Sādhanā Vol. 42, No. 5, May 2017, pp. 641–653
DOI 10.1007/s12046-017-0633-9

Improving EEG signal peak detection using feature weight learning of a neural network with random weights for eye event-related applications

ASRUL ADAM[1], ZUWAIRIE IBRAHIM[2], NORRIMA MOKHTAR[1,*], MOHD IBRAHIM SHAPIAI[3], PAUL CUMMING[4,5] and MARIZAN MUBIN[1]

[1]Applied Control and Robotics (ACR) Laboratory, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
[2]Faculty of Electrical and Electronic Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[3]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia Kuala Lumpur, Jalan Semarak, 54100 Kuala Lumpur, Malaysia
[4]School of Psychology and Counseling, Queensland University of Technology, Brisbane 4000, Australia
[5]QIMR Berghofer, Brisbane 4006, Australia
e-mail: asrul.adam@siswa.um.edu.my; zuwairie@ump.edu.my; norrimamokhtar@um.edu.my; md_ibrahim83@utm.my; paul.k.cumming@gmail.com; marizan@um.edu.my

MS received 17 March 2016; revised 18 September 2016; accepted 4 November 2016

Abstract. The optimization of peak detection algorithms for electroencephalogram (EEG) signal analysis is an ongoing project; previously existing algorithms have been used with different models to detect EEG peaks in various applications. However, none of the existing techniques perform adequately in eye event-related applications. Therefore, we aimed to develop a general procedure for eye event-related applications based on feature weight learning (FWL), through the use of a neural network with random weights (NNRW) as the classifier. The FWL is performed using a particle swarm optimization algorithm, applied to the well-studied Dumpala, Acir, Liu and Dingle peak detection models, where the associated features are considered as inputs to the NNRW with and without FWL. The combination of all the associated features from the four models is also considered, as a comprehensive model for validation purposes. Real EEG data recorded from two channels of 20 healthy volunteers were used to perform the model simulations. The data set consisted of 40 peaks arising in the frontal eye field in association with a change of horizontal eye gaze direction. It was found that the NNRW in conjunction with FWL has better performance than NNRW alone for all four peak detection models, of which the Dingle model gave the highest performance, with 74% accuracy.

Keywords. Neural network with random weights (NNRW); feature weight learning (FWL); electroencephalogram (EEG); peak detection algorithm; pattern recognition; particle swarm optimization (PSO).

1. Introduction

The utilization of peak detection algorithms has emerged as a useful tool in several physiological signal applications, such as the detection of epileptic activity [1], photoplethysmogram (PPG) monitoring [2] and the detection of eye gaze direction from activity in the frontal eye field [3]. In these applications, the peak detection algorithm is typically implemented in the first step of the signal classification process. For example, in an important application for clinical neurology, epileptiform activity in cerebral cortex is identified from recurrent spikes in the electroencephalogram (EEG) recording during a given time interval [4]. A similar approach is used in procedures for detecting horizontal eye gaze direction, which has applications for brain–machine interfacing [5]. The existing works that use the combination of eye gaze and EEG for brain–machine interfacing have been briefly reviewed in [6]. Furthermore, in the case of PPG signal monitoring, peak detection algorithms serve to measure heart rate variability, which can be predictive of risk for heart disease [7]. In all these applications, it is essential to have peak detection algorithms with high performance.

One approach for optimizing the performance of peak detection algorithms is to identify the most relevant combination of peak features. A number of research groups have previously defined peak features based on the characteristic of peaks in a time domain analysis of EEG signal

*For correspondence

[1, 8–12]. Each true EEG signal peak is revealed by several signal parameters, including amplitude, width and slope, from which a variety of peak features can be calculated. These features include the peak-to-peak amplitude of the first and second half waves, peak width, ascending peak slope of the first half wave and descending peak slope of the second half wave. These and other features can be used as inputs to the classification process aiming to differentiate between peak and non-peak. In one literature study, Dumpala *et al* [8] used the defined peak features and introduced a classification process to detect a peak signal in the analysis of gastric electrical activity (ECA) signals. The groups of Acir *et al* [1], Liu *et al* [11] and Dingle *et al* [12] also used the defined peak features and different classification processes for peak detection in EEG recordings from patients with epilepsy. In the present work we refer to the various peak features as the Dumpala, Acir, Liu and Dingle peak models. In general, these four pre-defined peak models have performed with high accuracy in their selected applications. However, the optimal features for one model may not generalize to other applications like eye event-related, such that there remains scope for developing a single robust platform.

The feature weight learning (FWL) method has extensively been used in diverse applications, such as digital image processing [13], text categorization [14], detection of heart disease [15] and industrial applications such as diagnosis of faulty bearings [16]. Also, FWL techniques have been widely used for the analysis of EEG signals associated with specific cognitive tasks [15] or defining the stage of sleep in polysomnography [17]. Using FWL, the EEG peak features are optimized to ensure that the selected peak detection algorithm is sufficiently accurate for the task at hand. In simple terms, the weight value of every feature is assigned a value between 0 and 1. By defining, the weights of relevant features will have higher weights, while irrelevant features will have lower weights [18].

In the present study, we test an EEG peak detection algorithm using FWL implemented in a neural network with random weights (NNRW). We performed FWL using a particle swarm optimization (PSO) algorithm. The general problem is considered as an optimization with continuous inquiry of the search space for identifying relevant and irrelevant weight features. We have applied the four different peak models and all defined features using real command-gated EEG data from healthy volunteers using the stand-alone NNRW classifier [19]; an earlier analysis of these data is to be presented in the experimental results section. We found that the selection of relevant features is necessary for providing the best and generalized performance in EEG signal peaks classification. However, we felt that the performance might be improved with FWL based on an NNRW procedure. Therefore, we further investigate through employing an existing advanced method for improving the performance of each peak model. Based on a well-established paradigm, the EEG data set consists of 40 peaks arising in the frontal eye field in association with a voluntary change of horizontal eye gaze direction. The four different peak models and all features were tested on the EEG data, so as to identify the most relevant peak model by evaluating the classification accuracy of the associated learning weights.

2. Existing peak models in the time domain analysis

Dumpala *et al* [8] introduced a peak model that comprised four features, namely (1) the peak-to-peak amplitude of the first half wave, (2) the peak width between the first and second half wave, (3) the ascending peak slope of the first half wave and (4) the descending peak slope of the second half wave.

Acir *et al* [1] introduced an additional feature of peak amplitude, defined as the peak-to-peak amplitude of the second half wave, and also two addition features of the peak width, i.e., the widths of the first and second half waves: this gives a total of six features, although Acir *et al* [1] did not use the width feature as introduced by Dumpala *et al* [8]. Liu *et al* [20] have used a similar definition of the peak amplitudes, widths and slopes as those of Acir *et al* [1], but with addition of the area of the peak as an extra feature: we did not consider peak area in the present study.

