

Noname manuscript No.
(will be inserted by the editor)

Improved Swarm-wavelet based Extreme Learning Machine for Myoelectric Pattern Recognition

Alrezza Pradanta Bagus Budiarsa,
Jenq-Shiou Leu, Kevin Kam Fung Yuen,
Xanno Sigalingging

Received: date / Accepted: date

Abstract Myoelectric signal generated by muscles is one of the bio-signals which is used by humans to control equipments. To achieve this purpose, a good myoelectric pattern recognition (M-PR) is required. The applied classifier and extracted feature sets greatly affect the success of M-PR. This paper proposes a hybrid and fast classifier, extreme learning machine (ELM) which is enhanced by improved hybrid particle swarm optimization with wavelet mutation (improved swarm-wavelet). ELM is, in essence, a single-hidden layer feed-forward neural network that keeps off iterative learning to save the training time. In addition to improving the actual performance of M-PR, we also evaluate the optimization of ELM using improved swarm-wavelet in this paper. The swarm-wavelet is improved by using the particle refresh and applying velocity improvement to avoid trapped in local minima. In ELM, the improved swarm-wavelet is used to find the most suitable parameters to increase the classification accuracy. Furthermore, this paper provides comparisons of improved swarm-wavelet-ELM, swarm-wavelet-ELM and standard swarm-ELM. The experimental results show that the improved swarm-wavelet-ELM, our proposed method, is the most accurate classifier with mean accuracy of 99.6%.

Keywords classification, myoelectric pattern recognition, extreme learning machine, improved swarm-wavelet.

Alrezza Pradanta Bagus Budiarsa, Jenq-Shiou Leu, and Xanno Sigalingging
National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Rd, Da'an District, Taipei City, 106
E-mail: m10402803, jsleu, d10502810@mail.ntust.edu.tw
Kevin Kam Fung Yuen
Hong Kong Polytechnic University
11 Yuk Choi Rd, Hung Hom, Hong Kong SAR, China
E-mail: kevinkf.yuen@gmail.com; kevin.yuen@polyu.edu.hk

1 Introduction

The hands are very important parts of the human body which have a very complex system. Having a disability or even losing it will causes some physical and mental illness. The uncomfortable feeling will easily arises if their ability to sense and explore the surrounding world are gone, also the same can be said with their ability to grasp and control things. Any physical difficulties and burdens may also emerge in consequence of patient's physical differences compared with intact-limbed people.

The common causes of hand disability is either problem on the motor system or amputation which mostly caused by trauma, cancer, or vascular disease. A flawless technology development for hand rehabilitation is a challenging task. The use of hand prosthesis is a representation of contemporary technological aid for hand amputation and imperfection. The artificial hands that circulate on the market nowadays mostly have only a few fixed set of gestures. Unconventional commercial technologies such as i-limb from Touch Bionics [1], bebionic hand from RSLSteeper [2], and Michelangelo prosthetic hand from Ottobock are the device that mimic the shape and functionality of the human hand. A sophisticated control system is need to be applied on these advanced prosthetic devices so that they are able to do more adept movements. Moreover, the desired control system in general has to be able to work in accordance with the wearer's wish to build up their comfort. Thus, to establish a flawless control system on the dexterous bionic hand, the myoelectric pattern recognition (MPR) can be employed.

A neuromuscular activity can be extracted into myoelectric signal which has valuable information using electromyography (EMG). The myoelectric signals acquired from forearm skin surface have been used to control a prosthetic hand or any other equipment for years. Many efforts to attain a deft hand prosthesis have been done and the main thing for this issue is the MPR. In fact, to achieve more dexterous prosthetic devices, a robust method for MPR is needed.

One of the classifier which is popular in the MPR system field is feed-forward neural networks (FFNNs). Uchida et al. implemented the FFNNs for the MPR to classify five finger movements. The proposed system used the feature of fast Fourier transform (FFT) from two EMG channels. The accuracy of the system is about 86% [3]. Tsenov et al. also proposed a MPR system using MLP for finger movement classification with two and four channels of EMG signal and accomplished a classification accuracy of 93% and 98%, respectively [4]. In addition to the works above, Tenore et al. utilized a MLP not only on healthy subjects, but also on amputees. Their MLP system reach the average accuracy of 90% on the finger flexion and extension classification. The result also indicates the results of their system are not able to differentiate between the able-bodied and the amputee subjects [5].

Oskoei and Hu have successfully classified six different upper limb motions with two different sets of multi-features. The classification accuracy using SVM method is around 96% and 97% by using four and seven feature sets, respec-

tively [6]. Moreover, Lee et al. conducted an EMG classification on stroke survivors using four feature sets with mean accuracy 71.3% [7]. Another upper limb movement classification was used for controlling hand prostheses [8] and other equipments.

The researchers in recent studies are getting interested in employing the artificial neural networks (ANNs) in altered phases. It is so-called extreme learning machine (ELM). ELM is a progression from the single-layer perceptron of feedforward neural networks which has strength in having shorter training time. Fast training speed of ELM is achieved by neglecting the repetitive learning process. The hidden parameters of ELM are estimated randomly while the output weights are computed analytically. However, defining input weights and hidden biases as random numbers for hidden node parameters in ELM will cause a non-optimal performance. In node-based ELM, the optimal result will be obtained when the optimal combination between the amount of hidden layers and the type of the activation function are achieved. Several attempts have been done to overcome this optimization problem. For example, to overcome the problem of the hidden node parameters optimization found in ELM method, advanced algorithms such as particle swarm optimization ELM (PSO-ELM) [9] and self-adaptive evolutionary ELM (SAE-ELM) [10] are proposed.

The particle swarm optimization (PSO) method seems to be a prominent solution to optimize the hidden node parameters in ELM so that the classifier can reach high accuracy. However, PSO inclined to get trapped in some local minimum. To overcome the local minimum problem, one technique that can be implemented is by adding the particle mutation algorithm in the swarm using a wavelet function [11]. More improvement by adding the refresh operator [12] to the particle in the swarm and by improving the velocity update formula [13] can also be applied to achieve a better convergence of PSO performance. These improvements to the PSO can increase the diversity of particles and improve the speed of the particle optimization as well. In this work, an improved hybrid PSO with wavelet mutation (improved swarm-wavelet) based ELM is proposed.

This paper evaluates and compares the implementation of ELM optimized using standard swarm (SS), swarm-wavelet (SW), and improved swarm-wavelet (ISW) to classify hand gestures. In the node-based ELM, the performance of the system relies on the amount of hidden nodes and the type of activation function which the PSO algorithm can optimize. The observation on the varying numbers of EMG channels and lengths of the window segment are also presented in this paper.

