Available at www.sciencedirect.com

ScienceDirect

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

Original Research Article

# EMGHandNet: A hybrid CNN and Bi-LSTM architecture for hand activity classification using surface EMG signals

*Naveen Kumar Karnam*[a], *Shiv Ram Dubey*[b], *Anish Chand Turlapaty*[a,*], *Balakrishna Gokaraju*[c]

[a] *Bio-Signal Analysis Group, Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Sri City, Andhra Pradesh, India*

[b] *Computer Vision and Biometrics Laboratory (CVBL), Department of Information Technology, Indian Institute of Information Technology, Allahabad, Uttar Pradesh, India*

[c] *Visualizations and Computing Advanced Research Center (ViCAR), Department of Computational Data Science and Engineering, North Carolina A and T State University, Greensboro, NC, United States*

ARTICLE INFO

*Article history:*
Received 18 September 2021
Received in revised form
12 February 2022
Accepted 13 February 2022
Available online 26 February 2022

*Keywords:*
CNN
Bidirectional LSTM
Deep Learning
Hand Activity Classification
sEMG Signal
Exoskeleton control

ABSTRACT

Recently, Convolutional Neural Networks (CNNs) have been used for the classification of hand activities from surface Electromyography (sEMG) signals. However, sEMG signal has spatial sparsity due to position of electrodes on hand muscle and temporal dependency due to performance of activity over a period of time. The CNN has the ability to extract spatial features and is limited in extracting temporal dependencies. Whereas, the Long Short-Term Memory (LSTM) aims to encode the temporal relations from sequential data. Hence, in this paper, we propose a hybrid CNN and Bidirectional LSTM (Bi-LSTM) based EMGHandNet architecture to encode the inter-channel and temporal dependencies of sEMG signals for hand activity classification. First, the CNN layers are used to extract deep features from sEMG signals, then these feature maps are processed by the Bi-LSTM to extract the sequential information in both the forward and backward directions. Thus, the proposed model learns both inter-channel and bidirectional temporal information in an end-to-end manner. The proposed model is trained and tested on five benchmark datasets, including the NinaPro DB1, NinaPro DB2, NinaPro DB4, BioPatRec DB2 and UCI Gesture. The average classification accuracies for the NinaPro DB1, NinaPro DB2, NinaPro DB4 and UCI Gesture are $95.77\%, 95.9\%, 91.65\%$, and $98.33\%$ respectively. They correspond to an improvement of $4.42\%, 12.2\%, 18.65\%$ and $1.33\%$ over the respective state-of-the-art models. Moreover, for the BioPatRec DB2 dataset, a comparable performance (91.29%) is observed. The experimental results and comparisons confirm the superiority of the proposed model for hand activity classification from the sEMG signals.

[*] Corresponding author at: Bio-Signal Analysis Group, Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Sri City 517646, Andhra Pradesh, India.
E-mail address: anish.turlapaty@iiits.in (A.C. Turlapaty).

https://doi.org/10.1016/j.bbe.2022.02.005

## 1. Introduction

Muscle computer interaction (MCI) is widely used in prostheses [1], robotic control [2], sign language recognition (SLR) systems [3] and human–machine interaction (HMI) [4]. The hand activity classification is a central problem in the MCI technology and it includes analyses of categories such as hand gestures, object grasping and hand movements [5]. The gesture classification is used in many application areas such as computer games, virtual reality, and robot assisted surgeries [6]. A robot assisted surgery requires precise specification of the posture of robotic hand with respect to a patient's body as well as requisite movements to be performed.

The human gestures can be identified directly by classifying gesture images [7] or indirectly through the corresponding surface Electromyography (sEMG) signals [8]. The latter approach has shown immense potential for developing control systems for the exoskeletons and prosthetic devices. The sEMG signal classification can be used to decode the intended motion to control the robotic arm [9]. Specifically, the force, torque and the direction that characterize the intended hand movements have to be decoded by the system [10]. The mapping between the hand movement and the corresponding force was studied in [11] using the ANN and LSTM architectures.

Classification of hand position during object grasping has direct applications in controlling robot arms for pick and place tasks [12,13]. Similarly, hand movement classification can provide improved degrees of freedom to the exoskeleton arm [14]. The focus of this paper is classification of hand activities using sEMG signals with the pattern recognition methods. The classification of sEMG signals is traditionally carried out by the machine learning (ML) algorithms and recently with the deep learning (DL) based approaches. A short review of these methods follows.

### 1.1. *Machine Learning Based Approaches*

In machine learning techniques, the classification of sEMG signals requires extraction of features, such as the time domain features [15], the frequency domain features [16] and the time–frequency domain features [15]. Shenoy et al. [17] classified 8 hand movements with the linear Support Vector Machine (SVM) using the Root Mean Square (RMS) as a feature and was able to control a robotic arm with four degrees of freedom. In Altimemy *et al.* [18], a classification of 15 hand movements for persons with intact-limbs and 12 hand movements for amputees is performed with the Linear Discriminant Analysis (LDA) and the SVM based on the Auto Regressive (AR) features. Shi *et al.* [19] have used features including the Mean Absolute Value (MAV), the Zero Crossing (ZC), the Slope Sign Change (SSC), and the Waveform Length (WL) for hand posture classification with the K-Nearest Neighbor (KNN) classifier to control a bionic hand. In [20], Waris *et al.* classified gesture data extracted through the surface as well as intramuscular EMG signals over a period of seven days and showed that the performance has improved over time with the Artificial Neural Network (ANN) classifier compared to the classical KNN and the SVM classifiers. Tuncer *et al.* [21] extracted the ternary features from the raw sEMG signals. The statistical moments have been used as a feature for the classification with the KNN and the cubic SVM classifiers. Recently, Fatimah *et al.* [22] have decomposed the sEMG signals into Fourier intrinsic band functions (FIBFs) and extracted statistical features for classification with the SVM and the KNN classifiers. The hand gesture identification can be enhanced by capturing the depth information using a leap motion device that improves the relabelling of gestures in training phase[14]. In [23], energy based features were used with the fine KNN for classification of hand gestures based on the sEMG signals. A key limitation of the machine learning approaches is a need for manual design of relevant feature sets for the said problem, which is a very tedious task and may not be sufficiently accurate. Another drawback is the challenge of selection of an optimal classifier for the chosen features.

