axioms

MDPI

Article

Hybrid Binary Particle Swarm Optimization Differential Evolution-Based Feature Selection for EMG Signals Classification

Jingwei Too [1,*], Abdul Rahim Abdullah [1,*] and Norhashimah Mohd Saad [2]

[1] Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[2] Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

* Correspondence: jamesjames868@gmail.com (J.T.); abdulr@utem.edu.my (A.R.A.); Tel.: +60-178-845-938 (J.T.)

Received: 21 May 2019; Accepted: 3 July 2019; Published: 5 July 2019

Abstract: To date, the usage of electromyography (EMG) signals in myoelectric prosthetics allows patients to recover functional rehabilitation of their upper limbs. However, the increment in the number of EMG features has been shown to have a great impact on performance degradation. Therefore, feature selection is an essential step to enhance classification performance and reduce the complexity of the classifier. In this paper, a hybrid method, namely, binary particle swarm optimization differential evolution (BPSODE) was proposed to tackle feature selection problems in EMG signals classification. The performance of BPSODE was validated using the EMG signals of 10 healthy subjects acquired from a publicly accessible EMG database. First, discrete wavelet transform was applied to decompose the signals into wavelet coefficients. The features were then extracted from each coefficient and formed into the feature vector. Afterward, BPSODE was used to evaluate the most informative feature subset. To examine the effectiveness of the proposed method, four state-of-the-art feature selection methods were used for comparison. The parameters, including accuracy, feature selection ratio, precision, *F*-measure, and computation time were used for performance measurement. Our results showed that BPSODE was superior, in not only offering a high classification performance, but also in having the smallest feature size. From the empirical results, it can be inferred that BPSODE-based feature selection is useful for EMG signals classification.

Keywords: electromyography; discrete wavelet transform; binary particle swarm optimization; binary differential evolution; feature selection; hybrid optimization; classification

1. Introduction

Electromyography (EMG) is a biomedical signal that records electric potential when there is a muscle contraction. Recently, the usefulness of EMG as a control source of myoelectric prosthetics has received much attention from biomedical researchers. The recognition of hand movements enables the application of multi-functional myoelectric prosthetics in engineering, rehabilitation, and clinical areas. However, myoelectric control is still limited by inadequate control techniques [1]. In addition, EMG signals are easily influenced by noise due to the fact of its complex nature [2]. Therefore, most researchers apply advanced signal processing, feature extraction, and feature selection techniques to extract only the useful information from the signal.

In previous studies, discrete wavelet transform (DWT) was found to be the most frequently used signal processing method due to its effectiveness in the analysis of EMG signals [3,4]. Intuitively, DWT offers the optimal time–frequency resolution by decomposing the signal into multi-resolution coefficients. However, by applying DWT, the number of extracting features is greatly increased [5]. This will not only improve the complexity of the recognition system, but also degrade the classification performance. In fact, it is difficult to identify which feature is optimal, as well as the best feature combination for producing the optimal classification result. Therefore, the feature selection technique is required to solve the feature selection problem.

Naturally, a feature set is made up of relevant, irrelevant, and redundant features. A relevant feature is able to enhance the prediction accuracy, whereas an irrelevant or redundant feature might be reducing the performance of system [6]. Feature selection attempts to select a subset of relevant features from a large available feature set. It is not only minimizing the number of features, but also evolving the performance of the system. In general, feature selection can be categorized into filter and wrapper approaches. The filter approach is independent of the learning algorithm. It makes use of statistical and mutual information for searching the potential features [7]. Unlike the filter approach, the wrapper approach applies a specific learning algorithm (classifier) to evaluate the best feature subset. Compared to the filter approach, the wrapper approach can often contribute to a better performance. Hence, wrapper approaches are widely used in feature selection problems [6,8–10].

In a previous study, Ahmed et al. [9] proposed a differential evolution (DE) with a wheel-based strategy to identify the subset of relevant features. Banka and Dara [11] proposed the hamming distance-based binary particle swarm optimization (HDBPSO) to solve the high-dimensional feature selection problem. The authors applied the hamming distance for velocity update, and the result obtained indicated that HDBPSO was superior to the genetic algorithm (GA) and the non-dominant sorting genetic algorithm (NSGA II). Later, Bharti and Singh [12] implemented the chaotic maps and opposition learning strategy to enhance the performance of BPSO. In addition, the authors proposed the fitness-based dynamic inertia weight strategy to control the value of inertia weight on particles. Furthermore, Zorarpaci and Ozel [13] developed a hybrid differential evolution and artificial bee colony (DEABC) for feature selection. The authors revealed that DEABC outperformed binary differential evolution (BDE) and artificial bee colony (ABC) in choosing the significant features. Another study [14] proposed the co-evolution binary particle swarm optimization with a multiple inertia weight strategy (CBPSO-MIWS) for feature selection. Previous works have shown the impact of feature selection before classification procedure.

According to the literature, conventional feature selection methods such as particle swarm optimization (PSO) and DE have the limitations of premature convergence and early stagnation. Therefore, different hybrid methods of PSO and DE have been developed for performance enhancement. However, most of them are designed to solve continuous optimization and numerical problems, which are different than feature selection problems [15,16]. Therefore, this study aimed to propose a hybrid version of binary particle swarm optimization differential evolution (BPSODE) for tackling the feature selection problem in EMG signals classification. The proposed BPSODE is the hybridization of binary particle swarm optimization (BPSO) and binary differential evolution (BDE). It not only inherits the advantages of BPSO and BDE in local and global search but is also good in escaping the local solution. In the proposed BPSODE, the BPSO and BDE algorithms are computed in sequence, and thus, no extra computation cost is required. Moreover, the dynamic inertia weight and dynamic crossover rate are introduced in BPSODE for improving the performance of the algorithm. The performance of BPSODE was validated using EMG data collected from 10 healthy subjects. To evaluate the effectiveness of the proposed method, the binary bat algorithm (BBA) [17], binary flower pollination algorithm (BFPA) [18], BPSO [14], and BDE [13] were used for performance comparison. Our experimental results showed that BPSODE outperformed the other algorithms in feature selection.

The organization of paper is as follows: Section 2 details the binary particle swarm optimization and binary differential evolution. Section 3 describes the proposed EMG pattern recognition system and the hybrid binary particle swarm optimization differential evolution algorithm. Section 4 discusses the experimental results and the conclusion is summarized in Section 5.

2. Preliminary

2.1. Binary Particle Swarm Optimization

Binary particle swarm optimization (BPSO) was first proposed by Kennedy and Eberhart [19] to solve binary optimization problems. In BPSO, the population is known as a swarm, which comprises N particles that flow through the multidimensional search space. The particle represents the potential solution, and it moves through the search space to seek out the best solution. Each particle searches for the global maximum or minimum according to its own experience and knowledge [20].

For a D dimensional problem, the velocity of the particle is expressed as $V = (v_{i1}, v_{i2}, \ldots, v_{iD})$ and the position of the particle is denoted as $X = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where i represents the order of the particle in the population. In BPSO, the optimal location of each particle is known as *Pbest* and the global best solution in the population is called *Gbest*. For each iteration t, the particle updates its velocity as follow:

$$v_i^d(t+1) = w(t) \times v_i^d(t) + c_1 \times r_1 \times \left(Pbest_i^d(t) - x_i{}^d(t)\right) + c_2 \times r_2 \times \left(Gbest^d(t) - x_i{}^d(t)\right) \tag{1}$$

where x is the position of the particle, v denotes the velocity of the particle, i is the order of the particle in the population, d is the dimension of the search space, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are the two independent random numbers uniformly distributed between 0 and 1. Then, the velocity is converted into a probability value using the sigmoid function as follow:

$$S\left(v_i^d(t+1)\right) = \frac{1}{1 + e^{-v_i^d(t+1)}} \tag{2}$$

Afterward, the position of the particle is updated as:

$$x_i^d(t+1) = \begin{cases} 1, & \text{If } \delta < S\left(v_i^d(t+1)\right) \\ 0, & \text{Otherwise} \end{cases} \tag{3}$$

where δ is a random number uniformly distributed between 0 and 1.

