Biomedical Signal Processing and Control 55 (2020) 101669

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

An experimental study on upper limb position invariant EMG signal classification based on deep neural network

Anand Kumar Mukhopadhyay [a], Suman Samui [b,*]

[a] Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India
[b] Advanced Technology Development Centre, Indian Institute of Technology, Kharagpur, India

ARTICLE INFO

Article history:
Received 30 May 2018
Received in revised form 29 April 2019
Accepted 17 August 2019

Keywords:
sEMG signal classification
Deep neural network
Electromyogram
Upper-limb invariant
Hand movement classification
Prosthetic application

ABSTRACT

The classification of surface electromyography (sEMG) signal has an important usage in the man-machine interfaces for proper controlling of prosthetic devices with multiple degrees of freedom. The vital research aspects in this field mainly focus on data acquisition, pre-processing, feature extraction and classification along with their feasibility in practical scenarios regarding implementation and reliability. In this article, we have demonstrated a detailed empirical exploration on Deep Neural Network (DNN) based classification system for the upper limb position invariant myoelectric signal. The classification of eight different hand movements is performed using a fully connected feed-forward DNN model and also compared with the existing machine learning tools. In our analysis, we have used a dataset consisting of the sEMG signals collected from eleven subjects at five different upper limb positions. The time domain power spectral descriptors (TDPSD) is used as the feature set to train the DNN classifier. In contrast to the prior methods, the proposed approach excludes the feature dimensionality reduction step, which in turn significantly reduce the overall complexity. As the EMG signal classification is a subject-specific problem, the DNN model is customized for each subject separately to get the best possible results. Our experimental results in various analysis frameworks demonstrate that DNN based system can outperform the other existing classifiers such as k-Nearest Neighbour (kNN), Random Forest, and Decision Tree. The average accuracy obtained among the five subjects for DNN, SVM, kNN, Random Forest and Decision Tree is 98.88%, 98.66%, 90.64%, 91.78%, and 88.36% respectively. Moreover, it can achieve competitive performance with the state-of-the-art SVM based model, even though the proposed DNN model requires minimal processing in feature engineering. This study provides an insight into the detailed step-by-step empirical procedure to achieve the optimum results regarding classification accuracy using the DNN model.

1. Introduction

The classification of Electromyogram (EMG) signal is an important task for various applications such as clinical diagnosis, Human-Computer Interaction (HCI) systems for prosthetic devices, myoelectric controllers used in wireless controlling devices such as video game controller and security check-in for authorized access [1]. In order to verify whether a person is suffering from any neuromuscular disorder, namely neuropathy and myopathy, the offline classification methods are often employed by collecting myoelectric signals from the diagnosed region of the muscles [2]. However, the real-time prediction is considered to be a far more challenging task in which the real-time signal is processed through the pre-processing blocks involving filters, feature extractor and classifier units in a short period (≈300 ms) to provide the desired output with acceptable accuracy. Therefore, it requires a proper selection of the size of the analysis window for real-time applicability [3–6]. The imposed time constraint is an essential design criterion to ensure the flawless functioning of the HCI systems for the end users.

The shape and firing rate of the Motor Unit Action Potentials (MUAPs), contained in the EMG signal, carry essential information about the characteristic of the signal which is often contaminated by noise. Henceforth, one can exploit the characteristic of MUAPs in analyzing the signal for clinical diagnosis by employing any efficient noise suppression technique which will allow the acquisition of useful content from surface EMG (sEMG) signal. Furthermore, with the advancement of signal processing and machine learning techniques, various new methods have been introduced for the

* Corresponding author.
E-mail addresses: anandmukh@iitkgp.ac.in (A.K. Mukhopadhyay), samuisuman@atdc.iitkgp.ac.in (S. Samui).

https://doi.org/10.1016/j.bspc.2019.101669
1746-8094/

EMG signal classification. In recent years, the supervised learning based methods are advancing by considering mainly three aspects: data collection techniques and its pre-processing, feature extraction methods and suitable classifier design. However, the proper deployment of these methods on the hardware also requires significant research attention for the real-time application [1,7].

A vital aspect lies in the acquisition of data through electrodes from the specific region of muscles of the subject which can either be invasive or non-invasive depending on the end user system. In invasive methods, the prosthetic device needs to be connected directly to the targeted muscle tissues surgically whereas, in non-invasive methods, signals are recorded from the surface of the muscle regions. Invasive methods prove to be successful for real-time control of artificial arms, but they are less preferred compared to the non-invasive case by the end users, so as to avoid the surgical process which may increase the chances of causing infection [8]. Now, for the non-invasive method of extracting the signals, it is quite evident that multi-channel electrodes are more effective compared to a single-channel case because it can cover more regions around the muscle area of a subject. Hence, in this work, we are mainly focusing on the sEMG data collected from the upper limb of the subjects through the multi-electrode channels at different limb positions. During the last decade, the EMG classification task is investigated extensively, which has led to the development of numerous methods which mainly includes various statistically induced mathematical models [9], discriminative learning models [1] and genetic algorithm [10] based techniques. The important approaches for classification of sEMG signals for controlling upper limb prostheses are Linear Discriminant Analysis (LDA), Support Vector Machines (SVMs), and Hidden Markov Models (HMM) which provide a marginal enhancement in performance regarding classification accuracy [11]. Even though these are proven classifiers, their shortcomings are, viz., (1) LDA needs more computational time for determination of eigenvalues which is required to reduce the feature dimension, (2) Though SVM gives good result with a proper combination of simple time domain features [12] such as MAV (Mean Absolute Value), WL (Waveform Length), ZC (Zero Crossing) and SSC (Slope Sign Change) which have a low variance for different choice of window lengths, data segmentation is performed for better results. In addition to that an appropriate choice of kernel for SVM is needed which is a challenging task [13], (3) HMM, on the other hand, does not need segmentation of EMG data and uses root mean square with Auto-Regressive (AR) coefficients as features, has shown higher accuracy than Multi-Layer Perceptron (MLP) and is suitable for real-time implementation [14]. The AR model helps in generating less number of features; and was also used on myoelectric signal collected from a single electrode channel for upper limb prostheses and later feeding it to a Fuzzy Clustering Neural Network Classifier which proved to be better than its counterpart neural networks consisting of MLP, back propagation NN and conic-section function NN [15,16].