The paper is organized as follows. Section II presents the preliminaries or the basic algorithms used in this work which are ELM and PSO. Section III provides the proposed method used in the experiments. Section IV provides the data acquisition protocol, feature extraction, data segmentation, and classification. Section V presents the result of experiments and analysis for classifiers on the different channel's number and different window lengths. This paper also compares optimized-ELM and other well-known classifiers such as

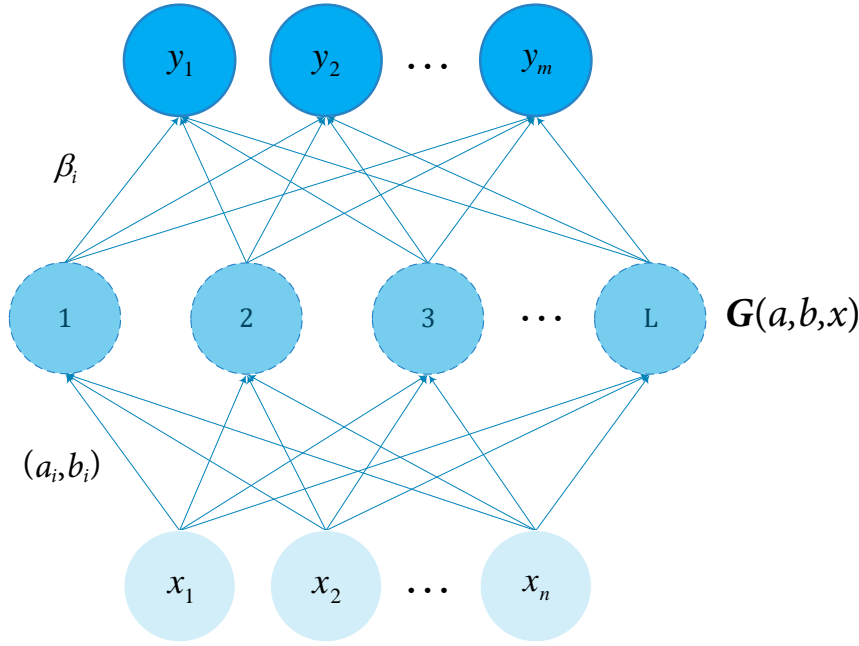


Fig. 1: Generalized single layer model of extreme learning machine.

k -nearest neighbors (kNN), support vector machine (SVM) type 1 (C-SVM), SVM type 2 (Nu-SVM), C5.0, and classification and regression trees (CART). Lastly, Section VI presents our conclusions from this work.

2 Preliminaries

2.1 Extreme Learning Machine (ELM)

The extreme learning machine (ELM) is an abstraction of the single-layer perceptron of feedforward neural network that keeps off repetitive learning in the determination of output weights and hidden nodes [14]. ELM is an improvement of artificial neural networks (ANN) that was designed to resolve their disadvantages, especially in the processing time. The appealing of the ELM features lies in the hidden nodes that tuning is not needed. The random implementation on the nodes is self-sufficient from the data training. The learning algorithm of the ELM is different with traditional SLFNs that the objective of ELM is to obtain the smallest error in the training process and criterion of output weights.

ELM is originally developed for regression and classification problem [14], but it used by many researchers for different intentions in varying area of

research fields. The current ELM implementation is extended to feature learning, clustering, and dimensionality reduction. The generality of its configuration makes the ELM can be applied in two ways, by random feature mappings (node-based) or by kernels (kernel-based). The node-based ELM has the hidden layer weights in its neural networks whereas the kernel-based ELM possesses a kernel for its feature mapping.

To obtain a good result from ELM, the randomization of the input weights and hidden biases required higher amount of hidden nodes. Nevertheless, the training process will take more time if the number of hidden neurons is infinite. There are different ways to achieve an optimum number of hidden nodes in the network such as differential evolutionary algorithm [15], particle swarm optimization [9], and self-adaptive differential evolution algorithm [10].

2.1.1 The concept of ELM

Essentially, ELM is single-layer perceptron of feedforward neural network as shown in Figure 1. For N random samples $(x_j, y_j) \in R^n \times R^m$, the output of ELM with L hidden nodes and activation function $G(x)$ is

$$\mathbf{F}(x) = \sum_{i=1}^L \beta_i \mathbf{G}(a_i, b_i, x_j), \quad j = 1, 2, \dots, N \quad (1)$$

where $\beta_i = [\beta_1, \beta_2, \dots, \beta_L]^T$ denotes the weight vector connecting the i th hidden node and the output node and (a_i, b_i) are hidden node parameters which randomly generated according to any continuous probability distribution. $a_i = [a_{i_1}, a_{i_2}, \dots, a_{i_n}]^T$ defines the vector of the weight linking the i th hidden node and the input node and $b_i = [b_{i_1}, b_{i_2}, \dots, b_{i_n}]^T$ is the threshold of the i th hidden node.

The activation function in the ELM roles as feature mapping. It can be a non-linear and continuous piecewise-defined function. The example of this function such as sigmoid function, hard-limit function, Gaussian function, and multi-quadratic function.

The single hidden layer feed-forward neural networks can also be defined in the concise form:

$$\mathbf{T} = \mathbf{H}\beta \quad (2)$$

where \mathbf{H} is the neural network's output matrix. The i th column of \mathbf{H} is the output of the i th hidden node with respect to the inputs x_1, x_2, \dots, x_N .

$$\mathbf{H} = \begin{bmatrix} \mathbf{G}(a_1, b_1, x_1) & \cdots & \mathbf{G}(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ \mathbf{G}(a_1, b_1, x_N) & \cdots & \mathbf{G}(a_L, b_L, x_N) \end{bmatrix}_{N \times L}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad (4)$$

and \mathbf{T} is the target

$$\mathbf{T} = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}. \quad (5)$$

The objective of the ELM is to minimize the norm of the output weights in addition to reach the smallest training error. So, that

$$\min_{a_i, b_i, \beta_i} : \|\mathbf{H}(a_i, \dots, a_L, b_i, \dots, b_L)\beta - \mathbf{T}\|^2. \quad (6)$$

The value of a_i and b_i are randomly determined, so that Eq. (6) becomes:

$$\min_{\beta} : \|\mathbf{H}\beta - \mathbf{T}\|^2 \quad (7)$$

The following equation is used to calculate the output weights β :

$$\beta = \mathbf{H}^\dagger \mathbf{T} \quad (8)$$

where \mathbf{H}^\dagger is the Moore-Penrose generalized inverse [16], [17] of hidden layer output matrix \mathbf{H} , which tends to obtain the smallest norm of the output weights β while minimizing the error in the training process.

In ELM, minimum norm of output weights achieves a better performance and more robust system [14], [18]. From the hidden layer to the output layer in the ELM, the output weights β can be calculated as:

$$\beta = \left(\frac{\mathbf{I}}{\mathbf{C}} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T} \quad (9)$$

or as:

$$\beta = \mathbf{H}^T \left(\frac{\mathbf{I}}{\mathbf{C}} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{T} \quad (10)$$

where \mathbf{C} is a parameter specified by the user. In the end, the output of function in Eq. (1) can be altered to:

$$\mathbf{F}(x) = \mathbf{G}(x)\beta = \mathbf{G}(x)\mathbf{H}^T \left(\frac{\mathbf{I}}{\mathbf{C}} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{T}. \quad (11)$$

Generally, the learning model procedure of the ELM is defined in Algorithm 1. For ELM, its only solving cost is the same as a linear system.