In BPSO, an inertia weight is gradually decreased from a higher to a lower value in order to ensure a well and stable balance between global and local exploration [21]. At each iteration, the inertia weight is computed as:

$$w(t) = w_{\max} - (w_{\max} - w_{\min})\frac{t}{T} \tag{4}$$

where $w_{\max}$ and $w_{\min}$ are the bounds on the inertia weight, t is the current iteration, and T is the maximum number of iterations. In this study, $w_{\max}$ and $w_{\min}$ were set to 0.9 and 0.4, respectively.

2.2. Binary Differential Evolution

Differential evolution (DE) is an evolutionary heuristic approach proposed by Storn and Price [22] to minimize the non-linear and continuous function. Originally, DE was designed to solve the continuous value problem; for feature selection, the DE is modified into binary differential evolution (BDE) according to Reference [13]. Binary differential evolution is a simple, direct use, and efficient feature selection method. It is composed of three main operators, which are mutation, crossover, and selection.

Firstly, BDE generates an initial population for a D dimensional problem randomly, where D is the number of features that need to be optimized. During the mutation stage, three random vectors x_{r1}, x_{r2}, and x_{r3} are randomly selected from the population for vector x_i. Note that $r_1 \neq r_2 \neq r_3 \neq i$. Then, the difference vector is computed as follow:

$$difference\ vector_i^d = \begin{cases} 0 & \text{If } x_{r1}^d = x_{r2}^d \\ x_{r1}^d & \text{Otherwise} \end{cases} \tag{5}$$

where i is the order of the vector in the population and d is the dimension of the vector. If the dth dimension of x_{r1} is equal to x_{r2}, then the difference vector will become 0. Otherwise, the differential vector will become the same as x_{r1}. Next, the mutation is performed as shown in Equation (6).

$$mutant\ vector_i^d = \begin{cases} 1 & \text{If } difference\ vector_i^d = 1 \\ x_{r3}^d & \text{Otherwise} \end{cases} \tag{6}$$

After that, the crossover process is executed as follow:

$$u_i^d = \begin{cases} mutant\ vector_i^d & \text{If } \delta \leq CR(t) \| \ d = d_{rand} \\ x_i^d & \text{Otherwise} \end{cases} \tag{7}$$

where u is the trial vector, x is the vector, d is the dimension of search space, $CR \in (0,1)$ is the crossover rate, d_{rand} is a random feature index distributed between 1 and D, and δ is a random number distributed between 0 and 1.

For the selection process, if the fitness value of the trial vector is better, then the current vector will be replaced. Otherwise, the current vector is kept for the next generation.

3. Materials and Methods

Figure 1 illustrates the flow diagram of the proposed EMG pattern recognition system. In the first step, the EMG data are acquired from the publicly accessible EMG database. Next, the discrete wavelet transform (DWT) is applied to decompose the EMG signals into multi-resolution coefficients. Then, the features are extracted from the wavelet coefficients and form the feature vector. After that, five feature selection methods including BBA, BDE, BFPA, BPSO, and BPSODE are used to evaluate the optimal feature subset. In the final step, the k-nearest neighbor (KNN) is employed for the classification process.

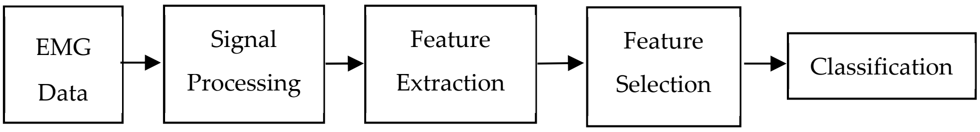

Figure 1. Flow diagram of the proposed electromyography (EMG) pattern recognition system.

3.1. EMG Data

The Non-Invasive Adaptive Prosthetics (NinaPro) project [23] is a publicly accessible EMG database that has previously been applied in EMG pattern recognition studies. In this study, the NinaPro database 4 (DB4), composed of the EMG signals of twelve different hand movement types (Exercise A), was utilized. The twelve hand movement types included index flexion, index extension, middle flexion, middle extension, ring flexion, ring extension, little finger flexion, little finger extension, thumb adduction, thumb abduction, thumb flexion, and thumb extension [24]. The DB4 contained the EMG data of 10 healthy subjects. In the experiment, 12 electrodes (12 channels) were implemented. The subjects were instructed to perform each movement type for 5 s, followed by a resting state of 3 s. In addition, each movement type was repeated six times, and the EMG signal was sampled at the rate of 2000 Hz [24]. Note that all resting states were removed before any further processing.

3.2. Discrete Wavelet Transform-Based Feature Extraction

Recently, discrete wavelet transform (DWT) has shown its potential and capability in biomedical signal processing. Discrete wavelet transform has the advantage of varying the time and frequency window, which can provide an optimal time–frequency resolution in EMG pattern recognition [25].

Basically, DWT decomposes the EMG signal into multi-resolution by filtering the signal with a high-pass filter, $h(n)$ and low-pass filter, $h(n)$. The first decomposition of DWT can be expressed as:

$$y_{high}(k) = \sum_n x(n) \cdot h(2k - n) \tag{8}$$

$$y_{low}(k) = \sum_n x(n) \cdot g(2k - n) \tag{9}$$

where $x(n)$ is the input EMG signal, and $y_{high}(k)$ and $y_{low}(k)$ represent the detail and approximation, respectively. In wavelet decomposition, detail (D) exhibits the signal at high frequency, whereas the low-frequency component is represented by the approximation (A) [26]. Previous works indicated that the selection of the mother wavelet and decomposition level were the main factors that can strongly affect the performance of DWT in EMG pattern recognition. According to the finding of Reference [27], DWT at the fourth decomposition level was employed in this work. An illustration of DWT is displayed in Figure 2.

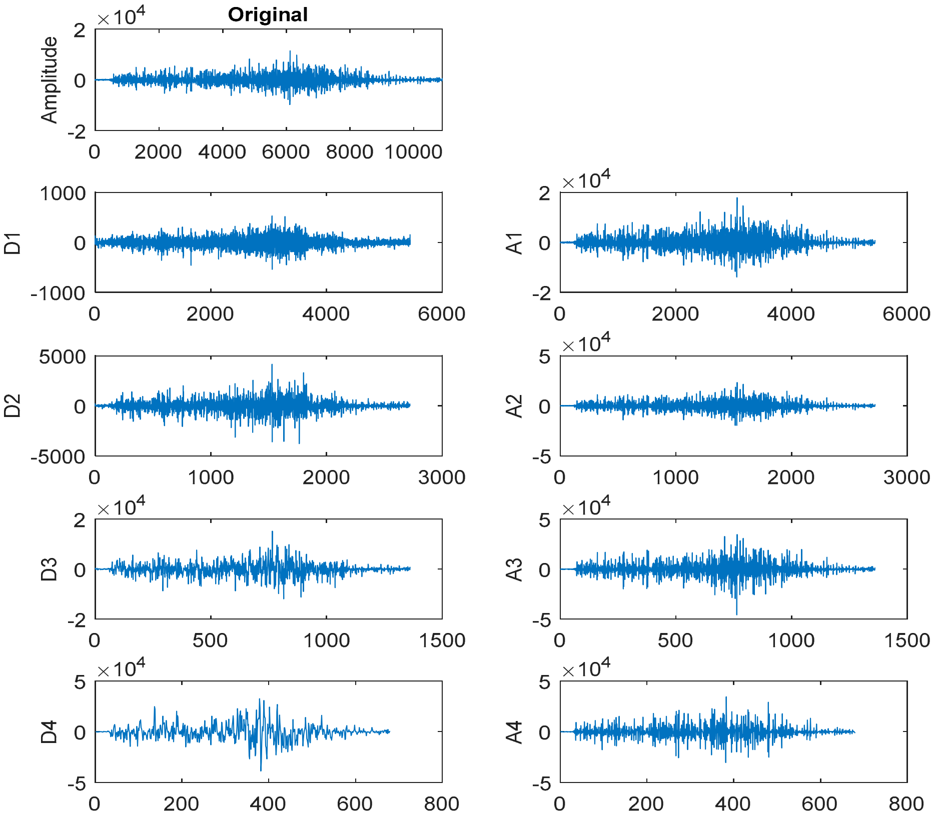

Figure 2. The illustration of the discrete wavelet transform (DWT) at fourth decomposition level.