The use of artificial neural networks (ANNs) for sEMG signal classification has been attempted earlier for hand gesture recognition, either by combination with Independent Component Analysis (ICA) or by using both time and frequency based features [17,18]. It is to be noted that the sEMG signal can be analyzed either by considering the gross signal or by extracting the useful information contained in the form of MUAPs from it. The gross signal is a combination of signals generated from overlapping muscles whereas the useful information content is obtained by separating the signals generated from the overlapping muscles. However, due to the similarity in the properties of the signals from different muscle regions, the task of separating the overlapping signals becomes a blind source separation (BSS) problem. ICA has been proven to be successful in solving BSS tasks in bio-signal applications [17]. In ICA, some prior knowledge of muscle anatomy is required for ensuring a distinct muscle activity pattern. ICA has been a useful tool for separating MUAPs from multi-channel sEMG [19], viz., to estimate the muscle force from surface HD-EMG with a significant reduction in error [20], to remove the contaminated ECG artifacts by using ICA with wavelet analysis [21,22] and hence is a superior feature extraction technique. However, ICA which is an iterative algorithm will increase the complexity of the feature extraction process and can work only under certain stringent conditions, for instance, the number of recording channels has to be more than or equal to the number of independent source signals and the source signals should be independent of each other.

Recently, a sophisticated deep neural network termed as deep time growing neural network (DTGNN) has been shown to classify biological signals having cyclic time series (CTS) properties with a one-versus-all classification approach [23]. CTS of a signal resembles a time series having repetitive characteristics over a set of successive temporal intervals. The DTGNN classifier can overcome the complexities of the CTS biological signals through the cyclic learning algorithms involving three different schemes namely forward, bilateral and backward time growing schemes which are then fed to a time growing neural network (TGNN) [24]. A TGNN classifier, which is a sub-component of DTGNN has shown better results when compared to time delay neural network (TDNN) and MLP for classification of short-duration heart sounds [24,25]. These classifiers (DTGNN, TGNN, and TDNN) increase the number of weight coefficients depending on the number of window segments for better learning and considers the spectral content of the signals for better performance. However, in this work, we have omitted enhanced feature extraction processes and analyzed the performance of DNN using TDPSD features extracted from the time domain [26] for sEMG classification. We have not chosen sophisticated deep neural networks having the capability to solve CTS problems because it will increase the complexity of the online classifier due to the increased number of weight coefficients and the complex feed-forward classification process. The main advantage of DNN lies in the fact that it does not need a very sophisticated feature extraction pipeline. In other words, it is an excellent tool for classifying multiple classes without undergoing a complex feature extraction process.

In this paper, we have examined the application of deep learning to EMG classification task. A recent work by Atzori et al. have used Convolutional Neural Network for classifying 50 hand movements for 67 intact subjects and 11 trans-radial amputees to find accurate results compared to the other classification methods like k-NN, SVM, Random Forest [27]. Just like any other supervised learning tasks in many other domains [28–30], deep learning has emerged to be a useful tool in the field of medical informatics in recent years due to its remarkable performance [31]. Furthermore, D. Yang et al. did a classification of multiple finger motions considering dynamic upper limb movements [32]. This study differs from previous studies in several ways. The major contributions of this manuscript are given as follows:

- We focused on fully connected deep neural network architecture for the classification of hand movements using EMG signal. For this study, we have used the dataset consisting of the myoelectric signal obtained from the upper limb of different subjects using multi-electrode channels at five different limb positions, which were recorded by Khushaba et al. [33].
- Moreover, most of the previous studies relied on some form of high-level features, while in this paper we strive for minimal signal processing and rely on deep learning to automate the process of feature extraction. We have chosen a fused time domain descriptors (fTDD) method with multiple myoelectric channels while avoiding the feature dimension reduction process [26,34]. fTDD is a fusion of the extracted features from the current and

Table 1
List of abbreviations.

Abbreviations	Full meaning
ANN	Artificial neural networks
AR	Auto regressive
BSS	Blind source separation
cTDD	Correlated time domain descriptors
CTS	Cyclic time series
DNN	Deep neural network
DTGNN	Deep time growing neural network
fTDD	Fused time domain descriptors
HCI	Human Computer Interaction
HMM	Hidden Markov Models
ICA	Independent Component Analysis
IF	Irregularity Factor
kNN	k-Nearest Neighbors
LDA	Linear discriminant analysis
MLP	Multi-Layer Perceptron
MUAPs	Motor unit action potentials
ReLU	Rectified linear unit
ROC	Receiver operating characteristics
sEMG	Surface electromyography
SSC	Slope Sign Change
SVM	Support vector machines
TDNN	Time delay neural network
TDPSD	Time domain power spectral descriptors
TGNN	Time growing neural network
WL	Waveform length
ZC	Zero crossing

previous window for enhancing the robustness of the extracted set of features [26].

- Furthermore, to the best of our knowledge, our work provides the first step-by-step detailed empirical exploration of deep learning methods applied to EMG classification task. Please note that our objective is not to achieve the state-of-the-art performance, but rather to explore the deep learning framework for EMG classification to the other existing EMG classification schemes whose performance often depends on the complex feature set. Further improvement may straightforwardly be attained with larger datasets and more complex models like RNN and CNN.

The paper is organized as follows: Initially, the motivation of classifying myoelectric signal for different applications and a brief survey of work related to it is elucidated in Section 1. In Section 2, a system level overview of sEMG classification in real time is described. The materials comprising of the sEMG dataset is presented in Section 3. The TDPSD features used are elaborated in Section 4. In Section 5, a brief theory of the deep learning architecture and the learning method is described. The analysis procedure leading to the experimental results is explained in a systematic manner in Section 6. In Section 7, the performance of DNN compared with four baseline classifiers is discussed. Finally, the paper is concluded by summarizing the key points in Section 8. Table 1 shows the list of abbreviations used in the article.

2. System level overview of real-time sEMG classification

An overview of the blocks used to describe the process of EMG classification is illustrated in Fig. 1. Initially, the training of a particular subject is done offline, and then the generated trained parameters are used in the online (embedded integrated circuit based) classifier. The function of the filter block is to retrieve the relevant portion of the sEMG signal (between 40 Hz to 450 Hz) by eliminating the high-frequency noise and power line interference frequency (50 Hz) from the raw sEMG signal. The process needs to be fast and accurate for better performance of the subsequent blocks. Recently, Hofmann proposed a Bayesian Filter which has shown to be superior to the conventional linear filters in estimating the amplitude of the sEMG signal for accurate simultaneous and proportional prosthetic control [35]. After, filtration of the raw sEMG signal, relevant features are to be extracted and fed to the classifier block for predicting the outcome class of movement, which is used by the following prosthetic device. To perform all the operations within a stipulated amount of time with acceptable performance, the time duration of the window size ($t_{winsize}$) and window increment time (t_{inc}) for analyzing the data should be chosen wisely.