### 1.2. *Deep Learning Based Approaches*

For the sEMG signal classification, though machine learning approaches have reported decent performance, however, in the recent literature, the deep learning approaches have become popular. It is because they automatically learn the important features [24] and tend to provide an improved performance. Hence with the application of deep learning techniques for the sEMG classification, the control mechanism of exoskeletons can be significantly improved. In this section, we review the recent deep learning techniques for classification of the sEMG signals.

#### 1.2.1. *CNN Approaches*

In [25], Atzori et al. performed the sEMG classification task over NinaPro DB1, NinaPro DB2 and NinaPro DB3 datasets with a deep Convolutional Neural Network (CNN) architecture consisting of two convolutional layers. The authors have shown a performance improvement of 2-5% compared to the existing machine learning classifiers such as the KNN, SVM, Random Forests and the LDA [26]. Among the 2D approaches, Geng *et al.* [27] considered each sample of dimension $1 \times 10$ (here, 10 refers to No. of channels) as an instantaneous image and provided as an input to the CNN model and showed that there are patterns within instantaneous image which are similar across samples of the same trial and discriminative across different trials.

Wei et al. [28] had split each segment of the data samples into patches of images and processed with a parallel multistream CNN architecture and provided patch wise analysis of images. The state of the art accuracies were achieved but the computational complexity is $10 - 20\%$ higher than the other state of the art architectures. In [29], an evolutionary algorithm is developed which generates the CNN topology to identify the right number of CNN layers, the number of kernels and the kernel size. In [30], deep learning architectures employing three different input modalities such as the raw EMG, the Spectrogram and the Continuous Wavelet Transform (CWT) are analysed with the help of a transfer learning technique. Qi *et al.* [31] used 3D CNN to classify the composite hand motions associated with digit writing. In [32], Betthauser *et al.* have implemented an EMG prediction model with temporal convolutional networks and shown that the sequential prediction of movements is better compared to the frame-wise prediction. In [33], a single channel is randomly selected from the multi-channel data achieving a classification accuracy close to 95% for a subset of NinaPro DB2. From this analysis, It is inferred that sEMG signals can also be classified with only a single channel of data which is very efficient for low memory devices.

#### 1.2.2. *RNN Approaches*

The recurrent neural network (RNN) is a variation of the neural network which works for sequential and temporal data. In [34], Koch et al. have used a ConvLSTM cascaded with the LSTM architecture for hand gesture sequence classification. In [35], a two stage network consisting of a fully connected network followed by a stacked RNN is implemented to classify the high density (HD) and sparse sEMG signals. Hu *et al.* [36] developed an attention based CNN-RNN architecture which is able to classify the sEMG images. In [37], a LSTM model is compared with a deep back-propagation (BP) LSTM by using the waveform based classification. In [38], a hybrid CNN-LSTM architecture named as the Long-term Recurrent Convolutional Networks (LRCNs) is used for the task of activity recognition and image and video description. Motivated by this work, Bao *et al.* [39] and Chen *et al.* [40] proposed another CNN-LSTM architecture. Specifically, [39], Bao *et al.* used this joint architecture to estimate wrist angles corresponding to four movements collected from six healthy subjects. Chen et al. [40] used hybrid CNN and LSTM architecture for gesture classification through a HD-sEMG dataset using the transfer learning approach. Note that the authors in [40] have only utilised the LSTM for target datasets.

### 1.3. *Motivation for Hybrid Architecture*

From the literature, it is observed that the CNN architectures has a capability to understand the spatial features of the human hand activity and the LSTM architectures can capture the temporal information. Moreover, the Bi-LSTM improves over the uni-directional LSTM through learning the forward and the backward inter-relations between the activities and the input sEMG signals. Thus, in this paper, a hybrid CNN and Bi-LSTM framework is proposed to learn both the spatial and bi-directional temporal relations. Our approach differs from existing architectures and methods as follows. The focus is on classification of hand activities with the CNN and Bi-LSTM in contrast to estimation of angles in [39]. The Bi-LSTM based architecture is included in each of the experiments in contrast to [40]. A flattening of activation maps is performed before the Bi-LSTM unit that is in contrast with the architecture in [40] where the dense layers are inserted in between the CNN and LSTM layers which may lead to loss of temporal information. Another difference is in consideration of inputs, specifically window overlap is not considered in contrast to many existing works [28,29,36].

### 1.4. *Objectives*

The main purpose of this study is to classify human hand activities based on the sEMG signals. The specific objective of this study is to identify the following characteristics of the deep learning model to achieve optimal classification performance: (1) a proper shape of the input data provided to the model, (2) the number of conv layers and/or the number of LSTM layers, (3) a proper set of hyper parameters for the model and (4) appropriate preprocessing techniques for the sEMG signals.

