure 21.

Figure 21

The labels export from Audacity showing stutter events. Column one represent stutter start time. Column 2 represents stutter end time. (Column 3 is names the author gave to the labels).

Classifying Stutter Events using fNIRS and EEG data

0.000000	0.000000	Prep signal 1
1.800527	1.800527	Cue to speak 1
8.309678	8.715028	Stutter 1
13.894502	14.097177	Stutter 2
19.727040	20.650338	Stutter 3
27.068382	27.496252	Stutter 4
81.407821	81.835691	Stutter 5
153.965497	154.663600	Stutter 6
171.327995	172.003579	Stutter 7
220.938349	222.762425	Stutter 8
227.919380	228.189613	Stutter 9
238.255808	239.674534	Stutter 10
293.676181	295.567815	Stutter 11
305.904244	307.165333	Stutter 12
311.038679	312.795196	Stutter 13
366.594169	368.193050	Stutter 13
372.539304	374.836288	Stutter 14
383.956666	386.050976	Stutter 15
389.834244	395.103796	Stutter 16
439.444598	441.381271	Stutter 16
456.852134	457.415121	Stutter 17
462.684572	464.170057	Stutter 18

A python script (see appendix 6 which includes notes on how this was done) was written by the researcher to convert these txt documents into a single column of either 1 or 0 values, which corresponded to each decisecond on the fNIRS and EEG data csv files. These columns were then copied into the fNIRS/EEG csv documents in a new column named “Stutter”. At this point, the csv files were ready (as per Figure 19) for input into the machine learning algorithms.

2.4 Analysis

The data and analysis were structured as follows.

2.4.1 Data combinations

“Data combinations” refer to different ways in which the 8 data sets (the separate social and naming tasks for each of the four participants) were combined in order to test the hypotheses.

The following data combination was created for the EEG *channel* data (pre-ICA), and separately for the fNIRS data, which allowed the testing of hypothesis 1 (that a stutter can be

Classifying Stutter Events using fNIRS and EEG data

classified using machine learning) and hypothesis 2 (that RF and SVM will be the highest performing classifiers). All samples available were used, to increase accuracy:

(1) **Four participants – both tasks** (006, 009, 017, 018, social and naming tasks).

Around 112,000 rows of data.

The following data combination was created for EEG only, creating two data sets; one in which *independent components* (ICs) were the columns (features) and one in which *channels* were the columns (features). This allowed us to test hypothesis 3 (that on equally sized data sets, independent component features classify stutters better than channel features).

(2) **Single participant – single task** (006 social task). Around 14,000 rows of data.

2.4.2 Data formats

“Data formats” refer to different ways in which the data combinations above were then manipulated to investigate how the machine learning classifiers were learning from the data and identify any problems. Two different data formats were created.

(1) “**Baseline data**”. This is the main data set created in the process explained in the Methods section. It includes full sets of both EEG (channel) and fNIRS data from all four participants combined (around 112,000 rows). In this data, no more than 10% of the rows represented stutter events.

(2) “**Balanced data**”. In this data set a number of non-stutter (“0”) rows were randomly removed (in python), so that the number of stutter (“1”) rows and non-stutter (“0”)

Classifying Stutter Events using fNIRS and EEG data

rows were the same. The resulting data set had around 10,000 rows, around 5,000 of which were stutters.

2.4.3 Machine learning classification

Each of these data combinations and formats were then run through the four machine learning algorithms KNN, SVM, LogReg and SVM (totalling 16 tests). This was done in a python script (using Python 3.7.0) written by the author of this study (see appendix 7) in Spyder (an open source cross-platform integrated development environment (IDE)). 10-fold cross validation was applied to all classifiers to reduce overfitting. Machine learning was self-taught from online resources (for example, Udemy, Deep Learning A-Z), books, and short university courses (for example a three-day full-time course at Kings College IOPPN) over a period of two years.

3. Results

All results are reported by accuracy rate without cross-validation (holdout method), accuracy with 10-fold cross validation, and by sensitivity (true positive rate), specificity (true negative rate), and AUC (the ability of the model to correctly classify those with and without a specific characteristic, as explained in the introduction).

3.1 Four participants combined (social and naming tasks)

Classifying Stutter Events using fNIRS and EEG data

Both tasks across four participants (8 data sets in total) were combined, creating a data frame with around 112,000 samples/rows for the baseline data (10% stutter events), and around 10,000 for the balanced data (50% stutter events).

The following table (Table 6) provides results to inform hypothesis one (stutters can be classified at an AUC accuracy at least 0.7) and two (SVM or RF will be the top performing classifiers).

Table 6. Results. Four participants combined.

fNIRS data					
	KNN	SVM	Log Reg	RF	Mean of classifiers
Baseline data					
Accuracy	0.93	0.94	0.94	0.93	0.94
Acc. (CV)	0.89	0.95	0.95	0.94	0.93
Sensitivity	0.0	0.0	0.0	0.0	0.0
Specificity	1.0	1.0	1.0	1.0	1.0
AUC	0.5	0.5	0.5	0.5	0.5
Balanced data					
Accuracy	0.84	0.63	0.50	0.90	0.72
Acc. (CV)	0.86	0.62	0.54	0.91	0.70
Sensitivity	0.97	0.56	1.0	0.98	0.88
Specificity	0.71	0.70	0.0	0.83	0.56
AUC	0.84	0.63	0.5	0.90	0.72
EEG data (channels)					
	KNN	SVM	Log Reg	RF	Mean of classifiers
Baseline data					
Accuracy	0.92	0.93	0.93	0.93	0.93
Acc. (CV)	0.94	0.95	0.95	0.95	0.95
Sensitivity	0.03	0.0	0.0	0.0	0.01
Specificity	0.98	1.0	1.0	1.0	1.0
AUC	0.51	0.5	0.5	0.5	0.5
Balanced data					
Accuracy	0.66	0.67	0.58	0.65	0.64
Acc. (CV)	0.67	0.67	0.58	0.64	0.64
Sensitivity	0.69	0.66	0.57	0.63	0.64
Specificity	0.63	0.68	0.60	0.67	0.65

AUC	0.66	0.67	0.58	0.65	<i>0.64</i>
-----	-------------	-------------	-------------	-------------	-------------

3.2 IC versus Channel Data (006 social task)

The table below (Table 7) compares the use of independent components (produced from ICA) to the use of channels as features. This is to inform hypothesis 3 (EEG IC data will outperform EEG channel data). Due to the inability to concatenate IC data between tasks and components (as explained in the introduction), only one task from one participant (one data set) is used in this comparison. This totalled around 14,000 samples/rows at baseline (around 10% of which were stutters) and around 2,500 when balanced (50% of which were stutters).

Table 7. Results. IC versus channel data.

EEG data (ICA – 006_Social)					
	KNN	SVM	Log Reg	RF	Mean of classifiers
Baseline data					
Accuracy	0.90	0.90	0.90	0.90	<i>0.90</i>
Acc. (CV)	0.90	0.91	0.91	0.91	<i>0.91</i>
Sensitivity	0.06	0.0	0.0	0.02	<i>0.02</i>
Specificity	0.99	1.0	1.0	1.0	<i>1.0</i>
AUC	0.53	0.5	0.5	0.51	0.51
Balanced data					
Accuracy	0.68	0.67	0.59	0.66	<i>0.65</i>
Acc. (CV)	0.69	0.67	0.57	0.66	<i>0.65</i>
Sensitivity	0.63	0.57	0.62	0.62	<i>0.61</i>
Specificity	0.71	0.77	0.56	0.71	<i>0.69</i>
AUC	0.67	0.67	0.59	0.66	<i>0.65</i>
EEG data (Channel – 006_Social)					
	KNN	SVM	Log Reg	RF	Mean of classifiers
Baseline data					
Accuracy	0.90	0.90	0.90	0.90	<i>0.90</i>
Acc. (CV)	0.90	0.91	0.91	0.91	<i>0.91</i>
Sensitivity	0.01	0.0	0.0	0.02	<i>0.01</i>

Classifying Stutter Events using fNIRS and EEG data

Specificity	0.99	1.0	1.0	1.0	<i>1.0</i>
AUC	0.5	0.5	0.5	0.51	0.5
Balanced data					
Accuracy	0.65	0.67	0.59	0.65	<i>0.64</i>
Acc. (CV)	0.63	0.65	0.57	0.61	<i>0.62</i>
Sensitivity	0.58	0.58	0.59	0.62	<i>0.61</i>
Specificity	0.71	0.74	0.59	0.67	<i>0.68</i>
AUC	0.64	0.66	0.59	0.65	<i>0.64</i>

Variable importance measures (VIMs)

VIMs refer to the extent each feature contributes to the prediction accuracy of the model. In neuroscience, this can tell us which channels, and as such, which brain regions, contribute most to the predictions. The VIMs of each feature in each analysis in the balanced data set were generated using the random forest classifier. The VIM table can be seen appendix 8. They have been omitted here as there were no substantial differences between VIMs.

4. Discussion

The results are summarised in relation to the hypotheses.

Hypothesis 1: When trained with fNIRS and EEG data which were labelled with stutter and non-stutter events, machine learning algorithms should be able to identify when a stutter occurred at an accuracy of at least 70% (measured by AUC).

The CV accuracy figures for the baseline data were exceptionally high, averaging 0.93 (fNIRS) and 0.95 (EEG). However, accuracy was not a useful performance statistic in this case due to the fact the data set was highly imbalanced. Since over 95% of the data represents non-stutters, the classifier could get a 95% accuracy rate simply by predicting “0” (non-

Classifying Stutter Events using fNIRS and EEG data

stutter) every time, which would hide its inability to correctly predict stutters. The problem with these results can be seen in the very low mean sensitivity figures of 0.02 (fNIRS) and 0.01 (EEG) and the very high mean specificity figures of 1.0 (fNIRS and EEG) showing that whilst the classifiers are predicting non-stutters very well, they are very poor at predicting stutters (doing so correctly only 1-2% of the time). If we look at a confusion matrix for the KNN model on the baseline EEG data for four participants, it can be seen that whilst this model has an excellent prediction accuracy of 94%, most of this figure is made up by the ability of the model to predict non-stutters. It correctly predicts non-stutters (“specificity”) 99% of the time ($N = 26,601$). However, it correctly predicts stutter events (1) only 3.5% ($N = 1,864$), which is very low. The sensitivity was equally low or lower on all four classifiers.

Table 8

Confusion matrix for baseline EEG data from all four participants on the KNN model.

26,180 True negative	421 False positive
1800 False negative	64 True positive

This poor sensitivity could have been the result of insufficient stutter events or too small a proportion of stutter events in the data, to allow the model to learn. As such, a new data format called *Balanced data*, was created in which non-stutter events were randomly removed (in python, see appendix 7 for script) until there was exactly the same number of stutter events as non-stutter events, giving the classifiers equal amounts of data for each class from which to learn.

We can see from the results in Table 5 that when using balanced data, the mean cross-validated accuracy of the classifiers reduced to 0.72 (fNIRS) and 0.64 (EEG) but the

Classifying Stutter Events using fNIRS and EEG data

sensitivity increased hugely, to 0.88 (fNIRS) and 0.64 (EEG). This means that by balancing the data, the ability of the classifiers to correctly predict stutters has increased by a mean of 88% (fNIRS) and 63% (EEG) across classifiers. This was despite the fact balancing reduced the data volume by up to 90%.

The overall accuracies have reduced, but the data is now meaningful as it able to predict both classes. For this reason, only balanced data will be used in addressing the hypotheses going forward.

To address hypothesis 1, the results of the top performing classifier in fNIRS and EEG are examined. For fNIRS data, the random forest classifier was most accurate in predicting stutters, doing so correctly 97% of the time, but also correctly predicting non-stutters 73% of the time. With an AUC of 0.90, it can be concluded that, when trained with balanced fNIRS data which had been labelled with stutter and non-stutter events, random forest can classify when a stutter occurred at an AUC accuracy of over 70% as per the hypothesis. An AUC of 0.9 is considered an excellent figure in the field of psychology.

