Developing a Two-state EEG-based Brain Computer Interface Using Curve Fitting and Time-Series Models as Feature Extraction Methods

1st Joseph Gill
Department of Computer Science
University of Exeter
Exeter, United Kingdom
jwdg201@exeter.ac.uk

2nd Dr Saptarshi Das
Department of Mathematics
University of Exeter
Exeter, United Kingdom
S.Das3@exeter.ac.uk

3rd Lamia Alyami
Department of Mathematics
University of Exeter
Exeter, United Kingdom
la424@exeter.ac.uk

***Abstract*—A brain computer interface (BCI) is a communication system that enables humans to interact with computers using cognitive actions. In this study, we have used the coefficients of the polynomial, Fourier and sum of sine curve fitting models, and the vector autoregression (VAR) timeseries model as the features of machine learning algorithms to classify left-and-right hand ERP trials from one of the largest EEG-based BCI datasets publicly available. The use of the sum of sine curve fitting model as a feature extraction method is a first in BCI literature. We found the polynomial curve fitting model to be the best feature extraction method, achieving an average accuracy score of 0.85. The Fourier, sum of sine and VAR models performed poorly, achieving average accuracy scores of 0.52, 0.56 and 0.58 respectively.**

***Index Terms*—BCI, EEG, ERP, feature extraction, classification**

I. Introduction

Brain-computer interfaces (BCIs) measure a subject's brain signals that relate to their intent to affect control and translate them into signals that are used to control external devices [1]. As BCIs make computer control possible without any physical activity, they can provide those with neuromuscular disorders, such as amyotrophic lateral sclerosis (ALS), brain or spinal cord injury and cerebral palsy, control of devices such as speech synthesizers and artificial limbs [2]. The effective use of BCIs will increase their independence, leading to an improved quality of life and reduced social costs.

The number of publications on BCIs has steadily increased, from 1300 in the decade 2001-2010 to 6630 in 2010-2020 [3]. This highlights the growing interest of the scientific community in BCI technology. The number of research applications of BCIs has expanded significantly over these years, with BCIs now being designed to detect drowsiness to improve human working performance [4], control quadcopters [5] and play video games [6]. Despite recent innovations, BCIs are not entirely ready to enter retail markets, mainly due to their limited capability in real world settings [7].

For a BCI to be operational, it must first undergo a training phase in which the system is calibrated, as summarised in Fig. 1. When a BCI is operational, it can recognize patterns in brain activity and translate them into commands for a computer. In the training phase of a BCI, the brain signals from a subject being stimulated to perform a motor imagery action are acquired. The brain signals then undergo signal preprocessing to remove noise. Features from the signals are extracted in order to represent the signals in a lower dimensional form [8]. These features form the descriptive variables and the motor imagery action stimuli form the target variable in the training of supervised machine learning algorithms.

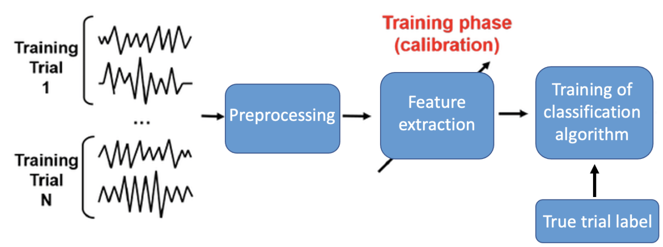

Fig. 1 The training phase of a BCI.

Once a BCI is trained, features extracted from the preprocessed brain signals are classified and translated into a command signal for a specific application. The operational phase of a BCI is summarized in Fig. 2.

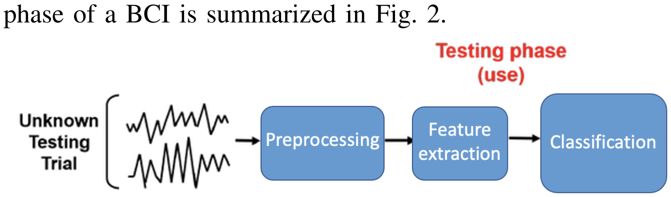

Fig. 2 The testing/operational phase of a BCI.

The modality of signal acquisition has been used to divide BCIs into invasive and non-invasive systems. In invasive BCIs, devices are implanted directly into the brain during neurosurgery. Invasive BCIs have shown to provide a signal-to-noise ratio and spatial localization of brain activity that is superior to non-invasive methods [3]. In non-invasive BCIs, brain signals are acquired through sensors that are external to the body, such as functional near infrared spectroscopy (fNIRS) [9] and functional magnetic resonance imaging (fMRI) [10]. The most popular method being electroencephalogram (EEG). Electrical impulses produced by electrons in the brain create an electric field around the head, the spatio-temporal voltage of which varies according to levels of neural activity inside areas of the brain. In an EEG scan, electrodes are placed on the scalp to measure the voltages within the electrical field, thus indirectly measuring levels of brain activity. EEG is the most studied signal acquisition method due to its fine temporal resolution, easy implementation, and low set-up costs [1].

Motor imagery Event Related Potential (ERP) EEG data is the EEG response as the direct result of a subject performing a mental rehearsal of a motor act, such as the clenching of a hand or the kicking of a leg. EEG-based BCIs are trained to identify and act upon the ERP response caused by a subject's specific motor imagery actions. EEG signals are formed of brain activities overlapping in space and time, contaminated with noise signals generated from electrical activity in the body and dependent on factors such as physical state, mood and posture. Such properties make ERP EEG data inherently noisy and difficult to classify. EEG data is also highly subject-specific, therefore a BCI system must be trained specifically for each subject and their BCI application. Therefore, researchers are required to take hundreds of ERP trials per motor imagery action, per subject, to accurately train a BCI. This makes the development of a BCI a time-consuming and expensive process. Current studies seek to improve the speed and accuracy of the signal preprocessing, feature extraction and classification methods to reduce training times and develop faster and more accurate BCIs.

II. Research Context

The design of ERP-EEG based BCIs has been extensively studied over the last 30 years. Papers reviewing the development of BCIs show that impressive results are being achieved on specific BCI problems [2]. An EEG-based BCI is formed of five main processes: the EEG signals are acquired via electrodes placed on the scalp, the signals are preprocessed, relevant features are extracted, the features are then classified using a machine learning algorithm to identify the motor imagery action and this information is transferred to a communication and control system. Researchers are developing more advanced signal preprocessing, feature extraction and classification algorithms to increase the capability of BCI systems.