### 1.5. *Contributions*

The main contributions of this work are:

1. A hybrid CNN and Bi-LSTM based EMGHandNet architecture is successfully demonstrated for classification of human hand activities using the sEMG signals.
2. The proposed method exploits the learning of the inter-channel and the temporal features using the 1-D convolutional layers and the Bi-LSTM layers respectively. Specifically, the spatial and short-term temporal relations are encoded by the convolutional layers and the long-term temporal relation are learned by the Bi-LSTM layers.
3. We have performed rigorous experiments on five benchmark sEMG datasets for the classification of hand movements from the sEMG signals. We have also analyzed the impact of the proposed method from different perspectives and compared the performance against that of the recent deep Learning methods.

The rest of this paper is organized as follows: Section II contains the preliminaries required; Section III presents the proposed methodology; Section IV gives details about the experimental setup; Section V demonstrates the experimental results; and Section VI provides a conclusion along with the future scope.

## 2. Preliminaries

In this section, the basic working of the CNN and the LSTM are illustrated.

### 2.1. *Convolutional Neural Networks (CNN)*

The CNNs are commonly used for the feature extraction from images [41]. A CNN consists of the convolution, the activation and the pooling layers. In the convolution (conv) layer, the input is convolved with the filter weights. The number of rows in the kernel is the kernel size and the number of columns in the kernel corresponds to the number of channels in the input data for the first conv layer. For the rest of the conv layers it is the number of filters. An illustration of 1-D convolution is shown in Fig. 1. Consider an input sEMG signal with a two channel data as shown in Fig. 1, the kernel is a two dimensional two column filter (since the number of channels are two in this example) and the first convolution output is calculated as $\mathbf{y}_0 = a_0 \times k_0 + a_1 \times k_1 + b_0 \times k_2 + b_1 \times k_3$. The remaining outputs are calculated by sliding the kernel in the vertical direction (time stamp direction). Thus, we get a single vector from each filter. The output vectors from each of the filters are concatenated column wise to obtain a 2D feature map which is further processed by the consecutive conv

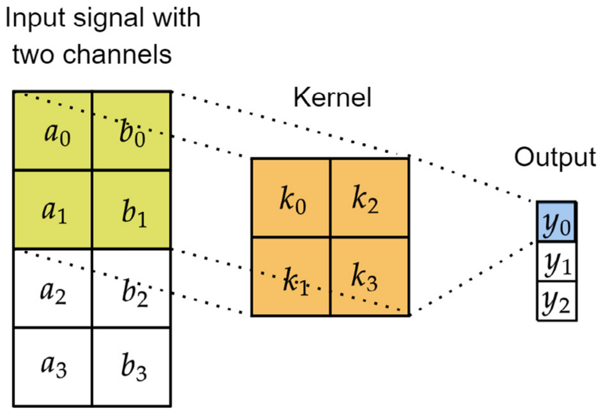


**Fig. 1 – Illustration of 1-D convolution using an example having two channels.**

layers. The conv layers are stacked until the abstract features of the signals are obtained. In the activation layer, the input is transformed by a non-linear function such as the $\tanh(\cdot)$ or the ReLU function. The pooling layer reduces the dimensionality of the feature map. The max-pooling is used for selecting prominent features from the feature map. Finally, the output is converted to probability values (classification score) by using the softmax function.

### 2.2. *Long Short-Term Memory (LSTM)*

The recurrent neural networks (RNNs) are a type of neural networks capable of analysing sequence of data in which a prediction is dependent on previously computed values. It is commonly used to analyze text, speech and DNA sequence datasets, where any current information is dependent on the preceding time steps. However, the vanishing and exploding gradient issues in RNNs [42] have limited its usability in long input sequence analysis. Hence, Long Short Term Memory (LSTMs) [43] are used for the long sequential data. Within the LSTM, there is a control gate to hold the current hidden state in a memory cell based upon preceding hidden states. The output predictions are encoded based on the previous hidden states. The encoded output, from each LSTM layer is recursively appended to form a complete feature vector. The Bidirectional LSTM (Bi-LSTM), deployed in this work, takes input data in the form of sequences and generates hidden state output predictions for each time step as depicted in the Fig. 2. It considers sequences in both forward and backward directions and encodes the features and concatenates them.

## 3. Methodology

### 3.1. *Problem Statement*

The total number of sEMG patterns in a dataset is $N = S \times N_A \times R$, where $S$ is to the total number of subjects, $N_A$ is the number of different hand activities, and $R$ corresponds to the number of activity repetitions per subject. A full sEMG dataset can be represented as:

$$\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^{N} \tag{1}$$

where each observation array $\mathbf{x}_n$ consists of multiple channels as:

$$\mathbf{x}_n = \{\mathbf{x}_{n,m}\}_{m=1}^{N_C}, \; n = 1, \cdots, N \tag{2}$$

and $N_C$ is the number of channels (from different electrodes) and each of these channels consists of an array:

$$\mathbf{x}_{n,m} = \{x_{n,m}(i)\}_{i=1}^{N_T} \tag{3}$$

where $N_T = N_s \times T$ is the number of values in one trial of duration $T$ and $N_s$ is the sampling rate (samples/s).