Algorithm 1 Learning Model of Extreme Learning Machine**Input:**Set of training data $\mathbf{X} = [(x_j, y_j) \in R^n \times R^m, i = 1, 2, \dots, N]$ Activation function $G(a, b, x)$ Number of hidden nodes L **Output:**ELM learning model $f(x)$ **Process:**Randomly generate parameters (a_i, b_i) Calculate the hidden output matrix $\mathbf{H} \leftarrow \text{Eq. (3)}$ Calculate the output weight $\beta \leftarrow \text{Eq. (8)}$ ELM learning model $\mathbf{F}(x_i) = [\mathbf{F}_1(x_i), \dots, \mathbf{F}_m(x_i)]^T$ Let $\mathbf{F}_j(x_i)$ denote the output value of the j th output node*2.1.2 ELM with Kernel*

The activation function or the feature mapping $G(x)$ is sometimes unknown. Therefore, a kernel function can be applied to map the network's input features. The definition of kernel matrix for ELM is

$$\Omega_{\text{ELM}} = \mathbf{H}\mathbf{H}^T : \Omega_{\text{ELM}_{i,j}} = G(x_i) \cdot G(x_j) = K(x_i, x_j). \quad (12)$$

Then, the output function of Eq. (11) is:

$$\begin{aligned} \mathbf{F}(x) &= G(x)\mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \\ &= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \Omega_{\text{ELM}} \right)^{-1} \mathbf{T} \end{aligned} \quad (13)$$

where matrix \mathbf{K} is a kernel matrix consisting kernel functions. The function used for the kernel matrix could be any kernel such as radial basis function, linear, and polynomial.

Similar to SVM and LS-SVM, the activation function (feature mapping) $G(x)$ and the number of hidden neurons L no need to be known by the users in this case; preferably the corresponding kernel $K(x_i, x_j)$ is given.

2.2 Particle Swarm Optimization (PSO)

The PSO is a population-based stochastic search algorithm that optimizes a problem by iteratively trying to improve the candidate solution based on the social behavior of bird flocking or fish schooling. Multiple particles in a swarm moves in a multi-dimensional search space to obtain the optimal solution of the problem [19]. All of particles have their own positions and velocities which control the movement speed and direction in the search space. Position is the result of the optimization and velocity is the process of the optimization. The

behavior of a particle in the swarm are affected by its local best known position and the practical knowledge of its neighbors. More specifically, particles within the swarm are moved through a problem space, in which the position of each particle is adjusted by its own experience and its neighbors.

PSO is initialized with a set of randomly determined particles and then searches for the optimal solution by updating generations. The particles have fitness values which will be evaluated by the fitness function in every iteration. Each particle keeps the information of its position in the search space which is associated to the best solution it has achieved so far. This value is called *pbest*. Moreover, another best value called *gbest* is the best value tracked by the global version of the particle swarm optimizer.

The optimization in PSO algorithm formulated to change the velocity and position of each particle in regard to its *pbest* and *gbest* location. The equation for updating the particle's velocity and position is shown in Eq. (14) and Eq. (15) respectively.

$$\begin{aligned} v_i(t+1) = & \chi v_i(t) \\ & + r_1 c_1 (p_i(t) - q_i(t)) \\ & + r_2 c_2 (p_g(t) - q_i(t)) \end{aligned} \quad (14)$$

$$q_i(t+1) = q_i(t) + v_i(t+1) \quad (15)$$

where:

- $q_i = [q_{i1}, q_{i2}, \dots, q_{in}]^T$ is the position of the i th particle in n -dimensional search space;
- $v_i = [v_{i1}, v_{i2}, \dots, v_{in}]^T$ is the velocity of the i th particle;
- p_i (*pbest*) is the best previous i th particle position;
- p_g (*gbest*) is the best particle among all particles;
- c_1 is personal acceleration coefficient;
- c_2 is social acceleration coefficient;
- r_1 and r_2 are random number $[0, 1]$ generated using the uniform distribution;
- and χ is the constriction coefficient

$$\chi = \frac{2\kappa}{\sqrt{2 - \phi - \sqrt{\phi^2 - 4\phi}}}, \quad \begin{cases} \kappa = 1 \\ \phi = \phi_1 + \phi_2 \\ \phi_1 = 2.05 \\ \phi_2 = 2.05. \end{cases} \quad (16)$$

3 Improved Swarm-wavelet based ELM

The standard particle swarm optimization algorithm tends to get trapped on some local minimum while trying to reach an optimal solution of the problem. Adding the particle mutation technique in the swarm may overcome the local minimum issue. The mutation operation using wavelet function [11] seems to

be one of the rising solution to get better convergence rate for optimization problem of myoelectric data. More improvement by adding the particle refresh [12] in the swarm and by improving the velocity update formula [13] can also be applied to achieve a better convergence of PSO performance. Such of improvements can improve the speed of the swarm optimization and increase the particle's diversity to get the better result.

3.0.1 Hybrid PSO with Wavelet Mutation

The hybrid PSO with wavelet mutation was proposed by [11]. Each particle in the swarm has probability $p_m \in [0, 1]$ to mutate its position. A random number $[0, 1]$ will be generated for each particle q_i . If the generated number is less than or equal to p_m , the particle's position will mutate to a new position q_i using following equation:

$$q_i = \begin{cases} q_i + \sigma \times (q_{i_{max}} - q_i), & \text{if } \sigma > 0 \\ q_i + \sigma \times (q_i - q_{i_{min}}), & \text{if } \sigma \leq 0 \end{cases} \quad (17)$$

where $q_{i_{max}}$ and $q_{i_{min}}$ is the maximum and minimum particle's position in the swarm, respectively. σ is the mother wavelet function, such as Meyer wavelet, Morlet wavelet, Gaussian wavelet, and Mexican hat wavelet.

$$\sigma = \psi_{\alpha,0}(\varphi) = \frac{1}{\sqrt{\alpha}} \psi\left(\frac{\varphi}{\alpha}\right) \quad (18)$$

According to [11], the Morlet wavelet was chosen as the mother wavelet because it offers the best performance in the nature of cost values. The definition of Morlet wavelet is as follows.

$$\sigma = \frac{1}{\sqrt{\alpha}} e^{-\left(\frac{\varphi}{\alpha}\right)^2 / 2} \cos\left(5\left(\frac{\varphi}{\alpha}\right)\right). \quad (19)$$

where the value of φ is generated randomly at the range $-2.5\alpha \leq \varphi \leq 2.5\alpha$ [11]. The value of the dilatation parameter α in the Morlet wavelet can be calculated as:

$$\alpha = e^{-\ln(g) \times \left(1 - \frac{t}{T}\right)^{\zeta_{wm}} + \ln(g)} \quad (20)$$

3.0.2 Particle Refresh and Velocity Improvement of PSO

The particle's velocity in the PSO algorithm influences the size of search space. the accuracy of the search, and also the precision of the optimal position. Therefore, the determination of particle's velocity for later iteration is greatly affected the performance of PSO. Improving the standard PSO velocity updating formula will balances the importance of the particle position in the swarm [13]. The improvement to the standard PSO velocity update formula will achieve prime position of the particle in the swarm and can learn its optimization information.