As for the mother wavelet selection, twelve mother wavelets including db4, db6, db8, sym4, sym6, sym8, bior2.2, bior3.3, bior4.4, coif3, coif4, and coif5 are investigated. From the experiment, we found that DWT with bior4.4 offered the optimal performance in the current work. Hence, only DWT with bior4.4 at the fourth decomposition level was applied in the rest of this paper.

In this work, five popular features, namely, mean absolute value (MAV), wavelength (WL), zero crossing (ZC), slope sign change (SSC), and maximum fractal length (MFL) were extracted from each wavelet coefficient to form the feature set. These features were selected due to their promising performances in previous works [3,4,28].

3.3. Proposed Hybrid Binary Particle Swarm Optimization Differential Evolution

In this paper, a hybrid binary particle swarm optimization differential evolution method (BPSODE) that combines the superior capability of BPSO and BDE algorithms is proposed to solve the feature selection problem in EMG signals classification. In the proposed BPSODE, the BPSO and BDE algorithms are computed in sequence. For example, BPSO is computed in the first, third, and fifth iterations, whereas the second, fourth, and sixth iterations are performed by BDE. In this way, BPSODE can fully take the advantages of BPSO and BDE without the additional computation cost. However, both BPSO and BDE have the limitations of premature convergence. To prevent the BPSODE from being trapped in the local optima, two simple schemes are introduced. The first scheme is dynamic inertia weight, which enables BPSODE to track the optimal solution in dynamic environment. The second scheme is the dynamic crossover rate. Instead of using a fixed crossover rate, a dynamic crossover rate is more capable of balancing the exploration and exploitation.

3.3.1. Dynamic Inertia Weight

The inertia weight is a parameter proposed by Shi and Eberhart [29] to enhance the performance of PSO. Generally, a larger inertia weight leads to good global exploration. On the contrary, a smaller inertia weight tends to promote the local exploration around the best solution [30]. In BPSO, the inertia weight linearly decreases from 0.9 to 0.4 for balancing the global and local exploration. However, in the experiment, we found that such a mechanism did not work very well in BPSODE. Thus, we applied a dynamic inertia weight as shown in Equation (10).

$$w(t) = 0.5 + \frac{r_3}{2} \tag{10}$$

where r_3 is a random number distributed between 0 and 1. Figure 3a illustrates an example of dynamic inertia weight. As can be seen, the inertia weight was generated uniformly between 0.5 and 1. Since it is difficult to estimate the exploration and exploitation stage, a random inertia weight is more appropriate to be used in this dynamic environment [31].

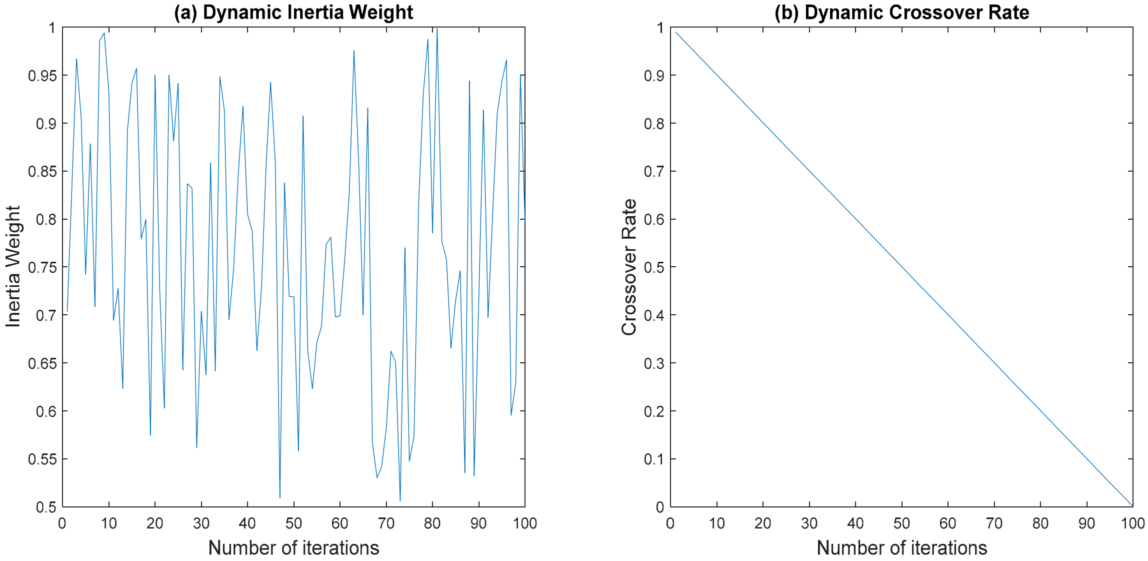

Figure 3. Samples of two schemes: (**a**) dynamic inertia weight and (**b**) dynamic crossover rate.

3.3.2. Dynamic Crossover Rate

The crossover rate (*CR*) is a parameter introduced in BDE. It controls the number of *d* dimension parameter values copied from the mutant vector [32]. A higher value of *CR* indicates more parameters are duplicated from the mutant vector. By contrast, a lower CR means less parameters are reproduced from the mutant vector. In BPSODE, the dynamic CR is proposed as shown in Equation (11).

$$CR(t) = 1 - \left(\frac{t}{T}\right) \tag{11}$$

where t is the current iteration and T is the maximum number of iterations. An example of dynamic crossover rate is exhibited in Figure 3b. One can see that the crossover rate was reduced from 1 to 0 as the number of iterations increased. At the beginning, a higher CR ensured more parameters were reproduced to improve the exploration (global search). As time passed, a lower CR guaranteed the exploitation process (local search).

Algorithm 1 demonstrates the pseudocode of BPSODE. Initially, the position of particles is randomly initialized in binary form (bit 1 or 0). The velocity of particles is initialized to zero. Next, the fitness of each particle is evaluated, and the *Pbest* and *Gbest* are defined. Then, BPSO (iteration with odd number) and BDE (iteration with even number) algorithms are computed in sequence. For the iteration with odd number, the inertia weight is updated as shown in Equation (10). Afterward, the position and velocity of particles are updated using Equations (1) and (3), respectively. Then, the fitness of each particle is evaluated. As for the iteration with even number, the crossover rate is updated as shown in Equation (11). After that, the mutation and crossover operations are computed as shown in Equations (6) and (7), respectively. From the mutation and crossover, the trial vector is generated. The fitness of newly generated trial vector is then evaluated and compared with current particle. If the trial vector results in better fitness, then the current particle will be replaced; otherwise, the current particle is kept for the next iteration. At the end of each iteration, the *Pbest* and *Gbest* are updated. The algorithm is repeated until the termination criteria (maximum number of iterations) is satisfied. At last, the global best solution is pointed out.