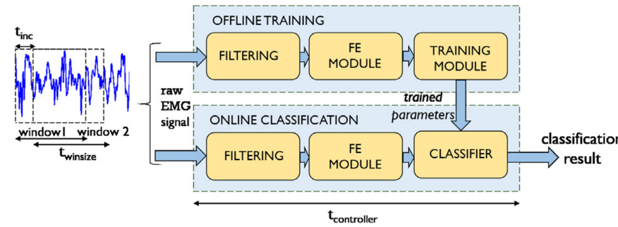

Fig. 1. The major blocks involved in the classification of EMG signal in real time.

In this work, an appropriate $t_{winsize}$ and t_{inc} is chosen for acceptable real-time controllability, i.e., controller delay $t_{controller}$ smaller than 300 ms. We have performed the classification offline while considering the constraints such as the $t_{winsize}$ and t_{inc} to be suitable for online classification. Another important point to be noted is that though $t_{winsize}$ corresponds to each of the data sample size, t_{inc} decides the throughput of processing the signal. We have considered $t_{winsize} = 100$ ms and $t_{inc} = 25$ms for creating the data samples for analysis which is an acceptable condition required for online classification. Further details for choosing the features from the data samples is described in Section 4.

3. EMG materials: dataset

We have used the dataset consisting of the myoelectric signal obtained from the upper limb of subjects via seven electrode channels at five different limb positions from different subjects, which were recorded by Khushaba et al. [33].[1] The eight different classes of hand movements considered are wrist flexion, wrist extension, wrist pronation, wrist supination, power grip, pinch grip, open hand and rest. This dataset is considered to be useful from the fact that it considers the variation in limb position which does affect the characteristics of the sEMG signal [36,37]. Here, the data has been recorded from the circumference of the forearm of healthy subjects. The data collection approach is valid for trans-radial amputees as well in which the data would be recorded from the muscles in their residual forearm. Hence, with appropriate training, while taking into account the different limb positions, we can expect to get a better test accuracy in real time scenarios where the user usually performs the different hand gestures at different limb positions. The applications benefited from this dataset are, viz., (1) different EMG based control applications such as exoskeleton/robotic hands, EMG based video game controllers, and (2) prosthetic hand controllers for trans-radial amputees.

4. Feature selection

Fused time domain descriptors (fTDD) are the features which combine the essential information between the current window (n^{th}) and the previous window ($(n-step)^{th}$), in which $step = 1, 2, \ldots, N$, determines the separation between the current and previous windows [26]. The advantage of this scheme is that if the two windows occur during the same class, it will enhance the correlation

[1] https://www.rami-khushaba.com/electromyogram-emg-repository.html.

between the fused features, whereas for non-similar classes the correlation will be diminished. In further analysis, we have considered fusion of features from two consecutive windows (i.e., $step = 1$) which will also determine the total number of data samples as given in Eq. (8).

fTDD is formed by the fusion of correlated time domain descriptors (cTDD) which is extracted from individual windows. The feature set of cTDD consists of root squared zero-order moment, root squared fourth and eighth order moments, sparseness, Irregularity Factor (IF), and WL ratio. The six-time domain descriptors used can be mathematically expressed as follows (Eqs. (1)–(7)):

$$f_1 = \log(m_0) \tag{1}$$

$$f_2 = \log(m_0 - m_4) \tag{2}$$

$$f_3 = \log(m_0 - m_8) \tag{3}$$

$$f_4 = \log(S) \tag{4}$$

$$f_5 = \log(IF) \tag{5}$$

$$f_6 = \log(WL) \tag{6}$$

$$m_n = \frac{\boldsymbol{m_n^\lambda}}{\lambda}; \quad n = 0, 2, 4, 8. \tag{7}$$

where the parameters $\lambda = 0.1$, $\boldsymbol{m_0} = \sqrt[2]{\sum_{j=0}^{N-1} x[j]^2}$, $\boldsymbol{m_2} = \sqrt[2]{\frac{1}{N}\sum_{j=0}^{N-1} \Delta(x[j])^2}$, $\boldsymbol{m_4} = \sqrt[2]{\frac{1}{N}\sum_{j=0}^{N-1} (\Delta^2 x[j])^2}$, $\boldsymbol{m_8} = \sqrt[2]{\frac{1}{N}\sum_{j=0}^{N-1} (\Delta^4 x[j])^2}$, $S = \frac{m_0}{\sqrt{m_0 - m_4}\sqrt{m_0 - m_8}}$, $IF = \frac{m_4}{\sqrt{m_0 m_8}}$, $WL = \frac{\sum_{j=0}^{N-1} |\Delta^2 x[j]|}{\sum_{j=0}^{N-1} |\Delta^4 x[j]|}$. The parameter N indicates the total number of samples of the EMG signal. In addition, Δ, Δ^2, Δ^3 and Δ^4 represent the first, second, third and fourth derivatives, respectively. The EMG signal and its nonlinear version extracted directly from the time-domain can aid in reducing the computational cost [38,39]. Basically, a set of six features were obtained per channel, implying a total of 42 (6 features × 7 channels) feature components representing each data sample which is formed by concatenation of features from all the channels present [26]. The set of 6 features from each channel is computed from two consecutive windows of size, $t_{winsize} = 100$ ms with an increment in shift of time window, $t_{inc} = 25$ ms. This is a reasonably good approximation of window length for real-time classification of myoelectric signals [4,5]. The generated feature set from the multiple channels are fed directly to the DNN trainer for finalizing the trained parameters (weight/bias coefficients) to be used by the DNN classifier. Each type of hand movement (class) was performed for 5 s with a rest of 3 s interval for 6 trials. Hence, for a particular subject, class, and limb position the number of data extracted can be calculated using Eq. (8) which results in 196 samples for the current case.

$$N_{data} = \left[\frac{T_{dur} - (t_{winsize} - t_{inc})}{t_{inc}} - step\right] \tag{8}$$

where N_{data}, T_{dur}, $t_{winsize}$ and t_{inc} are the number of data samples, the total duration of movement, window size, and window increment time respectively. Further, for 8 different type of movement classes and 6 trials for a particular limb position, the number of data samples is 9408 (8 classes × 6 trials × 1 limb position × 196 data samples). The features representing the 8 classes of hand movements are non-linearly separable which can be confirmed from the scatter plot as shown in Fig. 2. Only the first three features from the feature vector are considered for visualization purpose for the five limb positions individually.