Algorithm 2 Improved Swarm-wavelet based ELM

```

1: for each particle  $i = 1, \dots, S$  do
2:   Initialize particle's position:  $q_i \sim [q_{min}, q_{max}]$ 
3:   Set  $p_{best}$  to its initial position:  $p_i \leftarrow x_i$ 
4:   Calculate the fitness value (ELM accuracy):  $f(q)$ 
5:   if  $f(p_i) < f(p_g)$  then
6:     update  $g_{best}$ :  $p_g \leftarrow p_i$ 
7:   Initialize particle's velocity:  $v_i \sim [v_{min}, v_{max}]$ 
8: while a termination criterion is not met do
9:   for each particle  $i = 1, \dots, S$  do
10:    Generate random numbers:  $r_1, r_2, r_3, r_4 \sim [0, 1]$ 
11:    Update particle's velocity:  $v_i \leftarrow \text{Eq. (21)}$ 
12:    Apply the refresh operator
13:    if  $|v_i| < \varepsilon$  then
14:      Re-locate the particle's position
15:    Update particle's position:  $q_i \leftarrow \text{Eq. (22)}$ 
16:    Perform wavelet mutation operation with  $p_m$ 
17:    Calculate the fitness value (ELM accuracy):  $f(q)$ 
18:    if  $f(x_i) < f(p_i)$  then
19:      Update  $p_{best}$ :  $p_i \leftarrow q_i$ 
20:      if  $f(p_i) < f(p_g)$  then
21:        Update  $g_{best}$ :  $p_g \leftarrow p_i$ 

```

In addition to examining the particle's best position p_{best} and the global best particle's position g_{best} in the swarm, the velocity update formula is referenced to the global best particle's velocity v_g and the best particle's velocity v_{nb} in its neighbor. The improved update formula of the velocity is defined by

$$\begin{aligned}
v_i(t+1) = & \chi v_i(t) \\
& + \left(c_1 - \frac{(c_1 - c_2) \times t}{T} \right) r_1 (p_i - q_i(t)) \\
& + \left(c_1 - \frac{(c_1 - c_2) \times t}{T} \right) r_2 (p_g - q_i(t)) \\
& + \left(c_2 - \frac{(c_1 - c_2) \times t}{T} \right) r_3 (v_g(t) - v_i(t)) \\
& + \left(c_2 - \frac{(c_1 - c_2) \times t}{T} \right) r_4 (v_{nb}(t) - v_i(t))
\end{aligned} \tag{21}$$

and the position update formula is defined as follows.

$$\begin{aligned}
q_i(t+1) = & \left(d_1 - \frac{(d_1 - d_2) \times t}{T} \right) q_i(t) \\
& + \left(d_1 - \frac{(d_1 - d_2) \times t}{T} \right) v_i(t+1)
\end{aligned} \tag{22}$$

where $v_i(t+1)$ and $v_i(t)$ are the updated velocity and the current velocity. v_g is the global best particle's velocity and v_{nb} is the best velocity strategy in its neighbor particle swarm. c_1 and c_2 are the acceleration factors with $c_1 > c_2$,

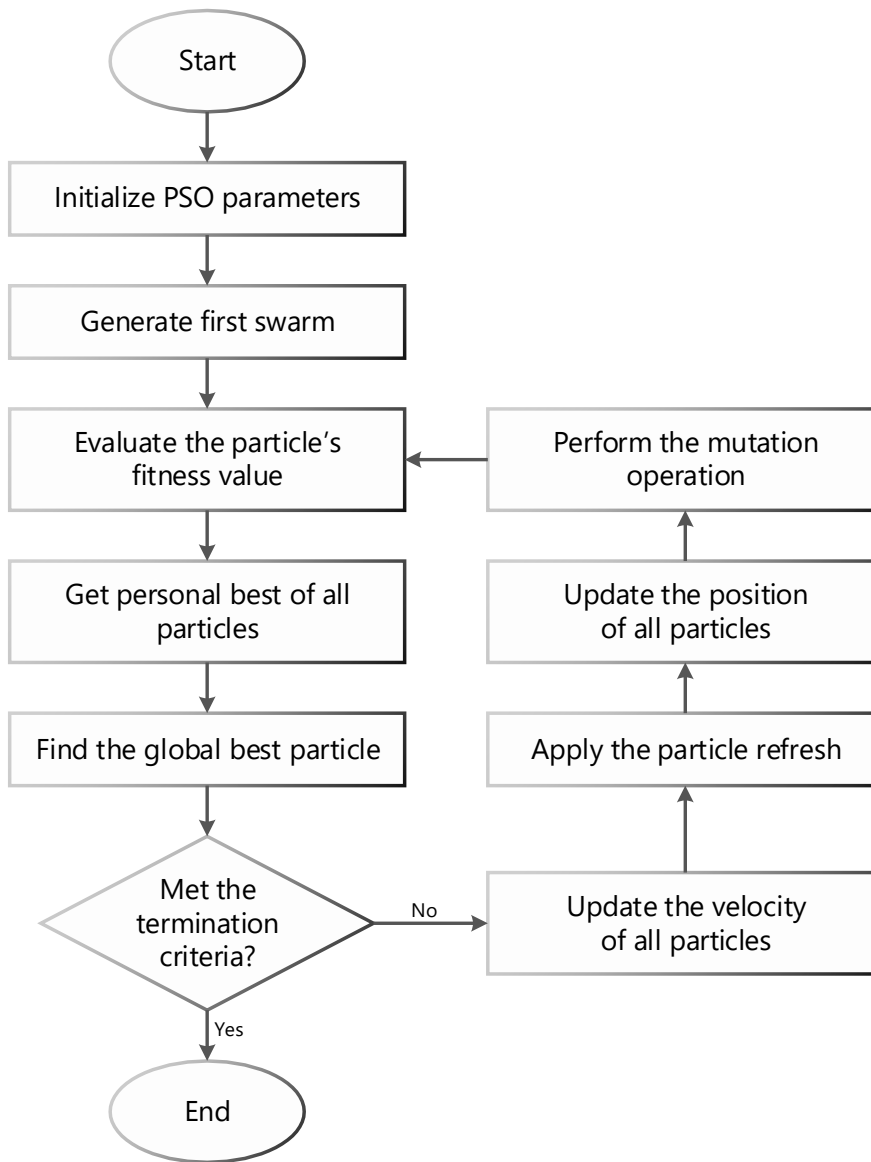


Fig. 2: Flowchart of the improved swarm-wavelet.

r_1, r_2, r_3 , and r_4 are randomly generated normal number at range $[0, 1]$, and d_1 and d_2 are weighting factors with $d_1 > d_2$.

The particle in a swarm may stuck in a position which not the global optimal position. Another improvement to the PSO algorithm can be done by adding the refresh operator. The operation of particle refresh in PSO will halt

particles and strengthen algorithm to avoid trapping in the local minima [12]. A particle is useless when it's a standstill particle. Therefore, when $|v_i| < \varepsilon$, randomly relocate the particle's position, where ε denotes a very small value [13]. The pseudo-code and flowchart of the improved swarm-wavelet are shown in Algorithm 2 and Figure 2, respectively.

The aim of the optimization using improved swarm-wavelet is to find the best node-based ELM parameters which maximize the classification accuracy of the hand gesture recognition. A five-fold cross validation was employed to evaluate the performance. Furthermore, the fitness function of the particle q is defined by

$$f(q) = \frac{\text{Number of correct samples}}{\text{Total number of testing samples}} \times 100\%. \quad (23)$$

4 Myoelectric Pattern Recognition with Improved Swarm-wavelet based ELM

4.1 Proposed System

The hand gestures classification in this paper uses a contemporary myoelectric pattern recognition system as shown in Figure 5. The proposed M-PR system comprised of several stages. The myoelectric signals were firstly recorded by EMG acquisition device. The windowing on the collected data then applied before extracting sets of the time-domain feature. The feature sets selection was conducted to the data to avoid a high computational complexity before being classified using improved swarm-wavelet-ELM, swarm-wavelet-ELM and standard swarm-ELM.