Algorithm 1. Hybrid Binary Particle Swarm Optimization Differential Evolution

```
Input Parameters: $N$, $T$, $c_1$, and $c_2$
Randomly initialize a population of particles, $x$
Evaluate the fitness of particles, $F(x)$
Set Pbest and Gbest
for $t$ = 1 to maximum number of iterations, $T$
        // BPSO Algorithm //
    if mod($t$,2) = 1
        $w = 0.5 + \frac{rand(0,1)}{2}$
        for $i$ = 1 to number of particles, $N$
            for $d$ = 1 to number of dimension, $D$
                $v_i^d(t+1) = w \times v_i^d(t) + c_1 \times r_1 \times \left(Pbest_i^d(t) - x_i^d(t)\right) + c_2 \times r_2 \times \left(Gbest^d(t) - x_i^d(t)\right)$
                $S\left(v_i^d(t+1)\right) = \frac{1}{1+exp\left(-v_i^d(t+1)\right)}$
                if $rand(0,1) \leq S\left(v_i^d(t+1)\right)$
                    $x_i^d(t+1) = 1$
                else
                    $x_i^d(t+1) = 0$
                end if
            end for
            Evaluate the fitness of new particle, $F(x_i(t+1))$
        end for
        // BDE Algorithm //
    else
        $CR = 1 - \left(\frac{t}{T}\right)$
        for $i$ = 1 to number of particles, $N$
            Random select vectors $x_{r1}, x_{r2}, x_{r3}$ and $d_{rand} = rand(1, D)$
            for $d$ = 1 to number of dimension, $D$
                if $x_{r1}^d = x_{r2}^d$
                    $difference\ vector_i^d = 0$
                else
                    $difference\ vector_i^d = x_{r1}^d$
                end if
                if $difference\ vector_i^d = 1$
                    $mutant\ vector_i^d = 1$
                else
                    $mutant\ vector_i^d = x_{r3}^d$
                end if
                if $rand(0,1) \leq CR\ or\ d = d_{rand}$
                    $u_i^d = mutant\ vector_i^d$
                else
                    $u_i^d = x_i^d(t)$
                end if
            end for
            Evaluate the fitness of trial vector, $F(u_i)$
            Perform greedy selection between current particle and trial vector
        end for
    end if
        // Pbest and Gbest Update //
    for $i$ = 1 to number of particles, $N$
        Update $Pbest_i$ and Gbest
    end for
end for
Output: Global best solution
```

3.4. Application of BPSODE for Feature Selection

In BPSODE, the position of the particle is expressed in binary form; either bit value 1 or 0, where bit 1 and bit 0 represent the selected feature and non-selected feature, respectively. For example, given a solution X= {0,1,1,0,0,1,1,0,0,0}, it shows that four features (2nd, 3rd, 6th, and 7th features) are selected.

As for the wrapper feature selection, the fitness function that maximizes the classification performance and minimizes the number of features is utilized, and it can be defined as:

$$\downarrow Fitness = \alpha E_R + (1 - \alpha)\frac{|R|}{|S|} \tag{12}$$

$$E_R = \frac{No.of\ wrongly\ predicted\ instances}{Total\ number\ of\ instances} \tag{13}$$

where E_R is the error rate computed by a learning algorithm, $|R|$ is the length of the feature subset, $|S|$ is the total number of features, and α is the parameter that control the weight between error rate and ratio of selected features. Considering the classification performance to be the most important measurement, the α was set to 0.9 in this work.

For fitness evaluation, the k-nearest neighbor (KNN) with a Euclidean distance and k = 1 was used as the learning algorithm. The KNN was chosen because it is a common, fast, and simple machine learning algorithm that has been widely applied in feature selection studies [33,34]. For performance evaluation, the 10-fold cross validation method was implemented. In this scheme, the data were randomly divided into 10 equal parts. Each part took turns testing while the remaining parts (nine parts) were used for the training set. The results obtained from the 10 folds were then averaged and recorded.

4. Results and Discussions

The EMG signals of the 10 subjects were gathered from the NinaPro database 4, comprising 10 different datasets. In the next step, DWT was applied to decompose the EMG signals into multi-resolution coefficients. It is worth noting that DWT produced eight coefficients (four details and four approximations) at the fourth decomposition level. Then, five features were extracted from each wavelet coefficient, and the feature set was formed. In total, 480 features (12 channels × 5 features × 8 coefficients) were extracted from each movement from each subject. On the other hand, 72 instances (12 hand movement types × 6 repetitions) were acquired from each subject. As a result, for each subject (dataset), a feature vector with a matrix of 72 × 480 was formed. In order to prevent numerical problems, the features were normalized between 0 and 1. Afterward, the feature selection algorithms were used to select the most informative feature subset. At last, the selected features were then fed into the KNN for the classification of twelve different hand movement types (12 classes). The classification process is critically important because it shows how accurate a myoelectric prosthetic can be. In a nutshell, a myoelectric prosthetic with higher accuracy allows the users to perform the hand movement types accurately.

4.1. Comparison Algorithms and Evaluation Metrics

In this study, five feature selection algorithms including BBA [17], BFPA [18], BPSO [14], BDE [13], and BPSODE were used to evaluate the best feature subset (best combination of features and channels). The specific parameter setting of utilized algorithms are given in Table 1. To ensure fair comparison, the maximum number of iterations (T) was fixed at 100. On the one hand, the population size (N) was chosen at 80. Note that the analysis for the selection of the population size will be discussed in Section 4.2.1. All analyses were conducted in MATLAB 9.3 using a computer with an Intel Core i5-9400F CPU 2.90 GHz and 16.0 GB RAM.

Table 1. Parameter settings of the hybrid binary particle swarm optimization differential evolution (BPSODE), binary bat algorithm (BBA), binary differential evolution (BDE), binary particle swarm optimization (BPSO) and binary flower pollination algorithm (BFPA).

Algorithm	Parameter	Value
BPSODE	Acceleration coefficient, c_1 and c_2	2
	Bound on velocity	(−6,6)
BDE	Crossover rate, CR	1
BPSO	Acceleration coefficient, c_1 and c_2	2
	Inertia weight, w	0.9–0.4
	Bound on velocity	(−6,6)
BFPA	Switch probability, P	0.8
	Levy component, λ	1.5
BBA	Maximum frequency, f_{max}	2
	Minimum frequency, f_{min}	0
	Control coefficient, α and γ	0.9
	Loudness, A	(1,2)
	Pulse rate, r	(0,1)

To evaluate the effectiveness of the proposed method, four statistical metrics including accuracy, feature selection ratio (FSR), precision, and F-measure were calculated, and they are defined as follows [35–37]:

$$\text{Accuracy} = \frac{No.of\ correctly\ classified\ instances}{Total\ number\ of\ instances} \times 100 \tag{14}$$

$$\text{FSR} = \frac{|R|}{|S|} \tag{15}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$F\text{-measure} = \frac{2TP}{2TP + FP + FN} \tag{17}$$

where $|R|$ is the length of the feature subset, $|S|$ is the total number of features, TP is the true positive, FP is the false positive, and FN is the false negative. To obtain the statistical results, each algorithm (i.e., BBA, BPSO, BDE, BFPA, and BPSODE) was executed for 20 independent runs. Then, the averaged results obtained from 20 runs were recorded for performance comparison.

