Automated Hand Gesture Recognition System Using Electromyography Signal and Empirical Mode Decomposition Technique

MARY VASANTHI S[1*], JAYASREE T[2], HAITER LENIN A[3]

*[1]*Department of Electronics and Communication Engineering, St Xavier's Catholic College of Engineering, Nagercoil, Tamilnadu - 629003, India.*

[2]Department of Electronics and Communication Engineering, Government College of Engineering, Tirunelveli, Tamilnadu - 627001, India.

[3]School of Mechanical and Chemical Engineering, WOLLO University, Kombolcha Institute of Technology, Kombolcha, Post Box No 208, Ethiopia.

**Corresponding Author: Mary Vasanthi S[1*]*

1

Abstract

Classification of Electromyography signals is used in the applications like myoelectric control and prosthetics. In biomedical engineering, these signals are fed as input for controlling prosthetic devices. Accurate classification of these signals is a tough task. In this paper, we propose a simple technique for classifying these signals efficiently using Empirical Mode Decomposition (EMD). Four state-of art classifiers like Artificial Neural Network (ANN), Cascaded ANN (CANN), Deep Neural Network (DNN) and Support Vector Machine (SVM) are selected to perform classification. The proposed method is validated using finger movement and hand grasp datasets. The experimental results and performance comparison of classifiers are also presented.

Keywords: *Electromyography (EMG), Empirical Mode Decomposition (EMD), Feature Extraction, Hand movement Classification, Artificial Neural Network (ANN).*

Introduction

A hand amputee person could perform basic hand gestures with the help of an exoskeleton prosthetic hand (EPH) [1]. But, controlling the EPH to perform hand gestures precisely is still a complex task [2]. Muscular contractions generate myoelectric potentials and they provide information about the extent of the muscle activity. Hence they are used to generate control commands for bio control applications like upper limb prosthetics [3, 4]. Surface Electromyography (EMG) signals have been exploited for pattern recognition of hand gestures of healthy subject [5]. It is easier for a hand amputee to wear a glove that comprises the EMG electrodes than using Electroencephalography (EEG) electrodes in the head [6]. This increases the use of EMG in clinical and biomedical applications such as human machine-interaction [7].

EMG signals retrieved using surface electrodes contain noise like electromagnetic and electrode-skin interface noise etc. [8]. Hence, accurate processing and classification of EMG signals is needed to achieve reliable results for myoelectric control [9]. The classification performance mainly depends on feature extraction and pattern recognition [10].

Different machine learning algorithms developed for pattern recognition have been generally unreliable and computationally complex [11]. Classifiers such as ANN [12], Linear discriminant analysis (LDA) [13], SVM [14] and self-organizing map (SOP) [15] were also used for this purpose. However, the existing methods introduced in the past decades have anyone of the following drawbacks such as: (1)

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963
Using increased number of EMG channels for recognition (2) Considering only a small number of movements (3) Low classification accuracy [16].

In this research work, to classify hand gestures using EMG signal, a computationally simple technique which does not compromise the classification accuracy and does not require any intrusive devices is proposed. The main contributions of this work are summarized below.

- To propose an efficient and simple technique based on Empirical Mode Decomposition (EMD) and to utilize the supervised classifiers to classify the hand movements from EMG signal.
- To analyse the performance and to validate the results of the proposed technique using two different datasets such as: Hand grasp dataset and Finger movement dataset.

Related works

This section reviews some of the recent works on hand gesture classification using SVM and Machine learning techniques.

Machine Learning based hand movement Classification

Machine learning algorithms have been successfully applied in many fields of research like Face recognition, Automatic recognition of a musical gesture, Automatic road-sign detection and Classifying human physical activity from on-body accelerometers. K-Nearest Neighbor (k-NN) was used in [17]. In this classifier, for calculating the distance, the standard Euclidean distance was normally used; however other metrics can be used as in [18]. An artificial neural network (ANN) used in [19] is a mathematical model for simulating the structure of biological neural systems. Features are fed as inputs to the ANN and they produce decisions as outputs. ANN was also used for gesture recognition in [20], for robotic soccer formations in [21], for traffic sign recognition in [22] and for hand gesture recognition to enable human-computer interaction in [23]. SVM is a technique that relies on statistical learning theory and it works very well with high-dimensional data [24]. It is used in real-time hand gesture recognition for human computer interaction [25]. SVM was used in conjunction with Self Organizing Maps (SOM) for the automatic learning of pattern recognition [26].