The preprocessing of EEG data eliminates the unwanted noise and artifacts from the EEG signals. Bashashati et al. identified 17 signal preprocessing techniques used in BCI research [11], highlighting the range of techniques currently being applied. Filtering algorithms such as notch [12] and bandpass [13] filters have been used to remove certain frequencies from EEG signals. The least mean squares (LMS) [14] and Kalman filter (KF) [15] are adaptive filtering techniques, in which the coefficients of self-learning digital filters are adjusted to minimize the difference between a reference signal and the output of the adaptive filter [16].

Feature extraction methods reduce the dimensionality of ERP data by only capturing specific aspects of the data that are useful in classification. There are many ways in which the features of an ERP signal can be extracted, the two most common ways are to use band power or time point features. Band power features represent the power (energy) of EEG signals for a given frequency band in each channel, and are useful in passive BCIs, such as those that decode mental states or emotions. Time point features are the typical features used to classify ERP data. They consider the temporal variations in the amplitude of the EEG signals related to a given stimulus, such as common spatial patterns [17] and wavelet packet decomposition [18]. These are the features used in most P300-based BCIs [19].

In a 2007 review of classification algorithms for EEG-based BCIs, five main families of classifiers were identified: linear classifiers, neural networks, non-linear Bayesian classifiers, nearest neighbour classifiers and classifier combinations [8]. Algorithms in these families, such as k-nearest neighbors [20], Support Vector Machines [21] and linear discriminant analysis [22] have been tested for BCI applications and achieved satisfactory results. In a more recent review of classification algorithms for EEG-based BCIs, three new families of classifiers were identified: adaptive classifiers, transfer learning and deep learning. Among these, adaptive classifiers and transfer learning achieved particularly high accuracies. On the other hand, deep learning methods have not yet achieved convincing results [19].

The processing of EEG data has significantly improved and BCIs have been demonstrated to be effective in specific application domains. Unfortunately, advances in the analysis of EEG data have been complicated by a lack of large and uniform datasets that can be used to design and evaluate different approaches. This issue has been addressed over recent years with the initiation of BCI competitions in 2001 [23] and the production of gold standard ERP EEG datasets in experimental research papers [24]. The competitions provide new challengers to BCI researchers, with the hope that winning entries may enhance the processing methods of future BCIs. Three BCI competitions were arranged in 2001, 2002, and 2004, with the number of submissions rising from 10 to 57 to 92, highlighting the growing research interest in BCIs.

III. Aims & Objectives

A. Aims

The aim of our research is to review the effectiveness of using the coefficients of the polynomial, Fourier and sum of sin curve fitting models, and the vector autoregression (VAR) time series model as the features of machine learning algorithms to classify ERP EEG data. The utility of the sum of sin curve fitting method has not been widely reported. Therefore, this paper will analyse the performance of the sum of sin method in context with other, more popular, feature extraction methods.

B. Objectives

To achieve this aim, we set the following objectives:

1) Import the EEG ERP datasets [24] into MATLAB and perform preliminary data analysis.
2) Apply the Butterworth filter to reduce the frequencies outside of the 1-45Hz band in the EEG signals.
3) Process the EEG datasets into ERP trials.
4) Fit the polynomial, Fourier and sum of sin curve fitting models and the VAR time series model to the ERP trials and extract the model parameters. The models should be fitted with varying levels of complexity.
5) For each participant and feature extraction method, create a dataset consisting of the parameters of the ERP trials and the target variable of the trials, i.e. if it is a right-or-left hand trial.
6) Use the datasets to train the logistic regression, decision tree, ensemble learning, support vector machine and K-nearest neighbours supervised machine learning algorithms. The hyperparameters of the machine learning algorithms should be optimised to achieve high accuracy scores.

IV. Methodology

For our study, we used ERP EEG datasets from an experimental research paper published in Nature in 2018, titled “A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces” [24]. Our data analysis was conducted in MATLAB version 9.10, with the use of the following Toolboxes:

- Curve Fitting Toolbox, version 3.5.13
- Econometrics Toolbox, version 5.6
- Signal Processing Toolbox, version 8.6
- Statistics and Machine Learning Toolbox, version 12.1

Our experimental design consisted of three main stages: data preprocessing, feature extraction and classification. In data preprocessing, we applied the Butterworth filter to remove unwanted frequencies in the EEG signals. In feature extraction, we fit the polynomial, Fourier and sum of sin curve fitting models, and the vector autoregression (VAR) model to the ERP trials and extracted the model parameters. In classification, we trained and tested the ability of the logistic regression, decision tree, ensemble learning, support vector machine and K-nearest neighbor machine learning algorithms to classify the features from the ERP trials. Fig. 3 shows a summary of the experimental design.

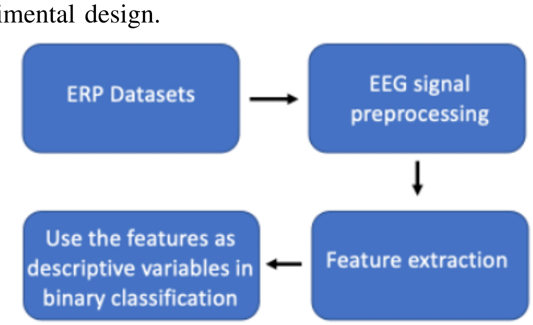

Fig. 3 A flow chart summarising the methodology of our research.

The EEG ERP data [24] we used in our study is one of the largest EEG-based BCI datasets publicly available to date, containing 60 hours of EEG recordings from 14 participants operating in 4 interaction paradigms. In our research, we consider the interaction paradigm in which the participants were stimulated to perform the motor-imagery action of clenching their left-or-right hand, while their brain activity was being recorded by an EEG. The EEG data acquisition was performed using a modified 10/20 EEG cap with 19 bridge electrodes in the 10/20 international configuration, as shown in Fig 4, at a sampling frequency of 200 Hz. Two ground leads labelled A1 and A2 and one lead for data synchronization were also recorded, for a total of 22 EEG channels.