4.2 Data Acquisition and Processing

4.2.1 Subjects

The electromyography data used in this experiment were acquired from ten healthy subjects. One female and nine males, aged 22-25 years old were recruited to perform the required hand movements. All subjects were normally limbed with no muscular or neurological disorders. A fixed EMG device was supported on the subject's right arm on certain position to avoid the effect of different limb positions on the generated EMG signals.

4.2.2 Signal acquisition device

The acquisition device for EMG signals should be able to maximize the quality of the obtained myoelectric signals in terms of signal-to-noise ratio (SNR) and must minimize possible distortion. An amplification chain made up for one or more low-noise, high-input impedance amplifiers, and filters are usually used to



Fig. 3: Myo Armband with eight EMG sensors.

acquire EMG signals. The amplitude of the EMG signal is usually determined by the size, the movement, and the speed of the muscle.

In this work, the data was collected using Myo Gesture Control Armband developed by Thalmic Labs. The Myo armband is a wearable gesture control and motion control device which consists of EMG and inertia measurement unit (IMU) sensors. The Myo armband has eight sensors as shown in Figure 3 which will generate eight channels on the recorded data with a sampling frequency of 25 Hz. The Bluetooth low energy (BLE) technology was used to connect the Myo armband to a personal computer.

4.2.3 Acquisition protocol

Ten healthy subjects were instructed to perform five actual hand gestures. The five classes of hand gestures implemented by the subjects consisted of a double tap (Dt), fingers spread (Fs), wave out (Wo), wave in (Wi), and fist (F). The sample of EMG signals for each gesture are presented in Table 1. Throughout the signal acquisition, the subjects were sitting on a chair wearing the signal acquisition device in front of a personal computer with the MyoDiagnostics interface screen to see all the EMG channels in real-time while performing the movements. The subjects put their arms on their resting state and produced different hand gestures subsequently as shown in Figure 4 with a rest of 1-2 seconds between two consecutive gestures. Moreover, ten trials were recorded for each gesture.

Table 1: Sample of EMG signals for each gesture.






Gesture	Signals							
 Dt	#1	#2	#3	#4	#5	#6	#7	#8
 Fs	#1	#2	#3	#4	#5	#6	#7	#8
 Wo	#1	#2	#3	#4	#5	#6	#7	#8
 Wi	#1	#2	#3	#4	#5	#6	#7	#8
 F	#1	#2	#3	#4	#5	#6	#7	#8



Fig. 4: Signal acquisition experiment using Myo Armband device.

Table 2
 MATHEMATICAL DEFINITION OF SOME FEATURES USED FOR M-PR.
 x_i IS THE i TH SIGNAL AND N IS THE NUMBER OF SAMPLES IN A SEGMENT.

TD Feature	Definition	References
Variance	$VAR = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$	[5], [20]
Mean absolute value	$MAV = \frac{1}{N} \sum_{i=1}^N x_i $	[21], [22], [20]
Root mean square	$RMS = \frac{1}{N} \sum_{i=1}^N x_i^2$	[23]
Simple square integral	$SSI = \sum_{i=1}^N x_i ^2$	[24]
Waveform length	$WL = \sum_{i=2}^N x_i - x_{i-1} $	[21], [25]
Slope sign change	$SSC = \sum_{i=2}^{N-1} f((x_i - x_{i-1}) \cdot (x_i - x_{i+1})), f(x) = \begin{cases} 1, & \text{if } x > \text{threshold} \\ 0, & \text{otherwise} \end{cases}$	[25]
Zero Crossing	$ZC = \sum_{i=1}^N f(x), f(x) = \begin{cases} 1, & x_i \cdot x_{i+1} < 0 \text{ and } x_i - x_{i+1} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases}$	[21]
Willison amplitude	$WAMP = \sum_{i=2}^N f(x_i - x_{i-1}), f(x) = \begin{cases} 1, & \text{if } x > \text{threshold} \\ 0, & \text{otherwise} \end{cases}$	[5]



Fig. 5: The proposed myoelectric pattern recognition for hand gestures classification using improved swarm-wavelet-ELM, swarm-wavelet-ELM, and standard swarm-ELM.

4.2.4 Data segmentation

The EMG data can be characterized into two states, a transient state and a steady state. A transient state is a state between two different muscle contractions, while a steady state is a condition where the muscle is continuously

under contraction. Several studies have been conducted in depth on the issue of selecting a good data segment [6], [26], [21], [27], [28].

The technique of data windowing deals with the continuous classifier development. This classifier must be able to produce a sequence of decisions using a sliding window of myoelectric signal activity. The acquired EMG signals can be segmented as a disjoint or overlapping window by data windowing technique. The disjoint window only relates with the window length, while the overlapping window associates with not only the window length but also the window increment. The overlapping window allows continuous classification on the transient and steady state of EMG data [26]. Moreover, the recorded EMG data in this work were divided into overlapping windows of 200 ms segment length.

4.2.5 Feature extraction

The raw EMG data can not be used as inputs for a classifier due to the signal's randomness and its too large to be used as inputs. Because of these reasons, smaller dimension vectors called feature vectors have to be extracted from raw EMG signals. The type and nature of the feature selected and extracted will surely affect the classification performance. Therefore, the feature extraction is an essential phase in the design of myoelectric pattern recognition system.

Various kind of features has been examined individually and in group [6], [29], [30], [22], [31], representing both amplitude and spectral content of EMG signals. The extracted Features must be able to present the characteristics or properties of the signal to a different hand gesture. The involved features of the time-domain (TD) are variance (VAR) [5], [20], mean absolute value (MAV) [21], [22], [20], root mean square (RMS) [23], simple square integral (SSI) [24], waveform length (WL) [21], [25], slope sign change (SSC) [25], zero crossing (ZC) [21], and Willison amplitude (WAMP) [5]. TD features are the most popular thanks to the ease of calculating based on EMG signal amplitude. The direct extraction of TD features without the need for mathematical transformation makes these features the best choice from a computational load point of view, especially on the real-time applications.

The extracted features from all channels of EMG are combined to create a large feature set. To avoid a high dimensional computation, the feature selection algorithm is applied. The multi-feature set consist of RMS, WL and MAV outperforms other features. The highest accuracy is acquired with just using these three features. RMS appears to be the best single feature in terms of high rate of accuracy and stability of segmentation method changes. The stability of TD features have been studied in [32] due to changes in EMG signals caused by the variation in muscle contraction strength, muscle fatigue, and location shift of the electrode.