4.2. Experimental Results and Analysis

4.2.1. Effect of Population Size

In the first part of the experiment, we studied the effect of population size. Briefly, population size is one of the key factors that can strongly affect the performance of BPSODE in feature selection. A higher population size can usually offer better performance; however, more computation time is required [38]. In this paper, five different population sizes (i.e., 20, 40, 60, 80 and 100) were investigated. Figure 4 illustrates the boxplot of BPSODE with five different population sizes across 10 subjects. We used accuracy as the evaluation metric since it is the most important measurement in this work. From Figure 4, it is noted that the optimal result was seen with the population size of 80. In comparison with other population sizes, the population size of 80 contributed to the highest median value (red line in the box) of 92.64%. The result shows that the population size of 80 overwhelmed its competitors in the current work. Thereafter, only the population size of 80 was applied in the rest of this paper.

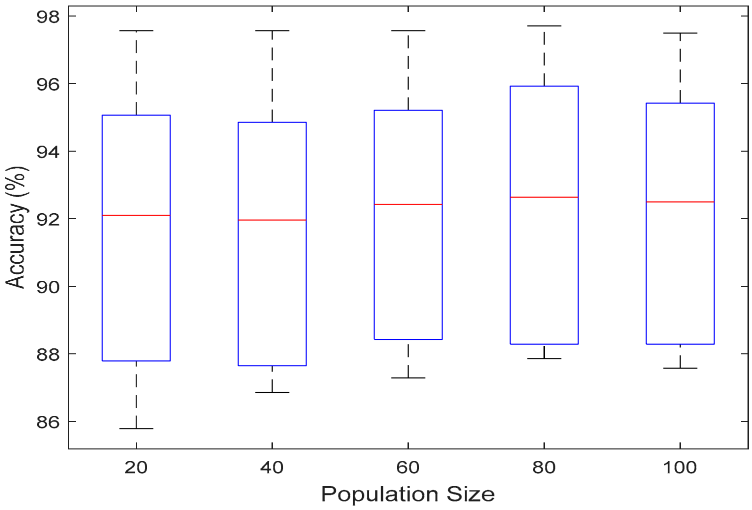

Figure 4. Boxplot of BPSODE with five different population sizes across 10 subjects.

4.2.2. Comparison Results

In the second part of the experiment, we examined the efficacy of BPSODE by comparing its performance with BBA, BDE, BPSO, and BFPA. Figure 5 illustrates the classification performance of five different feature selection methods on 10 subjects (detailed results on accuracy can be found in Table 2). As can be seen, the accuracies achieved by BDE and BPSO were relatively poor. This result highlights that the features selected by BDE and BPSO might contain redundant and irrelevant information, which caused them to be trapped in the local optima at the early stagnation.

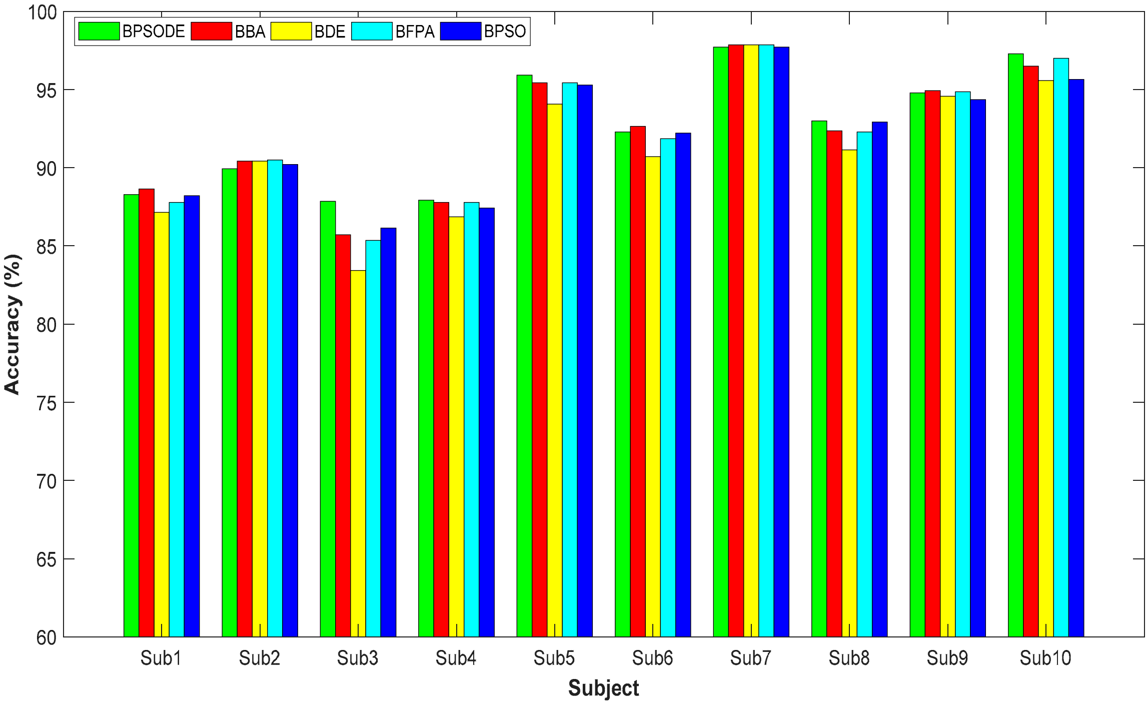

Figure 5. Accuracy of the five different feature selection methods on 10 subjects.

Table 2. Experimental results of five different feature selection methods on 10 subjects. The best result of each metric is highlighted in bold text.