For the channel EEG data set, the SVM classifier was most accurate in predicting stutters, doing so correctly 66% of the time, but also correctly predicting non-stutters 68% of the time. With an AUC of 0.67, it can be concluded that, in this study, when trained with balanced EEG channel data that has been labelled with stutter and non-stutter events, the machine learning classifiers used were not able to classify when a stutter occurred at an accuracy (measured by AUC) of over 70%. That doesn't mean it is not possible however. This will be discussed in detail in relation to hypothesis 3.

Classifying Stutter Events using fNIRS and EEG data

Hypothesis 2: SVM and random forest will show the highest performances in predicting stutter versus non stutter events.

The top three highest performing classifiers across 16 tests, were (1) Random forest for fNIRS ($AUC = 0.90$), (2) KNN for fNIRS ($AUC = 0.84$), (3) Jointly, KNN and SVM on the full set of EEG IC data, and SVM on the full set of EEG channel data (all $AUC = 0.67$).

As expected, random forest was among the top performing classifiers. This supports the work of Fernandes-Delgado (2014), who identified random forest as a joint best machine learning classifier out of 179 tested. SVM was also among the top three performers, similar to findings by Hosseini et al (2018), Jiang et al (2012) and Fernandes-Delgado (2014). As such, this hypothesis was confirmed.

However, at 0.67, SVM was quite substantially behind the next best algorithm at 0.84 and below the benchmark of 0.7 set for this project. The low performance on the IC data here could be due to the fact the balanced data set for all participants was around a quarter of the size of the data for all four participants (10,000 versus 2,500 samples). If SVM had been performed on an equally sized data set of IC data, it could have been higher (potentially over the 0.7 threshold), so SVM cannot be ruled out as a top performer on IC data. Another explanation for the lower than expected performance of SVM could be because SVM is known to work better after feature selection which was not done in this project. As seen in Myers et al (2018), the decision tree classifier performed best before feature selection, but the SVM performed better after, increasing accuracy by 10%,

Classifying Stutter Events using fNIRS and EEG data

On all data sets, 10-fold cross validation was applied, which reduced the likelihood of overfitting throughout (where the data predicts well on training data but does not generalise to new data). Nevertheless, we cannot rule out that some overfitting occurred.

We know KNN is more prone to overfitting and SVM is more resistant to it (Raschka & Morjalili, 2017). This means there is a chance overfitting could partly explain the unexpectedly high KNN accuracy (in that SVM has lower accuracy here, but may have better generalisability on unseen data). However, because the likelihood of overfitting increases when the ratio of features-to-samples reduces, we would have likely seen higher figures in the single data sets of EEG IC data (than for EEG channel data on all four participants) since the number of features was the same, but the samples were a quarter of those used on the full set. This reduces the features-samples ratio, making overfitting more likely. As can be seen in the results, the KNN performed better on the larger data set (with higher features-to-samples ratio).

There are two potential reasons as to why the fNIRS data showed superior results to the EEG data. One is that the fNIRS data all came from the LIFG and the LIFG has repeatedly been shown to be a key region related to stuttering (as was seen in the introduction). These results cannot be taken as an indication that LIFG contributes more to stuttering than other brain regions however, due to the fact different neuroimaging technologies were used on LIFG (fNIRS) versus the rest of the brain (EEG). This leads to the second reason; that higher fNIRS results could simply be a reflection of fNIRS as a superior method, independent of brain regions being studied. This could be due to the superior spatial resolution of fNIRS (2–3 cm versus EEG, ~5–9 cm), however in a study that is not aiming to differentiate between brain regions, this is of less value. The superior performance of fNIRS could also be due to the fact it is more robust to movement, and that even after filtering and movement removal

Classifying Stutter Events using fNIRS and EEG data

(done on both), fNIRS provides cleaner neural signals, and a more accurate reflection of brain activity. It is worth noting however, that EEG has a better temporal resolution, measuring in milliseconds versus fNIRS in deciseconds. In an experiment where events have very short durations, often less than a second, it could be assumed that a technology with higher temporal resolution would provide more accurate readings. fNIRS performed better despite its lower temporal resolution, suggesting its high performance could indeed be due to its superior spatial resolution or movement resistance. However, it can't be ruled out that the difference was due to stronger neural activity in relation to stuttering from the LIFG (suggestions on how to test this are below).

As expected, logistic regression performed poorest across all balanced data sets of fNIRS and EEG channel data, averaging AUC 0.53 (almost the same as chance). This confirmed that non-parametric classifiers are needed when using correlated neural data. It also performed best on the EEG IC data as expected (AUC = 0.59) but only by 0.01 over EEG channel data from four participants (AUC = 0.58). This small difference could be due to the low data volume of the IC data (around 2,500 samples) compared to the channel data (around 10,000). With more data, the LogReg on IC data would be expected to be higher.

Hypothesis 3: On equally sized data sets, IC features from the EEG data should have a higher prediction rate than channel features.

This hypothesis was designed to test whether ICs provide better insights to neural activity than channel data (a claim by Makeig et al, 1996; Oveis, 2009), even when used in non-parametric models, that do not require independence between predictor variables.

Classifying Stutter Events using fNIRS and EEG data

One of the most interesting findings of this study was that, on equally sized data sets, the models using IC data versus the models using channel data performed almost exactly the same, reaching a top AUC score of 0.67. This strongly suggests that ICA is not needed when using non-parametric models such as SVM. Perhaps the most significant implication of removing the need for ICA, would be that it improves the likelihood of achieving real-time EEG processing, which would be needed for real-time predictions (which in turn would be needed for the real-time intervention systems discussed in the introduction). Another advantage of using channel data is that it is possible to concatenate multiple data sets, unlike IC data, leading to larger data volumes for training. As such, it is suggested that ICA is not needed when developing real-time prediction-intervention neural applications.

It is worth noting that the IC data was a quarter of the size of the full EEG channels data set, yet still matched the accuracy of the full data set at 0.67. Had the models been tested on an IC data set the same size as the full data set, it could have outperformed the channel data. However, given that on equally sized smaller data sets (Table 6), the best IC classifier only outperformed the best EEG classifier by AUC 0.67 to AUC 0.65, a substantially superior performance is unlikely.

None of the EEG models reached the benchmark of AUC 0.7, however at AUC 0.67 for SVM, the EEG channel data was not far off. This means that with more data and some fine tuning (see Future Studies section below), EEG channel data *could* be used to classify stutters in machine learning.

In summary, more work needs done to fully conclude that stutters can be correctly classified using machine learning algorithms applied to neural data. This work would mainly be in the

Classifying Stutter Events using fNIRS and EEG data

form of reliably improving AUC above 0.7 which could be done via larger data sets and sophisticated feature and hyperparameter tuning (see Future Studies below). However, the results in this study provide good evidence to suggest it is possible, and also provide some guidelines for future study. These guidelines are 1) that balanced data sets are more useful than unbalanced, 2) that random forests, KNNs and SVMs are almost always worth including in comparison studies, and 3) that ICA is not necessarily needed when using non-parametric models.

This study has provided new and significant findings that suggest real-time prediction of stutters from neural data could be possible. This is partly due to the good classifier performances seen, but also because ICA has been highlighted as a step which can be removed to increase the likelihood of achieving real-time processing. This means that, when prediction is paired with a suitable, neural stimulation system, stutters could potentially be prevented in real-time. If hardware advances accordingly (which is already happening at a fast rate), this could lead to a situation where, by wearing a kit carried in a baseball cap and backpack, people could prevent themselves stuttering for extended periods of time, such as in job interviews, business meetings or social situations. Ultimately this could give PWS control over their stutters through technology.

Future studies

Data collection

This study showed that balanced data can be a better input than unbalanced data for certain machine learning classifiers. As such, it is recommended that balanced data sets are used in

Classifying Stutter Events using fNIRS and EEG data

future studies. The down side of this that a lot of the majority class data (in this case non-stutters) needs removed and ultimately, wasted. A way to get balanced data sets without having to lose as much non-stutter data (as happened in this study), would be to gather data by conducting interviews or conversations with participants, where they talk more (and as such, stutter more) than they did in the experiment design used in this study. Recruiting participants with severe stuttering could improve the ratio of stuttered to fluent speech also. Another advantage of interviews is that non-stutter events (which here included pauses and preparation as well as fluent speech) could be refined to include only non-stuttered *speech*. This would make the data more uniform and may enable the classifiers to learn better. Lastly, interviews are more similar to real-life situations and would induce a more typical rate of dysfluencies, making the study more useful in real life settings.

As stutters were recorded in audio and not video, only stutters where a sound was made when the participant was stuck on a word were included in the analysis. As stutters can be silent as well as audible, manifesting as unfilled pauses in speech (DSM-5 American Psychiatric Association, 2013; Carlson, 2013), the exclusion of silent stutters (and PWS silently) limited the data set in this study. A recommendation for future studies would be to video-record the experiments rather than just audio record them, to allow for inclusion of silent but visible stutters.

Although the steps above wouldn't create perfectly balanced data sets, they would help generate vastly more useable data and reduce data waste. If the above was done on 20 participants for example, whereby baseline data was comprised of, on average 20% stutter events (rather than 10%), this could lead to ten times the amount of data used in this study.

Classifying Stutter Events using fNIRS and EEG data

To measure whether EEG or fNIRS data is most effective for classifying stutters through machine learning, it is recommended that future studies compare EEG on LIFG and fNIRS on LIFG. To measure which brain *regions* provide the most useful data, fNIRS on LIFG versus fNIRS on other brain regions is suggested.

Pre-processing and labelling

A worthwhile subsequent study would be to test out the least-processed data with which machine learning algorithms can classify stutters. It may be that some classifiers can predict stutters from raw fNIRS and EEG data, which would significantly increase the possibility of real-time prediction.

A comparison could also be done between types of stutters as identified by Jiang et al (2018). For example, less typical (LT) stutters maybe easier to predict than more typical (MT) stutters (common to PWS and FS). To enable this comparison the different stutters would need classified and labelled before going into the final classification model.

Analysis

Whilst it was beyond the scope of this study, the performance of the classifiers could be improved in a number of ways. The first is through feature selection, whereby the features which contribute the least to predictions are removed (these can be seen in the VIMs table in appendix 8). Another method of improving performance could be through hyperparameter tuning, which involves fine-tuning the settings of the model such as number of iterations and learning rate (Dangeti, 2017). Regularization could also be applied to further minimize the

Classifying Stutter Events using fNIRS and EEG data

risk of overfitting. This is a useful method to handle collinearity, filter out noise from data, and eventually prevent overfitting. The concept behind regularization is to introduce additional information (bias) to penalize extreme parameter (weight) values. The most common form of regularization is called L2 regularization (Raschka & Mirjalili, 2017).

It is also recommended that in future studies, performance metrics for machine learning are always reported with sensitivity, specificity and AUC, as well as accuracy. As was seen in the baseline results, accuracy (or cross-validated accuracy) alone can be skewed by imbalanced data and does not tell us the true performance of a classifier.

This study showed that a stutter could be correctly classified by labelling the stutters and non-stutters in the data. To move towards a classifier that can predict, in advance, when a stutter is about to occur, the same analysis would need to be done on the data prior to the stutter event. This could be as simple as moving the stutter labels in the csv files back by 20 rows (2 seconds). Once prediction has been shown possible, a subsequent research project would be to pair up the prediction model with an intervention system, to test out whether tDCS (for example) works in preventing stutters in real time.

References

American Psychiatric Association. Diagnostic and Statistical Manual of Mental Disorders.

DSM-5. 5th ed. Washington, D.C.; London: American Psychiatric Pub., 2013.