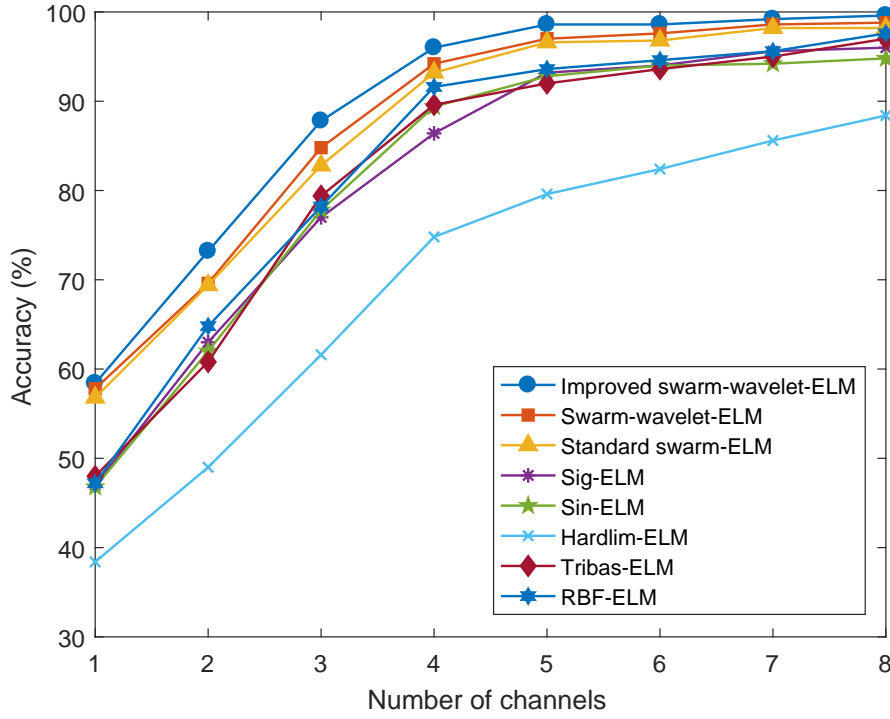


Fig. 6: The average classification accuracy of the different channel number experiments.

4.2.6 Classification

The performance of optimized node-based ELMs in hand gesture recognition was investigated in this paper. The node-based ELM uses hidden nodes to map the feature whereas the kernel-based ELM applies a kernel function. In this work, the activation function in node-based ELM are sigmoidal function, sine function, hard-limit function, triangular basis function and radial basis function. Three different optimization methods were employed to find the optimal parameter value in the node-based ELM. In addition, some experiments were conducted to make a performance comparison among different optimization methods of ELM in the term of classification accuracy.

The classification process was done in the offline mode and a five-fold cross-validation is used to validate the result in all experiments. The classification accuracy formulated as a fitness function of the PSO in Eq. (23) was used to evaluate the performance of the classifier in hand gesture recognition.

5 Experiments and Results

In this section, several experiments and results are presented and analyzed. All experiments were done on a 3.4 Ghz Intel Core i7 based with 16 GB RAM running MATLAB and R [33], [33], [34], [35] as tools for all experiments. The classification performance was validated using a five-fold cross-validation in all experiments.

5.1 Number of Channels

The purpose of this experiment is to look into the classification performance based on how many channel used from the surface EMG. The number from one up to eight channels were explored across ten healthy subjects. In this experiment, a feature set consists of RMS, WL and MAV was employed with 200 ms length of window. The performance of the classification are evaluated using average classification accuracy as presented in Figure 6.

Figure 6 shows the merit comparison of the different channel number experiments using different PSO methods to optimize ELM classifier. All three methods achieved very low accuracy rate when the small number of channels employed, but they increased swiftly as the number of channel increased to four channels. The classification accuracy consistently stayed at roughly 99% when the channel used is more than five for the improved swarm-wavelet-ELM. The maximum accuracy of the improved swarm-wavelet-ELM on eight channels outperforms the other two PSO methods at 99.6%, while the swarm-wavelet-ELM and standard swarm-ELM reached the maximum accuracy of 98.8% and 98.2%, respectively.

5.2 Window Length

This experiment was conducted to obtain the optimal number of window length for hand gesture classification by differing its window length from 20 to 200 ms with window increment of 20 ms. This experiment applied six channels of EMG signal from all subjects. The set of feature used in this experiment consists of RMS, WL, and MAV. Three different method of ELMs were used to investigate the classification performance. Figure 7 shows the result of this experiment.

As presented in Figure 7, the accuracy of the classification rises up as the window length increases. Therefore, the best accuracy achieved when the experiment used maximum window length. Nevertheless, that does not mean that the highest number of window length always determine as the optimal window length. According to [36], the optimal length of the window used should be ranged between 150 and 250 ms. In this range, Figure 7 shows the average accuracy is around 99%–99.2% for the improved swarm-wavelet-ELM, 97.4%–97.8% for the swarm-wavelet-ELM and 97%–97.2% for the standard

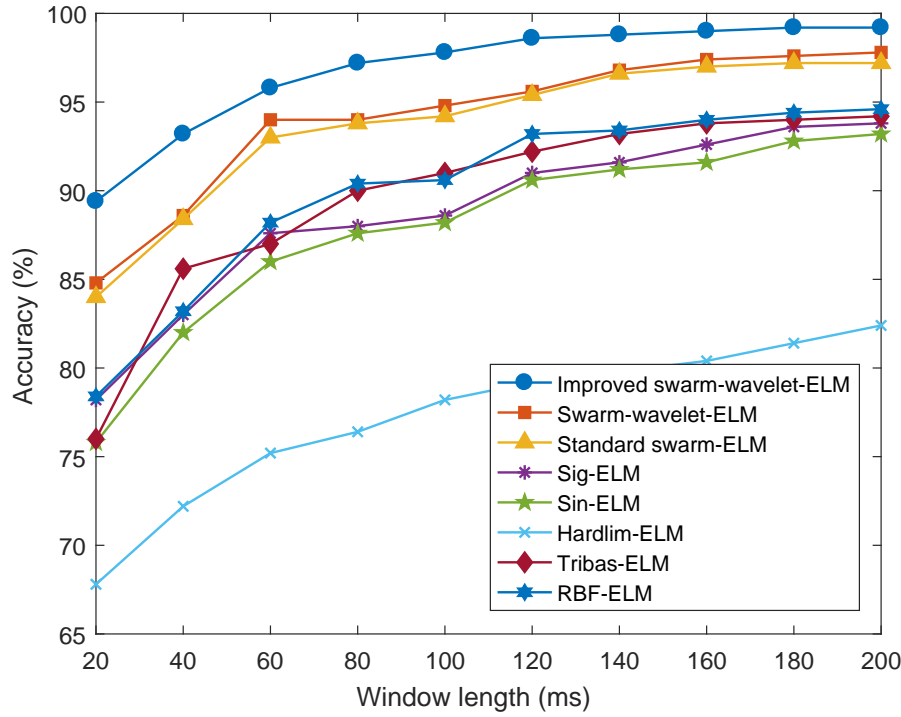


Fig. 7: The average accuracy of the different window length experiments.

swarm-ELM. Moreover, RBF-ELM achieved the highest accuracy among other ELM classifiers without optimization with 200 ms window length.

5.3 Final Result and Convergence Rate

In this experiment, the final result of hand gesture classification is presented as well as the convergence rate of three different swarm optimization methods. Figure 8 shows the accuracy of different swarm-ELM methods using five-fold cross-validation. The number of channels used in this experiment is eight channels with 200 ms window length and using RMS, WL, and MAV feature set.