Subject	Metrics	Feature Selection Method				
		BPSODE	**BBA**	**BDE**	**BFPA**	**BPSO**
1	Accuracy (%)	88.29 ± 1.58	**88.64 ± 1.64**	87.14 ± 1.67	87.79 ± 1.70	88.21 ± 1.60
	Feature selection ratio (FSR)	**0.4196 ± 0.0283**	0.4462 ± 0.0260	0.4926 ± 0.0329	0.5525 ± 0.0420	0.4486 ± 0.0250
	Precision	0.9034 ± 0.0108	**0.9049 ± 0.0128**	0.8941 ± 0.0156	0.9001 ± 0.0130	0.9023 ± 0.0120
	F-measure	0.8852 ± 0.0148	**0.8880 ± 0.0165**	0.8732 ± 0.0180	0.8798 ± 0.0169	0.8843 ± 0.0152
2	Accuracy (%)	89.93 ± 1.35	90.43 ± 1.54	90.43 ± 1.24	**90.50 ± 1.25**	90.21 ± 1.16
	FSR	**0.4079 ± 0.0324**	0.4424 ± 0.0167	0.4920 ± 0.0342	0.5594 ± 0.0550	0.4467 ± 0.0184
	Precision	0.9143 ± 0.0098	0.9182 ± 0.0132	0.9169 ± 0.0101	**0.9190 ± 0.0106**	0.9153 ± 0.0108
	F-measure	0.9018 ± 0.0120	0.9063 ± 0.0146	0.9061 ± 0.0111	**0.9069 ± 0.0115**	0.9042 ± 0.0105
3	Accuracy (%)	**87.86 ± 3.09**	85.71 ± 2.22	83.43 ± 1.70	85.36 ± 1.60	86.14 ± 1.68
	FSR	**0.4386 ± 0.0289**	0.4528 ± 0.0210	0.4981 ± 0.0489	0.5811 ± 0.0488	0.4636 ± 0.0185
	Precision	**0.8949 ± 0.0303**	0.8727 ± 0.0245	0.8519 ± 0.0181	0.8709 ± 0.0188	0.8779 ± 0.0186
	F-measure	**0.8810 ± 0.0326**	0.8588 ± 0.0227	0.8359 ± 0.0168	0.8555 ± 0.0160	0.8632 ± 0.0182
4	Accuracy (%)	**87.93 ± 1.35**	87.79 ± 1.43	86.86 ± 1.89	87.79 ± 1.35	87.43 ± 1.36
	FSR	**0.4196 ± 0.0343**	0.4393 ± 0.0207	0.4859 ± 0.0233	0.5581 ± 0.0523	0.4444 ± 0.0179
	Precision	**0.8891 ± 0.0139**	0.8870 ± 0.0146	0.8782 ± 0.0205	0.8880 ± 0.0136	0.8854 ± 0.0136
	F-measure	**0.8754 ± 0.0135**	0.8736 ± 0.0146	0.8653 ± 0.0194	0.8742 ± 0.0140	0.8705 ± 0.0136
5	Accuracy (%)	**95.93 ± 1.41**	95.43 ± 1.19	94.07 ± 1.49	95.43 ± 0.99	95.29 ± 1.32
	FSR	**0.4157 ± 0.0318**	0.4333 ± 0.0189	0.4820 ± 0.0168	0.5546 ± 0.0402	0.4426 ± 0.0221
	Precision	**0.9654 ± 0.0103**	0.9603 ± 0.0101	0.9505 ± 0.0121	0.9607 ± 0.0086	0.9593 ± 0.0111
	F-measure	**0.9593 ± 0.0141**	0.9544 ± 0.0119	0.9411 ± 0.0144	0.9546 ± 0.0099	0.9525 ± 0.0134
6	Accuracy (%)	92.29 ± 1.42	**92.64 ± 1.56**	90.71 ± 1.50	91.86 ± 1.24	92.21 ± 1.57
	FSR	**0.4225 ± 0.0295**	0.4436 ± 0.0216	0.4876 ± 0.0176	0.5663 ± 0.0434	0.4507 ± 0.0158
	Precision	0.9355 ± 0.0113	**0.9389 ± 0.0121**	0.9234 ± 0.0098	0.9338 ± 0.0087	0.9351 ± 0.0121
	F-measure	0.9258 ± 0.0136	**0.9289 ± 0.0147**	0.9111 ± 0.0152	0.9215 ± 0.0123	0.9251 ± 0.0151
7	Accuracy (%)	97.71 ± 0.97	**97.86 ± 0.87**	**97.86 ± 0.98**	**97.86 ± 0.98**	97.71 ± 1.17
	FSR	**0.3824 ± 0.0361**	0.4031 ± 0.0165	0.4627 ± 0.0111	0.4717 ± 0.0348	0.4149 ± 0.0200
	Precision	0.9788 ± 0.0087	0.9798 ± 0.0079	**0.9799 ± 0.0092**	**0.9799 ± 0.0092**	0.9789 ± 0.0102
	F-measure	0.9774 ± 0.0098	0.9785 ± 0.0092	**0.9786 ± 0.0104**	**0.9786 ± 0.0104**	0.9772 ± 0.0121
8	Accuracy (%)	**93.00 ± 1.60**	92.36 ± 1.33	91.14 ± 1.44	92.29 ± 1.34	92.93 ± 1.43
	FSR	**0.4426 ± 0.0285**	0.4536 ± 0.0272	0.5169 ± 0.0522	0.5676 ± 0.0485	0.4593 ± 0.0202
	Precision	**0.9338 ± 0.0143**	0.9278 ± 0.0125	0.9164 ± 0.0132	0.9268 ± 0.0121	0.9323 ± 0.0131
	F-measure	**0.9295 ± 0.0160**	0.9229 ± 0.0133	0.9110 ± 0.0143	0.9225 ± 0.0135	0.9287 ± 0.0141
9	Accuracy (%)	94.79 ± 1.25	**94.93 ± 1.57**	94.57 ± 1.94	94.86 ± 2.24	94.36 ± 1.64
	FSR	**0.4065 ± 0.0411**	0.4139 ± 0.0203	0.4813 ± 0.0270	0.5217 ± 0.0557	0.4332 ± 0.0207
	Precision	0.9563 ± 0.0097	**0.9575 ± 0.0115**	0.9543 ± 0.0161	0.9566 ± 0.0180	0.9518 ± 0.0143
	F-measure	0.9472 ± 0.0132	**0.9483 ± 0.0165**	0.9453 ± 0.0197	0.9478 ± 0.0235	0.9428 ± 0.0172
10	Accuracy (%)	**97.29 ± 1.13**	96.5 ± 1.64	95.57 ± 1.38	97.00 ± 0.92	95.64 ± 1.35
	FSR	**0.4214 ± 0.0380**	0.4343 ± 0.0177	0.4985 ± 0.0433	0.5785 ± 0.0314	0.4408 ± 0.0199
	Precision	**0.9758 ± 0.0101**	0.9680 ± 0.0154	0.9598 ± 0.0132	0.9731 ± 0.0087	0.9612 ± 0.0117
	F-measure	**0.9731 ± 0.0114**	0.9652 ± 0.0167	0.9561 ± 0.0142	0.9705 ± 0.0094	0.9570 ± 0.0129

From Figure 5, it can be seen that BPSODE reached the highest accuracy in most cases (five out of ten subjects). Especially for Subject 3, a great increment of 4.43% accuracy was found as compared to BDE. Based on the results obtained, the best feature selection method was found to be BPSODE, followed by BBA. On average across 10 subjects, the experimental result showed that BPSODE overtook other the algorithms with the highest mean accuracy of 92.5%. Obviously, BPSODE has proven its capability in effectively searching for significant features in the feature space.

Table 2 demonstrates the experimental results of accuracy, feature selection ratio (FSR), precision, and *F*-measure of the five different feature selection methods on 10 subjects. In this table, the best result of each metric is highlighted in bold text. A higher FSR means that more features are selected, while a lower FSR indicates less feature are selected by the algorithm. On the one hand, the higher the accuracy, precision, and *F*-measure, the better the performances.

Inspecting the result on FSR, it is seen that roughly half of the original features were eliminated, especially for BPSODE, which facilitated a smaller number of features while keeping a high classification performance. Based on the result obtained, the lowest FSR was achieved by BPSODE in all cases. The reduction in the number of features not only decreased the complexity of the recognition system, but also enhanced the prediction accuracy.

From Table 2, it can be seen that BPSODE scored the highest precision and *F*-measure values for five subjects. These findings suggest that BPSODE is more capable of solving feature selection problems in EMG signals classification. The superiority of BPSODE mainly comes from the hybridization strategy, which adopts the advantages of both BDE and BPSO for searching significant features in the feature space.

Furthermore, the statistical *t*-test with 95% confidence level was used to examine whether there was a significant difference in the classification performance between BPSODE and other competitors. The results of the *t*-tests with *p*-values are presented in Table 3. In this table, the symbols "*w*/*t*/*l*" indicate that BPSODE was significantly better to (win), equal to (tie), and significantly worse to (lose) other feature selection methods. By applying the *t*-test, it shows that BPSODE was significantly better than BDE and BPSO (p-value < 0.05) with at least three subjects. In addition, BPSODE did not provide any significant worse results against its competitors. This again validates the efficiency of BPSODE for solving the feature selection problem in EMG signals classification.

Table 3. *p*-Value of the *t*-test on 10 subjects.