5. Classification method: deep learning architecture

Deep learning in neural networks is the approach of composing networks into multiple layers of processing to meaningfully transform data to learn a useful representation of the input data. The essence of deep learning lies in this idea of successive layers of representations which in turn helps to learn higher-level features from low-level ones in a hierarchical manner, nullifying the over-dependence of shallow networks on feature engineering [40]. For this reason, the deep learning is often referred to as *hierarchical representations learning* in several contexts in the literature.

In the current work, we have adopted a feed-forward neural network architecture which comprises of multiple layers of transformations and non-linearity with the output of each layer feeding the subsequent layer. Mathematically, this DNN model can be interpreted as given in Eq. (9) and Eq. (10):

$$\boldsymbol{Z}^{(l)} = \boldsymbol{y}^{(l-1)}\boldsymbol{W}^{(l)} + \boldsymbol{b}^{(l)} \tag{9}$$

$$\boldsymbol{y}^{(l)} = g(\boldsymbol{Z}^{(l)}) \tag{10}$$

where $l \in [1,\ldots, L]$ denotes the l^{th} layer, $\boldsymbol{Z}^l$ is the vector of pre-activations of layer l, $\boldsymbol{y}^{(l-1)}$ is the output of previous layer $(l-1)$ and input to layer l. $\boldsymbol{W}^l \in \mathbb{R}^{n_i \times n_o}$ is a matrix of learnable biases of layer l, $\boldsymbol{y}^{(l)} \in \mathbb{R}^{n_o}$ is the output of layer l, $\boldsymbol{y}^{(0)}$ is the input to the model and $\boldsymbol{y}^{(L)}$ is the output of the final layer L of the model, $g(.)$ denotes the non-linear activation function. ReLU has been used in the hidden layers due to its computational efficiency and faster learning convergence over other activation functions. To provide a probabilistic interpretation of the model's output, the output layer L utilizes a softmax non-linearity as in Eq. (11):

$$softmax(\boldsymbol{Z}^{(L)}) = \frac{\exp^{\boldsymbol{Z}_k}}{\sum_{k=1}^{K} \exp^{\boldsymbol{Z}_k}} \tag{11}$$

where K is the number of output classes, i.e., the output layer contains K number of neurons.

5.1. Learning

The learning of a DNN is formulated as an optimization problem to minimize a cost-function [41]. As we are dealing with classification problem, the cross-entropy loss function has been used in this work as given in Eq. (12):

$$C = -\sum_{k=1}^{K} \hat{y}_k \log({y_k}^{(L)}) \tag{12}$$

where $\hat{y} \in \{0, 1\}^K$ is one-hot-encoded label and $y^{(L)}$ is the output of the model. In order to train the model, the gradients are computed by differentiating the cost-function w.r.t. the model parameters using a mini-batch of data sampled from the training data and back-propagated to prior layers using the backpropagation algorithm [41].

6. Experimental results

The earlier studies by Khushaba et al. have shown the importance of the feature extraction methods and compared between different types of classifiers such as SVM, LDA, kNN and Extreme Learning Machine (ELM) with a good amount of accuracy [33]. The current work is not to compete with the state of the art results obtained, but to do an extensive analysis with essential TDPSD features without undergoing the feature dimension reduction process, using DNN based classifier with an optimum configuration and a set of parameters on the similar set of data. In Fig. 3, a schematic model of the DNN classifier is depicted which takes the feature set as input

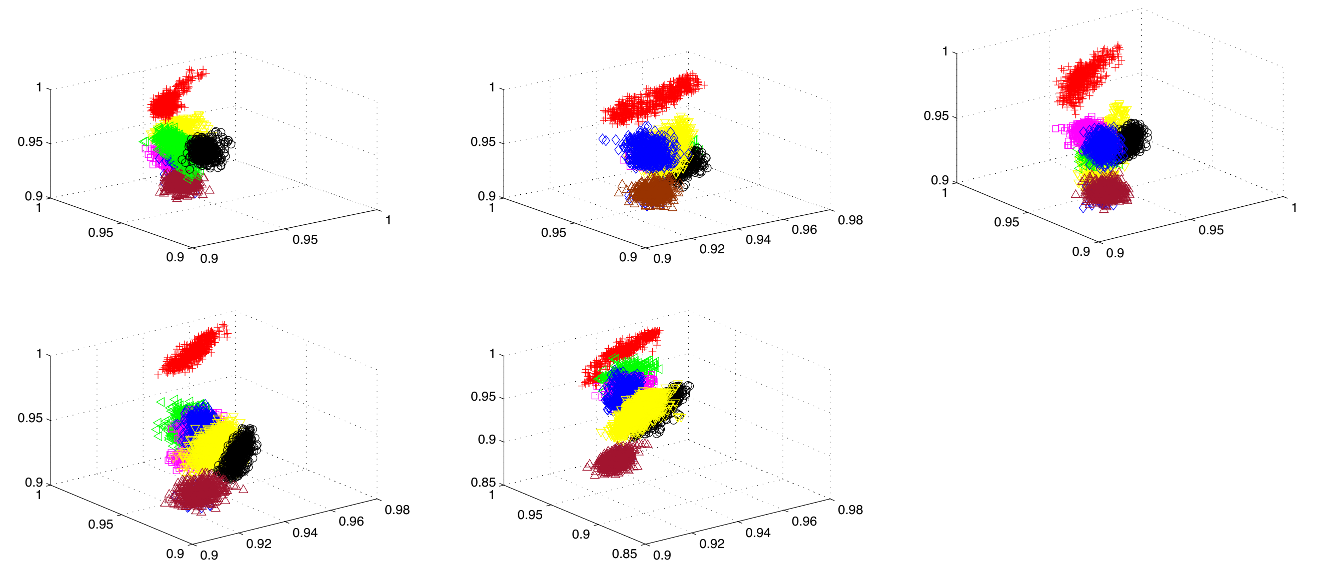

Fig. 2. Scatter plot of first three components of fTDD feature vector for limb positions (1–5). Each colour pattern represent one of the 8 class of hand movement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

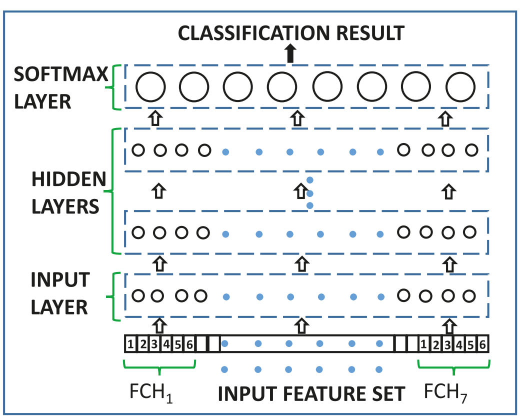

Fig. 3. The schematic of the DNN classifier receiving the feature set as an input and predicting the result at the output. As it is a multi-class classification problem a softmax function is used at the output layer. The activation function in the hidden layer can be relu, tanh or sigmoid.

to the first layer, followed by the hidden layers and output layer to give the classification result.