Deep learning based hand movement Classification

Deep learning algorithms automatically learn discriminant features from large amounts of data. Learning discriminant features from large datasets by applying transfer learning on data retrieved from multiple users was proposed in [27]. A systematic feature learning method for Human Activity Recognition using Convolutional Neural Networks (CNN) was proposed in [28]. Deep Neural Networks use raw EMG signals with intrinsic feature extraction capabilities and the classification was performed using CNN in [29]. The deep learning method for controlling prosthetic hands using surface EMG using many intact subjects and amputees was tested in [30].

EMD based hand movement classification

The classification approach consists of few discrete stages. Initially several EMG recordings were collected and the input data is prepared for processing. Then a preprocessing stage was involved to improve the quality of the dataset by excluding the presence of noise in the dataset. For this purpose, a simple Discrete Wavelet Transform (DWT) was used in this work. The wavelets acquired after performing DWT is then decomposed into N number of Intrinsic Mode functions (IMF) by applying EMD. A feature vector is created by extracting the available information from the IMFs which are further used for classification. Fig.1 shows the schematic representation of the proposed work.

Volume 91, No. 5, 2022

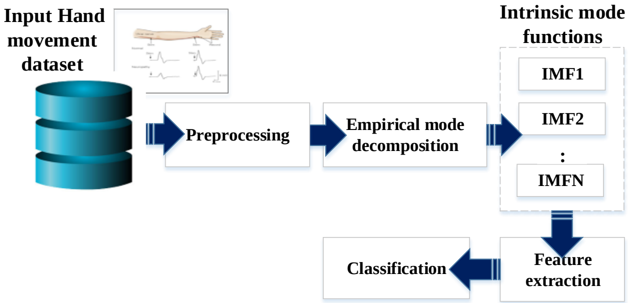


Fig. 1 Schematic Representation of the proposed work

A detailed description of this mentioned process is presented in the rest of this section.

Data Preparation

The input dataset used for this work is obtained by placing the electrodes on the group of muscles of individuals and each-individual are asked to produce continuous movements. The individuals are asked to repeat the same movements to create the dataset. If l numbers of hand movements are repeated by k number of individuals, then the matrix representation of the created dataset D can be written as:

$$D = \begin{bmatrix} X_{11} & X_{21} & .. & X_{l1} \\ X_{12} & X_{22} & .. & X_{l2} \\ : & : & : & \\ X_{1k} & X_{2k} & .. & X_{lk} \end{bmatrix} \quad (1)$$

where X_{lk} is the l^{th} hand movement created by the k^{th} user.

Pre-processing

Preprocessing mainly involves the process of denoising the input dataset in which the noise and unwanted redundancies in the input are eliminated. DWT is used for this purpose. The DWT of the dataset D can be written as:

$$DWT(D) = \frac{1}{2kl} \sum_{i=1}^{k} \sum_{j=1}^{l} D_{ij} \text{ where } \begin{matrix} i = 1,2,...k \\ j = 1,2,...l \end{matrix} \quad (2)$$

DWT requires decomposition of noised input signal to get the wavelet coefficients of the signal. The algorithmic steps used in DWT are explained in Fig.2.

https://doi.org/10.37896/pd91.5/91536

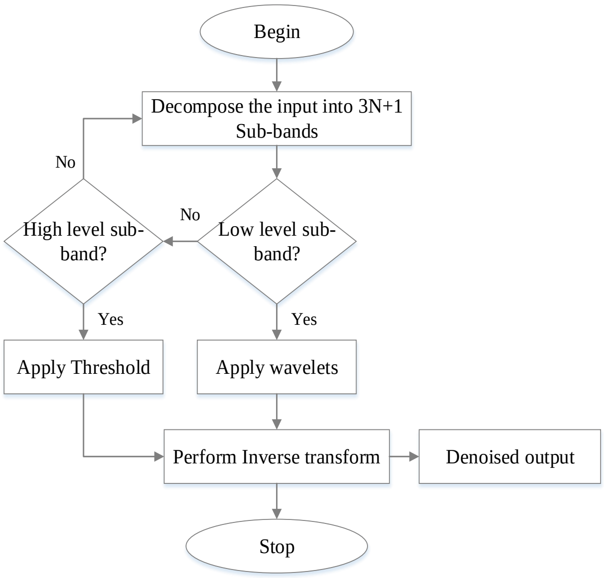


Fig. 2 Process of DWT

Empirical Mode Decomposition (EMD)

Classification of hand gestures is a non-linear process which is challenging to decompose it using decomposition algorithms without misclassification. Empirical Mode Decomposition can be used to solve this problem in which the hand movements are considered as non-linear and non-stationary.