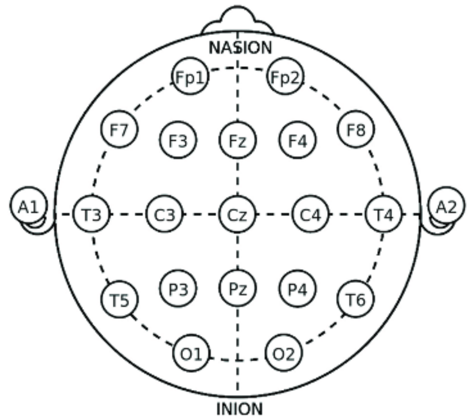

Fig. 4 Electrode locations of International 10-20 system for recordings.

The recording session of each participant lasted between 50 and 55 minutes and the EEG signals, along with the timings of the motor imagery stimuli were recorded continuously throughout. Each trial began with the presentation of a motor imagery stimulus to the participant, which remained on screen for 1s. During this time, the participant implemented the indicated mental imagery action once. A pause of the variable duration of 1.5-2.5s followed, concluding the trial. Each participant performed roughly 640 ERP trials of the left-and-right hand interaction paradigm. The EEG recordings, along with the timings of the motor imagery stimuli, are publicly available as 14 MATLAB structured arrays via a Fig share data deposition service [25-38].

A. Preprocessing

We downloaded the MATLAB structured arrays containing the experimental data and saved them locally. The raw EEG data and the timings of the motor imagery stimuli were then extracted from the structured arrays. Using the Signal Processing Toolbox in MATLAB, the Butterworth filter with an order of 4 was applied to the raw EEG data to remove frequencies outside of the range 1-45Hz. As shown in Fig. 5, the Butterworth filter reduces the power spectral density of unwanted frequencies, thus increasing the signal to noise ratio of the signal. After the preprocessing of the EEG data, the EEG datasets were processed to form left-and-right hand ERP trials.

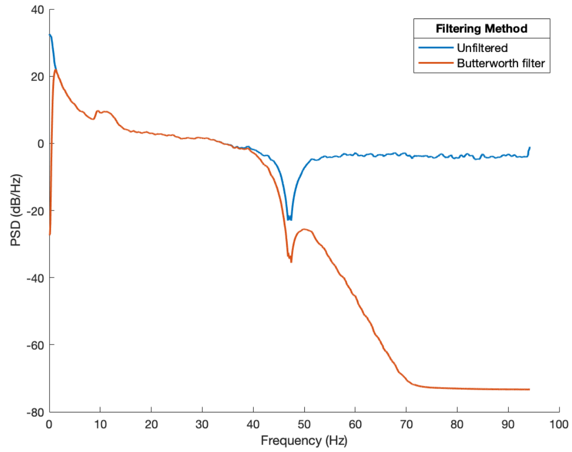

Fig. 5 The effect of the Butterworth filter on a sample of EEG ERP data. The filter was designed to reduce frequencies outside of the range 1-45Hz.

B. Feature extraction

Curve fitting is where the parameters of a mathematical function are tuned to fit a set of data points. Based on a preliminary analysis of the ability of curve fitting methods to accurately fit ERP trials, we explored the use of three curve fitting methods, which were the polynomial, Fourier and sum of sin methods. Time series models capture the relationship between multiple quantities as they change over time. Due to the number of channels in the ERP trials, we fitted the vector autoregressive model, which is a multivariate time-series model. As can be seen in Table 1, the degree of the curve fitting models and the number of lags that were used in the time-series model were systematically varied to explore the effect that the complexity of the feature extraction methods had on BCI performance.

Feature extraction method	Degree/lag	Method of fit
Polynomial	1 to 9	Least squares
Fourier	1 to 6	Least squares
Sum of sine	1 to 9	Least squares
VARM	1 to 2	Least squares

Table 1: Feature extraction methods with degree/lag range.

The curve fitting methods were fit independently to each of the 22 ERP signals using the Curve Fitting Toolbox. The coefficients of the curve fitting models were optimised using the method of least squares. The method of least squares minimizes the sum of the squared residuals, a residual being the difference between an observed ERP value and the fitted ERP value provided by a model. The coefficients of the 22 fitted curves were then extracted and concatenated.

In total, 25 feature extraction methods were applied to the ERP data of each participant. The parameters of each model were stored in tables and formed the features for the classification algorithms. A target column was appended to each table, consisting of the true classification of each trial, i.e. if the trial was created using a right-or-left hand motor imagery stimulus. This was done to put the features in a suitable format to apply supervised machine learning algorithms.

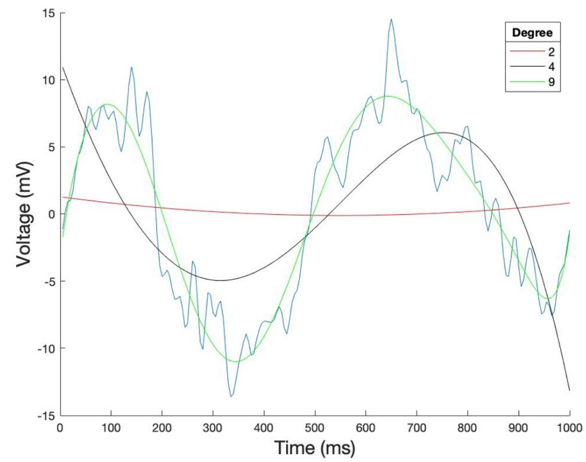

Fig. 6 A plot of EEG ERP data with polynomial curves of degrees 2, 4 and 9 fitted to an ERP trial.

1) Polynomial curve fitting: The equation for fitting a polynomial curve to a timeseries dataset is given by equation (1), where n is the degree of the polynomial being fitted.

$$f(x) = \sum_{i=1}^{n+1} P_i x^{n+1-i} \tag{1}$$

One of the main disadvantages of polynomial fitting is that the fit might be very good within a data range but can diverge outside this data range. However, this should not be a cause for concern since we are using the coefficients of the curve as features for classification. Fig. 6 shows that a higher degree of the polynomial increases the goodness of fit with respect to the data.

2) Fourier curve fitting: A Fourier series is a sum of sine and cosine functions, as described by equation (2). It is usually used in BCIs to extract band power features from an ERP signal, however here we use it as a curve fitting method. In the function, a_0 is the intercept, w is the fundamental frequency of the signal and n is the degree of the model

$$f(x) = a_0 + \sum_{i=1}^{n} a_i \cos(iwx) + b_i cos(iwx) \tag{2}$$

Fig. 7 shows that an increase in the number of terms in the Fourier series increases the models ability to accurately fit to the data.