The objective of this study is to map the sEMG signals to the corresponding activity ($\alpha$ - Target labels), which can be formulated as:

$$f\{\mathbf{x}_n\} \rightarrow \alpha \tag{4}$$

The mapping function in (4) can be implemented either by a machine learning or a deep learning classifier. For the mapping function, appropriate features are required that represent the underlying inverse kinematic relationships between the sEMG signals and the corresponding activity performed. The feature extraction and mapping is achieved by a hybrid deep learning model described below (see Fig. 3)

### 3.2. *Hybrid EMGHandNet Model*

The proposed hybrid CNN and Bi-LSTM based EMGHandNet implementation scheme is illustrated in Fig. 3 with the internal architecture depicted in Fig. 4. The sEMG signals are preprocessed, suitably segmented and provided as inputs to a deep learning architecture for classification. The EMGHandNet consists of four one dimensional (1-D) CNN layers with a time distributed wrapper followed by two Bi-LSTM layers and finally two dense layers as shown in Fig. 4. The outputs of the final dense layer are the categories of the hand activities. The deep learning model learns the mapping of neuro muscle activity (sEMG signals) to the limb kinematics. It is able to classify hand activities appropriately by bypassing the need for theoretical neuro-mechanical models (eg. differential equations). This ML based learning of kinematic mapping is demonstrated in [44–47].

The first step is the EMG data setup and preprocessing for feeding to the CNN and is described as follows. The number of samples $N_T$ from each trial is factorized as:

$$\{S_L, T_S\} = fact(N_T) \tag{5}$$

where $S_L$ is No. of sub sequences, $T_S$ is sub-sequence time steps and $fact(N_T)$ is the factorization of $(N_T)$ such that $N_T = S_L T_S$. $T_S$ is chosen such that the minimum $(T_S)$ greater than zero. The class labels are one-hot encoded to compute the cross entropy loss for the multi-class problem. The input shape of the data $N_T \times N_C$ is reshaped to the $S_L \times T_S \times N_C$ using the factorization method in (5). This 3D data is split into train and test samples and sent to 1-D CNN layers. The output size $N_o$ for each CNN layer with a kernel size $k$ is given by:

$$N_o = \left\lfloor \frac{i + 2p - k}{s} \right\rfloor + 1 \tag{6}$$

where $i$ is the input size, $s$ is the stride and $p$ is the padding given by $p = \lfloor \frac{k}{2} \rfloor$.

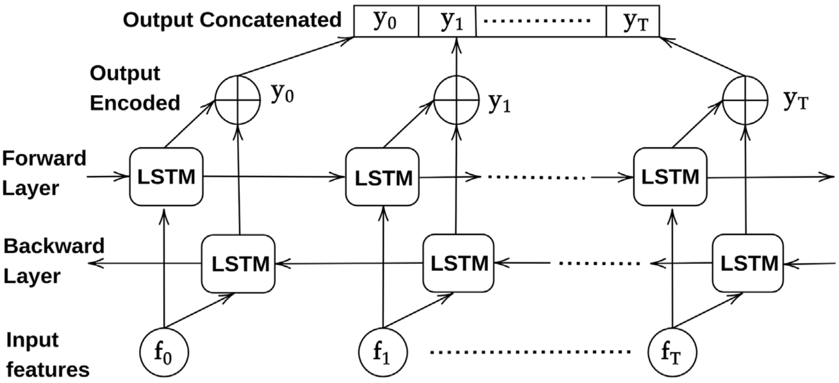


Fig. 2 – A block diagram of the Bi-directional LSTM.

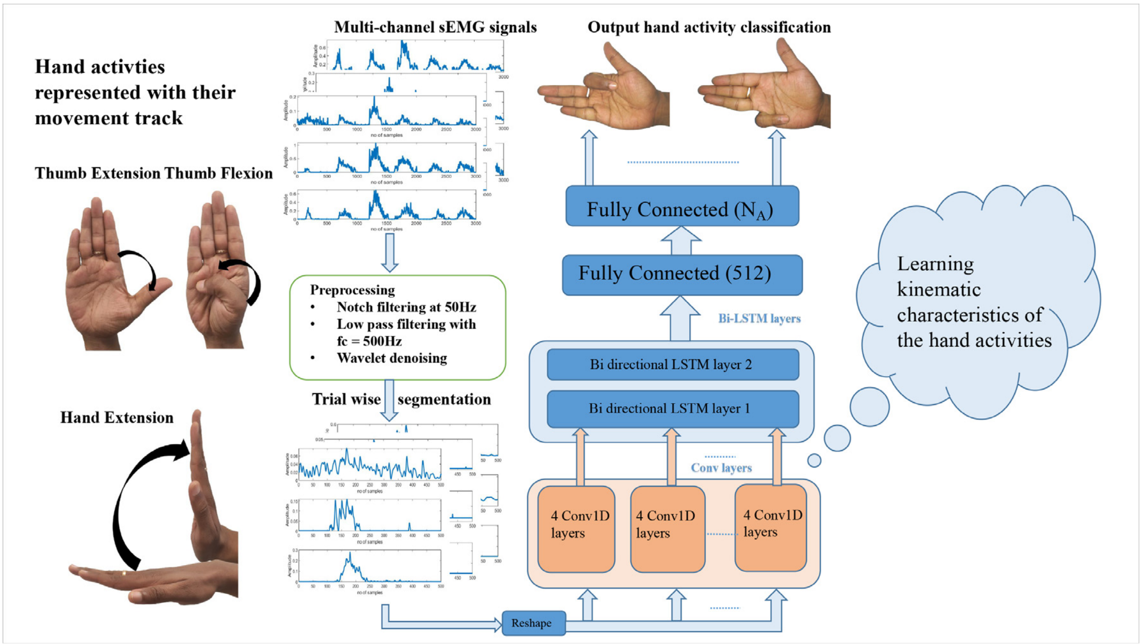

Hand activties represented with their movement track
Thumb Extension Thumb Flexion
Hand Extension
Multi-channel sEMG signals
Preprocessing
Notch filtering at 50Hz
Low pass filtering with fc = 500Hz
Wavelet denoising
Trial wise segmentation
Reshape
4 Conv1D layers
4 Conv1D layers
4 Conv1D layers
Conv layers
Bi directional LSTM layer 1
Bi directional LSTM layer 2
Bi-LSTM layers
Fully Connected (512)
Fully Connected (N_A)
Output hand activity classification
Learning kinematic characteristics of the hand activities

Fig. 3 – The block diagram depicts learning steps involved in proposed hybrid CNN and Bi-LSTM based EMGHandNet architecture.