Ambrose, N., Cox, N., & Yairi, E. (1997). The Genetic Basis of Persistence and Recovery in

Stuttering. *Journal of Speech Language and Hearing Research*, 40(3), 567. doi:
10.1044/jslhr.4003.567

Andrews, G., Morris-Yates, A., Howie, P., & Martin, N. (1991). Genetic Factors in Stuttering

Confirmed. *Archives of General Psychiatry*, 48(11), 1034. doi:
10.1001/archpsyc.1991.01810350074012

Andrews, G., & Harris, M. (1964). *The Syndrome of stuttering*. London: The Spastics Society.

Aslin, R.N. (2012). Questioning the questions that have been asked about the infant brain using near-infrared spectroscopy. *Cognitive Neuropsychol.* 29, 7–33.

Audacity Team (2019, June-August). Audacity(R): Free Audio Editor and Recorder. Version 2.3.1. Retrieved from [<https://www.audacityteam.org/>]

Barker, J. W., Aarabi, A., & Huppert, T. J. (2013). “Autoregressive model-based algorithm for correcting motion and serially correlated errors in fNIRS,” *Biomed. Opt. Express* 4(8), 1366–1379 (2013).

Classifying Stutter Events using fNIRS and EEG data

- Barker, J.W., Rosso, A.L., Sparto, P.J., & Huppert, T. J. (2016). Correction of Motion Artifacts and Serial Correlations for Real-time Functional Near-infrared Spectroscopy. *Neurophotonics* 3.3: 031410.
- Bashir, N. (2019). Using tDCS to improve speech processes in typical speakers and people who stutter. *UCL PhD Thesis*.
- Beal, D., Gracco, V., Lafaille, S., & De Nil, L. (2007). Voxel-based morphometry of auditory and speech-related cortex in stutterers. *Neuroreport*, 18(12), 1257-1260. doi: 10.1097/wnr.0b013e3282202c4d
- Belyk, M., & Brown, S. (2014). Somatotopy of the extrinsic laryngeal muscles in the human sensorimotor cortex. *Behavioural Brain Research*, 270, 364-371.
- Bloodstein, O. (1995). *A handbook on stuttering*. San Diego, Calif.: Singular Pub. Group.
- Bloodstein, O., & Ratner, N. (2008). *A handbook on stuttering*. Clifton Park, NY: Thomson Delmar Learning.
- Boksa, P. (2013). A Way Forward for Research on Biomarkers for Psychiatric Disorders. *Journal of Psychiatry & Neuroscience*: JPN 38.2: 75-7.
- Brainsight NIRS Manual v2.3 (2018). *Rogue Resolutions*.

Classifying Stutter Events using fNIRS and EEG data

Brigadoi, S., Ceccherini, L., Cutini, S., Scarpa, F., Scatturin, P., Selb, J., Gagnon, L., Boas, D.A., & Cooper, R.J. (2014). Motion artifacts in functional near-infrared spectroscopy: A comparison of motion correction techniques applied to real cognitive data. *Neuroimage*, 85, 181–191.

Broca, P. (1861). Remarks on the seat of the faculty of articulated language, following an observation of aphemia (loss of speech). *Bulletin De La Société Anatomique*, 6, 330-57.

Brown, S., Ingham, R., Ingham, J., Laird, A., & Fox, P. (2005). Stuttered and fluent speech production: An ALE meta-analysis of functional neuroimaging studies. *Human Brain Mapping*, 25(1), 105-117. Doi: 10.1002/hbm.20140

Budde, K., Barron, D., & Fox, P. (2014). Stuttering, induced fluency, and natural fluency: A hierarchical series of activation likelihood estimation meta-analyses. *Brain and Language*, 139, 99-107. Doi: 10.1016/j.bandl.2014.10.002

Carlson, N. (2013). *Physiology of behavior*. Harlow: Pearson.

Carmichael, G. (2004). EEG Signal Can Be Separated by Use of New Technique. *Lancet Neurology* 3.8: 450.

Chang, S.E., Kenney, M. K., Loucks, T. M. J., & Ludlow, C. L. (2009). Brain activation abnormalities during speech and non-speech in stuttering speakers. *NeuroImage*, vol. 46, no. 1, pp. 201–212, May 2009.

Classifying Stutter Events using fNIRS and EEG data

Cooper, R.J., Selb, J., Gagnon, L., Phillip, D., Schytz, H.W., Iversen, H.K., Ashina, M., & Boas, D.A. (2012). A systematic comparison of motion 72rtefact correction techniques

For functional near-infrared spectroscopy. *Frontiers Neuroscience*. 2012, 6, 147.

Couronné, R., Probst, P. & Boulesteix, A. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics* volume 19, Article number: 270.

Dangeti, P. (2017). *Statistics for Machine Learning*. Birmingham: Packt, Limited.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1), 9-21.

DEYMED Diagnostic (2019, July). TruScan Neurofeedback User guide. Retrieved from [\[https://manualzz.com/doc/6294259/truscan-qeeg-user-manual\]](https://manualzz.com/doc/6294259/truscan-qeeg-user-manual)

Diamantidis, N.A, Karlis, D., & Giakoumakis. E.A. (200). Unsupervised Stratification of Cross validation for Accuracy Estimation. *Artificial Intelligence* 116.1 (2000): 1-16.

Dunteman, G. H. & Moon-Ho, R. H. (2006). *An Introduction to Generalized Linear Models*. London: SAGE.

Classifying Stutter Events using fNIRS and EEG data

Dwyer, D.B., Falkai, P., & Koutsouleris, N. (2018). Machine Learning

Approaches for Clinical Psychology and Psychiatry. *Annual Review of Clinical Psychology* 14.1: 91-118. Web.

Ehman, T. (2013). The functional significance of delta oscillations in Cognitive processing. *Frontiers in Integrative Neuroscience* 7: 83.

Etchell, A., Johnson, B., & Sowman, P. (2014). Behavioral and multimodal neuroimaging evidence for a deficit in brain timing networks in stuttering: a hypothesis and theory. *Frontiers in Human Neuroscience*, 8. Doi: 10.3389/fnhum.2014.00467

Felsenfeld, S., Kirk, K., Zhu, G., Statham, D., Neale, M., & Martin, N. (2000). A study of the genetic and environmental etiology of stuttering in a selected twin sample. *Behavior Genetics*, 30(5), 359-366. doi: 10.1023/a:1002765620208

Felsenfeld, S. (2002). Finding susceptibility genes for developmental disorders of speech: the long and winding road. *Journal of Communication Disorders*, 35(4), 329-345. doi: 10.1016/s0021-9924(02)00088-6

Fernandez-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do We Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal Of Machine Learning Research* 15: 3133-181.

Florkowski, C. M. (2008). Sensitivity, Specificity, Receiver-operating Characteristic

Classifying Stutter Events using fNIRS and EEG data

(ROC) Curves and Likelihood Ratios: Communicating the Performance of Diagnostic Tests. *The Clinical Biochemist. Reviews* 29 Suppl 1: S83-7.

Foundas, A.L., Corey, D.M., Angeles, V., Bollich, A.M., Crabtree-Hartman, E., & Heilman, K.M. (2003). Atypical cerebral laterality in adults with persistent developmental stuttering. *Neurology* 61:1378–1385.

Goal, A. & Mahajan, S. (2017). Comparison of the KNN and SVM Algorithms.

International Journal for Research in Applied Science & Engineering Technology (IJRASET). Volume 5 Issue XII December.

Google Developers (2019, July 29). ROC Curve. Retrieved from

[<https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>]

Google Home Assistant (2019, August 4th). Retrieved from
[<https://support.google.com/assistant/answer/9071681?co=GENIE.Platform%3DAndroid&hl=en>]

Guenther, F., & Ghosh, S. (2003). A model of cortical and cerebellar function in speech. In *Proceedings of The Xvth International Congress of Phonetic Sciences*, 169-173.

Guitar, B. (2006). *Stuttering*. Philadelphia: Lippincott Williams & Wilkins.

Hillis, A., Work, M., Barker, P., Jacobs, M., Breese, E., & Maurer, K. (2004). Re-examining the brain regions crucial for orchestrating speech articulation. *Brain*, 127(7), 1479-1487. Doi: 10.1093/brain/awh172

Classifying Stutter Events using fNIRS and EEG data

Hocke, L. M., Ibukunoluwa, K.O., Duszynski, C.C., Corrigan, A.V., Frederick, B.D., & Dunn, J.F. (2018). Automated Processing of FNIRS Data-A Visual Guide to the Pitfalls and Consequences. *Algorithms*, May 2018, Vol.11(5).

Homae, F. (2014). A brain of two halves: Insights into interhemispheric organization provided by near-infrared spectroscopy. *NeuroImage*, vol. 85, pp. 354–362, Jan.

Hosseini, R., Walsh, F.T., & Shouyi, W. (2018). *An FNIRS-Based Feature Learning and Classification Framework to Distinguish Hemodynamic Patterns in Children Who Stutter*. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26.6: 1254-263.

Howard, P.D, Helmus, M.L., & Babchishin, K.M. (2017). The Effect of Sample Heterogeneity and Risk Categorization on Area Under the Curve Predictive Validity Metrics. *Criminal Justice and Behavior* 44.1: 103-20. Web.

Howie, P. (1981). Concordance for Stuttering in Monozygotic and Dizygotic Twin Pairs. *Journal of Speech Language and Hearing Research*, 24(3), 317. doi: 10.1044/jshr.2403.317

Hu, X.S., Arredondo, M.M., Gomba, M., Confer, N., DaSilva, A.F., Johnson, T.D., Shalinsky, M., & Kovelman, I. (2015). Comparison of motion correction techniques applied to functional near-infrared spectroscopy data from children. *J. Biomed. Opt.* 20, 126003.

Classifying Stutter Events using fNIRS and EEG data

Huppert, T. J., Diamond, S. G., Franceschini, M. A., & Boas, D. A. (2009). HomER: a review of time-series analysis methods for near-infrared spectroscopy of the brain. *Applied optics*, 48(10), D280-D298.

Huppert, T.J., Diamond, S.G., Franceschini, M.A., Boas, D.A. (2009). Homer: A review of Time series analysis methods for near-infrared spectroscopy of the brain. *Appl. Opt.* 48, D280–D298.

Huppert, T. J. (2016). Commentary on the statistical properties of noise and its implication on general linear models in functional near-infrared spectroscopy. *Neurophotonics* 3:010401. Doi: 10.1117/1.NPh.3.1.010401

Izenman, A.J., (2008). *Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning*. New York: Springer. Print.

Jia, X, & Adam Kohn. (2011). Gamma Rhythms in the Brain. *PloS Biology* 9.4: E1001045.

Jobsis, F. F. (1977). Noninvasive infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters. *Science*, 198: 1264–1267.

Kabdebon, C., Leroy, F., Simmonet, H., Perrot, M., Dubois, J., and Dehaene-Lambertz, G. (2014). Anatomical Correlations of the International 10–20 Sensor Placement System in Infants. *NeuroImage* 99: 342-56. Web.

Classifying Stutter Events using fNIRS and EEG data

Kings College (2019, May 20th – May 22nd). Machine Learning for Health and Bioinformatics (3-day course). Retrieved from

[<https://www.kcl.ac.uk/ioppn/depts/biostatisticshealthinformatics/teaching/courses/machine-learning>]

Klimesch, W. (1999). EEG Alpha and Theta Oscillations Reflect Cognitive and Memory Performance: A Review and Analysis. *Brain Research Reviews* 29.2-3: 169-95. Web.

Klimesch, W. (2012). Alpha-band Oscillations, Attention, and Controlled Access to Stored Information. *Trends in Cognitive Sciences* 16.12: 606-17. Web.

Kumra, S., Ashtari, M., Wu, J., Hongwanishkul, D., White, T., Cervellione, K., Cottone, J., & Szeszko, P.R. (2011). Gray Matter Volume Deficits Are Associated with Motor and Attentional Impairments in Adolescents with Schizophrenia. *Progress in Neuropsychopharmacology & Biological Psychiatry* 35.4: 939-43. Web.