When EMD is applied to the denoised output $DWT(D\ )$, N number of IMFs is created and the process is termed as decomposition. EMD is applied to each data present in the dataset.
If EMD is applied to $X_{lk}(t)$, which is the l^{th} denoised hand movement created by the k^{th} user at time t, then the upper and lower envelope of the denoised input can be calculated as:

$$U_E = e_{\max}\{X_{lk}(t)\} \qquad (3)$$

$$L_E = e_{\min}\{X_{lk}(t)\} \qquad (4)$$

where L_E is the lower envelope and U_E is the upper envelope.

The mean of the denoised input $\mu\{X_{lk}(t)\}$ can be calculated using Equation (3) and Equation (4) as:

$$\mu\{X_{lk}(t)\} = \frac{U_E + L_E}{2} \qquad (5)$$

To get the decomposed output $d(t)$, the mean calculated from Equation (5) is subtracted from the input as:

$$d(t) = X_{lk}(t) - \mu\{X_{lk}(t)\} \qquad (6)$$

In practical, Equation (6) may not be valid. So, to create the original IMFs, the process of shifting takes place. To assure that IMF components preserve satisfactory physical sense of both amplitude and frequency modulation, we need to determine the threshold value δ for shifting. This is carried out by restricting the size of the standard deviation computed from the two consecutive shifting results. Usually, δ is set between 0.2 and 0.3.

ISSN: 0369-8963

The standard deviation $\sigma_i(t)$ of the input data $X_{ij}(t)$ can be computed as:

$$\sigma_i(t) = \frac{\sum_{i=1}^{k} | h_{i-1}(t) - h_i(t) |^2}{h_{i-1}(t)^2} \text{ where } h_i(t) \leftarrow r_i(t) \quad (7)$$

where $r_i(t)$ is the residue of input data.

The shifting is repeated several times to obtain true IMFs until the following stopping criteria met.

1) The number of extrema and the number of zero intersections may not vary by more than 1.
2) The normal estimation of the envelope characterized by the nearby maxima and the envelope characterized by the local minima is zero.

The algorithmic steps of EMD are explained below.

EMD Algorithm
Input: *Denoised data* ***Output:*** *IMFs of denoised data* ***Begin*** ***Step 1:*** *Initialize number of IMFs to be created, local minima and local maxima of the input.* ***Step 2:*** *Perform interpolation to calculate upper and lower envelopes.* ***Step 3:*** *Calculate mean using Eqn. (5).* ***Step 4:*** *Decompose the input using Eqn. (6)* ***Step 5:*** *If step 4 is not valid, perform shifting to create the IMFs.* ***Step 5.1:*** *Set a threshold value and extract the residue from the input.* ***Step 5.2****: Calculate* $h_i(t)$ ***Step 5.3:*** *Calculate standard deviation using Eqn. (7).* ***Step 5.4: If*** $\sigma_i < \delta$ *check if* $r_i(t) \leq N$ ***Step 5.5:*** *do* $IMF_i(t) \leftarrow h_i(t)$ ***Step 5.6:*** *else do* $h_i(t) \leftarrow h_{i+1}(t)$, $r_i(t) \leftarrow r_{i+1}(t)$ ***Step 5.7:*** *repeat steps 5.1 to 5.6 until the stopping criteria is met.* ***Step 6:*** *Repeat steps 2 to 4 until the desired number of IMFs created.* ***End***

At the end of EMD process, the decomposed signal may be expressed as follows:

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963

$$d\{X_{lk}(t)\} = \sum_{i=1}^{k} \sum_{j=1}^{l} IMF_{ij}(t) + r_{ij}(t) \tag{8}$$

This creates N number of IMFs which can be processed independently. The IMFs extracted from the denoised input can be written in the form of array as:

$$IMF(X_{lk}) = \begin{bmatrix} IMF_{1,1}, IMF_{2,1}, \ldots IMF_{l1} \\ IMF_{1,2}, IMF_{2,2}, \ldots IMF_{l2} \\ : \\ IMF_{1k}, IMF_{2,k}, \ldots IMF_{lk} \end{bmatrix} \tag{9}$$

Feature Extraction

From the created IMFs, features such as Mean, Standard deviation, Energy, Entropy and their corresponding first and second order derivatives are extracted. These extracted features form a feature vector which can be used as input for the classifiers. The mathematical expressions used to calculate these 12 features are presented below from Equation (10) to Equation (21).