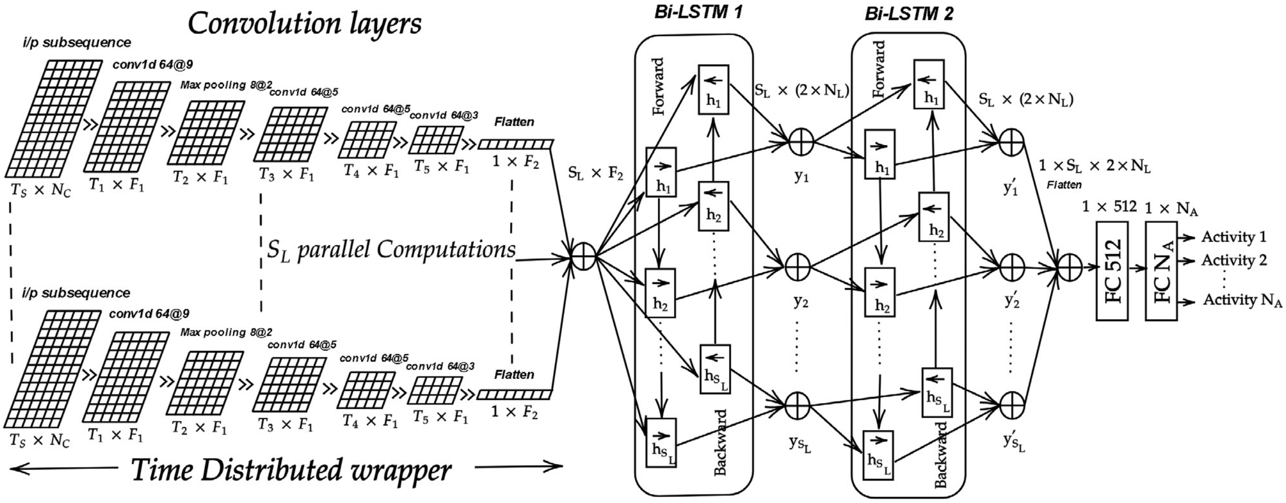


Fig. 4 – The expanded view of hybrid CNN and Bi-LSTM based EMGHandNet architecture.

**Table 1 – Action categories mapped to robotic arm control.**

| Action Category | Robotic or exoskeleton arm control mapping |
|---|---|
| Gestures | To control finger motions of the robotic arm and bionic hand [50,51] |
| wrist movements | To control degrees of freedom [51] |
| grasping objects | To guide an arm for the task such as pick and place objects [12,13] |
| hand movements | To provide multiple degrees of freedom for exoskeleton arm [14] |

The obtained feature map is further processed by three more convolution layers which lead to the generation of an output array of size $T_5 \times F_1$, which is then flattened to a vector size $1 \times F_2$. The feature sets computed in $S_L$ parallel streams are concatenated into a feature array of dimension $S_L \times F_2$, which is provided as input to a stack of Bi-LSTM layers.

Intuitively, Bi-LSTM is able to encode the long-term temporal information better than the uni-directional LSTM. It is because the forward LSTM of Bi-LSTM encodes the sequence from start to end of a trial and the backward LSTM encodes the sequence from the end to start of a trial. The model is better equipped with the temporal representation of the final activity posture from the backward LSTM layer while the forward LSTM layer is able to predict the intermediate movement of activity thus contributing to the improved classification accuracy of the model. The intermediate movement prediction from the backward LSTM layer also reduces the number of possibilities for subsequent movements in terms of prediction and enforces improved separability in the abstract feature space and thus minimizes the class ambiguity. The output dimension of the Bi-LSTM layer with all the sub sequences is $S_L \times 2 \times N_L$ that is flattened and subsequently processed by the two dense layers. Finally, the last dense layer consists of a softmax activation function to obtain output probabilities for each of the hand activity classes.

## 4. Experimental setup

### 4.1. Dataset Description

In this section, we briefly discuss the five publicly available benchmark datasets used in the experiments; NinaPro DB1, NinaPro DB2, NinaPro DB4, BioPatRec DB2 and UCI Gesture. In the respective measurement sessions, the sEMG sensors were placed on various muscle locations on the upper limbs. The muscle sites included are Flexor carpi radialis, Flexor carpi ulnaris, Extensor digitorum, Extensor carpi radialis brevis muscle, and Palmaris longus [48]. The datasets consist of hand activities broadly categorized into gestures, wrist movements, grasping objects and hand movements. The hand action categories that can be mapped to robotic or exoskeleton arm controls are provided in Table 1. It is assumed that the difference between static and dynamic gestures is the angular velocity of the fingers during activity [49]. In static actions, the angle between fingers does not vary during activity but varies in dynamic activities. The position of the hand can vary in both static and dynamic actions. As per this assumption, the datasets consist of static hand actions only. The datasets are illustrated in detail in the references provided in the descriptions given below.