Initially, similar analysis as mentioned in [33] were performed on the considered features derived from the dataset comprising of upper limb EMG signal, using DNN for validation purposes. The results obtained were following similar trends when compared with their results obtained using SVM classifier [33]. Next, the dataset was divided appropriately based on analysis 1 and 2 to continue the experiment process. Afterwards, a comparative analysis between the DNN classifier with other classifiers, viz., SVM, kNN, random forest and decision tree was performed in terms of classification accuracy. The different analysis undertaken are mentioned henceforth, out of which analysis 1 and 2 are done for validation purpose. The accuracy mentioned in our analysis is the average accuracy considering the five subjects.

6.1. Analysis 1: Training in single limb position

The train/test data were divided in equal proportion, i.e., for each subject at a particular limb position, there are (4704 + 4704) samples of the train and test data consisting of eight classes of hand movements in a balanced manner. Here, data from each position

Table 2
Analysis 1: Inter-position classification accuracy (in %), averaged across 5 subjects and 8 classes of hand movements.

		Testing positions				
		POS1	POS2	POS3	POS4	POS5
Training positions	POS1	98.85	70.29	62.44	71.95	61.69
	POS2	76.62	95.27	76.81	74.99	67.13
	POS3	71.66	73.62	96.38	69.03	61.59
	POS4	70.63	66.70	66.75	97.46	86.10
	POS5	68.45	71.11	66.01	88.93	97.29

was trained separately to see the effect of each position on the test set as shown in Table 2. Understandably, the maximum accuracy will be in the cases when the train/test data are taken from the same position. From this matrix, we can get the idea of the correlation between different positions. For instance, P1 and P5, which are the two extreme arm positions in which data is collected from the subjects, will have a poor correlation due to the fact that the angle of deviation between the two arm positions is the maximum ($\approx 135^{\circ}$). Also, P1 is in a relatively relaxed position as compared to P5 which will have a strong impact in the variation of the EMG signals generated.

6.2. Analysis 2: Training in multiple limb position

In this analysis, testing of data is done on individual limb positions separately, in which the training dataset consists of data from all the arm positions except the position which is being tested. The test positions which show low accuracy do not have a good likeness with the other positions and hence must be considered in training set for better accuracy. We can confirm that positions 2 (P2), 4 (P4) and 5 (P5) are having better test accuracy as compared to other positions from the analysis as depicted in Table 3.

6.3. Analysis 3: Dividing the train/test set appropriately

As collecting the data at all the different positions is a tiring task for the subject, it is preferable to train at lesser limb positions while keeping in mind its effect on the accuracy of classification. The positions which should be considered in the train data are P1, P3, and P5 which can be understood from the combined result of analysis 1

Table 3
Classification accuracy when training on EMG data from all positions, except the one being tested on. These results are the average across all 8 movements and all 5 subjects.

Testing positions				
POS1	POS2	POS3	POS4	POS5
69.37	76.61	73.53	74.73	74.87

and 2. From now onwards, further analysis will be done considering the train data from the mentioned limb positions (P1, P3, and P5). To get the best possible classification accuracy of sEMG using DNN classifier with the optimal set of design parameters, the dataset is divided into three sets, viz., train set, validation set and test set. The total number of data samples for a particular limb position is divided in (7:3) ratio i.e. (6592 + 2816 = 9408) corresponding to the train and (validation + test) set respectively. Considering positions P1, P3 and P5 for the train set and all positions for validation and test set, the data distribution in the train, validation and test set are 19,776 (≈58%), 7040 (≈21%) and 7040 (≈21%) respectively. A point to be noted is that the data samples are chosen randomly from the 5s duration and allocated to train/ validation/ test set such that all the eight possible classes are distributed in a balanced fashion.

Please note that EMG classification problem is absolutely subject specific. Henceforth, the parameters of each DNN has been optimized for a specific subject. In other words, the subject is not considered as a dimension of generalization; rather we want to achieve trained generalized model regarding various limb positions. In our experiment, we have used DNN classifier which follows

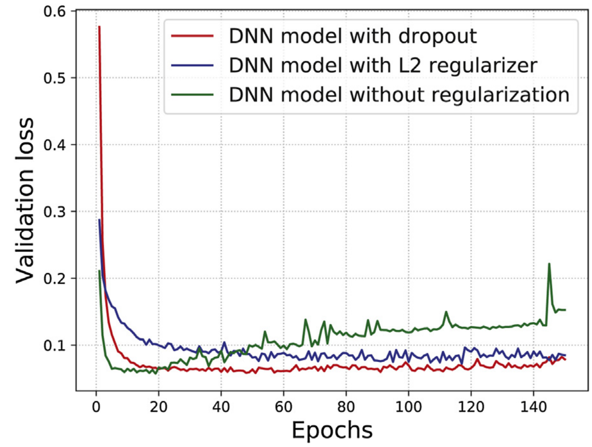

Fig. 4. Validation loss for different DNN models.

a fully connected feed-forward architecture comprising three hidden layers. The number of units in the input layer is equal to the feature dimension, and the output layer has a dimension equal to the number of classes (i.e., 8). The choice of activation functions in deep networks has a significant effect on the training dynamics and task performance. We have used the rectified linear unit (ReLU) as non-linearity in the hidden layers as ReLU yields much better optimized deep network with faster learning [42]. To prevent overfitting, we employed both L2 regularization and dropout separately as shown in Fig. 4. As the performance of DNN model with dropout