3.1.1 Mean:

$$M = \frac{1}{N} \sum_{i=1}^{N} X_i^2 \tag{10}$$

3.1.2 Standard Deviation:

$$STD = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} X_i^2} \tag{11}$$

3.1.3 Energy:

$$E = \sqrt{\sum_{i=1}^{N} X_i^2} \tag{12}$$

3.1.4 Entropy:

$$E = \sum_{i=1}^{N} X_i^2 - \ln(X_i)^2 \tag{13}$$

3.1.5 First oder derivative of Mean:

$$FM = \frac{d(M)}{dN} \tag{14}$$

3.1.6 First order deivtaive of Standard deviation:

$$FSTD = \frac{d(STD)}{dN} \tag{15}$$

3.1.7 First order derivative of energy:

$$FE = \frac{d(E)}{dN} \tag{16}$$

3.1.8 First order derivative of entropy:

$$FEn = \frac{d(En)}{dN} \tag{17}$$

3.1.9 Second order derivative of mean:

$$SM = \frac{d(FM)}{dN} \tag{18}$$

3.1.10 Second order derivative of standard deviation:

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963

$$SSTD = \frac{d(FSTD)}{dN} \tag{19}$$

3.1.11 Second order derivative of energy:

$$SE = \frac{d(FE)}{dN} \tag{20}$$

3.1.12 Second order derivative of entropy:

$$SEn = \frac{d(FEn)}{dN} \tag{21}$$

The feature vectors created for N number of IMFs can be written as:

$$\begin{aligned} F_i(N) = \{ & M_i, STD_i, E_i, En_i, \\ & FM_i, FSTD_i, FE_i, FEn_i, \\ & SM_i, SSTD_i, SE_i, SEn_i \} \end{aligned} \tag{22}$$

where $F_i(N)$ is the feature vector created for the i^{th} IMF and $i = 1,2 \ldots N$.

Classification

The feature vectors created using Equation (22) are further used for classification. For effective classification four state-of-art classifiers such as Artificial Neural Network (ANN), Cascaded ANN (CANN), Deed Neural Network (DNN) and Support Vector Machine (SVM) are selected in this work.

Artificial Neural Network (ANN):

Artificial Neural Network acts like a human brain. It consists of number of neurons and mainly three layers (input, hidden and output) in it. Fig. 3 shows the structure of ANN used.

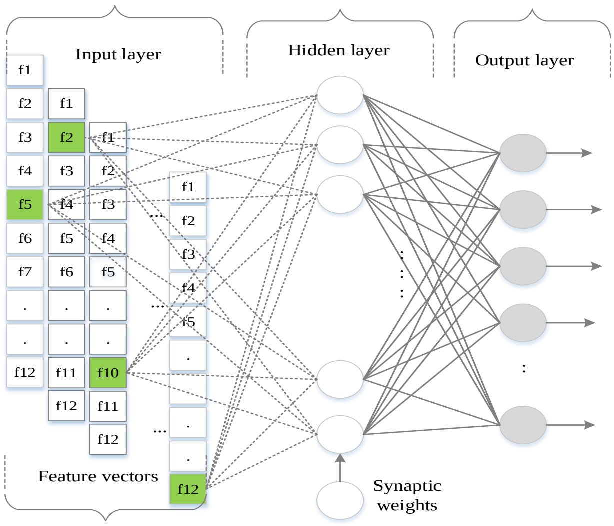


Fig. 3 Structure of ANN used in the proposed work

An input is given to an ANN, and an output is produced utilizing its activations and learning methods. With an input that was not used throughout its training, it can produce a desired output.

The synapses transmit the values from the input layer to the subsequent layer. The input vector for the first layer (input) is [$F_i(N)$]. The inputs for the following layer's neuron are created by multiplying each

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963

component of this vector by a synaptic weight $w_i(N)$ and then adding all of the results. This input is subjected to the activation function, which produces the neuron output. The calculation for each neuron's output is:

$$Y_i(N) = \nu\left(\sum_{i=1}^{N} F_i(N-1) * w_i(N-1)\right) \qquad (23)$$

where ν is the sigmoid function used as the activation function of the neurons and is given by,

$$\nu = \frac{1}{1+e^{x}} \qquad (24)$$

The neuron output $Y_i(N)$ is the targeted output of ANN which is calculated by Feed Forward algorithm in this work.