Figure 8 shows that the improved swarm-wavelet-ELM achieved the highest accuracy on every fold in spite that the swarm-wavelet-ELM also reach the same accuracy with the improved swarm-wavelet-ELM on fold 3 and fold 5. Actually, all three methods get the same accuracy on fold 5 with accuracy of 100%. Figure 9 presents the comparison between different swarm-ELM methods and other well-known classifiers such as k -nearest neighbors (kNN), support vector machine (SVM) type 1 (C-SVM), SVM type 2 (Nu-SVM), C5.0, and classification and regression trees (CART). The node-based ELM using 100 hidden neurons and radial basis function (RBF) as the activation func-

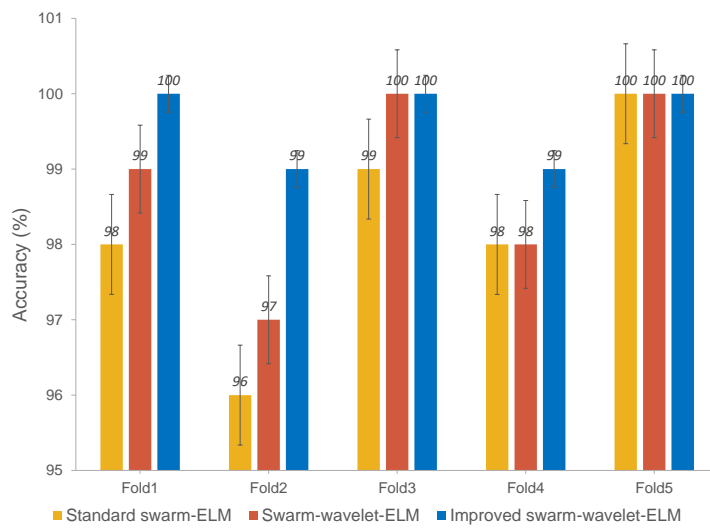


Fig. 8: Accuracy of different swarm-ELM methods using five-fold cross-validation.

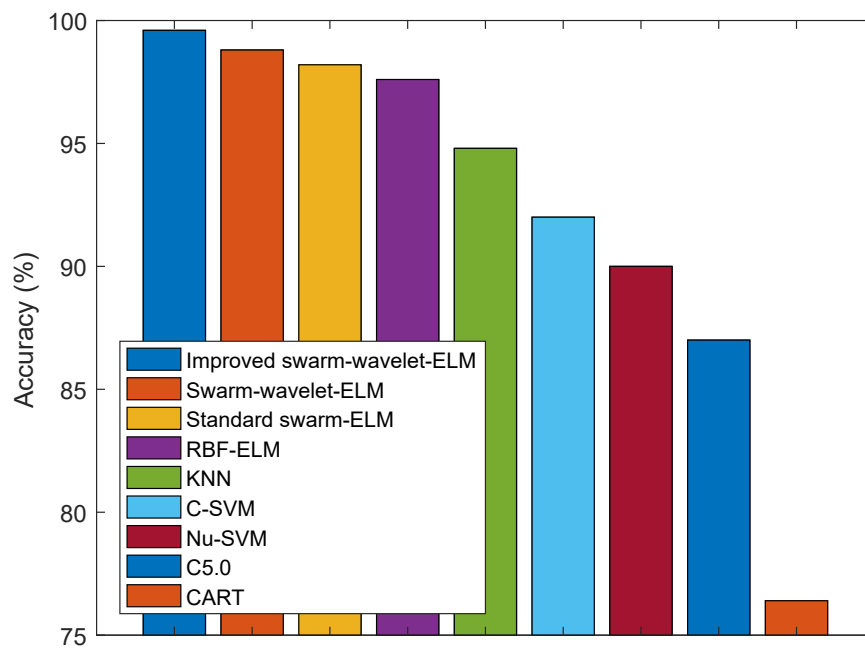
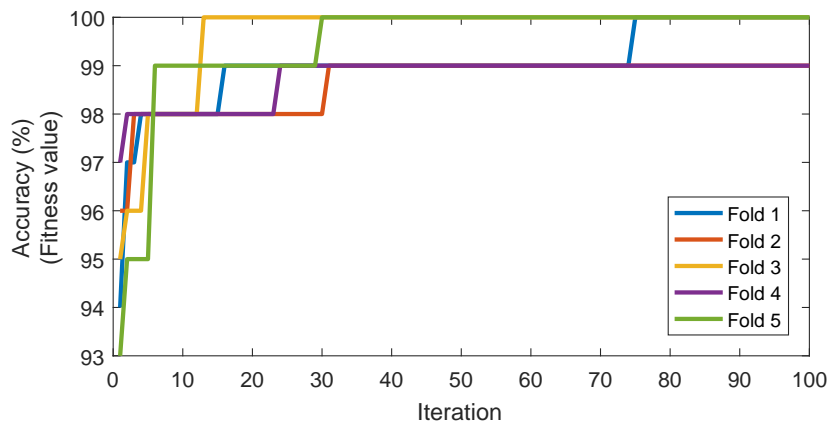
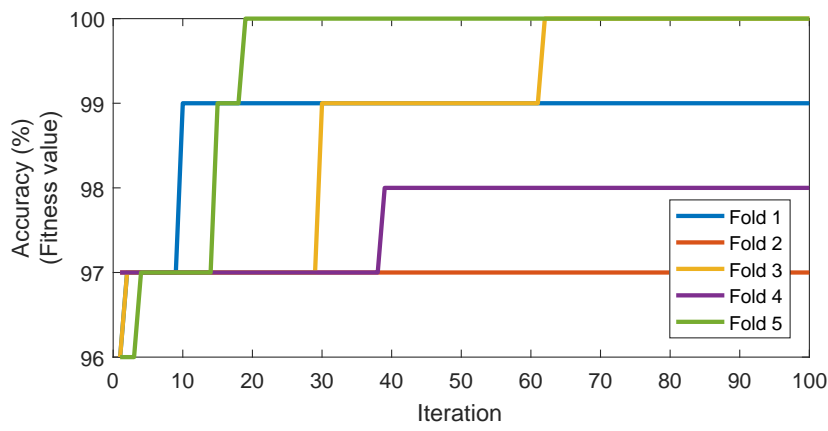


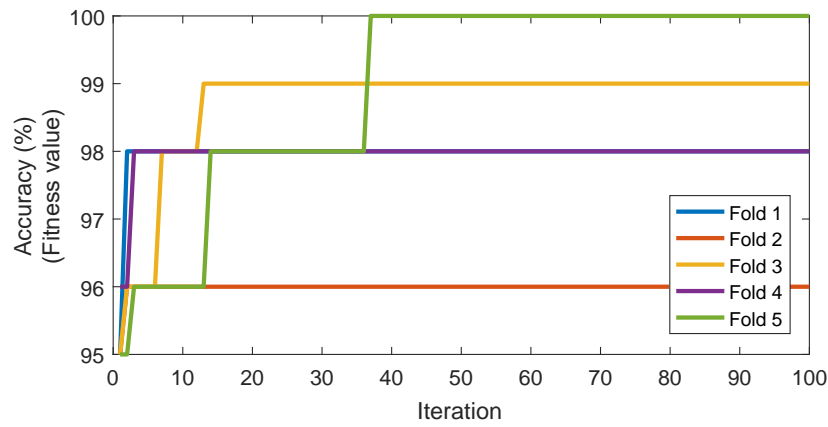
Fig. 9: Average accuracy comparison of different swarm-ELM methods and other well-known classifiers using five-fold cross-validation.



(a) Improved swarm-wavelet-ELM



(b) Swarm-wavelet-ELM



(c) Standard swarm-ELM

Fig. 10: The convergence rate of different swarm-ELM methods for hand gesture classification.

tion achieved the highest accuracy among all node-based ELM without any optimization method. The kNN classification is run using library class [37] in R with $k = 10$, while the e1071 package [33] is used to deal with multi-class SVM classifications. The C5.0 and CART classifications are using C50 [34] and rpart [35] package for R, respectively. All of the parameters used in those classifiers are set to the default value from the R package except for kNN which used 10 neighbors for parameter k .