The considerably more complex model of Liu *et al* [11] entails 11 features. Here, the peak model consists of four amplitudes: (1) the peak-to-peak amplitude of the first (2) and the peak-to-peak amplitude of the second half waves, as well as (3) the amplitudes of the peak relative to the turning points of the first and second half waves (4). The turning points is defined as the point where the slope decreases more than 50% as compared with the tangent at the preceding point. The model also consists of three widths: (1) the width between the first and second half wave, (2) the width between the turning point of the first and second half waves and (3) the width between the halfway points of the first and second half waves and it is also known as full width at half maximum (FWHM). Four slopes are also measured in Liu model, i.e., the ascending peak slope of the first half wave, the descending peak slope of the second half wave and two turning point slopes of the first and second half waves, thus to a total of 11 features.

The final peak model, as introduced by Dingle *et al* [12], consists of four features. Here, the peak amplitude is the difference between the peak and the floating mean, which is the mean EEG centred, that is, centred at the peak, and is also called moving average curve (MAC) [21]. The MAC is calculated over a specific time interval. The peak width is calculated based on the difference between the first and second half waves, and two slopes are defined from the first and second half waves to the peak. Recently, Elgendi *et al*

[2] used MAC in a procedure for detecting systolic peaks for heart rate analysis.

In general, all peak features in the time domain analysis can be formulated based on the eight points shown in figure 1. The set of points comprising the ith peak point, PP_i, the two associated valley points, $VP1_i$ and $VP2_i$, the half point of first half wave ($HP1_i$), the half point of second half wave ($HP2_i$), the turning point of first half wave ($TP1_i$), the turning point of second half wave ($TP2_i$) and the moving average curve point ($MAC(PP_i)$). As such, all peak features can be categorized into three groups, namely amplitude, width and slope, resulting in five different amplitudes, seven different widths and four different slopes, to a total of 16 features, as shown in table 1.

Details of the different peak detection algorithms in different peak models are tabulated in table 2, which is generalized to signals such as PPG and ECA as well as EEG. However, the main goal of peak detection is to determine the event, not describe the fine structure of entire peaks. Therefore, for the event detection systems highlighted in table 1, detection performance serves only to identify the best performing algorithm for capturing an event.

3. Employed methods

3.1 *Neural network with random weights (NNRW)*

NNRW is a randomized learner approach entailing a single-layer feed-forward neural network (SLFN), as first introduced by Wouter F Schmidt [22]. In essence, the network of NNRW consists of input, hidden and output layers, where the parameters of input weights and the biases at the hidden layer in the network are randomly chosen for a specific time interval. The weights at the output layer in the network can be estimated by calculating the Moore–Penrose generalized matrix inverse [23]. A similar randomized learner concept introduced by Pao *et al* [24], known as the random vector functional-link (RVFL) networks, has been extended in various tests of the RVFL concept.

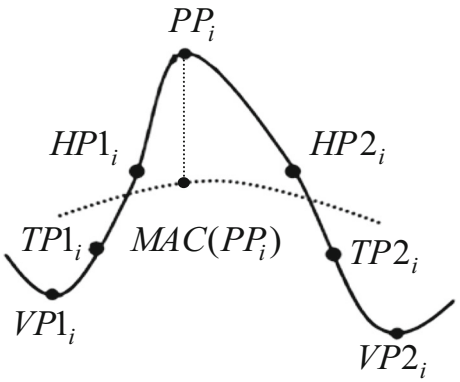

Figure 1. Eight-point locations of a peak.

Lying between the input and hidden layers of the NNRW are the input weights, whereas output weights lie between the hidden and output layers. The training process of the NNRW consists of three stages. First, the input weights are assigned random values between -1 and $+1$. Next, the biases in the hidden layer are assigned random values between 0 and $+1$. The approach to assign the input weights and biases is based on the experimental procedure in [25]. During the training process, the input weights and biases remain fixed. Afterwards, the output matrix of the hidden layer, H, is calculated as follows:

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g\left(\sum_{i=1}^{d} a_{i1}x_{1i} + b_1\right) & \cdots & g\left(\sum_{i=1}^{d} a_{iL}x_{1i} + b_L\right) \\ \vdots & \ddots & \vdots \\ g\left(\sum_{i=1}^{d} a_{i1}x_{Ni} + b_1\right) & \cdots & g\left(\sum_{i=1}^{d} a_{iL}x_{Ni} + b_L\right) \end{bmatrix}_{N\times L} \tag{1}$$

where g is an activation function of the hidden neuron, x is the $N \times L$ matrix of inputs, a is the $d \times L$ matrix of random input weights, b is the $1 \times L$ matrix of random biases in the hidden layer, N is an arbitrary distinct sample, L is the number of hidden neurons and d is the number of inputs. The ith column of H is the output of the ith hidden neuron with respect to inputs x_1, x_2, until x_d.

The NNRW can be represented as a linear system, modelled mathematically as

$$H\beta = T \tag{2}$$

where β is the $L \times m$ matrix of output weights and T is the $N \times m$ matrix of target outputs; m is the number of output neurons. The β and T matrixes are denoted as

$$\beta = \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_L^{\mathrm{T}} \end{bmatrix}_{L\times m} \tag{3}$$

and

$$T = \begin{bmatrix} t_1^{\mathrm{T}} \\ \vdots \\ t_N^{\mathrm{T}} \end{bmatrix}_{N\times m} \tag{4}$$

respectively. To find the least-square solution, β, of the linear system, $H\beta = T$, the minimum-norm least-square solution is computed as follows:

$$\|H(a_1, \ldots, a_L, b_1, \ldots, b_L)\beta - T\| = \min_{\beta} \|H(a_1, \ldots, a_L, b_1, \ldots, b_L)\beta - T\|. \tag{5}$$

Table 1. Equations and descriptions of peak features.

Peak feature	Feature name	Equation	Description
Amplitudes	Peak-to-peak amplitude of the first half wave	$f_1 = \lvert x(PP_i) - x(VP1_i)\rvert$	Amplitude between peak and valley of the first half wave
	Peak-to-peak amplitude of the second half wave	$f_2 = \lvert x(PP_i) - x(VP2_i)\rvert$	Amplitude between peak and valley of the second half wave
	Turning point amplitude of the first half wave	$f_3 = \lvert x(PP_i) - x(TP1_i)\rvert$	Amplitude between peak and turning point of the first half wave
	Turning point amplitude of the second half wave	$f_4 = \lvert x(PP_i) - x(TP2_i)\rvert$	Amplitude between peak and the turning point of the second half wave
	Moving average amplitude	$f_5 = \lvert x(PP_i) - MAC(PP_i)\rvert$	Amplitude between peak and moving average
Widths	Peak width	$f_6 = \lvert VP1_i - VP2_i\rvert$	Width between valleys of the first and second half waves
	First half wave width	$f_7 = \lvert PP_i - VP1_i\rvert$	Width between peak and valley of the first half wave
	Second half wave width	$f_8 = \lvert PP_i - VP2_i\rvert$	Width between peak and valley of the second half wave
	Turning point width	$f_9 = \lvert TP1_i - TP2_i\rvert$	Width between turning points of the first and second half waves
	First half wave turning point width	$f_{10} = \lvert PP_i - TP1_i\rvert$	Width between turning point of the first half wave and peak
	Second half wave turning point width	$f_{11} = \lvert PP_i - TP2_i\rvert$	Width between turning point of the second half wave and peak
	FWHM	$f_{12} = \lvert HP1_i - HP2_i\rvert$	Width between the half-way points of the first and second half waves
Slopes	Peak slope of the first half wave	$f_{13} = \left\lvert \frac{x(PP_i)-x(VP1_i)}{PP_i-VP1_i} \right\rvert$	Ascending peak slope of the first half wave
	Peak slope of the second half wave	$f_{14} = \left\lvert \frac{x(PP_i)-x(VP2_i)}{PP_i-VP2_i} \right\rvert$	Descending peak slope of the second half wave
	Turning point slope of the first half wave	$f_{15} = \left\lvert \frac{x(PP_i)-x(TP1_i)}{PP_i-TP1_i} \right\rvert$	Ascending peak slope between peak and turning point of the first half wave
	Turning point slope of the second half wave	$f_{16} = \left\lvert \frac{x(PP_i)-x(TP2_i)}{PP_i-TP2_i} \right\rvert$	Descending peak slope between peak and turning point of the second half wave

Table 2. Summary of the previous studies using various types of peak detection algorithms for different peak models.