Cascaded ANN (CANN):

Cascaded ANN is a series connection of ANN in which the output of the ANN is supplied as input to the next ANN connected in series. This reduces the classification error as resulted in ANN. The output of Cascaded ANN depends on the output of each ANN block. The structure of Cascaded ANN is shown in Fig. 4.

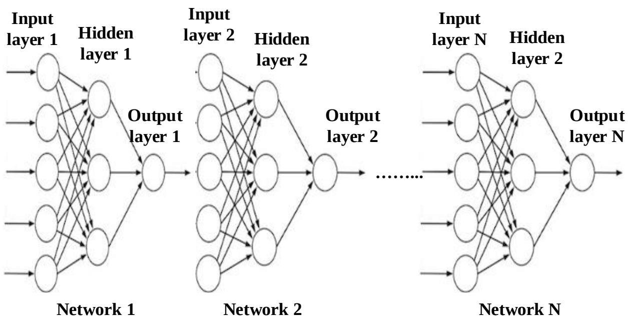


Fig. 4 Structure of Cascaded ANN

The output of cascaded ANN can be written as:

$$CNN_{output} = \sum_{i=1}^{n} Y_i(N) \qquad (25)$$

Deep Neural Network (DNN):

A deep architecture based on Convolutional Neural Network (CNN) is used in this work. The structure of DNN used in this work is shown in Fig.5.

 https://doi.org/10.37896/pd91.5/91536

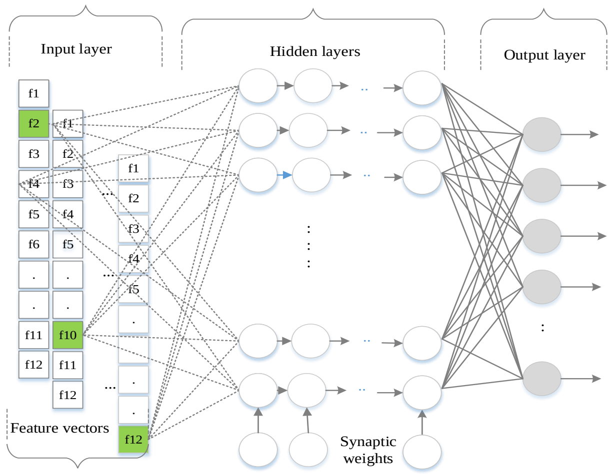


Fig. 5 Structure of DNN

DNN is a variant of ANN that includes a number of hidden layers. They operate using three variables, including pooling, shared weights, and the local receptive field. In the convolutional layer, filters are combined with input patches (receptive field) in such a way that each individual filter uses the same learning weights across all patches. The activation unit sends the dot product of filters with patches, and pooling reduces the output size. DNN uses multiple layers to deeply learn the features or inputs.

Support Vector Machine(SVM):

SVM is a binary classifier that uses supervised learning to make classification decisions. SVM uses feature space mapping to convert input data into solutions for non-linear problems. In feature space, the mapping information can typically be separated. To steer the training feature vector into the high-dimensional kernel feature space, SVM needs a function data $\phi(x_i)$:

$$K(x_i, y_i) = \phi(x_i)^T \phi(x_j) \qquad (26)$$

The group of data that is mapped in high dimensional linear space is divided into two labeled classes by a hyper plane. The conclusion can be taken as:

$$D(x) = \text{sgn}\left\{ \sum_{i=1}^{n} \alpha_i y_i K(x_i, x\ ) + b \right. \qquad (27)$$

where α_i is the Lagrangian multiplier. The portion capacity utilized for isolating examples is a quadratic capacity having box limitation level equivalent to 1. SVM is a methodology of area isolating with ideal hyperplanes in a multidimensional information space. In any case, SVM is utilized for the separation undertakings to get a parallel subjective variable. Its motivation is to discover probabilistic cutoff points. These breaking points are intended to limit blunders and boost the edges of partition to accomplish ideal hyperplanes.

Simulation Results

This section presents the simulation results and performance analysis of the proposed work.