A brief description of each dataset analyzed is as follows:

1. NinaPro DB1 [26]: The dataset consists of sEMG signals extracted from 27 subjects performing various hand activities are grouped into three exercises namely exercise A, exercise B and exercise C. The exercise A consists of single finger flexion and extension movements. The exercise B consists of multiple finger flexion and extension movements as well as wrist movements. The exercise C consists of grasping of household objects. Each activity is performed for a duration of 5s with a rest period of 3s and repeated over 10 trials.
2. NinaPro DB2 [26]: The dataset consists of sEMG signals extracted from 40 subjects while they performed finger gestures and grasping of objects. The 49 classes of actions are grouped into exercise B, exercise C and exercise D. The exercises B and C are same as in NinaPro DB1. The exercise D consists of 9 force patterns obtained while pressing fingers on force sensors.
3. NinaPro DB4 [52]: In this dataset, the sEMG signals are collected from 10 subjects. The hand activities are same as in the NinaPro DB1; the only difference is the data is obtained at a higher sampling rate.
4. BioPatRec DB2 [53]: This dataset consists of 26 hand movements obtained from 17 subjects. These activities include six basic hand movements such as open/close, hand pronation/supination and wrist flexion and extension and 20 movements made up of combinations of the six basic movements. Each movement is performed for a duration of 3s with a rest duration of 3s.
5. UCI Gesture [54]: This dataset consists of 7 hand movements obtained from 36 subjects. The activities include seven basic movements such as a hand clenched in a fist, the wrist flexion and extension, radial and ulnar deviation of the wrist, the extended palm, and the rest state. Since the extended palm is not performed by each of the subjects, we have considered first six classes of data for the classification task.

### 4.2. Dataset Preprocessing and Preparation

As summarized in the Table 2, each trial of the data $\mathbf{x}_n$ is preprocessed differently for each of the dataset used. The data pre-processing is performed in MATLAB R2020 to obtain the train and test data files. As the sEMG signals are contaminated by unwanted components such as the line-noise and the receiver noise. To mitigate their effects, the following three steps are used: a) Line noise filtering at 50 Hz to remove power supply noise, b) First order Butterworth low pass filter-

Table 2 – Preprocessing steps for each of the datasets.

| Processing step | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| Power line noise filtering at 50 Hz | Not applied | Applied | Applied | Not Applied | Not Applied |
| Low pass filtering with cutoff frequency fc = 500 Hz | Not applied | Applied | Applied | Not Applied | Not Applied |
| Wavelet Denoising | Applied | Applied | Applied | Applied | Applied |

Table 3 – Dataset characteristics and setup for numerical experiments.

| Parameter | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| No. of subjects (S) | 27 | 40 | 10 | 17 | 36 |
| No. of classes ($N_A$) | 52 | 49 | 52 | 26 | 6 |
| No. of channels ($N_C$) | 10 | 12 | 12 | 8 | 8 |
| No. of repetitions (R) | 10 | 6 | 6 | 3 | 4 |
| Sampling frequency ($N_s$) (samples/s) | 100 | 2000 | 2000 | 2000 | 1000 |
| Trial length ($N_T$) | 500 | 10000 | 10000 | 6000 | 1500 |
| Activity duration (T) (sec) | 5 | 5 | 5 | 3 | 3 |
| Rest period (sec) | 3 | 3 | 3 | 3 | 3 |
| Sensor type | Otto Bock | Delsys Trigno Wireless | Cometa MiniWave | Bipolar Silver Electrodes | MYO Thalmic bracelet |
| Total Patterns (N) | 14040 | 11760 | 3120 | 1326 | 864 |
| Patterns per class | 270 | 240 | 60 | 51 | 24 |
| Train patterns (Aggregate data) | 9828 | 7840 | 2080 | 884 | 648 |
| Test patterns (Aggregate data) | 4212 | 3920 | 1040 | 442 | 216 |
| Train patterns (Sub-wise data) | 364 | 196 | 208 | 52 | 18 |
| Test patterns (Sub-wise data) | 156 | 98 | 104 | 26 | 6 |
| Train trial numbers | 1,3,4, 6,8,9, 10 | 1,3,4, 6 | 1,3,4, 6 | 1,3 | 1,3,4 |
| Test trial numbers | 2,5,7 | 2, 5 | 2,5 | 2 | 2 |

ing at a cut-off frequency of 500 Hz (since sEMG signal lies in the band of $20 - 380$Hz), c) Wavelet denoising at an order of 8, with the symlet mother wavelet. The prepossessing steps are applied as per requirement on each of the datasets. Different preprocessing steps are applied to different datasets due to the following reasons. For the NinaPro DB1, since the data is already filtered with 50 Hz line noise only wavelet denoising is applied. As the data is sampled at 100 Hz, it does not require low pass filtering with a 500 Hz cut-off frequency. For the NinaPro DB2 and NinaPro DB4, since the sampling rate is 2 kHz, filters for 50 Hz line noise and a low pass filter with a cutoff at 500 Hz are applied. The filtered data is denoised with a wavelet-based method. Since the BioPatRec DB2 is available pre-filtered for the 50 Hz line noise and band-pass filtered with a pass-band frequencies of 20 Hz and 400 Hz, no further filtering is required. Similarly, the UCI Gesture data is also available pre-filtered; only wavelet denoising is applied.

For a given trial, if the number of samples are exceeding $N_T$, the excess data is truncated. In other case, if the number of samples is less than $N_T$, zero padding is used. The data is split trial-wise into 70% for training as mentioned in Table 3 and 30% for testing [29]. The train and test data sets are standardised channel wise, so that the resultant data has zero mean and unit variance. These files are converted into pandas data frames with TensorFlow back-end. The tensor shapes for each layer are given in the Table 4.