Makeig, S., Bell, A.J., Jung T.P., & Sejnowski, T.J. (1996). Independent component analysis of electroencephalographic data. *Adv Neural Inf Process Syst*;8:145–51.

Mock, J., Foundas, A.L, & Golob, E.J. (2016). Cortical Activity during Cued Picture Naming Predicts Individual Differences in Stuttering Frequency. *Clinical Neurophysiology* 127.9: 3093-101. Web.

Molavi, B., & Dumont, G.A. (2012). Wavelet-based motion 77rtefact removal for functional near infrared spectroscopy. *Physiol. Meas.* 33, 259–270.

Classifying Stutter Events using fNIRS and EEG data

Mossman, D. (1994). Assessing predictions of violence being accurate about accuracy.

Journal of Consulting and Clinical Psychology, 62, 783–792.

Motamedi-Fakhr, S., Moshrefi-Torbat, M., Hill, M., Hill, C. M., & White, P. R. (2014).

Signal Processing Techniques Applied to Human Sleep EEG Signals—A Review.

Biomedical Signal Processing and Control 10.1: 21-33. Web.

Myers, J.C., Irani, F., Golob, E.J., Mock, J.R., Robbins, K.A. (2018). Single-Trial Classification of Disfluent Brain States in Adults Who Stutter. *IEEE International Conference on Systems, Man, and Cybernetics*. DOI 10.1109/SMC.2018.00019

Neef, N., Anwander, A., & Friederici, A. (2015). The Neurobiological Grounding of Persistent Stuttering: from Structure to Function. *Current Neurology and Neuroscience Reports*, 15(9). Doi: 10.1007/s11910-015-0579-4

Neef, N. E., Bütferring, C., Anwander, A., Friederici, A. D., Paulus, W., & M. Sommer (2016). Left posterior-dorsal area 44 couples with parietal areas to promote speech fluency, while right area 44 activity promotes the stopping of motor responses. *NeuroImage*, vol. 142, pp. 628–644, Nov.

Neef, B., Anwander, F., Paulus, and Sommer. (2016). Left Posterior-dorsal Area 44 Couples With Parietal Areas to Promote Speech Fluency, While Right Area 44 Activity Promotes the Stopping of Motor Responses. *NeuroImage* 142: 628-44. Web.

Classifying Stutter Events using fNIRS and EEG data

Neumann, E., Gudenberg, G., Lanfermann, G., & Preibisch. (2003). The Nature and Treatment of Stuttering as Revealed by FMRI: A Within- and Between-group Comparison. *Journal of Fluency Disorders* 28.4: 381-410. Web.

Onton, J. & Makeig, S. (2006). Information-based Modeling of Event-related Brain Dynamics. *Progress in Brain Research* 159: 99-120..

Orihuela-Espina, F., Leff, D.R., James, D.R., Darzi, A.W., & Yang, G.Z. (2010). Quality control and assurance in functional near infrared spectroscopy (fnirs) experimentation. *Phys. Med. Biol.* 55, 3701–3724.

Oveisi, F. (2009). ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings: 361-64.

Pinti, P., Aichelburg, C., Gilbert, S., Hamilton, A., Hirsch, J., Burgess, P., & Tachtsidis, I. (2018). A Review on the Use of Wearable Functional Near-Infrared Spectroscopy in Naturalistic Environments. *Japanese Psychological Research*, 60 (4), 347-373.
doi:10.1111/jpr.12206

Pinti, P., Scholkmann, F., Hamilton, A., Burgess, P., & Tachtsidis, I. (2019). Current Status and Issues Regarding Pre-processing of fNIRS Neuroimaging Data: An Investigation of Diverse Signal Filtering Methods Within a General Linear Model Framework. *Frontiers in Human Neuroscience*. January. Volume 12. Article 505

Pion-Tonachini, L. (2019, July 26). ICLLabel Tutorial. Retrieved from

Classifying Stutter Events using fNIRS and EEG data

[<https://labeling.ucsd.edu/tutorial/labels>]

Putze, F., Hesslinger, S., Tse, C., Huang, Y., Herff, C., Guan, C., and Schultz, T. (2014).

Hybrid FNIRS-EEG Based Classification of Auditory and Visual Perception Processes. *Frontiers in Neuroscience* 8: 373. Web.

Raschka, S., & Vahid M. (2017). *Python Machine Learning* – Second Edition.

Birmingham: Packt, Limited.

Radüntz, T., Scouten, J., Hochmuth, O. & Meffert, B. (2015). EEG Artifact Elimination by Extraction of ICA component Features Using Image Processing Algorithms. *Journal of Neuroscience Methods* 243: 84-93.

Reilly, S., Onslow, M., Packman, A., Cini, E., Conway, L., & Ukoumunne, O. (2013).

Natural History of Stuttering to 4 Years of Age: A Prospective Community-Based Study. *Pediatrics*, 132(3), 460-467. doi: 10.1542/peds.2012-3067

Rice, M. E., & Harris, T. G. (2005). Comparing Effect Sizes in Follow-Up Studies: ROC Area, Cohen's D, and R. *Law and Human Behavior* 29.5: 615-20. Web.

SCCN (2019, July 9). EEGLAB Guide. Chapter 09. Decomposing Data Using ICA.

Retrieved from

[https://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA]

Schiweck, C., Piette, D., Berckmans, D., Claes, S., and Vrieze, E. (2019). Heart Rate and

Classifying Stutter Events using fNIRS and EEG data

High Frequency Heart Rate Variability during Stress as Biomarker for Clinical Depression. A Systematic Review. *Psychological Medicine* 49.2: 200-211. Web.

Shimadzu (2019, August 1). Portable fNIRS systems. Retrieved from

[<https://www.ssi.shimadzu.com/products/imaging/lightnirs.html>]

Smith, S. (2003). *Digital signal processing: A practical guide for engineers and scientists* (Demystifying technology series). Amsterdam; Oxford: Newnes.

Sommer, M., Koch, M., Paulus, W., Weiller, C., & Büchel, C. (2002). Disconnection of speech-relevant brain areas in persistent developmental stuttering. *The Lancet*, 360(9330), 380-383. doi: 10.1016/s0140-6736(02)09610-1

Subasi, A., & Erçelebi, E. (2005). Classification of EEG Signals Using Neural Network and Logistic Regression.” *Computer Methods and Programs in Biomedicine* 78.2: 87-99.

Swets, J. A., Dawes, R. M., & Monahan, J. (2000). Psychological science can improve diagnostic decisions. *Psychological Science in the Public Interest: A Journal of the American Psychological Society*, 1, 1–26.

Tachtsidis, I., & Scholkmann, F. (2016). False positives and false negatives in functional near-infrared spectroscopy: Issues, challenges, and the way forward. *Neurophotonics*, 3, 031405.

Takahashi, K.G., Nicholas, G. D., Hatsopoulos, M.S., & Penn, R.D. (2011). Propagating

Classifying Stutter Events using fNIRS and EEG data

Waves in Human Motor Cortex. *Frontiers in Human Neuroscience* 5. April, (2011): 40.

The MathWorks, Inc (v2018b). MATLAB and Statistics Toolbox. Natick, Massachusetts, United States.

Tourville, J., & Guenther, F. (2011). The DIVA model: A neural theory of speech acquisition and production. *Language and Cognitive Processes*, 26(7), 952-981. doi: 10.1080/01690960903498424

Pion-Tonachini, L., Kreutz-Delgado, K., & Makeig., S. (2019). ICLLabel: An Automated Electroencephalographic Independent Component Classifier, Dataset, and Website. *NeuroImage* 198: 181-97.

The Brain Driver (2019, July 31). Portable tDCS systems. Retrieved from

[<https://thebraindriver.com/#slidepro>]

Udemy (2018, July-September). Deep Learning A-Z Online Course. Retrieved from

[<https://www.udemy.com/deeplearning/>]

Vanhoutte, S., Cosyns, M., van Mierlo, P., Batens, K., Corthals, P., De Letter, M., Van Borsel, J., and Santens, P. (2016). When Will a Stuttering Moment Occur? The Determining Role of Speech Motor Preparation. *Neuropsychologia* 86: 93-102. Web.

Van Riper, C. (1982). *The nature of stuttering*. Englewood Cliffs, Nj: Prentice-Hall.

Classifying Stutter Events using fNIRS and EEG data

- Walsh, B., Tian, F., Tourville, J.A., Yücel, M. A., Kuczak, T., & Bostian, A. J. (2017). Hemodynamics of speech production: An fNIRS investigation of children who stutter. *Sci. Rep.*, vol. 7, Jun. Art. No. 4034.
- Ward, D. (2006). *Stuttering and cluttering*. Hove, England: Psychology Press.
- Watkins, K., Smith, S., Davis, S., & Howell, P. (2008). Structural and functional abnormalities of the motor system in developmental stuttering. *Brain*, 131(1), 50-59. doi: 10.1093/brain/awm241
- Watkins, C., & Möttönen. (2017). Investigating the Feasibility of Using Transcranial Direct Current Stimulation to Enhance Fluency in People Who Stutter. *Brain and Language* 164: 68-76.
- Wikipedia (2019, August 1). Infrared. Retrieved from [<https://en.wikipedia.org/wiki/Infrared>]
- Wu, J., Maguire, G., Riley, G., Fallon, J., LaCasse, L., & Chin, S. (1995). A positron emission tomography [18F] deoxyglucose study of developmental stuttering. *Neuroreport*, 6(3), 501- 505. Doi: 10.1097/00001756-199502000-00024
- Yairi, E., & Ambrose, N. G. (1999). Early childhood stuttering I: Persistency and recovery rates. *Journal of Speech, Language, and Hearing Research*, 42(5), 1097-1112.
- Yairi, E., & Ambrose, N. (2013). Epidemiology of stuttering: 21st century advances. *Journal of Fluency Disorders*, 38(2), 66-87. doi: 10.1016/j.jfludis.2012.11.002

Appendices

Contents

1. Neural regions most associated with stuttering - expansion
2. Neuronavigation data example
3. EEG electrodes and related neural regions
4. Experiment procedure - full
5. Time points and the time-syncing procedure
6. Python script for labelling the data
7. Python script for machine learning classification
8. Variable importance measures (VIMs)

1. Neural regions most associated with stuttering - expansion

Left Interior frontal gyrus (LIFG). As can be seen in able 1, the brain region most associated with differences between FS and PWS is the LIFG (also known as “Broca’s area”). In a recent review of the past 20 years of neuroimaging with PWS, LIFG dysfunction was highlighted as a key theme (Etchell et al., 2018). It has been suggested that the LIFG may develop atypically in children who stutter (Hosseini et al, 2018), and evidence has been found to suggest that the LIFG is underactive during speaking in PWS compared to FS, a difference which is linked to increased stuttering severity (Wu et al., 1995; Fox et al., 1996; Neumann et al., 2005; Watkins et al., 2008; Kell et al., 2009; Toyomura et al., 2011; Lu et al., 2010; Sowman et al., 2012; Budde et al., 2014; Neef et al., 2015; Etchell et al., 2018). For example, evidence of reduced activation of LIFG during speech production from fMRI studies was found in adults who stutter (Change et al, 2009, Neef et al, 2016). fNIRS studies

Classifying Stutter Events using fNIRS and EEG data

have also found reduced LIFG activation in PWS, for example Walsh et al (2013). One suggested explanation for this is that reduced activation could represent a shift in blood flow to regions outside of the LIFG area to compensate for functional deficits in the LIFG. This compensation argument is supported by Sommer et al (2012), who found right-hemisphere language areas were overactive in people who stutter and proposed that this reflects a compensatory mechanism for a disturbed temporal pattern of activation in the premotor and motor cortex.