Dataset Description

To validate the proposed work, we use two datasets namely Finger movement dataset and Hand grasp dataset. The finger movements dataset contains 10 movements of n individuals such as Hand close (HC),

 https://doi.org/10.37896/pd91.5/91536

Thumb (T), Little (L), Middle (M), Index (I), Ring (R), Thumb-Index (TI), Thumb-Little (TL), Thumb-Middle (TM) and Thumb-Ring (TR). Similarly, the Hand grasp dataset contains 6 movements of n individuals such as Palmar Class (PC), Hook Class (HC), Spherical Class (SC), Lateral Class (LC), Tip Class (TC) and Cylindrical Class (CC). Each data in the dataset were retrieved at a sampling rate of 500 Hz.

Experimental setup

The two datasets used in this work are created by placing the electrodes on the muscles of individuals. Fig. 6 shows the experimental setup of dataset creation.

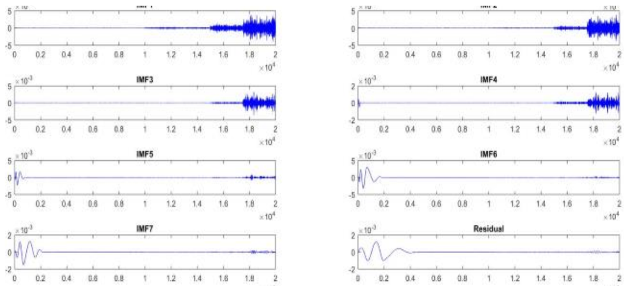
Fig. 6 Experimental setup for dataset creation

Simulation Results

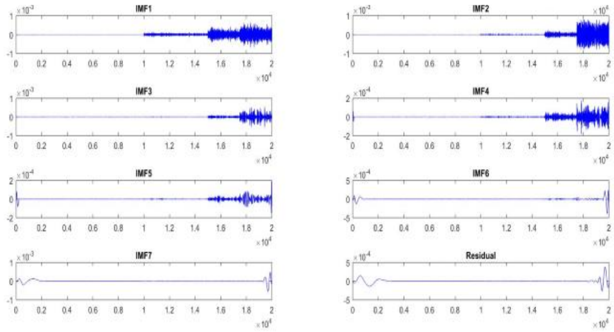
The four selected classifiers are tested with both datasets.

Fig. 7 (a) shows the IMFs obtained for Hand Close (HC) movement and 7(b) shows the IMFs obtained for Thumb (T) movement taken from Finger movement dataset.

Fig. 8 (a) shows the IMFs obtained for Palmer Class (PC) movement and 8(b) shows the IMFs obtained for Hook Class (HC) movement taken from Hand grasp dataset.

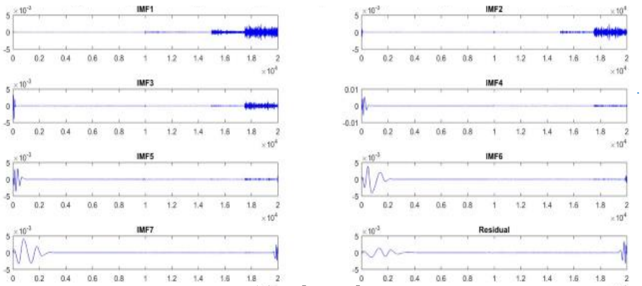


(a) Handclose

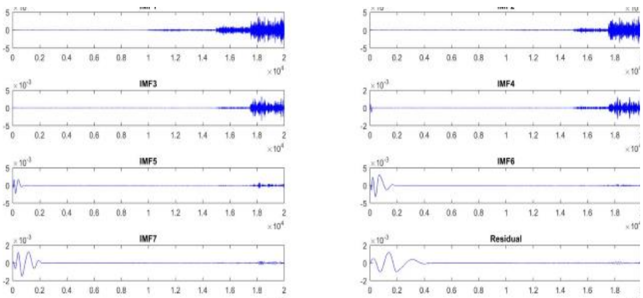


(b) Thumb

Fig. 7 IMFs of Finger movement dataset



(a) Palmer class



(b) Hook class

Fig. 8 IMFs of Hand grasp dataset

In this work, the input data are decomposed into 7 IMFs.

Features such as Mean, Standard deviation, Energy, Entropy and their corresponding first and second order derivatives are extracted from the IMFs. From the features of each input data, the classifiers can easily classify the hand gestures. Fig. 9 and 10 shows the classified output images of the finger movements.