3) Sum of sin: The use of the sum of sine as a curve fitting method in feature extraction has not been widely studied in BCI literature. As the name suggests, the sum of sine is a linear equation of sine curves, the equation for which is shown in equation (3), where n is the degree of the model.

$$f(x) = \sum_{i=1}^{n} a_i \sin(b_i x + c_i) \tag{3}$$

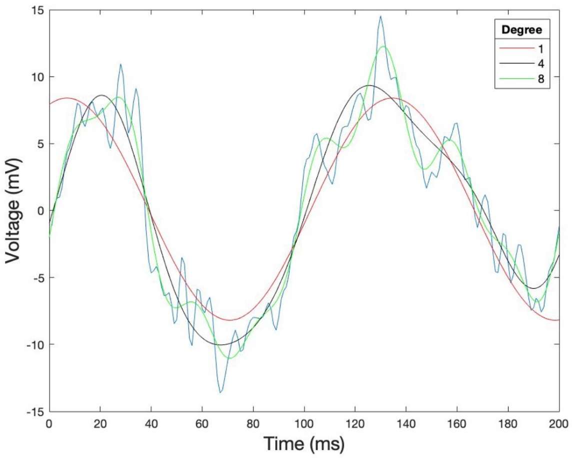

Fig. 7 A plot of EEG ERP data with Fourier curves of degrees 1, 4 and 8 fitted to an ERP trial.

Increasing the degree in the sum of sine curve fitting method improves its accuracy, as shown in Fig. 8.

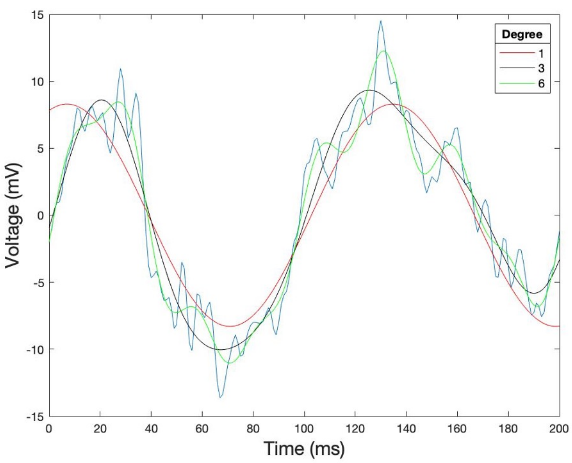

Fig. 8 A plot of EEG ERP data with sum of sine curves of degrees 1, 3 and 6 fitted to an ERP trial.

4) Vector autoregressive model: The vector autoregressive (VAR) model is one of the most successful, flexible and easy to use models for the analysis of multivariate time series. It is a natural extension of the univariate autoregressive model to dynamic multivariate time series. The VAR model has proven to be especially useful for describing the dynamic behaviour of economic and financial time series for forecasting. The equation of a VAR model with one lag is displayed in equation (4), and the equation of a VAR model with two lags is displayed in equation (5).

$$y_t = c + A_1 y_{t-1} \tag{4}$$

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} \tag{5}$$

In the VAR model, the 22 ERP channels that form the variable y are endogenous, such that the variables are allowed to influence each other equally. The number of previous terms in the model is called the number of lags. For each lag i, the term at time $t-i$ is used in the prediction of the current term, y_t. The coefficients, A, of each lag are 22x22 matrices which capture the influence of the lag on the current term. The A matrices are fitted using the principle of least squares. The A matrices of the VAR models were concatenated to form a set of extracted features for classification.

C. Classification

A BCI must be trained to identify specific motor imagery brain signals, this is done through supervised machine learning algorithms. A supervised machine learning algorithm is trained to learn a function that maps input features to their target variable. In supervised machine learning, the dataset is split into training, validation and test sets. The training set is the dataset that is used to train the model, the validation set is used to provide an unbiased evaluation of the model's fit on the training dataset while tuning model hyperparameters, and the test data provides an unbiased evaluation of a final model fit on the training dataset.

To obtain fair and robust results, we reviewed the performance of the algorithms using 10-fold cross validation. In 10-fold cross-validation, the dataset is randomly partitioned into 10 equal size subsets. Of the 10 subsets, a single subset is retained as the test set for testing the model, and the remaining 9 subsets are used as training data. The cross-validation process is repeated 10 times, with each of the 10 subsets used exactly once as the test data. The 10 results are averaged to produce a single estimation. The advantage of this method is that all observations are used for both training and testing, and each observation is used for testing exactly once. To allow for a fair comparison between the classification algorithms, the same test and train sets were used for all the classification algorithms.

The Classification Learner App in the Statistics and Machine Learning Toolbox was used to implement and optimise the hyperparameters of the classification algorithms. Based on preliminary performance testing of the classifiers available in the Classifier Leaner App, we chose 5 classifiers to use in our study. The classifiers being logistic regression, decision tree, ensemble learning, support vector machine and k-nearest neighbors.

1) Logistic regression: In logistic regression, a linear combination of the features is developed and passed through a logit function, shown in equation (6), to predict the probability of the target variable belonging to a certain class. The coefficients of the features are tuned according to the method of least squares.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}} \tag{6}$$

For an unlabelled sample from the test set, the output of logistic regression is the probability of the sample belonging to each of the classes. Therefore, the classification thresholds of linear regression can be easily adjusted.

2) Decision tree: The decision tree is a popular supervised machine learning algorithm due to its speed and easy interpretability. A decision tree creates a model that predicts the target variable by learning simple decision rules inferred from the data features. The decision tree algorithm starts with the training samples and their associated class labels. This training set is recursively partitioned based on features, chosen to divide the data into subsets with the highest level of purity. There are several purity metrics for finding the feature that best divides the data, with no one metric being significantly superior to the others. Decision tree complexity increases with the depth of the tree. In our study, both the decision tree depth and the purity metric were automatically optimised in the Classification Learner App, with the optimal tree depth being 15 and the optimal purity metric being the Gini index.