It is important to note that whilst a sizeable body of literature shows abnormalities in LIFG functioning contribute to stuttering, the mechanism by which this impairment results in stuttering is not clear (Brown et al., 2005). Indeed, the LIFG is most likely part of a wider network of regions in which dysfunction results in stuttering (Etchell et al., 2018; Neef et al., 2018), a network which likely also includes motor areas.

Motor areas. Brown et al (2005) found, in an eight-study meta-analysis, over-activation in the primary motor cortex, supplementary motor area, cingulate motor area, and cerebellar vermis in stuttered speech compared to fluent speech. More recently, stuttered speech has been related to increased bilateral supplementary motor area (SMA) activation and reduced bilateral primary auditory cortex activation in PWS compared to FS. (Belyk et al., 2014; Budde et al., 2014).

Auditory. Research has suggested stuttered speech is related to reduced bilateral primary auditory cortex activation in PWS compared to FS; whereas fluent speech production is linked to reduced left auditory cortex activity (Belyk et al., 2014; Budde et al., 2014). Brown

Classifying Stutter Events using fNIRS and EEG data

et al (2005) also found absence of activation in auditory areas bilaterally in people who stutter.

2. Neuronavigation example (from 006_Social).

Version: 8
Coordinate system: MNI
Created by: Brainsight 2.3.9
Units: millimetres, degrees, milliseconds, and microvolts
Encoding: UTF-8
Notes: Each column is delimited by a tab. Each value within a column is delimited by a semicolon.
Target Name Loc_X Loc_Y Loc_Z m0n0 m0n1 m0n2 m1n0 m1n1 m1n2 m2n0 m2n1 m2n2
Left IFG -40 24 -10 -0.366 -0.928 -0.072 0 -0.077 0.997 -0.931 0.365 0.028
Right IFG 40 24 -10 0.371 0.926 -0.072 0 0.077 0.997 0.929 0.37 -0.029
Electrode Name Electrode Ty Session Name Optode Nam Optode Connections Loc_X Loc_Y Loc_Z m0n0 m0n1 m0n2 m1n0 m1n1 m1n2 m2n0 m2n1 m2n2
Fp1 (1) EEG 006PWS SOCIAL (null) (null) -43 58 47.372 -0.779 -0.285 -0.559 -0.19 -0.742 0.642 -0.598 0.606 0.524
AF7 (24) EEG 006PWS SOCIAL (null) (null) -59.792 47 34 -0.596 -0.638 -0.487 -0.061 -0.569 0.82 -0.801 0.518 0.3
F7 (3) EEG 006PWS SOCIAL (null) (null) -76.301 22 21 -0.31 -0.699 -0.645 -0.087 -0.654 0.751 -0.947 0.289 0.142
T3 (8) EEG 006PWS SOCIAL (null) (null) -89 -26.543 -25 0.018 -0.775 -0.632 -0.243 -0.616 0.749 -0.97 0.14 -0.199
T5 (13) EEG 006PWS SOCIAL (null) (null) -70.536 -84 -10 0.528 -0.847 0.053 -0.066 0.021 0.998 -0.846 -0.531 -0.045
O1 (18) EEG 006PWS SOCIAL (null) (null) -32 -112.716 -47 0.118 -0.108 0.987 -0.915 0.373 0.15 -0.385 -0.921 -0.055
FCS (20) EEG 006PWS SOCIAL (null) (null) -83.495 -22 38 -0.171 -0.215 -0.961 -0.071 -0.971 0.23 -0.983 0.107 0.151
CP5 (25) EEG 006PWS SOCIAL (null) (null) -80.435 -52 39 0.047 -0.69 -0.723 0.313 -0.677 0.666 -0.948 -0.258 0.184
F3 (4) EEG 006PWS SOCIAL (null) (null) -50 21 70.482 -0.759 -0.447 -0.473 0.107 -0.802 0.587 -0.642 0.395 0.657
C3 (9) EEG 006PWS SOCIAL (null) (null) -65.001 -8 69 -0.506 -0.831 -0.232 0.447 -0.483 0.753 -0.737 0.277 0.616
P3 (14) EEG 006PWS SOCIAL (null) (null) -57 -86.647 52 0.513 -0.003 0.858 -0.556 0.761 0.335 -0.654 -0.649 0.389
RA24 (B1-D6) NIRS-Detect006PWS SOCIAL B1D6 B153 82.022 -26 46 -0.225 0.742 0.632 -0.167 -0.668 0.725 0.96 0.058 0.274
FC1 (21) EEG 006PWS SOCIAL (null) (null) -32 7 89.203 -0.765 0.365 -0.53 -0.471 -0.879 0.074 -0.439 0.306 0.845
CP1 (26) EEG 006PWS SOCIAL (null) (null) -26 -74.285 88 -0.015 0.825 0.564 -0.963 0.141 -0.231 -0.271 -0.547 0.792
RA34 (B1-D7) NIRS-Detect006PWS SOCIAL B1D7 B153;B154 75.168 9 43 -0.446 0.711 0.544 -0.091 -0.64 0.763 0.89 0.291 0.35
Fz (5) EEG 006PWS SOCIAL (null) (null) -6 42.565 81 0.435 0.737 -0.517 -0.895 -0.417 0.159 -0.099 0.532 0.841
Cz (10) EEG 006PWS SOCIAL (null) (null) -1 -31 104.509 -0.833 0.553 0 -0.553 -0.833 0 0 0 1
Pr (15) EEG 006PWS SOCIAL (null) (null) 1 -98 67.673 0.078 0.539 0.839 -0.997 0.019 0.08 0.027 -0.842 0.539
RA34 (B1-D5) NIRS-Detect006PWS SOCIAL B1D5 B153 85.49 -39 12 0.072 0.651 0.755 -0.133 -0.744 0.654 0.988 -0.148 0.033
Oz (30) EEG 006PWS SOCIAL (null) (null) -2 -121.527 3 0.999 0.007 0.052 -0.052 0.029 0.998 0.005 -1 0.029
Fp2 (2) EEG 006PWS SOCIAL (null) (null) 41 62.649 45 0.818 0.503 0.279 -0.055 0.55 0.833 0.572 0.667 0.478
AF8 (29) EEG 006PWS SOCIAL (null) (null) 56 54.655 32 -0.601 0.791 -0.111 -0.331 -0.12 0.936 0.727 0.599 0.335
F8 (7) EEG 006PWS SOCIAL (null) (null) 70.703 42 4 -0.274 0.426 0.862 0.37 -0.781 0.504 0.888 0.457 0.056
LA14 (B1-D1) NIRS-Detect006PWS SOCIAL B1D1 B151 -83.961 -48 12 0.206 -0.897 -0.391 0.127 -0.372 0.919 -0.97 -0.239 0.037

3. EEG electrodes and related neural regions

Data from thirty EEG channels were collected. The brain regions corresponding to these channels can be seen below. Labels are coded as follows: Pre-frontal (Fp), frontal (F), temporal (T), parietal (P), occipital (O), and central I, between parietal and central (CP), between frontal and central (FC), between frontal and pre-frontal (AF), electrodes placed on the the midline sagittal plane of the skull for measurement for reference/measurement points (FpZ, Fz, Cz, Oz). Kabdebon, Leroy, Simmonet, Perrot, Dubois, and Dehaene-Lambertz (2014).

Channel	Brain Region
Fp1	Orbitary superior frontal gyrus Orbitary inferior frontal gyrus Middle frontal gyrus
Fp2	Orbitary superior frontal gyrus Orbitary inferior frontal gyrus Middle frontal gyrus
F7	Inferior frontal gyrus, triangular part Orbitary middle frontal gyrus
F3	Middle frontal gyrus
Fz	Supplementary motor area Medial superior frontal gyrus
F4	Middle frontal gyrus
F8	Inferior frontal gyrus, triangular part Orbitary middle frontal gyrus
T7	Mid-temporal left
C3	Postcentral gyrus Supramarginal gyrus
Cz	Paracentral lobule

Classifying Stutter Events using fNIRS and EEG data

	Supplementary motor area
C4	Postcentral gyrus Supramarginal gyrus
T8	Mid-temporal right
P7	Inferior parietal region
P3	Angular gyrus Middle occipital gyrus
Pz	Superior parietal gyrus Superior occipital gyrus
P4	Angular gyrus Middle occipital gyrus
P8	Inferior parietal region
O1	Middle occipital gyrus Inferior occipital gyrus
O2	Middle occipital gyrus Inferior occipital gyrus
FC5	Front-central
FC1	Front-central
FC2	Front-central
FC6	Front-central
AF7	Between frontal and pre-frontal
CP5	Between parietal and central
CP1	Between parietal and central
CP2	Between parietal and central
CP6	Between parietal and central
AF8	Between frontal and pre-frontal
Oz, EXG1, EXG2, E33, Sync, Gnd-Ref, Gnd-Ref1, Gnd-Ref2	Neuronavigation points that were exported from EEGLAB but not include in the analysis

4. Experiment procedure - full.

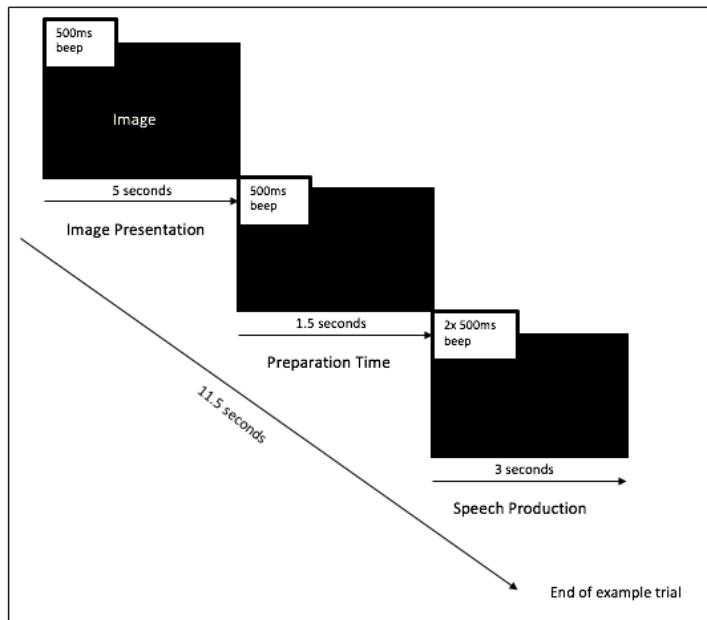
Picture Naming Task. 100 different images were employed in the Picture Naming task. To facilitate audio analysis, images were specifically selected so the names of the objects represented were longer than one syllable. Stimuli were selected collectively from the Bank of Standardised Stimuli (BOSS) (Brodeur et al., 2010) database (Version 1.3). The 100 images were randomly presented in 20 blocks of five images per block, with 30s breaks in between each block.

The basic structure of each trial involved the image presentation, a preparation period for the participants' responses, and a verbal response window (Figure 1). Blocks opened with a 30s

Classifying Stutter Events using fNIRS and EEG data

break and a fixation cross preceding the image. Images were displayed for 500ms and replaced with blank screen after a 500ms auditory cue (beep). A 1.5s preparation period followed where participants were given time to recognise and prepare to name the object. Two 500ms beeps then cued a 3s speech production interval during which the participants named the object. A fixation cross then followed before the next trial. Each block had five trials, after which the next 30s rest period was initiated to re-establish resting haemodynamic response.

Figure 1. Visual example of a trial in the Picture Naming task.