ISSN: 0369-8963

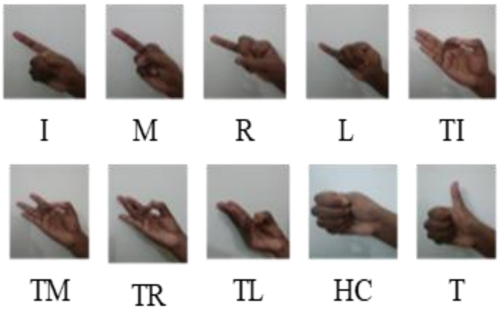


Fig. 9 Finger movements

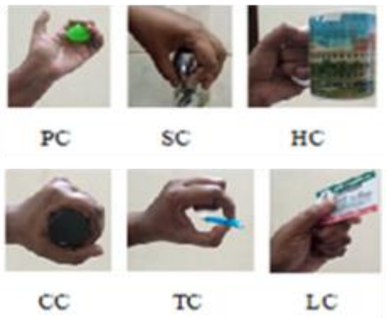


Fig. 10 Hand grasp movements

Each classifier performs its own way to produce the classification results. The performance of the classifiers is analyzed and compared in the next section.

Performance Analysis

The performance of the four selected classifiers is evaluated in terms of accuracy, sensitivity and specificity.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (28)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (29)$$

$$Specificity = \frac{TN}{TN + FP} \quad (30)$$

where TN is True negative (correctly rejected), TP is True positive (correctly identified), FN is False Negative (incorrectly rejected) and FP is False Positive (incorrectly identified).

Performance of the four selected classifiers for both Finger movement and Hand grasp dataset are presented in Table 1.

Table 1 Performance of the classifiers

Dataset Used Parameters		**CLASSIFIERS USED**			
		ANN	**CANN**	**DNN**	**SVM**
Finger movement	Accuracy (%)	89.92	87.69	89.81	98.77
	Sensitivity (%)	80.09	90.51	86.76	91.64

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963

dataset	Specificity (%)	91.11	85.56	84.48	93.35
Hand grasp dataset	Accuracy (%)	94.01	86.16	89.67	99.69
	Sensitivity (%)	81.16	83.95	87.27	92.79
	Specificity (%)	90.43	93.34	85.25	93.92

It is clear from Table 1 that SVM achieves the highest accuracy of 98.771% for the dataset of finger movement and 99.694% for the dataset of hand grasp. Similarly, SVM shows the optimum results in terms of sensitivity and specificity. This demonstrates that SVM classifier is better than the chosen classifiers (ANN, CANN and DNN).

Conclusion

The results of hand gesture classification using myoelectric signals with different traits have been presented. In this work, four different classifiers such as ANN, CANN, DNN and SVM are used for classification.

- The suggested work decomposes the input signal using EMD, which simplifies the proposed classification scheme. The outcome of the experiment demonstrates that the SVM classifier is more effective than the other classifiers at differentiating the desired movements.
- The finger movement dataset and the hand grasp dataset were successfully classified by the SVM classifier, on average, with success rates of 98.77% and 99.69%, respectively. As a result, our suggested approach is the greatest option for using prosthetic devices with basic, cutting-edge techniques.

Reference

[1] Andrianesis K and Tzes A 2008 Design of an anthropomorphic prosthetic hand driven by shape memory alloy actuators 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics 517-522.

[2] Carroll D and Subbiah A 2012 Recent Advances in Biosensors and Bio sensing Protocols *Journal of Biosensors & Bioelectronics* **3(3).**

[3] Kaniusas E 2012 Fundamentals of Bio Signals. In Biomedical Signals and Sensors I, Berlin, Heidelberg. Springer.

[4] Kiguchi K and Hayashi Y 2012 A study of EMG and EEG during perception-assist with an upper-limb power-assist robot *In IEEE International Conference on Robotics and Automation 2012 May 14* 2711-2716.

[5] Oskoei M A and Hu H 2007 Myoelectric control systems-A survey *Biomedical Signal Processing and Control* **2(4)** 275-294.

[6] Castellini C Gruppioni E Davalli A and Sandini G 2009 Fine detection of grasp force and posture by amputees via surface electromyography *Journal of Physiology-Paris* **103(3-5)** 255-62.

[7] Delagi E F Perotto A O Iazzetti J and Morrison D 2011Anatomical Guide for the Electromyographer the Limbs and Trunk. Charles C Thomas Publisher.

[8] Saponas T S Tan D S Morris D Balakrishnan R Turner J and Landay J A 2009 Enabling always-available input with muscle-computer interfaces *In Proceedings of the 22nd annual ACM symposium on User interface software and technology* 167-176.