3) Ensemble learning: Machine learning algorithms output their best solution to the classification problem. In ensemble learning, k machine learning algorithms, called an ensemble, are trained and can vote to predict the label of a new sample. More precisely, an ensemble learning method constructs a set of predictions $p_1, ..., p_k$, chooses a set of weights $w_1, ..., w_k$ and constructs a vote based classifier, $H(x) = w_1 h_1(x) + ... + w_K h_K(x)$. The classification decision of the combined classifier H is $+1$ if $H(x) \geq 0$ and -1 otherwise [39]. In our ensemble learning method, we used boosting to create 100 classification trees. The idea being that the majority vote of the trees will achieve a better predictive performance than could be obtained from any of the trees alone.

4) Support vector machine: A support vector machine (SVM) determines the location of a decision boundary, also known as a hyperplane, that produces the optimal separation of classes. The algorithm is based on the notion of a margin existing either side of a hyperplane that can separate the two classes. The algorithm's goal is to maximise the margin and thereby creating the largest possible distance between the hyperplane and the instances on either side of it.

Preliminary performance testing showed that a linear SVM was the most suitable SVM for our datasets. A linear SVM operates under the assumption that the training data (x_i, y_i) is linearly separable [40], such that a pair (w, b) exists where a weight vector w and a bias b can be found such that:

$$\begin{aligned} w \cdot x_i + b &\geq 1 \quad for \quad y_i = 1 \\ w \cdot x_i + b &< 1 \quad for \quad y_i = -1 \end{aligned} \tag{7}$$

5) K-nearest neighbours: The k-nearest neighbor (KNN) algorithm is amongst the simplest of all machine learning algorithms. It is based on the principal that samples with the same target label lie in close proximity. KNN is a non-parametric, lazy classifier. In the KNN algorithm, the training samples are stored in an n-dimensional space where n is the number of features of the dataset. Given an unlabelled sample from the test set, it's k-nearest neighbors are the k data points in the training set that are closest to the sample based on a distance metric. The sample is then labelled according to the majority class of its k-nearest neighbors. The distance metric and the value of k have a large effect on algorithm performance. By using the optimised Classification Learner App to optimise the model's hyperparameters, an optimised choice of $k = 5$ and the Euclidean distance were used in our study.

V. Results

Distinguishing between left-and-right hand ERP data is a binary classification problem as there are only two possible classes. In our binary classification problem, a right-hand ERP sample is termed positive and a left-hand ERP sample is termed negative. So, when a classification algorithm correctly identifies a right-hand ERP, it is called a True Positive (TP) and when a right-hand ERP is correctly classified, it is called a True Negative (TN). The detection of a Positive, when it is a Negative, is called a False Positive (FP). The detection of a Negative, when it is a Positive, is called a False Negative (FN). This is summarised in a confusion matrix, as displayed in Fig. 1 [41].

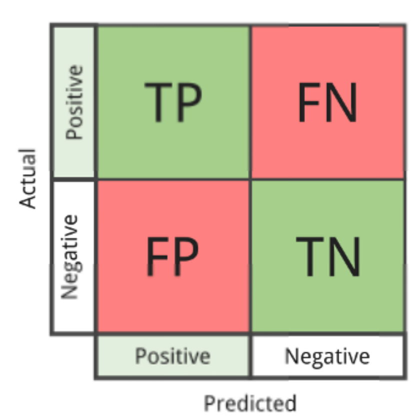

Fig. 9 A confusion matrix

Selecting an appropriate performance metric is essential to understanding the ability of a classification algorithm. We used accuracy as the performance metric in our study because the target classes are well balanced across all participant experiments and the cost of false positives and false negatives is the same. Accuracy is the most intuitive performance metric and is simply the ratio of the correctly predicted observations to the total observations, as shown in equation (8).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

We averaged the performance of the feature extraction and classification algorithms across the 14 participants. This was done to understand how the BCI would performs on the average participant. The full results obtained in our study are displayed in Table 2 in the Appendix.

A. Polynomial

Fig. 10 shows the results of using the parameters of polynomial curve fitting as the features of machine learning algorithms. Each line in the graph represents the results achieved by having a different degree in the polynomial model. The graph shows that overall, good accuracy scores were achieved, with scores ranging from 0.72 to 0.85. The machine learning algorithm used has a rather small effect on the accuracy achieved. However, the degree of the polynomial model used in feature extraction has a large effect, with higher degree polynomials achieving higher accuracy scores. The highest accuracy of 0.85 was achieved by the polynomial model with degree 9 and a SVM classifier. This is the highest accuracy score achieved by any feature extraction method.

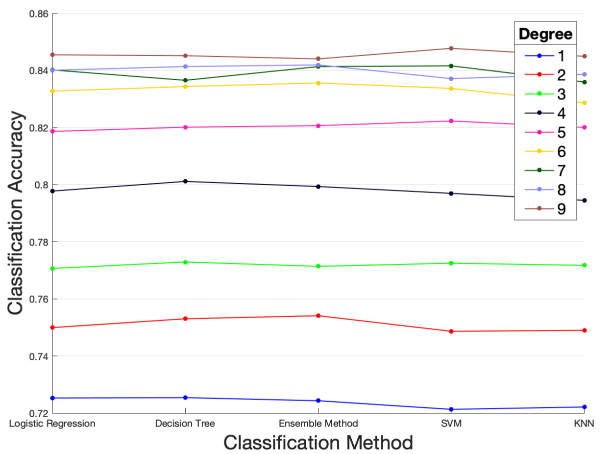

Fig. 10 The classification accuracies of using the parameters of polynomial curve fitting methods with degrees 1 to 9 as the features of machine learning algorithms.

B. Fourier

The results of using the parameters of Fourier curve fitting as the features of machine learning algorithms is shown in Fig. 11. The classification algorithms have a small effect on the accuracy scores. The effect of model degree is relatively small too, with the the Fourier models with 2 to 8 degrees obtaining accuracies ranging between between 0.51 and 0.53. These are the lowest accuracy scores achieved by any feature extraction method in our study. The Fourier with 1 degree performs slightly better and achieves an accuracy score of 0.56.

C. Sum of sine

As shown in Fig. 12, the classification of the sum of sine models achieve similar results to the classification of the Fourier models. The sum of sine model with 1 degree achieves the highest accuracy score of 0.61. The methods with degrees 2 to 6 achieve accuracy scores ranging from 0.53 to 0.56. The sum of sine models achieve the same accuracy scores across the machine learning algorithms.