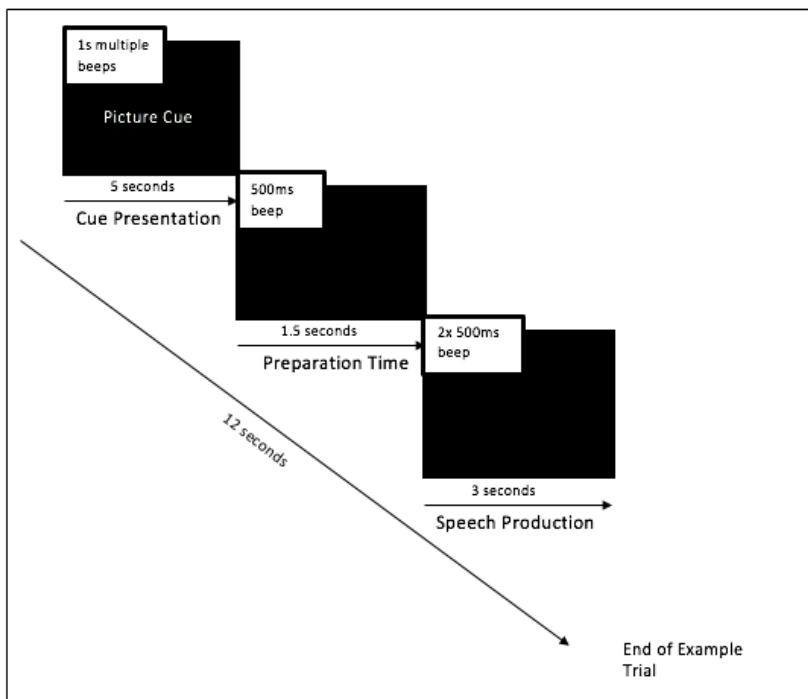
Social Task. The Social task had 20 verbal cues that appeared on-screen asking participants to list five items relevant to the cue (e.g “items you would find in a fridge”). Such verbal cues, shown in a randomised order, were accompanied by visual prompts that provided a visual representation related to the cue. The participant was thus tasked to list 100 items in

Classifying Stutter Events using fNIRS and EEG data

total across all blocks. A trial consisted of the presentation of visual and verbal cue, a preparation time and a verbal response time (Figure 2).

A series of 1s multiple beeps indicated the start of a block after a 32s rest period. A picture and verbal cue (e.g Underground Lines) appeared for 5s. A 500ms beep indicated a preparation time of 1.5s for the participant to prepare their answer, and eye contact was held with the researcher. Two 500ms beeps then cued a 3s timeslot when the participant was to speak. The preparation and production cues were then repeated four more times and were followed by a 32s break indicating the end of the block. Eye contact during the break was encouraged, but participants could take short breaks from sustained eye contact until the next block.

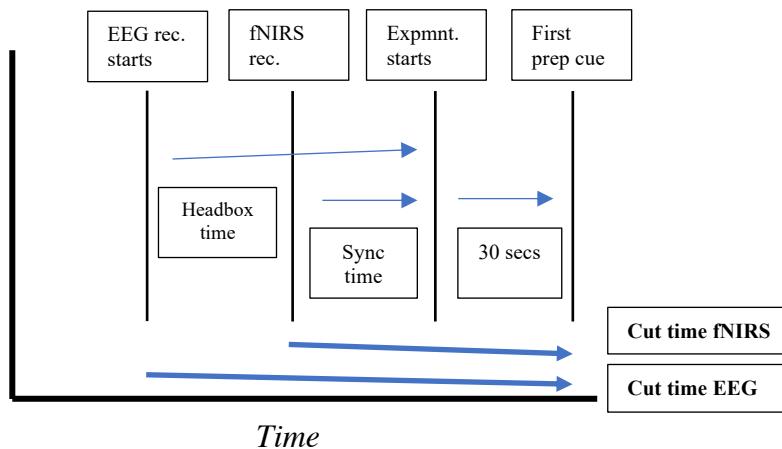
Figure 2. Visual example of a trial in the Social task.



Classifying Stutter Events using fNIRS and EEG data

5. Time points and the time-syncing procedure

During the experiment, time points (in seconds) were extracted to show the time difference between the start of the EEG recording and the experiment start (“headbox time”), and between the fNIRS recording and the experiment start (“sync time”). The distance between the experiment start and the first prep cue was always 30 seconds. Therefore, by combining the headbox time and 30 seconds, we could calculate how many rows to remove from the EEG data. By combining the sync time and 30 seconds, we could calculate how many rows to remove from the fNIRS data. This is illustrated in the diagram below (the EEG always started recording first).



The “sync” times and “headbox” times for each participant in each task are below.

	Sync time	Headbox time	Number of lines to cut from csv	
		Seconds to cut	fNIRS*	EEG**
	fNIRS	EEG		
006Naming	12s	23s	420	530
006Social	12s	30s	420	600
009Naming	7s	18s	370	480
009Social	11s	19s	410	490
017Naming	4s	16s	340	460
017Social	6s	10s	360	400
018Naming	5s	13s	350	460
018Social	5s	13s	350	400
			$*(Sync \times 10) + 300$	$**(\text{Headbox} \times 10) + 300$

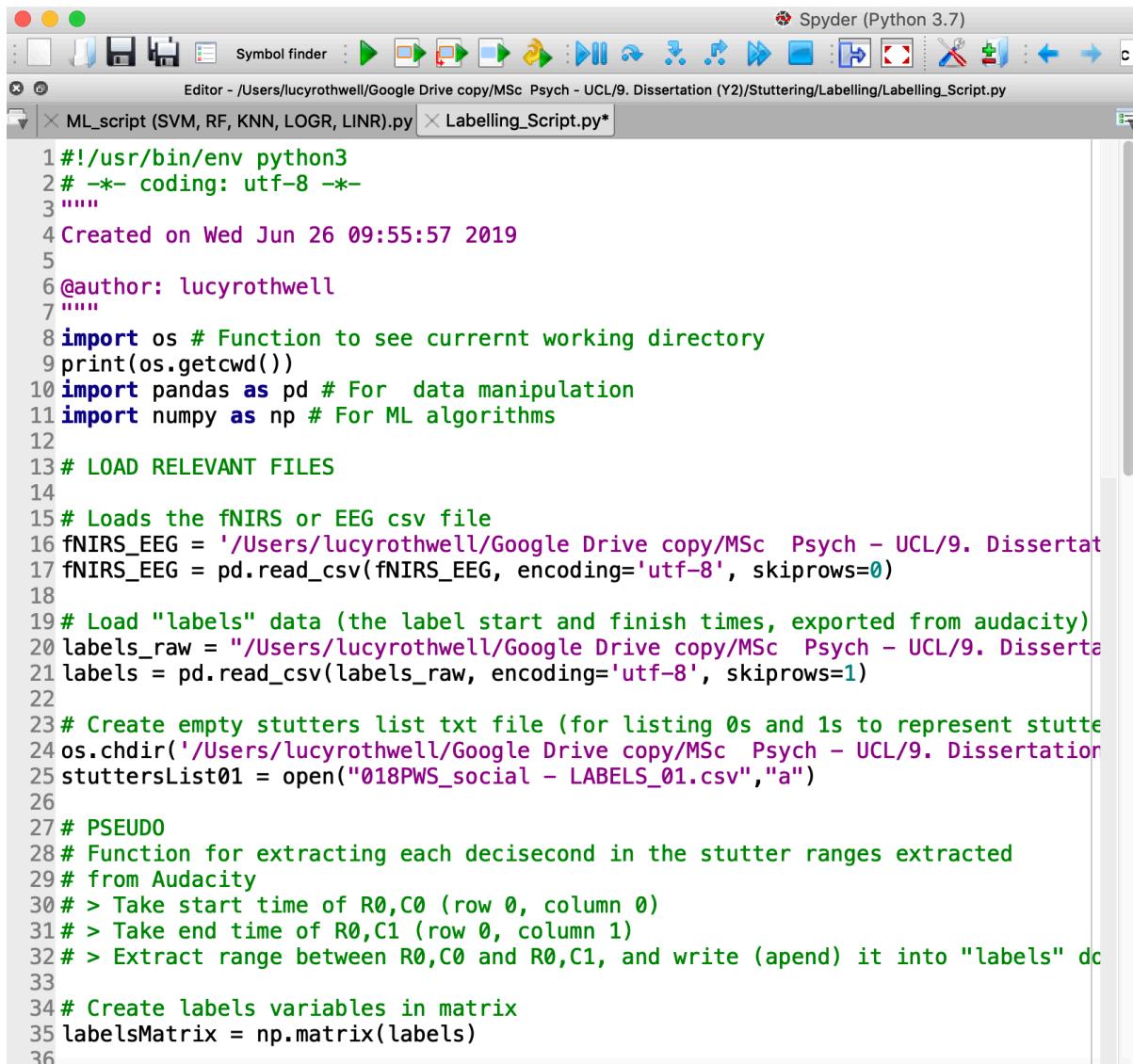
Classifying Stutter Events using fNIRS and EEG data

- “**Sync time**” = The distance in seconds between the time fNIRS started recording and “Experiment Start”. For example on 006_Social, it’s 12 seconds.
- “**Headbox time**” = The distance in seconds between the time EEG started recording and “Experiment Start” (Headbox). For example on 006_Social, it’s 23 seconds.
- **30 seconds** = The time between the experiment start and the “first preparation cue”
- **Example for 006_Social, fNIRS.**

12 seconds + 30 seconds = 42 seconds to remove

42 seconds * 10 (for deciseconds) = 420 rows to delete from the fNIRS csv export

6. Python script for data labelling



The screenshot shows the Spyder Python IDE interface. The title bar reads "Spyder (Python 3.7)". The main window displays a Python script titled "Labelling_Script.py". The code itself is as follows:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Wed Jun 26 09:55:57 2019
5
6 @author: lucyrothwell
7 """
8 import os # Function to see current working directory
9 print(os.getcwd())
10 import pandas as pd # For data manipulation
11 import numpy as np # For ML algorithms
12
13 # LOAD RELEVANT FILES
14
15 # Loads the fNIRS or EEG csv file
16 fNIRS_EEG = '/Users/lucyrothwell/Google Drive copy/MSc Psych - UCL/9. Dissertation'
17 fNIRS_EEG = pd.read_csv(fNIRS_EEG, encoding='utf-8', skiprows=0)
18
19 # Load "labels" data (the label start and finish times, exported from audacity)
20 labels_raw = "/Users/lucyrothwell/Google Drive copy/MSc Psych - UCL/9. Dissertation"
21 labels = pd.read_csv(labels_raw, encoding='utf-8', skiprows=1)
22
23 # Create empty stutters list txt file (for listing 0s and 1s to represent stutters)
24 os.chdir('/Users/lucyrothwell/Google Drive copy/MSc Psych - UCL/9. Dissertation')
25 stuttersList01 = open("018PWS_social - LABELS_01.csv", "a")
26
27 # PSEUDO
28 # Function for extracting each decisecond in the stutter ranges extracted
29 # from Audacity
30 # > Take start time of R0,C0 (row 0, column 0)
31 # > Take end time of R0,C1 (row 0, column 1)
32 # > Extract range between R0,C0 and R0,C1, and write (append) it into "labels" dc
33
34 # Create labels variables in matrix
35 labelsMatrix = np.matrix(labels)
36
```

Classifying Stutter Events using fNIRS and EEG data