Peak model	Input signals	Event	Classification method	Mean accuracy test of event (%)
Elgendi *et al* [2]	PPG	Heart rate analysis	Thresholds, rule-based	Sensitivity: 99.89 Selectivity: 99.84
Liu *et al* [20]	EEG	Epilepsy	AdaBoost	93.5
Acir [9]	EEG	Epilepsy	Radial basis function network (RBFN)	Sensitivity: 91.1 Selectivity: 89.2
Acir *et al* [1]	EEG	Epilepsy	Radial basis support vector machine (RB-SVM)	Sensitivity: 89.1 Selectivity: 85.9
Liu *et al* [11]	EEG	Epilepsy	ANN, Expert system	90
Dingle *et al* [12]	EEG	Epilepsy	Thresholds, rule-based, Expert system	80
Dumpala *et al* [8]	ECA	Gastric activity	Thresholds, Rule-based	100

It is well known that the smallest norm least-square solution of Eq. (5) is

$$\beta = (H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}T = H^{+}T \tag{6}$$

where H^{+} is the Moore–Penrose pseudo-inverse of H. The summary of the training stages of the NNRW classifier is as follows.

Stage 1 Assign randomly the input weights a_i and biases in the hidden neurons, b_i

Stage 2 Calculate the output matrix of the hidden layer, H

Stage 3 Calculate the output weights, $\beta = H^{+}T$

The output function of NNRW classifier of a given unknown sample x is

$$f(x) = h(x)\beta. \quad (7)$$

In the output layer, two neurons are used in the network for binary classification of the output into class 0 and 1. The predicted class label of a given unknown sample x is defined as follows:

$$label(x) = \arg\max_{i\in\{1,\ldots,m\}} f_i(x)\,. \quad (8)$$

3.2 *PSO algorithm*

The PSO algorithm was first introduced by James Kennedy and Russell Eberhart in 1995 as a representation of the movement of flocking birds or schooling fish [26]. In this study, the PSO algorithm iteratively attempts to find the best solution in a specific search space and it becomes the learning process of features weights for signal peak detection problem. In general, the PSO algorithm follows several steps as denoted in Algorithm 1 below: (1) initialize, (2) calculate the fitness evaluation function, (3) update the personal best (*pbest*) for each particle and global best (*gbest*), (4) update the particle's velocity and position and (5) terminate based on a stopping criterion.

In the initialization step, several initial PSO parameters such as the initial weight, ω, the cognitive and social components, c_1 and c_2, the random values r_1 and r_2, the initial velocity vector and *pbest* score for each particle and the *gbest* score for a population at each iteration are all set to a specific initial value.

Algorithm 1: PSO algorithm

```
Initialization
while not stopping criteria do
    for each ith particle in a population do
        calculate fitness evaluation function
        update pbest and gbest
    end for
        for each ith particle in a population do
            update the ith particle's velocity and
            update the ith particle's position
        end for
end while
```

4. EEG signals peak detection algorithm

The algorithm of EEG signal peak detection (figure 2) involves training and testing phases to determine the best fit model for the collected EEG data. The data used in this study had been recorded using two scalp electrodes from clinical EEG recordings from 20 healthy young make volunteers, who gave informed consent to participate in the study, in accordance with institutional regulations. In the first stage of peak detection, the input training and testing EEG signals must be filtered so as to remove noises. The filtering process in this study uses the Daubechies 4 wavelet at level 5. The training phase of the algorithm involves several processes, including peak candidate identification, feature extraction and feature weight with data learning processes implemented by the NNRW classifier. The FWL is performed during the training phase in order to identify the optimal network weights using the PSO algorithm described earlier. Also, the estimation process is executed during this initial phase so as to train the network for adjusting NNRW parameters using the learning algorithm of the NNRW classifier. In the testing phase, the algorithm follows the same sequence of processes, but the optimal feature weights and NNRW parameters emerging from the training phase are used in the classification process of the testing phase. The final output of the training and testing phase is the predicted peak points and non-peak points of the identified peak candidates.

4.1 *Peak candidate identification*

The first process of the detection algorithm is to recognize candidate peaks, which must be parsed into groups consisting of true non-peaks and true peaks. The group of true peaks is reassigned to the category of candidates peaks, which can be further classified into predicted true non-peaks and predicted true peaks using the NNRW classifier.

The process to determine peak candidates is as follows. By considering a discrete-time signal, $x(I)$, of L points, the ith candidate peak point, PP_i, is identified using the three-point sliding window method [8, 27]. The three points are denoted as $x(I-1)$, $x(I)$ and $x(I+1)$ for $I = 1, 2, 3,\ldots, L$. A candidate peak point is identified when $x(PP_i - 1)\langle x(PP_i)\rangle x(PP_i + 1)$ and two associated valley points, $VP1_i$ and $VP2_i$, are found between. Both valley points exist when $x(VP1_i - 1) > x(VP1_i) < x(VP1_i + 1)$ and $x(VP2_i - 1) > x(VP2_i) < x(VP2_i + 1)$.

4.2 *Feature extraction*

The entire set of 16 peak features listed in table 1 are used to define the four different peak models, as tabulated in table 3.

4.3 *FWL-NNRW*

In integrating FWL into NNRW, one layer is added to the right side of the network, as shown in figure 3. The weights in the layer represent the feature weights for input neurons. As noted earlier, the weights are initially assigned as a random number between 0 and 1. The output matrix of the hidden layer for integrated FWL, H', is modified as follows:

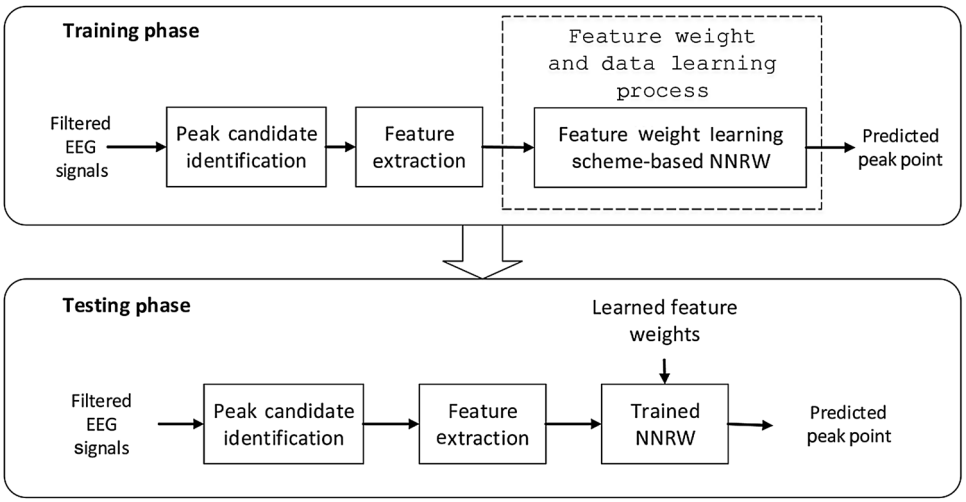

Figure 2. Training and testing phase of EEG signals peak detection algorithm.