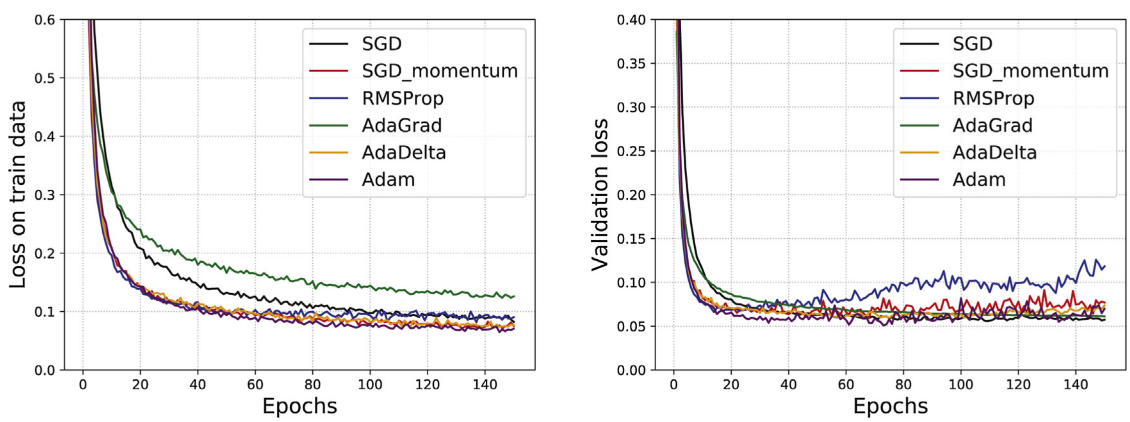

Fig. 5. Effect of various optimizer is illustrated in terms of loss on (a) train data and (b) validation data.

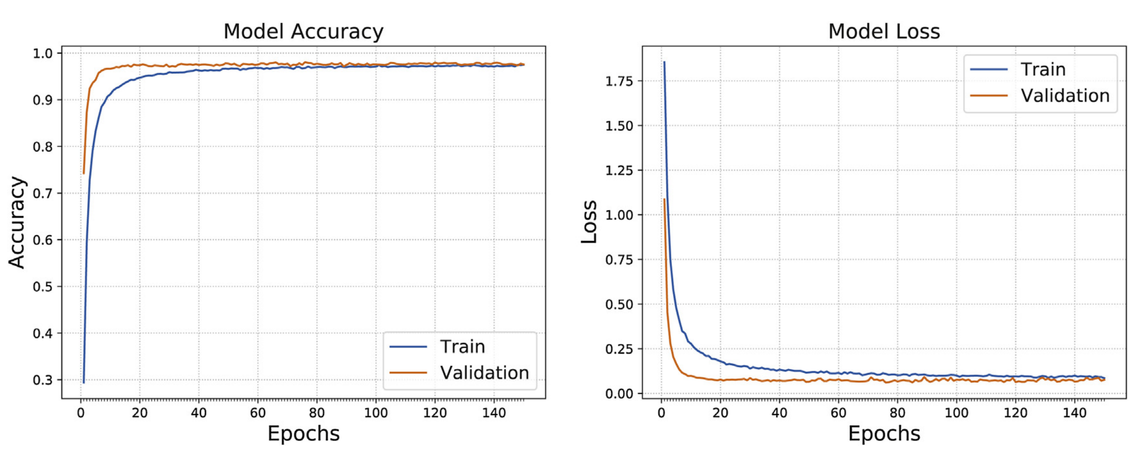

Fig. 6. DNN model accuracy and loss.

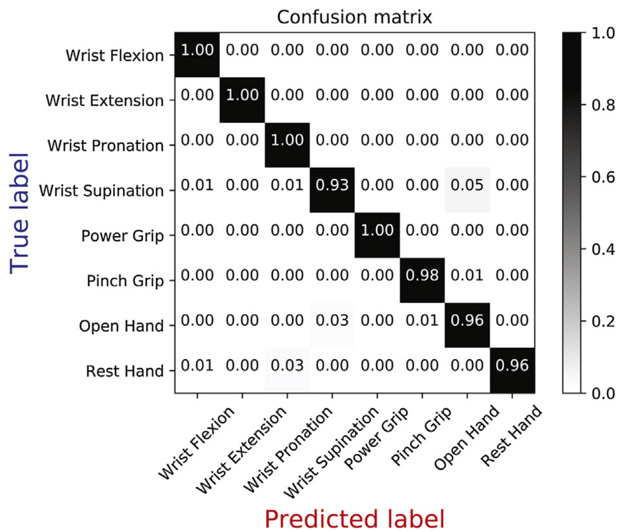

Fig. 7. Normalized confusion matrix.

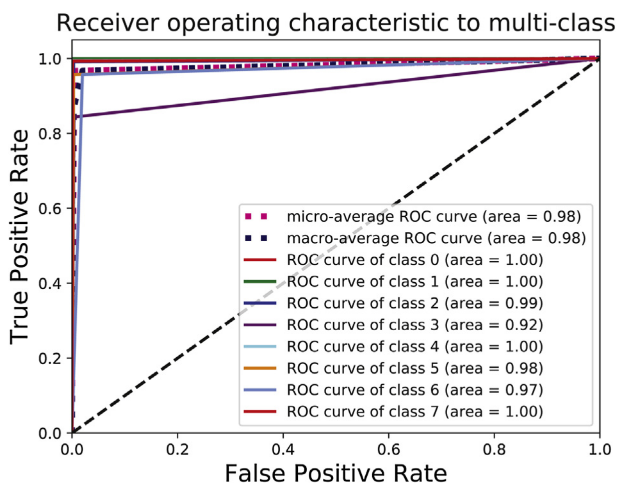

Fig. 8. ROC curve for multi-class classification.

(with probability = 0.5) perform significantly better than L2 regularized model for most of the subjects under consideration, we used dropout for almost all experiments. Fig. 5 illustrates the effect of different types of optimization algorithms in terms of loss on train and validation data. It is quite evident from Fig. 5 that Adam optimizer performs the best followed by SGD with momentum method and the rest. Hence, we choose the Adam optimizer for evaluation. In addition, batch-normalization technique [43] has also been applied during fine-tuning which helps to make DNN more robust to the changes in the hyperparameters [44]. For implementing the DNN model, we have used Keras (with Tensorflow backend) [45], a popular deep learning programming framework.