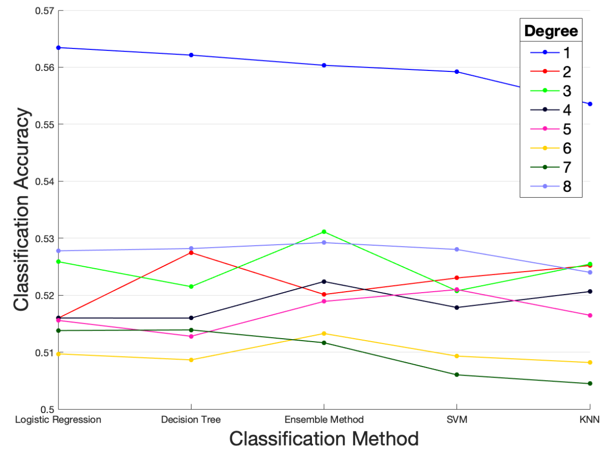

Fig. 11 The classification accuracies of using the parameters of Fourier curve fitting methods with degrees 1 to 8 as the features of machine learning algorithms.

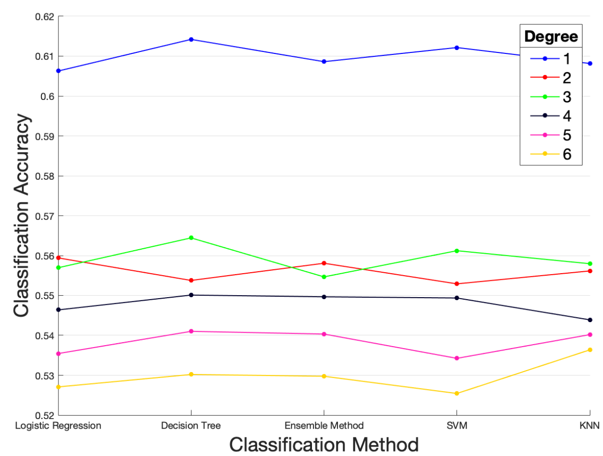

Fig. 12 The classification accuracies of using the parameters of sum of sine curve fitting methods with degrees 1 to 6 as the features of machine learning algorithms.

D. VAR

Fig. 13 shows that the VAR models with 1 lag and 2 lags achieve similar accuracy scores across the machine learning algorithms. The VAR model with 1 lag obtains an accuracy score of around 0.57. The model with 2 lags outperforms this by 0.01, with an accuracy score of around 0.58.

E. Comparison

We selected the degree or lag value that optimises the performance of each feature extraction method. Their classification accuracies are displayed in Fig. 14 to compare their performance. This graph shows that the polynomial model, with a degree of 9, outperforms the other methods with an accuracy score of 0.85. The second best feature extraction method is the sum of sine model with 1 degree (0.61). The

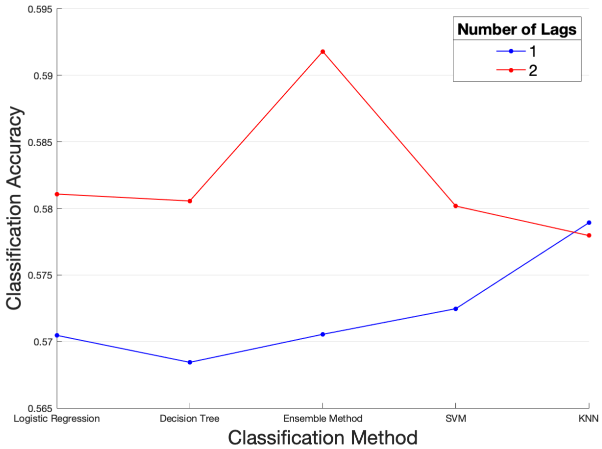

Fig. 13 The classification accuracies of using the parameters of the VAR model with 1 and 2 lags as the features of machine learning algorithms.

feature extraction method that achieves the third highest results is the VAR model with 2 lags (0.58). The feature extraction that performed the worst is the Fourier model with 1 degree (0.53).

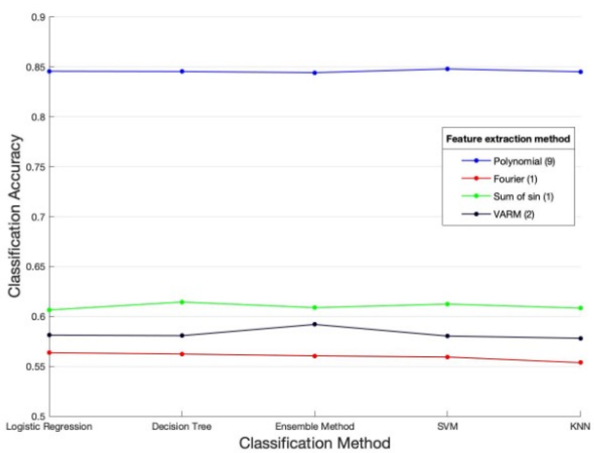

Fig. 14 The classification accuracies obtained by the optimised feature extraction methods.

VI. Discussion of Results

In the last 30 years, there has been extensive research into the preprocessing, feature extraction and classification algorithms that constitute BCIs. The development of such algorithms is vital in providing those with neuromuscular disorders control of computers and robotics that will increase their independence, leading to an improved quality of life and reduced social costs. Unfortunately, advances in BCI technology have been complicated by a lack of large and uniform datasets that can provide an objective analysis of preprocessing, feature extraction and classification algorithms.

In this study, we have applied a range of feature extraction and classification algorithms to one of the largest uniform EEG ERP datasets. By doing so, we have attempted to provide an objective measure of feature extraction and classification algorithm performance. Our results showed that the feature extraction method has a significant impact on classification performance, with the best feature extraction method, the polynomial model, achieving an average accuracy score of 0.85 and the worst feature extraction method, the Fourier model, achieving an average accuracy score of 0.52. In contrast to this, the classification method had little effect on the classification accuracy, with the feature extraction methods obtaining consistent accuracy scores across the classification algorithms.