Table 3. List of different peak models and corresponding sets of features.

Peak model	Set of features	Number of features
Dingle *et al* [12]	f_5, f_6, f_{13}, f_{14}	4
Dumpala *et al* [8]	f_1, f_6, f_{13}, f_{14}	4
Acir *et al* [1]	$f_1, f_2, f_7, f_8, f_{13}, f_{14}$	6
Liu *et al* [11]	$f_1, f_2, f_3, f_4, f_6, f_9, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}$	11
All features	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}$	16

$$H' = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g\left(\sum_{i=1}^{d} a_{i1} w_i x_{1i} + b_1\right) & \cdots & g\left(\sum_{i=1}^{d} a_{iL} w_i x_{1i} + b_L\right) \\ \vdots & \ddots & \vdots \\ g\left(\sum_{i=1}^{d} a_{i1} w_i x_{Ni} + b_1\right) & \cdots & g\left(\sum_{i=1}^{d} a_{iL} w_i x_{Ni} + b_L\right) \end{bmatrix}_{N \times L} \tag{9}$$

where w_i is the feature weights of the input neurons. The estimated output weights of the modified NNRW are denoted as

$$\beta' = H'^{+} T. \tag{10}$$

Subsequently, the output function and the predicted class label are reformulated as follows:

$$f'(x) = h'(x)\beta' \tag{11}$$

$$label(x) = \arg\max_{i \in \{1,\ldots,m\}} f_i'(x) \tag{12}$$

Table 4 illustrates the representation of particle position. The ith particle at the kth iteration represents the feature weight w_i^k. The position of particle i at iteration k is denoted as $w_i^k = \left\{w_{i,1}^k, w_{i,2}^k, w_{i,j}^k, \ldots, w_{i,d}^k\right\}$, where $j = 1, 2, 3, \ldots, d$ is a jth dimension of feature weights.

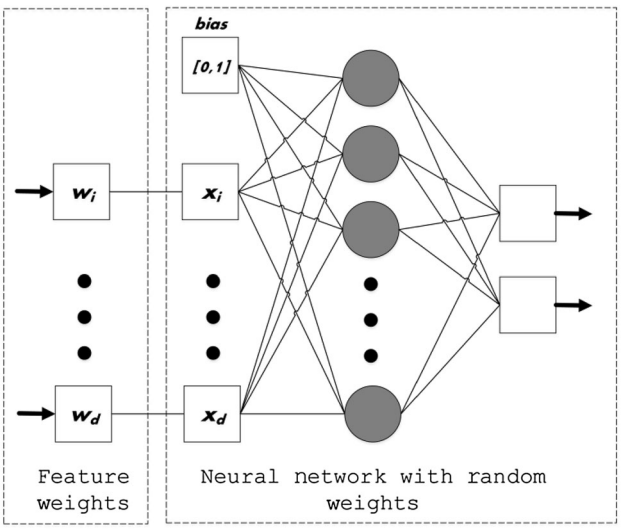

Figure 3. Feature weights learning NNRW.

The PSO algorithm adapts the feature weight parameters of the NNRW during a training process. Our primary

Table 4. Representation of feature weight parameters.

Feature weights			
Particle *i*th's position			
$w_{i,1}^k$	$w_{i,2}^k$	...	$w_{i,d}^k$

objective is to find the value of the feature weights for producing the highest classification performance in the training process. The detailed process of the evaluation of fitness evaluation function in PSO algorithm is illustrated in figure 4. To this end, the fitness evaluation function is based on the classification output of FWL-NNRW classifier and *Gmean* [28]. The *Gmean* is calculated as follows:

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$TNR = \frac{TN}{TN + FP} \tag{14}$$

$$Gmean = \sqrt{TPR \times TNR} \tag{15}$$

where any true peak (*TP*) is the correctly detected apex point of a peak candidate, a true non-peak (*TN*) is any correctly detected non-peak point of a peak candidate, a false peak (*FP*) is an incorrectly designated non-peak point of a peak candidate, a false non-peak (*FN*) is any incorrectly detected true peak point of peak candidate, *TPR* is the true peak rate and *TNR* is the true non-peak rate.

5. Experiments

In this section, we describe the two main experiments for detecting peaks in our EEG signals. In the first experiment, all 16 features and the four different peak models are considered as inputs to NNRW without any FWL. In the second experiment, all 16 features and the four different peak models in table 3 are used as inputs to the FWL of the NNRW.

5.1 *Experimental set-ups*

Each experiment was conducted in 30 independent runs. To prepare the experiment data, the first half of the filtered EEG signals was assigned to training data, and the second half of testing data. The first half of training data consists of the series of EEG recordings in 10 subjects, while the second half of testing data consists of the series of EEG recording in other 10 subjects As shown in table 5, for the NNRW classifier of each experiment, the number of hidden neurons was selected by a trial and error method, to a maximum of 500 neurons. The sigmoid $[-1, 1]$ is used as an activation function in the hidden layer for normalization, whereas a linear function is located inside the neuron in the output layer. Other settings for the NNRW classifier, such as the number of neurons in the input layer and the number of neurons in the feature weight layer, are dependent on the number of the associated features of the particular peak model. The number of output neurons was set to 2. We note that the input weights and the biases remained fixed during the training, but the values of these two NNRW parameters changed for each run.

For each run of the NNRW with FWL, we used 30 particles for performing the FWL. For each of these particles, the total number of dimensions is equal to the number of features in the feature set. The maximum number of iterations for the PSO algorithm was set to 1000.

For the initial value of PSO parameters, the maximum inertia weight, $\omega_{\max}$, was 0.9 and the minimum inertia weight, $\omega_{\min}$, was 0.4. The cognitive component, c_1, and the social component, c_2, were set to 2, as proposed by Shi and Eberhart [29]. The random values, r_1, and r_2, are assigned random values ranging from 0 to 1. The initial velocity vector for each particle, the *pbest* score for each particle and *gbest* score were all set to 0. These parameters settings of the PSO are presented in table 6.

To evaluate the peak detection algorithm for training and testing phases, four different measurements are used, including the average *Gmean*, the maximum *Gmean*, the minimum *Gmean* and the standard deviation (STDEV).

5.2 *Experimental protocols*

The experimental protocol was approved by the medical ethics committee of the University of Malaya Medical Centre. All subjects signed informed consent forms in advance of the EEG session. The 20 healthy volunteers were undergraduate and postgraduate students in the Faculty of Engineering. The filtered EEG signals in this study were obtained in the Applied Control and Robotic (ACR) Laboratory, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Malaysia.