An instance of the result of DNN training regarding loss function and accuracy have been illustrated in Fig. 6. The normalized confusion matrix for a particular subject (S2 considered) is plotted in Fig. 7 which depicts that the DNN model can perform significantly well regarding accuracy. As the process of training is subject specific, similar confusion matrix can be generated for other subjects once the network parameters are optimized. It can be seen that class 4 (wrist supination) has the lowest accuracy or in other words is having less number of true positives which is confirmed from Fig. 8 which depicts the receiver operating characteristics (ROC) to a multi-class problem.

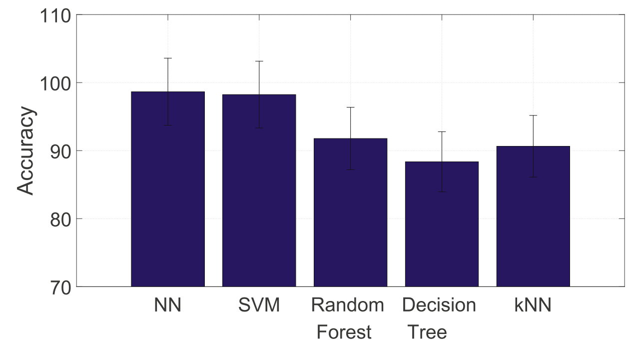

Fig. 9. Average test accuracy using the classifiers: NN, SVM, Random Forest, Decision Tree, kNN.

7. Discussion

The dataset considered in analysis 3 (described in Section 6.3) is evaluated with four other classifiers, viz., (1) SVM, (2) kNN, (3) Random Forest and (4) Decision Tree to compare with the performance of DNN. Due to the subject specificity of the problem the classifiers are fine-tuned to get the best possible results for each subject considered. A Python based machine learning toolbox ‘Scikit-learn’ has been used to evaluate the performance based on analysis 3. The complexity parameter C of the SVM classifier (LIBSVM library), as well as the gamma parameter in the kernel function (‘linear’), were optimized for each subject. In addition, we have used the LIBSVM classifier with one-versus-rest approach as this method is providing better performance than one-versus-one approach. Similarly, we have used kNN with ‘Brute-force’ algorithm and ‘minkowski’ distance metric. The average accuracy obtained among the five subjects for SVM, kNN, Random Forest and Decision Tree are 98.66%, 90.64%, 91.78%, and 88.36% respectively as evident from Fig. 9. It can be inferred that the accuracy obtained using DNN and SVM are comparable and relatively higher than the other classifiers used. One important point to be noted is that DNN can reduce the classification error for any subject by a significant amount compared to SVM by tweaking the parameters, which will be different for each subject. The other existing deep learning models such as DTGNN, have the ability to extract rich contextual information from CTS signals. However, the use of DTGNN would increase the computational overhead in the online classifier and hence is not explored in the current work. However, the performance of DTGNN could be studied in the future on EMG signal classification performance for clinical diagnosis of a patient in which the inference is performed offline.

8. Conclusions

This study mainly focused on the classification of the upper limb invariant myoelectric signal. The position of the limb effects the signal generated through the muscle surface and hence is a vital point to consider during the data collection process. In this article, we delved into the detailed empirical exploration of DNN based EMG classification system. Our approach has avoided refining the EMG signal to obtain the MUAPs using methods such as ICA and considered only the relevant features which can be directly obtained from the time domain signal. The primary reason is that DNN can itself help in feature engineering aspects, hence discriminative features requiring complex procedures is not necessary. Experimental results in various analysis frameworks have demonstrated that DNN based system can outperform other classifiers such as kNN, Random Forest and Decision Tree. Moreover, it can achieve competitive performance with the state-of-the-art SVM based model, even though the proposed DNN model requires minimal computation on feature engineering. The future work will be devoted to the exploration of other complex DNN models such as RNN (LSTM, GRU, etc.) to improve the performance further. Also, the performance of complex deep learning models such as DTGNN could be studied on different bio-medical applications having signals with CTS properties. Although the dataset analyzed in this work has advantages because of the non-invasive nature of recording the sEMG signals from the circumference of the forearm of the subjects at different upper limb positions, it has considered only healthy subjects. Therefore, in future work there is scope for exploring other datasets [46] consisting of amputees along with different DNN models.

Acknowledgments

This work has been supported by the Ministry of Human Resource Development (MHRD), Govt. of India. The authors are thankful to the editor and the anonymous reviewers for their insightful suggestions and constructive feedback throughout the review process, which have considerably helped to improve the content of the article.

Declaration of Competing Interest

None declared.