Subject	***p*-Value**			
	BBA	**BDE**	**BFPA**	**BPSO**
1	0.506287	0.056945	0.413356	0.847362
2	0.217023	0.109897	0.088031	0.384724
3	0.010163	2.00×10^{-5}	0.018040	0.001725
4	0.629456	0.031698	0.666264	0.049260
5	0.109897	0.000358	0.129670	0.058264
6	0.425133	0.000462	0.249168	0.870789
7	0.605826	0.605826	0.605826	1.000000
8	0.131348	0.000265	0.135088	0.803685
9	0.693922	0.651311	0.894854	0.186411
10	0.085574	0.000499	0.329877	0.000490
Win (*w*)/tie (*t*)/lose(*l*)	1/9/0	6/4/0	1/9/0	3/7/0

Figure 6 illustrates the convergence curve of the five different feature selection methods on 10 subjects. Note that the fitness is the average fitness values obtained from 20 runs. As can be seen, BPSODE achieved the lowest fitness value on most subjects, followed by BBA. Through the observation in Figure 6, BDE converged faster, but without acceleration. This explains why BDE did not work very well for high-dimensional feature selection. On the one side, one can see that BPSO and BBA converged faster at the initial stage. However, as time (iteration) passed, BPSO and BBA were trapped in the local optima, even though BPSODE did not give the fastest convergence speed. Nevertheless, BPSODE kept tracking for the global optimum, thus leading to a very good diversity. As a result, BPSODE overtook BBA, BFPA, BPSO, and BDE in evaluating the most informative feature subset.

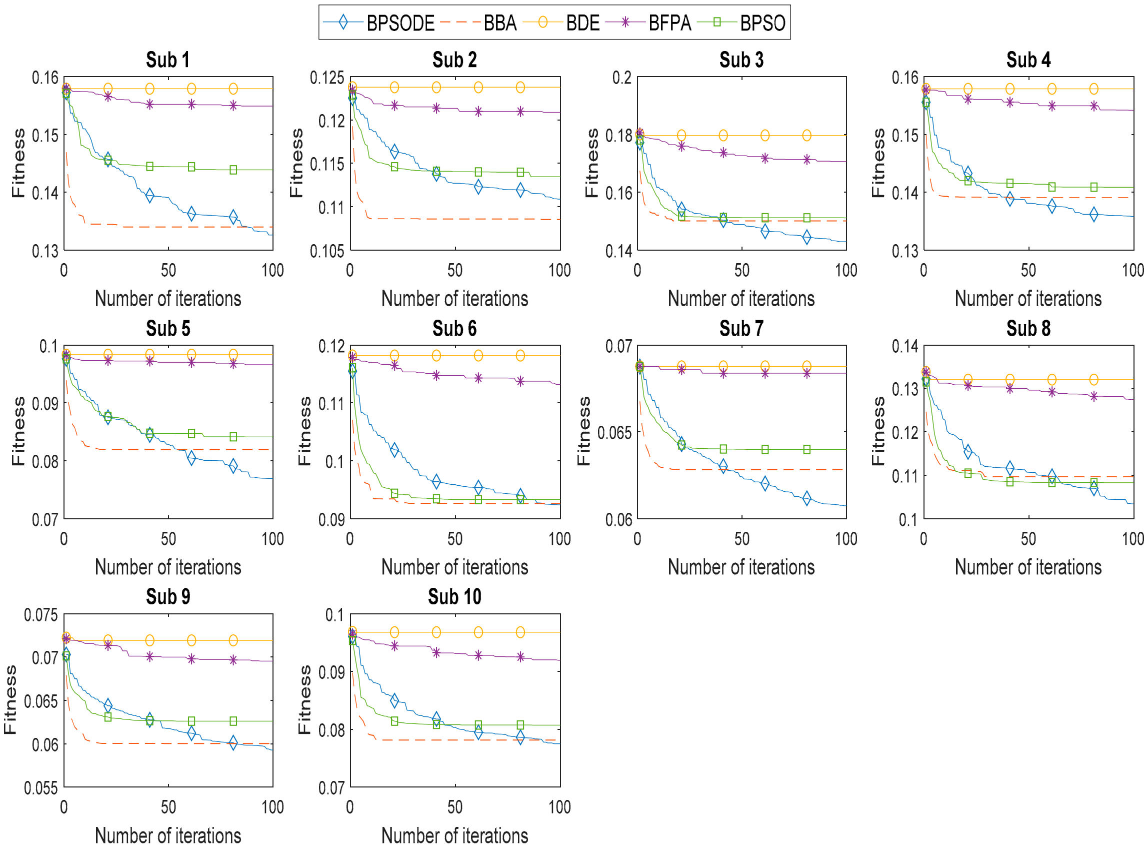

Figure 6. Convergence curves of BBA, BDE, BFPA, BPSO, and BPSODE on 10 subjects.

Table 4 outlines the computational cost of the five different feature selection methods on 10 subjects. As can be observed, BPSODE was computationally efficient in finding the best feature subset. In comparison with BPSO and BDE, BPSODE did not show much of an increment in computation time. This was because in BPSODE, the BPSO and BDE algorithms are computed in sequence, and thus, no additional computation cost is needed for the evaluations.

Table 4. Computational cost of the five different feature selection methods on 10 subjects.

Subject	**Average Computational Time(s)**				
	BPSODE	**BBA**	**BDE**	**BFPA**	**BPSO**
1	11.2170	9.6904	11.0745	10.9240	13.9385
2	11.4169	9.5965	11.3315	10.9586	13.6228
3	11.5064	9.7207	11.0580	11.4189	13.6346
4	11.6714	9.3344	11.2869	10.6936	13.4868
5	11.3360	9.3501	11.5490	11.0549	13.3068
6	11.5847	9.3415	11.2253	11.3795	13.2607
7	11.5799	9.2611	11.5731	11.1553	13.0535
8	11.8117	9.2501	11.9112	11.3934	13.3610
9	11.5575	9.1336	11.8026	11.2800	13.4043
10	11.7899	9.2060	11.7631	11.4669	13.3526

In this paper, we proposed BPSODE to solve feature selection problems in EMG signals classification. The BPSODE is the hybridization of BPSO and BDE, which inherits the advantages of both BPSO and BDE in feature selection. From the experiments, it can be inferred that the performance of BPSODE was superior against BPSO and BDE. In terms of accuracy, FSR, precision, and *F*-measure values, BPSODE proved to be the most powerful algorithm in this work.

The following observations explain why BPSODE outperformed BPSO and BDE in feature selection. Firstly, the hybridization of BPSO and BDE allowed a good exchange between exploitation and exploration. This restricts BPSODE from being trapped in the local optima. Secondly, a dynamic crossover rate offered a high diversity in the searching process. Lastly, the implementation of a dynamic inertia weight improved the convergence, which enhanced the performance of BPSODE in searching for the potential solution.

On the whole, the proposed BPSODE outperformed other conventional feature selection methods in exploring the feature search space. The BPSODE not only enhanced the BPSO algorithm with the aid of BDE in exploration, but also prevented itself from being trapped in the local solution. The present study showed that proper hybridization was able to overcome the limitations of two different algorithms leading to promising results.

5. Conclusions

In this study, a hybrid binary particle swarm optimization differential evolution (BPSODE) was proposed to solve the feature selection problem in EMG signals classification. In BPSODE, the BPSO and BDE algorithms are computed in sequence, hence, no extra computational cost is required. Additionally, two simple schemes, the dynamic inertia weight and dynamic crossover rate were introduced to improve the convergence and diversity of BPSODE in the searching process. In comparison with BBA, BFPA, BDE, and BPSO, our BPSODE can effectively remove the redundant features and maximize the classification accuracy. Successively, BPSODE overtook other algorithms in terms of the classification performance, FSR, precision, and *F*-measure values. Therefore, it can be inferred that BPSODE is a powerful feature selection tool, and BPSODE can be useful in engineering, rehabilitation, and clinical applications. In the future, the hybridization of other feature selection methods is recommended for tackling feature selection problems.

Author Contributions: Conceptualization, J.T.; Formal analysis, J.T.; Funding acquisition, A.R.A.; Investigation, J.T.; Methodology, J.T.; Software, J.T.; Supervision, A.R.A.; Validation, J.T.; Writing—original draft, J.T.; Writing—review and editing, J.T., A.R.A., and N.M.S.