```
37 def addRange (labels, labelsMatrix):
38     labels = open(labels_raw, "a")
39     row = 0
40     for i in labelsMatrix:
41         x = round(labelsMatrix[row,0],1)
42         y = labelsMatrix[row,1]
43         labels.write("\n" + str(x))
44         while x < y:
45             x = x + 0.1
46             x = round((x),1)
47             labels.write("\n" + str(x))
48         row = row + 1
49     labels.close()
50
51 addRange(labels, labelsMatrix)
52
53 # Updating the labels variable with the updated labels document
54 labels = labels_raw
55 labels = pd.read_csv(labels, encoding='utf-8', skiprows=1)
56
57 # PSEUDO
58 # Function for turning stutter labels into a column of 1s and 0s
59 # Take R1-C0 in fNIRS_EEG
60 # If the number exists in labels, then print "1" in R1-C0 in stutterList.
61 # If not print "0" in R1-C1 in stutterList and go to R2-C0 in fNIRS_EEG.
62 # Then take R2-C1 in fNIRS_EEG
63 # If the number exists in Labels, then print "1" in R2-C0 in stutterList.
64 # If not print "0" in R2-C0 in stutterList and go to R3-C0 in fNIRS_EEG.
65 |
66 # Create stuttersList variable
67 labelsC1 = list(labels.iloc[:,0]) # Column 1 - containing stutter start times.
68 # ^ Must be list so it can be read by addEvents function.
69 #labelsC2 = labels.iloc[:,1] # Column 2 - containing stutter end times.
70
71 # Create a "time column" variable "fNIRS_EEG_CTime variable" (i.e., index the
72 #fNIRS_EEG_CTime = fNIRS_EEG.iloc[:,0] # Column 1 (time column) of fNIRS_EEG doc
73
74 # Create a column of 1s and 0s, correspondong to the fNIRS/EEG time points,
75 # based on the labels doc
76 def addEvents(labelsC1, stuttersList01, fNIRS_EEG_CTime):
77     stuttersList01
78     for i in fNIRS_EEG_CTime:
79         if i in labelsC1:
80             stuttersList01.write("\n" + "1")
81         else:
82             stuttersList01.write("\n" + "0")
83         print(i)
84     stuttersList01.close()
85
86 # Execute function "addEvent" using variables created above
87 addEvents(labelsC1, stuttersList01, fNIRS_EEG_CTime)
88
89 # The document stutterList01 should now be a column of 1s and 0s, with
90 # the same number of rows as your fNIRS_EEG data file (15k or so). This
91 # can now be pasted into your fNIRS_EEG data file as labels for supervised
92 # learning . NOTE - A line may have been skipped at the top of the
93 # stuttersLIst01 doc - delete this to ensure numbers are aligned.
94
```

7. Python script for machine learning

```
1#!/usr/bin/env python3
2# -*- coding: utf-8 -*-
3"""
4Created on Thu Aug  8 17:47:38 2019
5
6@author: lucyrothwell
7"""
8
9#!/usr/bin/env python3
10# -*- coding: utf-8 -*-
11"""
12Created on Thu Jun 27 15:39:19 2019
13
14@author: lucyrothwell
15"""
16
17# CONTENTS
18# Data visualisation / missing values / downsampling
19# K Nearest Neighbour (KNN)
20# Support Vector Machine (SVM)
21# Randomforest
22# Logistic Regression
23# Linear Regression
24
25
26#####Import all the necessary libraries
27#####
28import pandas as pd #Pandas for managing data
29import matplotlib.pyplot as plt #Matplotlib for plotting
30import matplotlib
31
32# K-fold cross validation
33import sklearn
34from sklearn.model_selection import cross_val_score
35#from sklearn.model_selection import KFold
36# from sklearn.model_selection import train_test_split
```

Classifying Stutter Events using fNIRS and EEG data

```
36 #from sklearn.model_selection import LeaveOneOut
37 from sklearn import datasets
38
39 print('The scikit-learn version is {}.'.format(sklearn.__version__))
40
41 #All the machine learning algorithms from the sklean libraries.
42 #Notice they are under different 'sublibraries' - why?
43 from sklearn.linear_model import LogisticRegression
44 from sklearn.linear_model import LinearRegression
45 from sklearn.ensemble import RandomForestClassifier
46 from sklearn.neighbors import KNeighborsClassifier
47 from sklearn.svm import SVC
48 # from sklearn.feature_extraction_text import CounterVectorizer # (YouTube: https://www.youtube.com/watch?v=JyDgXWzIwU)
49 # from sklearn.cross_validation import train_test_split # (YouTube: https://www.youtube.com/watch?v=JyDgXWzIwU)
50 # from sklearn.naive_bayes import MultinomialNBnp.array([0])np.array([0])
51
52 #The regression machine learning algorithms from the sklean libraries.
53 from sklearn.ensemble import RandomForestRegressor
54
55 # Import tools needed for visualization
56 from sklearn.tree import export_graphviz
57 import pydot
58
59 #scoring metrics from sklearn
60 from sklearn.metrics import accuracy_score, mean_squared_error, r2_score, roc_curve
61 from sklearn.model_selection import cross_val_predict
62 # > good explanation: https://github.com/justmarkham/scikit-learn-videos/blob/master/08_cross-validation.ipynb
63
64 # Use numpy to convert to arrays
65 import numpy as np
66
67 # For imputation of missing values
68 from sklearn.impute import SimpleImputer
69 from fancyimpute import KNN
70
```

Classifying Stutter Events using fNIRS and EEG data

```
71
72
73 ##### Read data
74 #####
75
76 # ---- DATA ----
77
78
79 #data_columns = list(dataFrame)
80 #imp.fit(dataFrame)
81 #SimpleImputer(copy=True, fill_value=None, strategy='mean', verbose=0)
82 ### Data
83 DataFrame = "/Users/lucyrothwell/Google Drive copy/MSc Psych - UCL/9. Dissertation"
84 DataFrame = pd.read_csv(DataFrame, encoding='utf-8', skiprows=0)
85 resultFileName = "Results - DELETE.csv"
86
87 # fNIRS or EEG ICA
88 colNum = DataFrame.shape[1]
89
90 ## EEG CHANNEL
91 #colNum = 31
92
93 # ONCE THE ABOVE INFO HAS BEEN ENTERED, THE PROGRAMME CAN RUN AS A
94 # A WHOLE, AND WILL OUTPUT THE RESULTS INTO A CSV
95
96 DataFrame_types = DataFrame.dtypes # variable types
97 DataFrame.shape
98
99 # Checking value counts (first)
100 valueCountsStut = DataFrame['Stutter'].value_counts()
101 print(valueCountsStut)
102 valueCountOnes = valueCountsStut[0]
103 valueCountZeros = valueCountsStut[1]
104 print(valueCountOnes)
105 print(valueCountZeros)
```

Classifying Stutter Events using fNIRS and EEG data

```
106
107 # Remove missing values
108 dataFrame.dropna(inplace=True)
109 dataFrame.shape
110
111 # Checking value counts (second)
112 valueCountsStut = dataFrame['Stutter'].value_counts()
113 print(valueCountsStut)
114 valueCountOnes = valueCountsStut[0]
115 valueCountZeros = valueCountsStut[1]
116 print(valueCountOnes)
117 print(valueCountZeros)
118
119
120 ## RESAMPLE to balance classes (making sure there are 50% stutters and 50%
121 ## non-stutters)
122
123 from sklearn.utils import resample
124
125 #Separate majority and minority classes
126 dataFrame_majority = dataFrame[dataFrame.Stutter==0]
127 dataFrame_minority = dataFrame[dataFrame.Stutter==1]
128
129 #DOWNSAMPLE majority class
130 dataFrame_majority_downsampled = resample(dataFrame_majority,
131                                         replace=False,      # sample with replacement
132                                         n_samples=valueCountZeros, # to match N of majori
133                                         random_state=123) # reproducible results
134
135 # Combine minority class with downsampled majority class
136 dataFrame_downsampled = pd.concat([dataFrame_majority_downsampled, dataFrame_minor
137
138 # Display new class counts
139 dataFrame_downsampled.Stutter.value_counts()
140
```

Classifying Stutter Events using fNIRS and EEG data

```
141 # Shuffle rows so that the 1s and 0s are in random order (previously the
142 # DataFrame showed all of class one then all of class two which distorts the
143 # test_training set split we do later)
144 DataFrame = DataFrame_downsampled.sample(frac=1)
145
146 # Testing the number of 1s and 0s have balanced
147 valueCountsStut = DataFrame['Stutter'].value_counts()
148 valueCountOnes = valueCountsStut[0]
149 valueCountZeros = valueCountsStut[1]
150 print(valueCountOnes)
151 print(valueCountZeros)
152
153
154
155 #####Make sure we have no missing data
156 percent_missing = DataFrame.isnull().sum()*100/len(DataFrame)
157 print(percent_missing)
158
159
160 ##### TRAIN
161 #We are going to apply the below four algorithms to this problem and evaluate its
162 #And finally choose the best algorithm and train it.
163 #Algorithms:
164     #K – Nearest Neighbour (KNN)
165     #Support Vector Machine (SVM)
166     #Randomforest
167     #Logistic Regression
168     #Linear Regression
169 #####
170
171 #Separating the data into training & testing set (holdout and cross-val).
172
173 # 1) Creating variables for "Holdout method" split
174 first75 = round(len(DataFrame)*0.75)
175 training_set = DataFrame[0:first75]
```

Classifying Stutter Events using fNIRS and EEG data

```
176 test_set = dataFrame[first75:len(dataFrame)]
177
178 x_train = training_set.iloc[:,2:colNum]
179 x_test = test_set.iloc[:,2:colNum]
180
181 y_train = training_set[['Stutter']]
182 y_test = test_set[['Stutter']]
183
184
185 # 2) Creating variables for cross validation
186 X = dataFrame.iloc[:,2:colNum]
187 y = dataFrame[['Stutter']]
188
189
190 # List features
191 feature_list = list(x_train.columns)
192 feature_list
193
194 print('Training Features Shape:', x_train.shape)
195 print('Training Labels Shape:', y_train.shape)
196 print('Testing Features Shape:', x_test.shape)
197 print('Testing Labels Shape:', y_test.shape)
198
199
200 #K-NEAREST NEIGHBOUR (KNN)
201 knnmodel = KNeighborsClassifier()
202 knnmodel.fit(x_train,np.ravel(y_train,order='C'))
203 knnpredictions = knnmodel.predict(x_test) # Why x_train not used here?
204 resultKNN = accuracy_score(y_test, knnpredictions)
205 print("K-nearest Neighbours accuracy: ", resultKNN)
206
207 #KNN with CV
208 knn_cv = KNeighborsClassifier()
209 #train model with cv of 5
210 cv_scores = cross_val_score(knn_cv, X, y, cv=10)
```

Classifying Stutter Events using fNIRS and EEG data

```
211 #print each cv score (accuracy) and average them
212 print(cv_scores)
213 KNN_cv = 'KNN CV (mean):{}'.format(np.mean(cv_scores))
214 print(KNN_cv)
215
216
217 # Sensitivity and specificity - HOLDOUT
218 confMatrixKNN = confusion_matrix(y_test, knnpredictions)
219 print(confMatrixKNN)
220
221 TP = confMatrixKNN[1, 1]
222 TN = confMatrixKNN[0, 0]
223 FP = confMatrixKNN[0, 1]
224 FN = confMatrixKNN[1, 0]
225
226 sensitivityKNN = TP / float(TP + FN)
227 print(sensitivityKNN)
228
229 specificityKNN = TN / float(TN + FP)
230 print(specificityKNN)
231
232 classReportKNN = classification_report(y_test, knnpredictions)
233 print(classReportKNN)
234 # > precision = what % of the observed cases were correctly predicted as negative
235 # > recall = what % of the observed cases were correctly predicted as positive (because)
236 # > f1-score = combination of precision and recall ("harmonic mean")
237
238 # Create ROC curve of the KNN model - HOLDOUT
239 # >>> What does the graph tell us? To read: https://www.medcalc.org/manual/roc-curve.html
240 false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, knnpredictions)
241 roc_auc = auc(false_positive_rate, true_positive_rate)
242 plt.title('Receiver Operating Characteristic - KNN Model')
243 plt.plot(false_positive_rate, true_positive_rate, 'b', label='AUC = %0.2f' % roc_auc)
244 plt.legend(loc='lower right')
245 plt.plot([0,1],[0,1], 'r--')
```

Classifying Stutter Events using fNIRS and EEG data