The filtered EEG signal recordings were conducted using the g.MOBIlab portable biological signals acquisition system. The scalp electrode arrangement was placed following the international convention for the 10–20 electrodes placement, but with the use of only four active electrodes. The EEG signals were recorded from channels C3 and C4, while channel CZ served as a reference. The ground electrode was located on the FPz channel. The sampling frequency was set to 256 Hz. The electrodes from the C3 and C4 channels are positioned on the central scalp for detecting EEG peaks associated with the brain activity associated with commanded horizontal eye gaze direction shifts. The eye gaze directions producing EEG peaks for channels C3 and C4 were archived.

The subjects were advised to wash their hair and rest before the day of data collection session. In the data collection session, the subjects were told to prepare for the

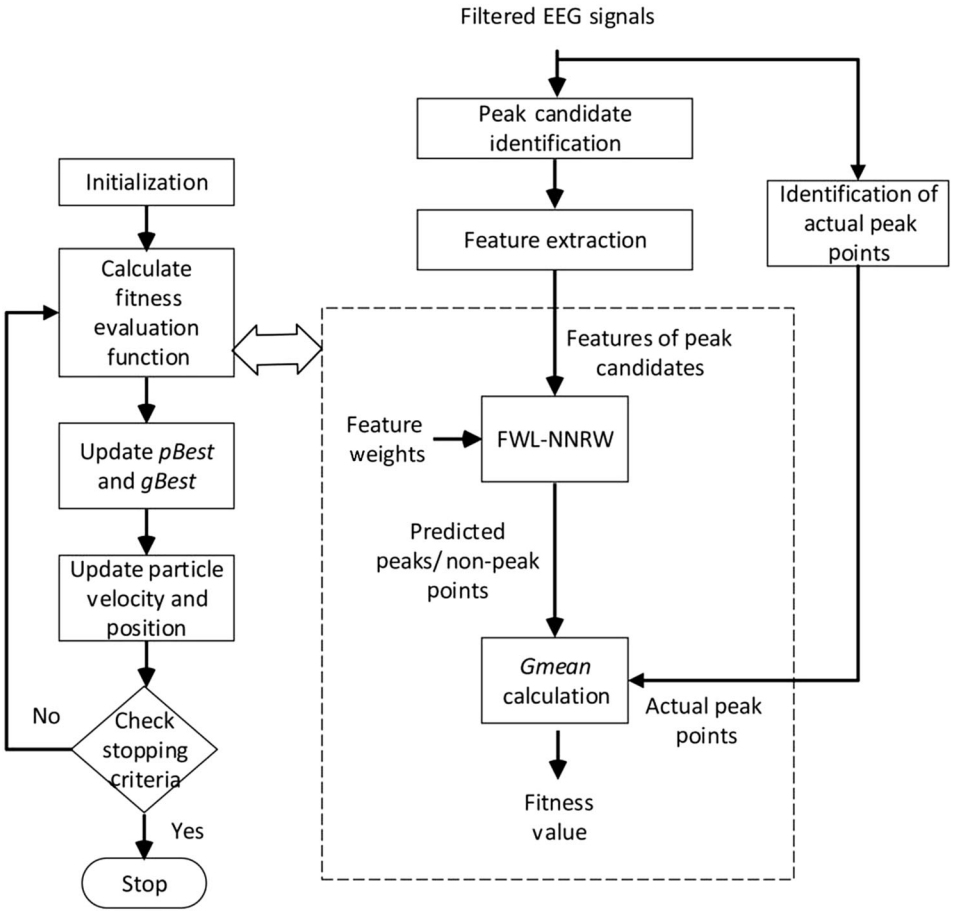

Figure 4. FWL-NNRW with PSO algorithm for peak detection algorithm.

Table 5. Parameters setting of NNRW.

Parameters	Value
Number of neurons in hidden layer	500
Activation function at hidden layer	Sigmoid [−1, 1]
Activation function at output layer	Linear function
Number of neurons in the input layer	Depends on number of features
Number of neurons in the output layer	2
Number of neurons in the feature scaling layer	Depends on number of features

external voice cue within up to 4 s. Appearance of the cue is voice command or verbal reminder for the subject to move his eyes, initially in forward fixation, to the left or right. At exactly 5 s from the beginning session, the external voice cue appears instructing the subject to follow the command. The subjects have only to follow the command to shift gaze to the left or right direction, and hold the new eye position from 5 until 10 s, which is the end of the EEG recording. This instruction is usually used to reduce the number of blink artifacts. In that case, the subjects are instructed to blink before and after recording session. The eye gaze directions that produce some peaks in the signals on channels C3 and C4 are archived as raw data for analysis.

Figure 5 shows an example of a filtered EEG signal that is linked to a voluntary horizontal eye movement event, either to the left or the right. The dotted red vertical lines show the peak point location, as assigned manually. The eye movement recordings consist of 40 signals from the C3 and C4 channels, each of 10 s sampled at 256 Hz, as specified in table 7.

Several previous studies have relied upon the C3 and C4 channels to record the EEG activation in association with commanded gaze shift [3, 30]. Also, the CZ channel is commonly used as a reference for EEG signals; we selected the C3, C4 and CZ EEG channels because they have relatively little contamination from artefacts due to eye blinking [31]. The C3 and C4 channels are of proven utility in human–machine interface (HMI) applications such as wheelchair navigation [30].

All recodings totalled 40 s in duration, and contained 102,400 sampling points, of which only 40 are actual peak locations. The first 51,200 sampling points were used as training data. Here, initial peak analysis recognized 1730

Table 6. Parameters setting of PSO algorithm.

Parameters	Value
Decreasing inertia weight, ω	0.9–0.4
Cognitive component, c_1	2
Social component, c_2	2
Random values, r_1 and r_2	Random number [0,1]
Initial velocity vector for each particle	0
Initial *pbest* score for each particle	0
Initial *gbest* score	0

candidate peaks, of which 20 are genuine peaks, and the remainder non-peaks.

6. Results and discussion

6.1 *Peak model without FWL*

The results of peak detection for training and testing performance without FWL are tabulated in table 8. The average, maximum, minimum and STDEV values for the training performance were 84.7%, 86.6%, 83.7% and 1.4%, respectively, for the Dumpala model, 88.3%, 89.4%, 86.6% and 1.4% for the Acir model, 78.9%, 83.7%, 74.1% and 2.6% for the Liu model, 99.3%, 100%, 97.4% and 0.9% for the Dingle model and 88.3%, 94.9%, 80.6% and 3.6% with all 16 peak features. Overall, the Dingle model had superior accuracy in the training stage compared with the Liu model.

For the testing performance, the average, maximum, minimum and STDEV values were 70.1%, 82.6%, 51.6% and 6.7% for the Dumpala model, versus 36.9%, 62.6%, 0% and 11.9% for the Acir model, 52.1%, 71.8%, 37.2% and 7.9% for the Liu model, 71.7%, 89.2%, 57.1% and 6.9% for the Dingle model and 36.9%, 58.1%, 0% and 11.8% when using all 16 features. Thus, the Dingle model gave the highest accuracy, whereas the Acir and 16-feature models gave the lowest accuracies.