We have identified one paper [22] that develops a BCI using the same EEG ERP datasets that we used in our study. In the paper, they experimented with several feature extraction methods including Power Spectral Density (PSD), Fourier transform amplitudes (FTA) and raw time series. By classifying the features of each trial using a SVM, they obtained an average accuracy of 0.90 across all 14 participants. Using a polynomial model with degree 9 and a SVM classifier, we obtained an average accuracy score of 0.85. The results are comparable, showing that our findings fit with existing literature. The more advanced feature extraction methods used in the paper could account for the higher level of accuracy that was achieved.

The average accuracy of the Fourier, sum of sine and VAR feature extraction methods are 0.52, 0.56 and 0.58 respectively. In research literature, BCIs designed on EEG ERP datasets using the same interaction paradigms have obtained average accuracy scores of 0.87 [42], highlighting the low standard of the accuracy scores achieved. Results on the sum of sine curve fitting being used as a feature extraction method in BCIs have not been reported in research literature. Therefore, the results that we achieved in this study can be used as a flag post to future researchers. The Fourier and VAR models have been used extensively in BCI literature, however in different ways in which we used them in our study. The Fourier model has been used to analyse the PSD of the EEG ERP signals [43] and is often used as a feature extraction method alongside other methods. The VAR model is usually used in the preprocessing stage of a BCI, as part of a filtering system such as the Kalman filter. Using a Kalman filter as part of a BCI has been shown to significantly improve the quality of the EEG ERP signals [44].

EEG signals are inherently noisy, therefore signal pre-processing is required to improve the signal to noise ratio. We only applied one preprocessing method, the Butterworth filter, which is a relatively simple filter. Research suggests that applying more advanced filters, such as adaptive filtering techniques, would reduce the noise of the signal and therefore improve the classification accuracies. Our research could be expanded to investigate using a larger range of preprocessing methods. In our study, we relied on MATLAB's Classification Learner App to automatically tune the hyperparameters of the classification algorithms. This was done due to the time constraints of the project, as the manual tuning of hyperparameters is a particularly time-consuming process. Future research into the tuning of hyperparameters could see a larger influence of the classification algorithms on the accuracy scores than we obtained.

This study has assessed the performance of feature extraction and machine learning algorithms on one of the largest EEG ERP datasets. Performing such studies is a vital part of developing operational BCIs that can help provide those with neuromuscular disorders control of computers and robotics. Using the parameters of polynomial curve fitting as the features of a SVM, we classified left-and-right hand ERP data with an average accuracy score of 0.85, which fits within existing research literature. This study has assessed the use of the sum of sine curve fitting as a feature extraction method in BCIs, which is a first in BCI literature. We observe that there is a lack of data preprocessing and hyperparameter tuning of classification algorithms in our study, providing topics of expansion for future research.

VII. References

[1] T. Kayikcioglu, O. Aydemir, "A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data," Pattern Recognition Letters, vol. 1, no.31, pp. 1207-1215, Aug 2010.

[2] T. Vaughan, W. Heetderks, L. Trejo, et al., "Brain-computer interface technology: a review of the Second International Meeting", IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society, vol. 11, no. 2, pp. 94-109, Jun 2003.

[3] S. Simanto, A. Khondaker, A. Khawza, et al., "Progress in Brain Computer Interface: Challenges and Opportunities," Frontiers in Systems Neuroscience, vol. 15, pp. 1-4, Feb 2021.

[4] G. Borghini, P. Aricò, I. Graziani, et al., "Quantitative assessment of the training improvement in a motor-cognitive task by using EEG, ECG and EOG signals," Brain topography, vol. 29, pp. 149-161, Jan 2016.

[5] k. LaFleur, K. Cassady, A. Doud, et al., (Jun 2013). "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface". Journal of neural engineering [online]. vol. 10, no. 4. available: https://iopscience.iop.org/article/10.1088/1741-2560/10/4/046003

[6] A. Singh, Y. Wang, J. King, et al., "Extended interaction with a bci video game changes resting-state brain activity," IEEE Transactions on Cognitive and Developmental Systems, vol. 12, no. 4, pp. 809-823, Apr 2020.

[7] P. Arico, G. Borghini, G. Di Flumeri, et al., "Passive BCI in operational environments: insights, recent advances, and future trends," IEEE Transactions on Biomedical Engineering, vol. 64, no. 7, pp. 1431-1436, Apr 2017.

[8] F. Lotte, L. Bougrain, A. Cichocki, et al., (Apr 2018). "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update". Journal of neural engineering [online]. vol.15, no. 3. available: https://iopscience.iop.org/article/10.1088/1741-2552/aab2f2/meta

[9] F. Matthews, B. Pearlmutter, T. Wards, et al., "Hemodynamics for brain-computer interfaces", IEEE Signal Processing Magazine, vol. 25, pp. 87-94, Dec 2007.

[10] G. Valente, A. Kaas, E. Formisano, et al., "Optimizing fMRI experimental design for MVPA-based BCI control: Combining the strengths of block and event-related designs", NeuroImage, vol.186, pp. 369-381, Feb 2019.

[11] A. Bashashati, M. Fatourechi, R. Ward, et al., "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals," Journal of Neural engineering, vol. 4, no. 2, Mar 2007.

[12] D. Olguin, F. Bouchereau, S. Martinez. "Adaptive notch filter for EEG signals based on the LMS algorithm with variable step-size parameter," in International Conference on Information Sciences and Systems, Baltimore, MD, 2005.

[13] S. Daud, R. Sudirman, "Butterworth bandpass and stationary wavelet transform filter comparison for electroencephalography signal," in Intelligent systems, modelling and simulation, Malaysia, 2015, pp. 123-126.

[14] D. Krusienski, W. Jenkins, "A particle swarm optimization-least mean squares algorithm for adaptive filtering" in Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2004, pp. 241-245.

[15] T. Paul, P. Karmakar, "Performances of Kalman filter and LMS Algorithm in Channel Estimation," International Journal of Engineering Research Technology, vol. 1, no. 5, pp. 2278-2283, Jul 2012.

[16] N. Thakor, Y. Zhu, "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection," IEEE transactions on biomedical engineering, vol. 38, no, 8, pp. 795-794, Aug 1991.

[17] G. Blanchard, B. Blankertz, "BCI competition 2003-data set IIa: spatial patterns of self-controlled brain rhythm modulations," IEEE Transactions on Biomedical Engineering, vol. 51, no.6, pp.1062-1066, May 2004.