Moreover, the overall result presented in Figure 9 shows that the highest accuracy for myoelectric pattern recognition was achieved by the improved swarm-wavelet-ELM with an average accuracy of 99.6%. The convergence rate comparisons among different swarm-ELM methods was presented in Figure 10.

6 Conclusion

In this pattern recognition system, all fundamental parts were estimated using experimental protocols. Thus, a robust pattern recognition system that can work well on hand gesture classification had been achieved. We evaluated the implementation of ELMs to recognize hand gestures using myoelectric signals. However, this study only investigates the classification of offline system. In spite of the fact that the subjects of our experiments were healthy with completely intact limbs rather than amputee subjects, the results of [21] and [38] shows that our results by no means invalidated the reached conclusions in the design of myoelectric pattern recognition.

In conclusion, this study presented the analysis of different optimization methods for ELM in MPR system. The proposed recognition system, which employs a hybrid scheme using the improved swarm-wavelet to optimized the node-based ELM, was able to recognize five different hand gestures on ten subjects with the mean accuracy of 99.6% using eight EMG channels and 200 ms window length. The other optimization methods for ELM, the swarm-wavelet achieved the mean accuracy of 98.8% and the standard swarm method achieved the mean accuracy of 98.2%. In average, the improved swarm-wavelet-ELM gives better performance than the ELM method optimized by the standard swarm or the swarm with wavelet mutation. Based on the promising results, the proposed system has a high potential to be implemented on a real-time application. Current limitation points to limited range of varieties utilized in ELM and ISW sub-methods, such as using several different mother wavelets and different optimization for ELM. We are considering these improvements for our future works.

Appendix A

The notations are summarized as follows:

(x, y)	inputs for ELM
\mathbf{L}	number of hidden nodes
\mathbf{G}	activation function for ELM
a, b	hidden node parameters for ELM
β	weight vector
β	ELM's output weights
\mathbf{H}	ELM's output matrix
\mathbf{H}^\dagger	Moore-Penrose generalized inverse
\mathbf{C}	parameter specified by the user for calculating β
Ω_{ELM}	kernel matrix for ELM
\mathbf{K}	kernel matrix consisting kernel functions
v	particle's velocity in the swarm
q	particle's position in the swarm
r	random number
p_i	(<i>pbest</i>) the best previous <i>i</i> th particle position
p_g	(<i>gbest</i>) the best particle among all particles
c_1, c_2	acceleration coefficient
χ	constriction coefficient
σ	mother wavelet
α	dilatation parameter
ζ_{wm}	shape parameter
g	upper limit of the parameter α
v_g	global best particle's velocity
v_{nb}	the best velocity strategy in its neighbor
d_1, d_2	weighting factors

References

1. TouchBionics, "The big picture: Bionic hand," *In Spectrum*, vol. 44, no. 22, 2007.
2. E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *Journal of rehabilitation research and development*, vol. 48, no. 6, p. 643, 2011.
3. N. Uchida, A. Hiraiwa, N. Sonehara, and K. Shimohara, "Emg pattern recognition by neural networks for multi fingers control," in *Engineering in Medicine and Biology Society, 1992 14th Annual International Conference of the IEEE*, vol. 3, pp. 1016–1018, IEEE, 1992.
4. G. Tsenov, A. Zeghibib, F. Palis, N. Shoylev, and V. Mladenov, "Neural networks for online classification of hand and finger movements using surface emg signals," in *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, pp. 167–171, IEEE, 2006.
5. F. V. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *IEEE transactions on biomedical engineering*, vol. 56, no. 5, pp. 1427–1434, 2009.
6. M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE transactions on biomedical engineering*, vol. 55, no. 8, pp. 1956–1965, 2008.
7. S. W. Lee, K. M. Wilson, B. A. Lock, and D. G. Kamper, "Subject-specific myoelectric pattern classification of functional hand movements for stroke survivors," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 5, pp. 558–566, 2011.
8. S. Micera, J. Carpaneto, and S. Raspopovic, "Control of hand prostheses using peripheral information," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 48–68, 2010.
9. Y. Xu and Y. Shu, "Evolutionary extreme learning machine-based on particle swarm optimization," *Advances in Neural Networks-ISNN 2006*, pp. 644–652, 2006.
10. J. Cao, Z. Lin, and G.-B. Huang, "Self-adaptive evolutionary extreme learning machine," *Neural processing letters*, pp. 1–21, 2012.
11. S.-H. Ling, H. H. Iu, K. Y. Chan, H.-K. Lam, B. C. Yeung, and F. H. Leung, "Hybrid particle swarm optimization with wavelet mutation and its industrial applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 3, pp. 743–763, 2008.
12. K. Kawakami and Z. Meng, "Improvement of particle swarm optimization," *PIERS Online*, vol. 5, no. 3, pp. 261–264, 2009.
13. Y. Wang and J. Liu, "Velocity improvement-particle swarm optimization," in *Foundations of Intelligent Systems*, pp. 1133–1142, Springer, 2014.
14. G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
15. Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.
16. D. Serre, *Matrices: Theory and applications*. Graduate texts in mathematics, Springer, 2002.
17. C. R. Rao and S. K. Mitra, "Generalized inverse of matrices and its applications," 1971.
18. G.-B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.
19. J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. IV, pp. 1942–1948, 1995.
20. S.-H. Park and S.-P. Lee, "Emg pattern recognition based on artificial intelligence techniques," *IEEE transactions on Rehabilitation Engineering*, vol. 6, no. 4, pp. 400–405, 1998.
21. B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
22. M. Zardoshti-Kermani, B. C. Wheeler, K. Badie, and R. M. Hashemi, "Emg feature evaluation for movement control of upper extremity prostheses," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 4, pp. 324–333, 1995.

23. D. Farina and R. Merletti, "Comparison of algorithms for estimation of emg variables during voluntary isometric contractions," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 337–349, 2000.
24. A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "A novel feature extraction for robust emg pattern recognition," *arXiv preprint arXiv:0912.3973*, 2009.
25. K. A. Farry, I. D. Walker, and R. G. Baraniuk, "Myoelectric teleoperation of a complex robotic hand," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 775–788, 1996.
26. K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 50, no. 7, pp. 848–854, 2003.
27. K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.
28. T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 15, no. 1, pp. 111–118, 2007.
29. M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews™ in Biomedical Engineering*, vol. 30, no. 4-6, 2002.
30. M. A. Oskoei and H. Hu, "Myoelectric control systems—a survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.
31. K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Medical engineering & physics*, vol. 21, no. 6, pp. 431–438, 1999.
32. D. Tkach, H. Huang, and T. A. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, p. 21, 2010.
33. D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien, 2017. R package version 1.6-8.
34. M. Kuhn, S. Weston, N. Coulter, and M. C. C. code for C5.0 by R. Quinlan, *C5.0: C5.0 Decision Trees and Rule-Based Models*, 2015. R package version 0.1.0-24.
35. T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive Partitioning and Regression Trees*, 2017. R package version 4.1-11.
36. L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, "Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 186–192, 2011.
37. W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*. New York: Springer, fourth ed., 2002. ISBN 0-387-95457-0.
38. Y. Huang, K. B. Englehart, B. Hudgins, and A. D. Chan, "A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1801–1811, 2005.