One of the 30 runs using the Acir and the 16 features models had zero contribution to detection performance during the testing phase, due to failure of the testing performance in that instance, presumably due to the presence of irrelevant features, which rendered the classifier unfit to recognize all true peak points in those signals. In this case, the Dingle model obtained the highest minimum testing accuracy (57.1%), indicative of its greater robustness.

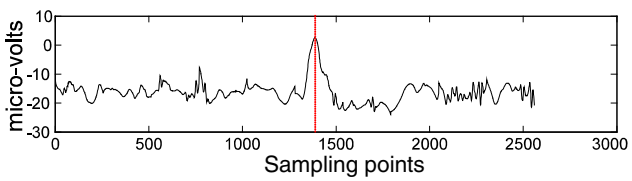

Figure 5. The filtered EEG-based voluntary eye movement signal (one peak point per signal).

Table 7. Signals specifications.

Specification	Channel C3	Channel C4
Sampling points per signal	2560	2560
Length per signal (s)	10	10
No. of signals	20	20
No. of peak point per signal	1	1
Sampling frequency (Hz)	256	256

6.2 *Peak model with FWL*

Results for 30 runs of the several peak models with FWL in the NNRW classifier are listed in table 9. These results include the training and testing performances for the four peak models and the 16-feature analysis. The four peak models inherently ignore some peak features, and use only designated features as inputs to the classifier. These peak models were selected based on the single characteristics, rather than via an experimental approach. Thus, we can now ascertain the effect of these peak models on the detection performance of the FWL- NNRW classifier. The training performance for average, maximum, minimum and STDEV values were 92%, 92.2%, 89.4%, 0.7% for the Dumpala model; 92.2%, 92.2%, 92.2%, 0% for the Acir model; 89.5%, 92.2%, 89.4%, 0.5% for Liu model and 100%, 100%, 100%, 0% for the Dingle model versus 98.4%, 100%, 94.9%, 2.1% for the 16-feature analysis. Overall, the training results of the Dingle model proved superior compared with the Dumpala, Acir and Liu models; training detection performance of the Dingle model was 100%. The Acir and Dingle models both contributed 0% of the STDEV, which is the lowest STDEV value when compared with other existing models. We conclude that that FWL greatly improved the learning process of the NNRW classifier.

Testing performances for average, maximum, minimum and STDEV values were 72.4%, 96.4%, 44.3%, 13.6% for the Dumpala model; 68.7%, 96.2%, 31.4%, 17.3% for the Acir model; 66.8%, 96.3%, 22.3%, 18.9% for the Liu model; 74.1%, 96.3%, 49.5%, 10% for the Dingle model and 52.9%, 70.7%, 31.6%, 3% for all 16 features. The highest testing detection performance was obtained by the Dingle model, whereas the all 16 features model gave the lowest performance. With respect to the 16-feature analysis, the selected features in the Dingle model enhanced detection performance from 52.9% to 74.1%.

6.3 *Comparison of results for NNRW with and without FWL*

We present comparison of peak model performance with and without FWL in table 10, for analysis of EEG data using four established peak models, and a complete 16-feature peak description. In general, we see improved performance of NNRW with the addition of FWL, which

Table 8. Peak detection training and testing performance without using feature weight learning.

	Training (%)				Testing (%)			
Peak model	Average	Max.	Min.	STDEV	Average	Max.	Min.	STDEV
Dumpala	84.7	86.6	83.7	1.4	70.1	82.6	51.6	6.7
Acir	88.3	89.4	86.6	1.4	36.9	62.6	0	11.9
Liu	78.9	83.7	74.1	2.6	52.1	71.8	37.2	7.9
Dingle	99.5	100	97.4	0.9	71.7	89.2	57.1	6.9
All features	88.3	94.9	80.6	3.6	36.9	58.1	0	11.8

Table 9. Peak detection training and testing performance using feature weight learning.

	Training (%)				Testing (%)			
Peak model	Average	Max.	Min.	STDEV	Average	Max.	Min.	STDEV
Dumpala	92	92.2	89.4	0.7	72.4	96.4	44.3	13.6
Acir	92.2	92.2	92.2	0	68.7	96.2	31.4	17.3
Liu	89.5	92.2	89.4	0.5	66.8	96.3	22.3	18.9
Dingle	100	100	100	0	74.1	96.3	49.5	10
All features	98.4	100	94.9	2.1	52.9	70.7	31.6	3

Table 10. Comparisons of peak models testing accuracy for NNRW without and with FWL.

Peak model	Set of feature	NNRW without FWL (%)				NNRW with FWL (%)			
		Average	Max.	Min.	STDEV	Average	Max.	Min.	STDEV
All features	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}$	36.9	58.1	0	11.8	52.9	70.7	31.6	3
Liu	$f_1, f_2, f_3, f_4, f_6, f_9, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}$	52.1	71.8	37.2	7.9	66.8	96.3	22.3	18.9
Acir	$f_1, f_2, f_7, f_8, f_{13}, f_{14}$	36.9	62.6	0	11.9	68.7	96.2	31.4	17.3
Dumpala	f_1, f_6, f_{13}, f_{14}	70.1	82.6	51.6	6.7	72.4	96.4	44.3	13.6
Dingle	f_5, f_6, f_{13}, f_{14}	71.7	89.2	57.1	6.9	74.1	96.3	49.5	10

increased average test results from 37% to 53% for the case of all 16 features. However, the 16-feature peak description was by no means the best performing input for the NNRW classifier with FWL. We suppose that the 16 features are overspecified due to the well-known noise properties of recorded EEG signals. Thus, the individual features were all vulnerable to large variance arising from various noise sources. We find that a selection of peak features gives superior input for the classifier to differentiate correctly between peak and non-peak.

As can be seen in table 10, over 30 runs, the Acir peak model, like the all-feature model, gave 0% accuracy of the minimum testing result as an input to NNRW without FWL. However, the addition of FWL to the NNRW avoided any instances of 0% testing accuracy of minimum testing accuracy for each peak model. This confirms the general superiority of FWL-NNRW classifier.

Furthermore, we find that the NNRW with FWL learning for all the four models (Dumpala, Acir, Liu and Dingle) showed significant improvement of detection performance compared with the stand-alone NNRW (table 10). Thus, the average testing accuracy of the Liu model for NNRW increased 14.7% with FWL. Similarly, FWL increased the average testing accuracy by a factor of 31.8% for the Acir model. With FWL, they all seem to converge at around 70% accuracy, although the increment in performance is low for Dumpala and Dingle. Therefore, 74% accuracy is the highest performance as we can get in this study.