[9] Guo Y Naik G R Huang S Abraham A and Nguyen H T 2015 Nonlinear multiscale maximal Lyapunov exponent for accurate myoelectric signal classification *Applied Soft Computing* **1(36)** 633-40.

[10] De Luca C J 2006 Electromyography Hoboken, NJ, USA John Wiley & Sons.

[11] Reaz M B Hussain M S and Mohd-Yasin F 2006 Techniques of EMG signal analysis: detection, processing, classification and applications *Biological procedures online* **1 8(1)** 11-35.

[12] Lahmiri S and Boukadoum M 2018 Improved Electromyography Signal Modeling for Myopathy Detection. In IEEE International Symposium on Circuits and Systems (ISCAS) 1-4.

[13] Fuglsang-Frederiksen A. 2006 The role of different EMG methods in evaluating myopathy *Clinical neurophysiology* **117(6)** 1173-89.

[14] Dostal O Vysata O Pazdera L Prochazka A Kopal J Kuchynka J Valis M 2018 Permutation entropy and signal energy increase the accuracy of neuropathic change detection in needle EMG *Computational intelligence and neuroscience* **(2018)** 1-5.

[15] Cler M J and Stepp C E 2015 Discrete versus continuous mapping of facial electromyography for human–machine interface control: Performance and training effects. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **23(4)** 572-80.

[16] Brunelli D Tadesse A M Vodermayer B Nowak M and Castellini C 2015 Low-cost wearable multichannel surface EMG acquisition for prosthetic hand control *In 6th international workshop on advances in sensors and interfaces (IWASI)* 94-99.

[17] Viana P L Fujii V S Lima L M Ouriques G L Oliveira G C Varoto R and Cliquet Jr A 2019 An Artificial Neural Network for Hand Movement Classification using Surface Electromyography *Biosignals.*

[18] Burigo A C 2014 Classification of hand movements using surface electromyography, logistic regression, neural networks, support vector machine and NinaPro database, Porto Alegre.

[19] Englehart K and Hudgins B 2003 A robust, real-time control scheme for multifunction myoelectric control *IEEE transactions on biomedical engineering* **50(7)** 848-54.

[20] Hall J E 2010 Guyton and Hall textbook of medical physiology e-Book. Elsevier Health Sciences.

[21] Haykin S. 1994 Neural networks: a comprehensive foundation Prentice Hall PTR.

[22] Pizzolato S Tagliapietra L Cognolato M Reggiani M Muller H and Atzori M 2017 Comparison of six electromyography acquisition setups on hand movement classification tasks *PloS one* **12(10)** 1-17.

[23] Tang X Liu Y Lv C and Sun D 2012 Hand motion classification using a multi-channel surface electromyography sensor *Sensors* **12(2)** 1130-47.

[24] Cheddad A Condell J Curran K and Mc Kevitt P 2010 Digital image steganography: Survey and analysis of current methods *Signal processing* **90(3)** 727-52.

[25] Djebbar F Ayad B Abed-Meraim K and Hamam H 2013 Unified phase and magnitude speech spectra data hiding algorithm *Security and Communication Networks* **6(8)** 961-71.

[26] Djebbar F Abed-Meraim K Guerchi D and Hamam H 2010 Dynamic energy based text-in-speech spectrum hiding using speech masking properties *In IEEE 2nd International Conference on Industrial Mechatronics and Automation* **(2)** 422-426.

[27] Cote-Allard U Fall C L Drouin A Campeau-Lecours A Gosselin C Glette K Laviolette F and Gosselin B 2019 Deep learning for electromyographic hand gesture signal classification using transfer learning *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **27(4)** 760-71.

[28] Yang J Nguyen M N San P P Li X L and Krishnaswamy S 2015 Deep convolutional neural networks on multichannel time series for human activity recognition *InTwenty-Fourth International Joint Conference on Artificial Intelligence*.

 https://doi.org/10.37896/pd91.5/91536

ISSN: 0369-8963
[29] Zia ur Rehman M Waris A Gilani S O Jochumsen M Niazi I K Jamil M Farina D and Kamavuako E N 2018 Multiday EMG-based classification of hand motions with deep learning techniques *Sensors* **18(8)** 2497.

[30] Atzori M Cognolato M and Muller H 2016 Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands *Frontiers in neurorobotics* **10 (9)** 1-10.

 https://doi.org/10.37896/pd91.5/91536