References

[1] M.B.I. Reaz, M. Hussain, F. Mohd-Yasin, Techniques of EMG signal analysis: detection, processing, classification and applications, Biol. Proced. Online 8 (1) (2006) 11.
[2] C.I. Christodoulou, C.S. Pattichis, Unsupervised pattern recognition for the classification of EMG signals, IEEE Trans. Biomed. Eng. 46 (2) (1999) 169–178.
[3] K. Englehart, B. Hudgins, A robust, real-time control scheme for multifunction myoelectric control, IEEE Trans. Biomed. Eng. 50 (7) (2003) 848–854.
[4] T.R. Farrell, R.F. Weir, The optimal controller delay for myoelectric prostheses, IEEE Trans. Neural Syst. Rehabil. Eng. 15 (1) (2007) 111–118.
[5] L.H. Smith, L.J. Hargrove, B.A. Lock, 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 Trans. Neural Syst. Rehabil. Eng. 19 (2) (2011) 186–192.
[6] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, O.C. Aszmann, The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges, IEEE Trans. Neural Syst. Rehabil. Eng. 22 (4) (2014) 797–809.
[7] K. Englehart, B. Hudgins, P.A. Parker, M. Stevenson, Classification of the myoelectric signal using time-frequency based representations, Med. Eng. Phys. 21 (6) (1999) 431–438.
[8] T.A. Kuiken, G. Li, B.A. Lock, R.D. Lipschutz, L.A. Miller, K.A. Stubblefield, K.B. Englehart, Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms, JAMA 301 (6) (2009) 619–628.
[9] Y. Zhang, W. Herzog, M. Liu, A mathematical model of myoelectric signals obtained during locomotion, in: Engineering in Medicine and Biology Society, 1995, IEEE 17th Annual Conference, vol. 2, IEEE, 1995, pp. 1403–1404.
[10] M.A. Oskoei, H. Hu, Ga-based feature subset selection for myoelectric classification, in: IEEE International Conference on Robotics and Biomimetics, 2006, ROBIO'06, IEEE, 2006, pp. 1465–1470.
[11] P. Erik Scheme, P. Kevin Englehart, Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use, J. Rehabil. Res. Dev. 48 (6) (2011) 643.
[12] B. Hudgins, P. Parker, R.N. Scott, A new strategy for multifunction myoelectric control, IEEE Trans. Biomed. Eng. 40 (1) (1993) 82–94.
[13] M.A. Oskoei, H. Hu, Support vector machine-based classification scheme for myoelectric control applied to upper limb, IEEE Trans. Biomed. Eng. 55 (8) (2008) 1956–1965.
[14] A.D. Chan, K.B. Englehart, Continuous myoelectric control for powered prostheses using hidden Markov models, IEEE Trans. Biomed. Eng. 52 (1) (2005) 121–124.
[15] B. Karlik, M.O. Tokhi, M. Alci, A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis, IEEE Trans. Biomed. Eng. 50 (11) (2003) 1255–1261.
[16] J. Yousefi, A. Hamilton-Wright, Characterizing EMG data using machine-learning tools, Comput. Biol. Med. 51 (2014) 1–13.
[17] G.R. Naik, D.K. Kumar, V.P. Singh, M. Palaniswami, Hand gestures for HCI using ICA of EMG Proceedings of the HCSNet Workshop on Use of Vision in Human-Computer Interaction, vol. 56, Australian Computer Society, Inc., 2006, pp. 67–72.
[18] M.R. Ahsan, M.I. Ibrahimy, O.O. Khalifa, Electromyography (EMG) signal based hand gesture recognition using artificial neural network (ANN), in: 2011 4th International Conference on Mechatronics (ICOM), IEEE, 2011, pp. 1–6.
[19] H. Nakamura, M. Yoshida, M. Kotani, K. Akazawa, T. Moritani, The application of independent component analysis to the multi-channel surface electromyographic signals for separation of motor unit action potential trains: Part I – Measuring techniques, J. Electromyogr. Kinesiol. 14 (4) (2004) 423–432.
[20] D. Staudenmann, A. Daffertshofer, I. Kingma, D.F. Stegeman, J.H. van Dieen, Independent component analysis of high-density electromyography in muscle force estimation, IEEE Trans. Biomed. Eng. 54 (4) (2007) 751–754.
[21] J. Taelman, S. Van Huffel, A. Spaepen, Wavelet-independent component analysis to remove electrocardiography contamination in surface electromyography, in: Engineering in Medicine and Biology Society, 2007, EMBS 2007. 29th Annual International Conference of the IEEE, IEEE, 2007, pp. 682–685.
[22] B. Mijovic, M. De Vos, I. Gligorijevic, J. Taelman, S. Van Huffel, Source separation from single-channel recordings by combining empirical-mode decomposition and independent component analysis, IEEE Trans. Biomed. Eng. 57 (9) (2010) 2188–2196.
[23] A. Gharehbaghi, M. Lindén, A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4102–4115.
[24] A. Gharehbaghi, T. Dutoit, P. Ask, L. Sörnmo, Detection of systolic ejection click using time growing neural network, Med. Eng. Phys. 36 (4) (2014) 477–483.
[25] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, Phoneme recognition using time-delay neural networks, in: Readings in Speech Recognition, Elsevier, 1990, pp. 393–404.
[26] R.N. Khushaba, A. Al-Ani, A. Al-Timemy, A. Al-Jumaily, A fusion of time-domain descriptors for improved myoelectric hand control, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–6.
[27] M. Atzori, M. Cognolato, H. Müller, Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands, Front. Neurorobot. 10 (2016) 9.
[28] C.-C. Chiu, T.N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R.J. Weiss, K. Rao, E. Gonina, et al., State-of-the-art speech recognition with sequence-to-sequence models, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4774–4778.
[29] I. Sutskever, J. Martens, G.E. Hinton, Generating text with recurrent neural networks, Proceedings of the 28th International Conference on Machine Learning (ICML-11) (2011) 1017–1024.
[30] S. Samui, I. Chakrabarti, S.K. Ghosh, Time-frequency masking based supervised speech enhancement framework using fuzzy deep belief network, Appl. Soft Comput. 74 (2019) 583–602.
[31] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, G.-Z. Yang, Deep learning for health informatics, IEEE J. Biomed. Health Inform. 21 (1) (2017) 4–21.
[32] D. Yang, W. Yang, Q. Huang, H. Liu, Classification of multiple finger motions during dynamic upper limb movements, IEEE J. Biomed. Health Inform. 21 (1) (2017) 134–141.
[33] R.N. Khushaba, M. Takruri, J.V. Miro, S. Kodagoda, Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features, Neural Netw. 55 (2014) 42–58.
[34] A. Phinyomark, P. Phukpattaranont, C. Limsakul, Feature reduction and selection for EMG signal classification, Expert Syst. Appl. 39 (8) (2012) 7420–7431.
[35] D. Hofmann, N. Jiang, I. Vujaklija, D. Farina, Bayesian filtering of surface EMG for accurate simultaneous and proportional prosthetic control, IEEE Trans. Neural Syst. Rehabil. Eng. 24 (12) (2016) 1333–1341.
[36] O.W. Samuel, X. Li, Y. Geng, M.G. Asogbon, P. Fang, Z. Huang, G. Li, Resolving the adverse impact of mobility on myoelectric pattern recognition in upper-limb multifunctional prostheses, Comput. Biol. Med. 90 (2017) 76–87.
[37] M. Jochumsen, A. Waris, E.N. Kamavuako, The effect of arm position on classification of hand gestures with intramuscular EMG, Biomed. Signal Process. Control (2018), http://dx.doi.org/10.1016/j.bspc.2018.02.013.
[38] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, Y. Laurillau, Feature extraction of the first difference of EMG time series for EMG pattern recognition, Comput. Methods Programs Biomed. 117 (2) (2014) 247–256.
[39] A.H. Al-Timemy, R.N. Khushaba, G. Bugmann, J. Escudero, Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees, IEEE Trans. Neural Syst. Rehabil. Eng. 24 (6) (2016) 650–661.
[40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[41] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533.
[42] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010) 807–814.
[43] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, International Conference on Machine Learning (2015) 448–456.
[44] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[45] F. Chollet, keras, https://github.com/fchollet/keras, 2015.
[46] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G.M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Müller, Electromyography data for non-invasive naturally-controlled robotic hand prostheses, Sci. Data 1 (2014), 140053.