[18] W. Ting, Y. Guo-Zheng, Y. Bang-Hua, et al., "EEG feature extraction based on wavelet packet decomposition for brain computer interface," Measurement, vol. 41, no. 6, pp. 618-625, Jul 2008.

[19] F. Beverina, G. Palmas, S. Silvoni, et al., "User adaptive BCIs: SSVEP and P300 based interfaces," PsychNology, vol.4, pp. 331-354, Dec 2003.

[20] A. Schlögl, F. Lee, H. Bischof, et al., (Aug 2005). "Characterization of four-class motor imagery EEG data for the BCI-competition 2005". Journal of neural engineering [online]. vol.2, no. 4. available: https://iopscience.iop.org/article/10.1088/1741-2560/2/4/L02/pdf

[21] K. Müller, M. Krauledat, G. Dornhege, et al. (Dec 2004). "Machine learning techniques for brain-computer interfaces," Biomed. Tech. Available: 10.1.1.69.3062rep=rep1type=pdf

[22] Y. Mishchenko, M. Kaya, E. Ozbay, et al., "Developing a three-to six-state EEG-based brain–computer interface for a virtual robotic manipulator control," IEEE Transactions on Biomedical Engineering, vol. 66, no. 4, pp. 977-987, Aug 2018.

[23] B. Blankertz, K. Muller, D. Krusienski, et al. "The BCI competition III: Validating alternative approaches to actual BCI problems," IEEE transactions on neural systems and rehabilitation engineering, vol. 14, no. 2, pp. 153-159, Jun 2006.

[24] M. Kaya, M. Binli, E. Ozbay, et al., "A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces," Scientific data, Vol. 5, pp. 1-6, Oct 2018.

[25] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectF-150916-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554531.v1. [Accessed: March 20, 2021].

[26] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectE-160122-3States, 2018 [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554534.v1. [Accessed: March 20, 2021].

[27] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectF-150928-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554537.v1. [Accessed: March 20, 2021].

[28] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectF-150917-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554540.v1. [Accessed: March 20, 2021].

[29] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectE-160119-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554528.v1. [Accessed: March 20, 2021].

[30] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectB-151215-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554510.v1. [Accessed: March 20, 2021].

[31] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectC-151223-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554513.v1. [Accessed: March 20, 2021].

[32] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectC-151216-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554516.v1. [Accessed: March 20, 2021].

[33] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectC-151126-3States, 2018. [Dataset] Available:

https://doi.org/10.6084/m9.figshare.5554519.v1. [Accessed: March 20, 2021].
[34] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectD-151125-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554522.v1. [Accessed: March 20, 2021].
[35] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectE-151225-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554525.v1. [Accessed: March 20, 2021].
[36] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectB-151019-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554501.v1. [Accessed: March 20, 2021].
[37] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectA-160108-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554504.v1. [Accessed: March 20, 2021].
[38] Y. Mishchenko, M. Kaya, M. Binli et al., Experiment CLA-SubjectB-151020-3States, 2018. [Dataset] Available: https://doi.org/10.6084/m9.figshare.5554507.v1. [Accessed: March 20, 2021].
[39] T. Dietterich, "Ensemble learning," in The handbook of brain theory and neural networksm M. Arbib: Bradford Books, pp. 110-125.
[40] H. Bhavsar, A. Ganatra, "A comparative study of training algorithms for supervised machine learning," International Journal of Soft Computing and Engineering, vol. 2, no. 4, pp. 2241-2307, Sep 2012.
[41] Chatterjee SK, Malik O, Gupta S. Chemical sensing employing plant electrical signal response-classification of stimuli using curve fitting coefficients as features. Biosensors. 2018 Sep;8(3):83.
[42] D. Irimia, R. Ortner, M. Poboroniuc, et al., "High classification accuracy of a motor imagery based brain-computer interface for stroke rehabilitation training," Frontiers in Robotics and AI, vol. 29 pp. 5-130, Nov 2018.
[43] K. Samiee, P. Kovacs, M. Gabbouj, "Epileptic seizure classification of EEG time-series using rational discrete short-time Fourier transform," IEEE transactions on Biomedical Engineering, vol. 62, no. 2, pp. 542-542, Sep 2014.
[44] N. Aznan, Y. Yang, "Applying Kalman filter in EEG-based brain computer interface for motor imagery classification" in The international conference on ICT convergence, Gumi-Si, South Korea, 2013, pp. 688-690

VIII. Appendix

	Classification Algorithm				
Feature extraction	Logistic regression	Decision tree	Ensemble methods	SVM	KNN
Polynomial 1	0.72	0.72	0.73	0.73	0.73
Polynomial 2	0.75	0.75	0.75	0.75	0.75
Polynomial 3	0.77	0.77	0.77	0.77	0.77
Polynomial 4	0.80	0.80	0.80	0.80	0.80
Polynomial 5	0.82	0.82	0.82	0.82	0.82
Polynomial 6	0.83	0.83	0.84	0.83	0.83
Polynomial 7	0.84	0.84	0.84	0.84	0.84
Polynomial 8	0.84	0.84	0.84	0.84	0.84
Polynomial 9	0.85	0.85	0.84	0.85	0.85
Fourier 1	0.56	0.56	0.56	0.56	0.55
Fourier 2	0.52	0.53	0.52	0.52	0.53
Fourier 3	0.53	0.52	0.53	0.52	0.53
Fourier 4	0.52	0.52	0.52	0.52	0.52
Fourier 5	0.52	0.51	0.52	0.52	0.52
Fourier 6	0.51	0.51	0.51	0.51	0.51
Fourier 7	0.51	0.51	0.51	0.51	0.50
Fourier 8	0.53	0.53	0.53	0.53	0.52
Sum of sine 1	0.61	0.61	0.61	0.61	0.61
Sum of sine 2	0.56	0.55	0.56	0.55	0.56
Sum of sine 3	0.56	0.56	0.55	0.56	0.56
Sum of sine 4	0.55	0.55	0.55	0.55	0.54
Sum of sine 5	0.54	0.54	0.54	0.53	0.54
Sum of sine 6	0.53	0.53	0.53	0.53	0.54
VAR 1	0.57	0.57	0.57	0.57	0.58
VAR 2	0.58	0.58	0.59	0.58	0.58

Table 2: The average results of classifying the EEG ERP datasets using feature extraction and classification algorithms.