```
246 plt.xlim([-0.1,1.2])
247 plt.ylim([-0.1,1.2])
248 plt.ylabel('True Positive Rate')
249 plt.xlabel('False Positive Rate')
250 plt.show()
251
252 AUC_KNN = label='%0.2f'% roc_auc
253 print(AUC_KNN)
254
255
256
257 #SUPPORT VECTOR MACHINE (SVM)
258 svcmodel = SVC(kernel='rbf', gamma = "scale") # SVC is the function in sklearn
259 #svcmodel.fit(x_train,y_train)
260 svcmodel.fit(x_train,np.ravel(y_train,order='C'))
261 svcpredictions = svcmodel.predict(x_test)
262 resultSVM = accuracy_score(y_test, svcpredictions)
263 print('SVM accuracy: ', resultSVM)
264
265 #SVM with CV
266 svc_cv = SVC(kernel='rbf', gamma = "scale")
267 #train model with cv of 5
268 # cv_scores = cross_val_score(svc_cv, X, y, cv=5) # old
269 cv_scores = cross_val_score(svc_cv, X, np.ravel(y,order='C'), cv=5)
270 #print each cv score (accuracy) and average them
271 print(cv_scores)
272 SVM_cv = ('SVM CV (mean):{}'.format(np.mean(cv_scores)))
273 print(SVM_cv)
274
275 # Sensitivity & specificity
276 confMatrixSVM = confusion_matrix(y_test, svcpredictions)
277 print(confMatrixSVM)
278
279 TP = confMatrixSVM[1, 1]
280 TN = confMatrixSVM[0, 0]
281 FP = confMatrixSVM[0, 1]
```

Classifying Stutter Events using fNIRS and EEG data

```
280 TN = confMatrixSVM[0, 0]
281 FP = confMatrixSVM[0, 1]
282 FN = confMatrixSVM[1, 0]
283
284 sensitivitySVM = TP / float(TP + FN)
285 print(sensitivitySVM)
286
287 specificitySVM = TN / float(TN + FP)
288 print(specificitySVM)
289
290 classReportSVM = classification_report(y_test, svcpredictions)
291 print(classReportSVM)
292
293 # Create ROC curve of the SVM model
294 false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, svcpredictions)
295 roc_auc = auc(false_positive_rate, true_positive_rate)
296 plt.title('Receiver Operating Characteristic - SVM Model')
297 plt.plot(false_positive_rate, true_positive_rate, 'b',
298 label='AUC = %0.2f' % roc_auc)
299 plt.legend(loc='lower right')
300 plt.plot([0,1],[0,1], 'r--')
301 plt.xlim([-0.1,1.2])
302 plt.ylim([-0.1,1.2])
303 plt.ylabel('True Positive Rate')
304 plt.xlabel('False Positive Rate')
305 plt.show()
306
307 AUC_SVM = label='%.2f' % roc_auc
308 print(AUC_SVM)
309
310
311
312
313 #LOGISTIC REGRESSION
314 logreg = LogisticRegression(C=1,solver = 'lbfgs')
```

Classifying Stutter Events using fNIRS and EEG data

```
315 logreg.fit(x_train,np.ravel(y_train,order='C'))
316 logpredictions = logreg.predict(x_test)
317 resultLogreg = accuracy_score(y_test, logpredictions)
318 print("Log Reg accuracy: ", resultLogreg)
319
320 #LOG REG with CV
321 logreg_cv = LogisticRegression(C=1, solver = 'lbfgs')
322 #train model with cv of 5
323 cv_scores = cross_val_score(logreg_cv, X, y, cv=10)
324 #print each cv score (accuracy) and average them
325 logReg_cv = ('logReg CV (mean):{}'.format(np.mean(cv_scores)))
326
327 # Sensitivity & specificity
328 confMatrix_logreg = confusion_matrix(y_test, logpredictions)
329 print(confMatrix_logreg)
330
331 TP = confMatrix_logreg[1, 1]
332 TN = confMatrix_logreg[0, 0]
333 FP = confMatrix_logreg[0, 1]
334 FN = confMatrix_logreg[1, 0]
335
336 sensitivity_logreg = TP / float(TP + FN)
337 print(sensitivity_logreg)
338
339 specificity_logreg = TN / float(TN + FP)
340 print(specificity_logreg)
341
342 classReportLR = classification_report(y_test, logpredictions)
343 print(classReportLR)
344
345 # Create ROC curve of the Log Reg model
346 false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, logpredict
347 roc_auc = auc(false_positive_rate, true_positive_rate)
348 plt.title('Receiver Operating Characteristic - Log Reg Model')
349 plt.plot(false_positive_rate, true_positive_rate, 'b',
```

Classifying Stutter Events using fNIRS and EEG data

```
350 label='AUC = %0.2f'% roc_auc)
351 plt.legend(loc='lower right')
352 plt.plot([0,1],[0,1],'r--')
353 plt.xlim([-0.1,1.2])
354 plt.ylim([-0.1,1.2])
355 plt.ylabel('True Positive Rate')
356 plt.xlabel('False Positive Rate')
357 plt.show()
358
359 AUC_logreg = label='%0.2f'% roc_auc
360 print(AUC_logreg)
361
362
363
364
365 #RANDOM FOREST
366 rfmodel = RandomForestClassifier(n_estimators=24) #rfmodel is the function in sklearn
367 rfmodel.fit(x_train,y_train)
368 rfpredictions = rfmodel.predict(x_test)
369 resultRfmodel = accuracy_score(y_test, rfpredictions)
370 print('Random Forest accuracy: ', resultRfmodel)
371
372
373 #RAND FOR with CV
374 rfmodel_cv = RandomForestClassifier()
375 #train model with cv of 5
376 cv_scores = cross_val_score(rfmodel_cv, X, y, cv=10)
377 #print each cv score (accuracy) and average them
378 print(cv_scores)
379 rfmodel_cv = ('RF CV (mean):{}'.format(np.mean(cv_scores)))
380 print(rfmodel_cv)
381
382 # Sensitivity & specificity
383 confMatrix_rfmodel = confusion_matrix(y_test, rfpredictions)
384 print(confMatrix_rfmodel)
```

Classifying Stutter Events using fNIRS and EEG data

```
385
386 TP = confMatrix_rfmodel[1, 1]
387 TN = confMatrix_rfmodel[0, 0]
388 FP = confMatrix_rfmodel[0, 1]
389 FN = confMatrix_rfmodel[1, 0]
390
391 sensitivity_rfmodel = TP / float(TP + FN)
392 print(sensitivity_rfmodel)
393
394 specificity_rfmodel = TN / float(TN + FP)
395 print(specificity_rfmodel)
396
397 classReportRF = classification_report(y_test, rfpredictions)
398 print(classReportRF)
399
400 # Create ROC curve of the Random Forest model
401 false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, rfpredictions)
402 roc_auc = auc(false_positive_rate, true_positive_rate)
403 plt.title('Receiver Operating Characteristic - SVM Model')
404 plt.plot(false_positive_rate, true_positive_rate, 'b',
405 label='AUC = %0.2f' % roc_auc)
406 plt.legend(loc='lower right')
407 plt.plot([0,1],[0,1], 'r--')
408 plt.xlim([-0.1,1.2])
409 plt.ylim([-0.1,1.2])
410 plt.ylabel('True Positive Rate')
411 plt.xlabel('False Positive Rate')
412 plt.show()
413
414 AUC_rfmodel = label='%.2f' % roc_auc
415 print(AUC_rfmodel)
416
417
418
419 # IMPORTANCES (IN RF MODEL)
```

Classifying Stutter Events using fNIRS and EEG data

```
420 # Get numerical feature importances
421 importances = list(rfmodel.feature_importances_)
422 # List of tuples with variable and importance
423 feature_importances = [(feature, round(importance, 2)) for feature, importance in
424 # Sort the feature importances by most important first
425 feature_importances = sorted(feature_importances, key = lambda x: x[1], reverse = True)
426 |
427
428 # WRITING RESULTS
429 results = open("/Users/lucyrothwell/Google Drive copy/MSc Psych - UCL/9. Dissertation/Classification Results/Classification Results.txt", "w")
430
431 results.write("K-nearest Neighbours accuracy: " + str(resultKNN))
432 results.write("\n" + str(KNN_cv))
433 results.write("\n" + "Sensitivity KNN: " + str(sensitivityKNN))
434 results.write("\n" + "Specificity KNN: " + str(specificityKNN))
435 results.write("\n" + "AUC KNN: " + str(AUC_KNN)+ "\n" + str(confMatrixKNN)+ "\n" +
436
437 results.write("\n" + "\n" + "SVM accuracy:" + str(resultSVM))
438 results.write("\n" + str(SVM_cv))
439 results.write("\n" + "Sensitivity SVM: " + str(sensitivitySVM))
440 results.write("\n" + "Specificity SVM: " + str(specificitySVM))
441 results.write("\n" + "AUC SVM: " + str(AUC_SVM)+ "\n" + str(confMatrixSVM)+ "\n" +
442
443 results.write("\n" + "\n" + ("Log Reg accuracy: " + str(resultLogreg)))
444 results.write("\n" + str(logReg_cv))
445 results.write("\n" + "Sensitivity LogReg: " + str(sensitivity_logreg))
446 results.write("\n" + "Specificity LogReg: " + str(specificity_logreg))
447 results.write("\n" + "AUC LogReg: " + str(AUC_logreg) + "\n" + str(confMatrix_logreg))
448
449 results.write("\n" + "\n" + ("Random Forest accuracy: " + str(resultRfmodel)))
450 results.write("\n" + str(rfmodel_cv))
451 results.write("\n" + "Sensitivity RF: " + str(sensitivity_rfmodel))
452 results.write("\n" + "Specificity RF: " + str(specificity_rfmodel))
453 results.write("\n" + "AUC RF: " + str(AUC_rfmodel) + "\n" + str(confMatrix_rfmodel))
454
455 #RFimportances_write = [((Variable: {:20} Importance: {})'.format(*pair)) + '\n' for pair in feature_importances]
456 #results.write("\n" + "\n" + str(RFimportances_write))
457 [print('Variable: {:20} Importance: {}'.format(*pair)) for pair in feature_importances]
458 # COPY& PASTE THIS FOR NOW
459
460 results.close()
461
```

8. Variable importance measures (VIMs)

VIMs from balanced data

EEG: Channel (x4)		EEG: ICA (006_Social)		EEG: Chanel (006_Social)		fNIRS (x4)	
Feature	VIM	Feature	VIM	Feature	VIM	Feature	VIM
AF7	0.09	IC2	0.11	C3, C4, P7, Cz, P4, O1, FC5, FC6	0.05	S1-D2 (Deoxy)	0.12
Fp1, F7	0.08	IC1, IC7	0.09	Fp1, Fp2, F7, F3, F4, F8, T7, Cz, T8, P3, P8, O2, FC2, AF7	0.04	S2-D3 (HbT)	0.11
Fp2, C3, C4	0.06	IC3, IC6, IC9, IC10	0.08	Fz, FC1	0.03	S1-D2 (HbT)	0.1
F8, P4	0.05	IC5, IC8, IC11, IC12	0.07			S1-D3 (Oxy)	0.08
Fz, P3, FC6	0.04	IC4	0.06			S3-D7 (Dexy), S3-D7 (Hbt)	0.07

Classifying Stutter Events using fNIRS and EEG data

F3, F4, T7, Cz, T8, P7, Pz, P8, O1, O2, FC5, FC1	0.03	IC13	0.05			S3-D5 (Hbt)	0.06
FC2	0.02					S1-D3 (Deoxy) S2-D3 (Deoxy)	0.05
						S1-D2 (Oxy), S3-D5 (Deoxy)	0.04
						S1-D1 (Deoxy), S1- D3 (HbT), S3- D6 (HbT)	0.03
						S1-D1 (HbT), S2-D4 (HbT), S3-D5 (Oxy), S3-D6 (Deoxy), S3- D7 (Oxy)	0.02
						S1-D1 (Oxy), S2-D4 (Oxy), S2-D4 (Deoxy), S3- D6 (Oxy),	0.01

Classifying Stutter Events using fNIRS and EEG data

						S2-d3 (Oxy)	
--	--	--	--	--	--	-------------	--