6.4 *Analysis of learned weights variation of every peak model*

The distributions of feature weights for 30 runs of every peak model are illustrated as boxplots in figure 6, so as to display variation in the samples without making any assumptions of the underlying statistical distributions. Overall, the boxplots show that use of FWL gave smaller and more symmetric variance as compared with the stand-

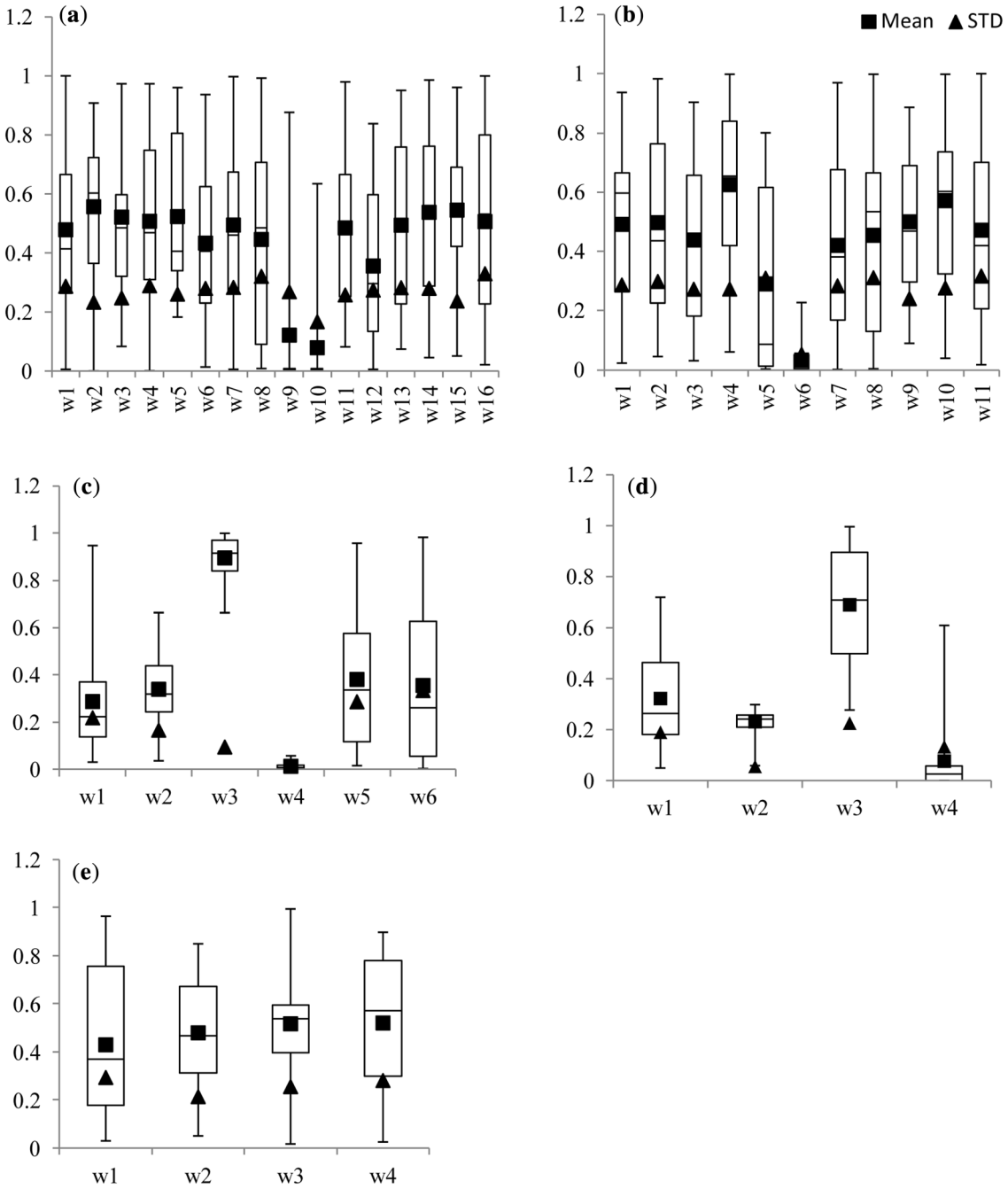

Figure 6. Learned weights variation of every peak model. (**a**) All features model, (**b**) Liu model, (**c**) Acir model, (**d**) Dumpala model and (**e**) Dingle model.

alone NNRW. This shows that the FWL successfully optimized the weighting of the associated features for getting the smaller variance of the models. The lower lying boxes in figure 6 correspond to the peak models containing some irrelevant or distracting features. The results in table 10 show clearly that a peak model containing some irrelevant features does not achieve the highest peak detection performance. For example, the most irrelevant features of the complete 16-feature model are f_9 and f_{10}, whereas f_9 proves irrelevant in the Liu model, versus f_7 and f_8 in the Acir model and f_6 and f_{14} in the Dumpala model. Comparing the Dingle and Dumpala models, all features of the Dingle model learned higher weight than did features of the Dumpala model. Therefore, we conclude that the Dingle model, with only four features, gives the most relevant available combination of peak features, resulting in the highest testing accuracy.

7. Conclusions and future works

In this study, we searched for the most relevant model for detecting peak in EEG signals, using an algorithm with FWL applied to an NNRW classifier. We obtained real EEG data of two channels responsive to frontal cortex from 20 healthy subjects who were instructed to direct their horizontal eye gaze upon command. The data were used to evaluate the performance of four established peak detection models as well as a complete 16-feature analysis. Our choice of four different peak models was based on the known peak features in EEG signals in the time domain analysis, and based upon their proven utility in various physiological signal applications [1, 2, 8, 11, 12]. In the experiments, various permutations of the 16 features were applied to a stand-alone NNRW, and an approach with FWL and NNRW. We evaluated the performances of NNRW with and without FWL for the various peak models, aiming to identify the approach giving the highest accuracy in the testing data set.

In general, the experimental results showed that the FWL-NNRW method performed better than stand-alone NNRW for all peak model inputs. The highest performance peak detection algorithm (Dingle) integrated with FWL-NNRW gave 74.1% accuracy. The highest classification rate of the minimum testing result (49.5%) was also obtained by the Dingle model, indicating a good generalized performance. On the other hand, we also analysed the learned weights variation of every peak model, aiming to give a physically deep understanding of relevant and irrelevant features. Based on the analysis of the learned weights variation, the most irrelevant features are from the group of peak width and slope, which are $f_6, f_7, f_8, f_9, f_{10}$ and f_{14}.

The utilization of the NNRW without the FWL classifier gave generally inferior accuracy of the peak detection performance. Results in table 9 show that the highest detection performance for stand-alone NNRW was 71.7%, and the highest of minimum accuracy was only 57.1%. Using the NNRW with FWL gave better performance, with an average testing accuracy at 74.1%. Based on these results, we will undertake comparisons of other variations of NNRW classifiers in the detection algorithm, aiming for still better peak detection performance. We feel that this study has significant implications for many applications. For example, the general peak detection problem is an issue in critical care medical diagnostics [4], human–machine interface (HMI), brain–computer interface (BCI) and harmonic detection in digital and audio signal processing. Our present case of EEG peak in response to a voluntary change of horizontal eye gaze direction might be useful for patients with locked-in syndrome or other disabilities for controlling the direction of computer cursor in BCI applications [32]. This approach might also be translatable for EEG-based command of the movement of a robotic arm or wheelchair in HMI applications [30, 33].

Acknowledgements

This research is supported by High Impact Research Fund (UM.C/HIR/MOHE/ENG/16 Account Code: D000016-16001) awarded by Ministry of Higher Education Malaysia to University of Malaya, and Research University Grant (Q.K130000.2643.10J98) by Universiti Teknologi Malaysia. The first author would like to thank the Ministry of Education Malaysia for supporting his study by awarding him a MyPhD scholarship.