Funding: This research and the Article Processing Charge were funded by the Ministry of Higher Education (MOHE) Malaysia under grant number GLuar/STEVIA/2016/FKE-CeRIA/l00009.

Acknowledgments: The authors would like to thank the Skim Zamalah UTeM and the Ministry of Higher Education Malaysia for funding the research under grant GLuar/STEVIA/2016/FKE-CeRIA/l00009.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Earley, E.J.; Hargrove, L.J.; Kuiken, T.A. Dual Window Pattern Recognition Classifier for Improved Partial-Hand Prosthesis Control. *Front. Neurosci.* **2016**, *10*. [CrossRef] [PubMed]
2. Naik, G.R.; Kumar, D.K.; Palaniswami, M. Signal processing evaluation of myoelectric sensor placement in low-level gestures: sensitivity analysis using independent component analysis. *Expert Syst.* **2014**, *31*, 91–99. [CrossRef]
3. Subasi, A. Classification of EMG signals using combined features and soft computing techniques. *Appl. Soft Comput.* **2012**, *12*, 2188–2198. [CrossRef]
4. Phinyomark, A.; Limsakul, C.; Phukpattaranont, P. Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification. *Meas. Sci. Rev.* **2011**, *11*, 45–52. [CrossRef]
5. Subasi, A. Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Comput. Biol. Med.* **2013**, *43*, 576–586. [CrossRef] [PubMed]
6. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276. [CrossRef]
7. Wang, D.; Zhang, H.; Liu, R.; Lv, W.; Wang, D. t-Test feature selection approach based on term frequency for text categorization. *Pattern Recognit. Lett.* **2014**, *45*, 1–10. [CrossRef]

8. Ghosh, A.; Datta, A.; Ghosh, S. Self-adaptive differential evolution for feature selection in hyperspectral image data. *Appl. Soft Comput.* **2013**, *13*, 1969–1977. [CrossRef]
9. Al-Ani, A.; Alsukker, A.; Khushaba, R.N. Feature subset selection using differential evolution and a wheel based search strategy. *Swarm Evol. Comput.* **2013**, *9*, 15–26. [CrossRef]
10. Too, J.; Abdullah, A.R.; Mohd Saad, N. Binary Competitive Swarm Optimizer Approaches for Feature Selection. *Computation* **2019**, *7*, 31. [CrossRef]
11. Banka, H.; Dara, S. A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognit. Lett.* **2015**, *52*, 94–100. [CrossRef]
12. Bharti, K.K.; Singh, P.K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. *Appl. Soft Comput.* **2016**, *43*, 20–34. [CrossRef]
13. Zorarpacı, E.; Özel, S.A. A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Syst. Appl.* **2016**, *62*, 91–103. [CrossRef]
14. Too, J.; Abdullah, A.R.; Mohd Saad, N. A New Co-Evolution Binary Particle Swarm Optimization with Multiple Inertia Weight Strategy for Feature Selection. *Informatics* **2019**, *6*, 21. [CrossRef]
15. Das, S.; Konar, A.; Chakraborty, U.K. Improving Particle Swarm Optimization with Differentially Perturbed Velocity. Proceedings of Genetic and Evolutionary Computation, New York, NY, USA, 25–29 June 2005; ACM: New York, NY, USA, 2005; pp. 177–184. [CrossRef]
16. Lin, G.-H.; Zhang, J.; Liu, Z.-H. Hybrid particle swarm optimization with differential evolution for numerical and engineering optimization. *Int. J. Autom. Comput.* **2018**, *15*, 103–114. [CrossRef]
17. Mirjalili, S.; Mirjalili, S.M.; Yang, X.-S. Binary bat algorithm. *Neural Comput. Appl.* **2014**, *25*, 663–681. [CrossRef]
18. Rodrigues, D.; Yang, X.-S.; Souza, A.N. de; Papa, J.P. Binary Flower Pollination Algorithm and Its Application to Feature Selection. In *Recent Advances in Swarm Intelligence and Evolutionary Computation*; Studies in Computational Intelligence; Springer: Cham, Switzerland, 2015; pp. 85–100. [CrossRef]
19. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 4104–4108.
20. Behera, H.S.; Dash, P.K.; Biswal, B. Power quality time series data mining using S-transform and fuzzy expert system. *Appl. Soft Comput.* **2010**, *10*, 945–955. [CrossRef]
21. Unler, A.; Murat, A. A discrete particle swarm optimization method for feature selection in binary classification problems. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [CrossRef]
22. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
23. NinaPro Database—Non-Invasive Adaptive Hand Prosthetics. Available online: https://www.idiap.ch/project/ninapro/database (accessed on 31 March 2019).
24. Pizzolato, S.; Tagliapietra, L.; Cognolato, M.; Reggiani, M.; Müller, H.; Atzori, M. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS ONE* **2017**, *12*, e0186132. [CrossRef]
25. Ahila, R.; Sadasivam, V.; Manimala, K. An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances. *Appl. Soft Comput.* **2015**, *32*, 23–37. [CrossRef]
26. Omari, F.A.; Hui, J.; Mei, C.; Liu, G. Pattern Recognition of Eight Hand Motions Using Feature Extraction of Forearm EMG Signal. *Proc. Natl. Acad. Sci. India Sect. Phys. Sci.* **2014**, *84*, 473–480. [CrossRef]
27. Chowdhury, R.H.; Reaz, M.B.I.; Ali, M.A.B.M.; Bakar, A.A.A.; Chellappan, K.; Chang, T.G. Surface electromyography signal processing and classification techniques. *Sensors* **2013**, *13*, 12431–12466. [CrossRef] [PubMed]
28. Phinyomark, A.; Phukpattaranont, P.; Limsakul, C. Fractal analysis features for weak and single-channel upper-limb EMG signals. *Expert Syst. Appl.* **2012**, *39*, 11156–11163. [CrossRef]
29. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
30. Jiao, B.; Lian, Z.; Gu, X. A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons Fractals* **2008**, *37*, 698–705. [CrossRef]

31. Taherkhani, M.; Safabakhsh, R. A novel stability-based adaptive inertia weight for particle swarm optimization. *Appl. Soft Comput.* **2016**, *38*, 281–295. [CrossRef]
32. Khushaba, R.N.; Al-Ani, A.; Al-Jumaily, A. Feature subset selection using differential evolution and a statistical repair mechanism. *Expert Syst. Appl.* **2011**, *38*, 11515–11526. [CrossRef]
33. Chuang, L.-Y.; Chang, H.-W.; Tu, C.-J.; Yang, C.-H. Improved binary PSO for feature selection using gene expression data. *Comput. Biol. Chem.* **2008**, *32*, 29–38. [CrossRef]
34. Tawhid, M.A.; Dsouza, K.B. Hybrid Binary Bat Enhanced Particle Swarm Optimization Algorithm for solving feature selection problems. *Appl. Comput. Inform.* **2018**. [CrossRef]
35. Gokgoz, E.; Subasi, A. Comparison of decision tree algorithms for EMG signal classification using DWT. *Biomed. Signal Process. Control* **2015**, *18*, 138–144. [CrossRef]
36. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, *2017*. [CrossRef] [PubMed]
37. Adeli, A.; Broumandnia, A. Image steganalysis using improved particle swarm optimization based feature selection. *Appl. Intell.* **2018**, *48*, 1609–1622. [CrossRef]
38. Taradeh, M.; Mafarja, M.; Heidari, A.A.; Faris, H.; Aljarah, I.; Mirjalili, S.; Fujita, H. An evolutionary gravitational search-based feature selection. *Inf. Sci.* **2019**, *497*, 219–239. [CrossRef]