### 4.3. *Network Settings and Model Training*

The EMGHandNet is implemented with the following parameters. The learning rate (*lr*) is initialized to $10^{-3}$ with a batch size of 16. The hyper parameters tuned for an improved performance are given in the Table 5. The number of filters at each convolution layer and the kernel size are mentioned in the Fig. 4, for example conv1d 64@9 refers to 64 filters with a kernel size 9. The EMGHandNet is trained and tested separately for each dataset and corresponding classification accuracy is evaluated as follows:

$$\text{Accuracy} = \frac{\text{Number of correctly classified trials}}{\text{Total number of trials}} \times 100. \quad (7)$$

The experiments are carried out on a machine consisting of Nvidia Quadro RTX 6000 24 GB RAM Graphical Processing Unit (GPU) card. The model deployment is done in TensorFlow 2.2.0 framework with python.

## 5. Results and discussion

The following models are trained and tested on the five benchmark datasets mentioned earlier.

1. EMGHandNet: The proposed hybrid CNN and Bi-LSTM architecture.

Table 4 – Tensor shapes obtained during implementation - Aggregate data scheme. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms)).

| Shape of Tensor | Values for NinaPro DB1 | Values for NinaPro DB2 and NinaPro DB4 | Values for BioPatRec DB2 | Values for UCI Gesture |
|---|---|---|---|---|
| $N_T \times N_C$ | $500 \times 10$ | $10000 \times 12$ | $6000 \times 8$ | $1500 \times 8$ |
| $S_L \times T_S \times N_C$ | $25 \times 20 \times 10$ | $25 \times 400 \times 12$ | $15 \times 400 \times 8$ | $30 \times 50 \times 8$ |
| $S_L \times T_1 \times F_1$ | $25 \times 10 \times 64$ | $25 \times 200 \times 64$ | $15 \times 200 \times 64$ | $30 \times 25 \times 64$ |
| $S_L \times T_2 \times F_1$ | $25 \times 2 \times 64$ | $25 \times 97 \times 64$ | $15 \times 97 \times 64$ | $30 \times 9 \times 64$ |
| $S_L \times T_3 \times F_1$ | $25 \times 1 \times 64$ | $25 \times 49 \times 64$ | $15 \times 49 \times 64$ | $30 \times 5 \times 64$ |
| $S_L \times T_4 \times F_1$ | $25 \times 1 \times 64$ | $25 \times 25 \times 64$ | $15 \times 25 \times 64$ | $30 \times 3 \times 64$ |
| $S_L \times T_5 \times F_1$ | $25 \times 1 \times 64$ | $25 \times 13 \times 64$ | $15 \times 13 \times 64$ | $30 \times 2 \times 64$ |
| $S_L \times 1 \times F_2$ | $25 \times 1 \times 64$ | $25 \times 1 \times 832$ | $15 \times 1 \times 832$ | $30 \times 1 \times 128$ |
| $S_L \times F_2$ | $25 \times 64$ | $25 \times 832$ | $15 \times 832$ | $30 \times 128$ |
| $S_L \times (2 \times N_L)$ | $25 \times (2 \times 200)$ | $25 \times (2 \times 200)$ | $15 \times (2 \times 200)$ | $30 \times (2 \times 200)$ |
| $1 \times S_L \times 2 \times N_L$ | $1 \times 10000$ | $1 \times 10000$ | $1 \times 6000$ | $1 \times 12000$ |
| $1 \times N_A$ | $1 \times 52$ | $1 \times 49$ /$1 \times 52$ | $1 \times 26$ | $1 \times 6$ |

Table 5 – Tuned hyper parameters for the EMGHandNet.

| Layer | Parameter Name | Value |
|---|---|---|
| Convolution hyper parameters | Kernel initializer: | he_normal |
| | Strides: | 2 |
| | Kernel regularizer: | $10^{-4}$ |
| Batch Normalization | $\epsilon$: | $10^{-6}$ |
| | Momentum: | 0.95 |
| Batch size | | 16 |
| Learning rate | | $l1 = 10^{-3}$ |
| Optimizer | Adam $\beta1$:, $\beta2$: | 0.9, 0.999 |
| Activation | | Tanh & Relu |
| Dropout | | 0.2093 |
| No. of epochs | | 200 |
| Bi-LSTM | Cells: | 200 |
| TensorFlow version | | 2.2.0 |

2. MsCNN: The Multistream-CNN architecture by Wei et al. [28].
3. EvCNN: Evolved CNN by Olsson et al. [29].
4. CNNLM: Combined CNN - LSTM architecture by Chen et al. [40].
5. Energy features: KNN classifier using energy features by Karnam et al. [23].

Specifically, these models are analyzed in the following two schemes:

- Subject-wise analysis:- In this scheme, the EMGHandNet is trained and tested with the subject wise data. The average classification accuracy across the subjects is computed.
- Aggregated data analysis:- In this analysis, the EMGHandNet is trained and tested with aggregated data from the available subjects with split-up scheme mentioned in data preprocessing stage. The average accuracy is calculated as per (7).