References

[1] Acir N, Oztura I, Kuntalp M, Baklan B and Guzelis C 2005 Automatic detection of epileptiform events in EEG by a three-stage procedure based on artificial neural networks. *IEEE Trans. Biomed. Eng.* 52: 30–40. doi:10.1109/TBME.2004.839630

[2] Elgendi M, Norton I, Brearley M, Abbott D and Schuurmans D 2013 Systolic peak detection in acceleration photoplethysmograms measured from emergency responders in tropical conditions. *PLoS One* 8: e76585. doi:10.1371/journal.pone.0076585

[3] Adam A, Shapiai M I, Mohd Tumari M Z, Mohamad M S and Mubin M 2014 Feature selection and classifier parameters estimation for EEG signals peak detection using particle swarm optimization. *Sci. World J.* 2014: Article ID 973063. doi:10.1155/2014/973063.

[4] Fürbassa F, Hartmann M M, Halford J J, Koren J, Herta J,Gruber A, Baumgartner C and Kluge T 2015 Automatic detection of rhythmic and periodic patterns in critical care EEG based on American Clinical Neurophysiology Society (ACNS) standardized terminology. *Clin. Neurophysiol.* 45: 203–213

[5] Zander T O, Gaertner M, Kothe C and Vilimek R 2011 Combining eye gaze input with a brain–computer interface for touchless human–computer interaction. *Int. J. Hum. Comput. Interact.* 27(1): 38–51

[6] Nikolaev A R, Meghanathan R N and van Leeuwen C 2016 Combining EEG and eye movement recording in free viewing: pitfalls and possibilities. *Brain Cogn.* 107: 55–83

[7] Shaffer F, McCraty R and Zerr C L 2014 A healthy heart is not a metronome: an integrative review of the heart's anatomy and heart rate variability. *Front. Psychol.* 5: 1040. doi:10.3389/fpsyg.2014.01040

[8] Dumpala S R, Reddy S N and Sarna S K 1982 An algorithm for the detection of peaks in biological signals. *Comput. Programs Biomed.* 14: 249–256. doi:10.1016/0010-468X(82)90030-7

[9] Acir N 2005 Automated system for detection of epileptiform patterns in EEG by using a modified RBFN classifier. *Expert Syst. Appl.* 29: 455–462. doi:10.1016/j.eswa.2005.04.040

[10] Acir N and Guzelis C 2004 Automatic spike detection in EEG by a two-stage procedure based on support vector machines. *Comput. Biol. Med.* 34: 561–575. doi:10.1016/j.compbiomed.2003.08.003

[11] Liu H S, Zhang T and Yang F S 2002 A multistage, multimethod approach for automatic detection and classification

of epileptiform EEG. *IEEE Trans. Biomed. Eng.* 49: 1557–1566. doi:10.1109/TBME.2002.805477

[12] Dingle A A, Jones R D, Carroll G J and Fright W R 1993 A multistage system to detect epileptiform activity in the EEG. *IEEE Trans. Biomed. Eng*. 40(12): 1260–1268

[13] Lee K M and Street W N 2002 Incremental feature weight learning and its application to a shape-based query system. *Pattern Recognit. Lett.* 23(7): 865–874

[14] Deng Z H, Tang S W, Yang D Q, Zhang M, Li L Y and Xie K Q 2004 A comparative study on feature weight in text categorization. In: *Advanced web technologies and applications*, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, vol. 3007, pp. 588–597

[15] Giraldo L F, Delgado E and Castellanos C G 2007 Feature weighting and selection using a hybrid approach based on Rademacher complexity model selection. *Comput. Cardiol.* 34(2007): 257–260

[16] Xu Z B, Xuan J P, Shi T L, Wu B and Hu Y M 2009 Application of a modified fuzzy ARTMAP with feature-weight learning for the fault diagnosis of bearing. *Expert Syst. Appl.* 36(6): 9961–9968

[17] Gunes S, Polat K and Yosunkaya S 2010 Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Syst. Appl.* 37(12): 7922–7928

[18] Kira K and Rendell L A 1992 The feature selection problem: traditional methods and a new algorithm. In: *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 129–134

[19] Adam A, Ibrahim Z, Mokhtar N, Shapiai M I, Cumming P and Mubin M 2016 Evaluation of different time domain peak models using extreme learning machine-based peak detection for EEG signal. *SpringerPlus* 5: 1036. doi: 10.1186/s40064-016-2697-0

[20] Liu Y C, Lin C C, Tsai J J and Sun Y N 2013 Model-based spike detection of epileptic EEG data. *Sensors (Basel)* 13: 12536–12547. doi:10.3390/s130912536

[21] Lu W, Nystrom M M, Parikh P J, Fooshee D R, Hubenschmidt J P, Bradley J D and Low D A 2006 A semi-automatic method for peak and valley detection in free-breathing respiratory waveforms. *Med. Phys.* 33: 3634–3636. doi:10.1118/1.2348764

[22] Schmidt W F 1992 Feed forward neural networks with random weights. In: *Proceedings of the 11th IAPR International Conference on Pattern Recognition Methodology and Systems*, pp. 1–4

[23] Rao C R and Mit S K 1971 *Generalized inverse of matrices and its applications*. New York: Wiley

[24] Pao Y H, Park G H and Sobajic D J 1994 Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6: 163–180. doi:10.1016/0925-2312(94)90053-1

[25] Cao F L, Ye H L and Wang D H 2015 A probabilistic learning algorithm for robust modeling using neural networks with random weights. *Inf. Sci.* 313: 62–78. doi:10.1016/j.ins.2015.03.039

[26] Kennedy J and Eberhart R 1995 Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks (ICW)*, pp. 1942–1948

[27] Billauer E 2012 *peakdet: peak detection using MATLAB*

[28] Guo X, Yin Y, Dong C, Yang G and Zhou G 2008 On the class imbalance problem. In: *Proceedings of the Fourth International Conference on Natural Computation (ICNC 08)*, pp. 192–201

[29] Shi Y and Eberhart R C 1999 Empirical study of particle swarm optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1945–1950

[30] Ramli R, Arof H, Ibrahim F, Mokhtar N and Idris M Y I 2015 Using finite state machine and a hybrid of EEG signal and EOG artifacts for an asynchronous wheelchair navigation. *Expert Syst. Appl.* 42: 2451–2463. doi:10.1016/j.eswa.2014.10.052

[31] Klados M A, Papadelis C, Braun C and Bamidis P D 2011 REG-ICA: a hybrid methodology combining blind source separation and regression techniques for the rejection of ocular artifacts. *Biomed. Signal Process. Contr.* 6: 291–300. doi:10.1016/j.bspc.2011.02.001

[32] Belkacem A N, Hirose H, Yoshimura N, Shin D and Koike Y 2014 Classification of four eye directions from EEG signals for eye-movement-based communication systems. *J. Med. Biol. Eng.* 34: 581–588. doi:10.5405/jmbe.1596

[33] Postelnicu C C, Talaba D and Toma M I 2011 Controlling a robotic arm by brainwaves and eye movement. In: *Technological innovation for sustainability*, IFIP Advances in Information and Communication Technology, vol. 349, pp. 157–164