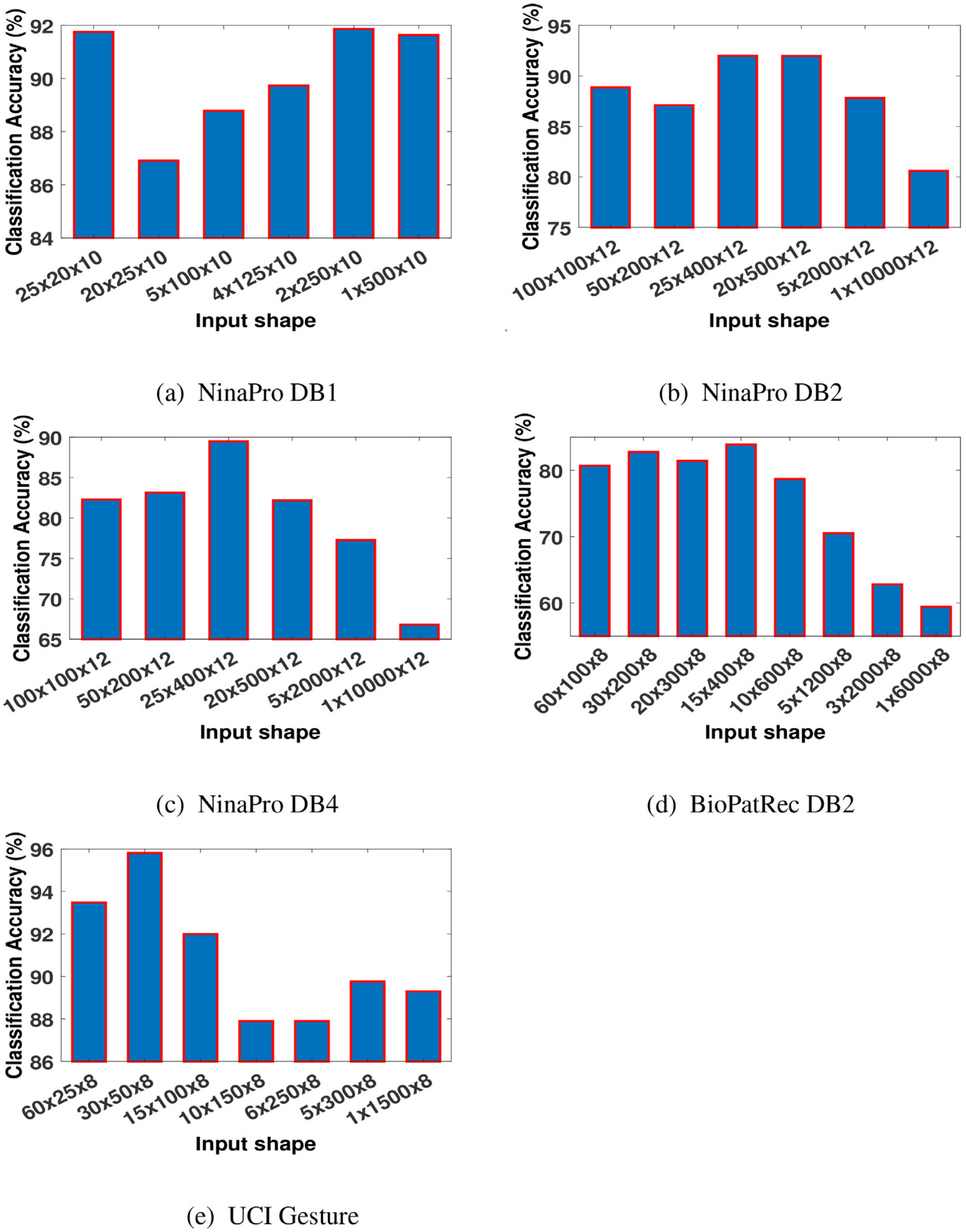
To validate the EMGHandNet, the following analyses are carried out: (1) analysis of the role of input shapes, (2) comparative analysis with the RNN layers, (3) analysis of the preprocessing methods, (4) loss curves for different datasets, (5) comparison with the state of the art models, and (6) time and space complexity analysis. For fair comparison, we have used the same dataset setup and input shape processing techniques both in the EMGHandNet and the other models.

### 5.1. *Role of Input Shapes*

The EMGHandNet is also analysed with respect to different input shapes of the data as shown in Fig. 5 for the five datasets. It is observed that for some datasets with input shape consisting of samples with 200*ms* time segment (for NinaPro DB1, $T_S = 20$ samples, for NinaPro DB2, NinaPro DB4, and BioPatRec DB2, $T_S = 400$ samples) and 50*ms* time segment (for UCI Gesture, $T_S = 50$ samples), the performance is superior compared to when other time segments are used. For the BioPatRec dataset, the highest accuracy of 83.9% is achieved for an input shape of $[15, 400]$ (corresponds to 200 ms time segment). This analysis suggests that the performance of the model varies with respect to different shapes and achieves an optimal average accuracy for the identified input shape. Finally, for the UCI Gestures, a time segment of 0.05s out of 3s trial data provides improved classification. It is observed that a similar time segment of 0.2s has provided better results for four datasets except UCI Gesture. Hence an optimal time segment is important for improved capture of temporal relations. The optimal time segment obtained is dependent on various factors such as sampling frequency (Ns), activity duration (T) and number of samples within a sub-sequence (Ts). The training and testing of datasets are done independently across the five datasets and observed that there is no direct linear relationship between No.of samples within a trial ($N_T$) and optimal time segment.

### 5.2. *Performance Analysis with RNN Layers*

The performance of the EMGHandNet is also analysed by varying the number of stacked Bi-LSTM layers and results are shown in Fig. 6. It is observed that the EMGHandNet model with 4 Conv and 2 Bi-LSTM layers outperforms the EMGHandNet model with 4 Conv layers and other Bi-LSTM



(a) NinaPro DB1

(b) NinaPro DB2

(c) NinaPro DB4

(d) BioPatRec DB2

(e) UCI Gesture

**Fig. 5 – Performance analysis from Aggregate Data scheme with different shapes for NinaPro DB1, NinaPro DB2, NinaPro DB4, BioPatRec DB2 and UCI Gesture datasets. For NinaPro DB1, $T_s = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_s = 400$ and for UCI Gesture, $T_s = 50$ (Time segment of 50 ms).**

configurations. The model with no Bi-LSTM layer corresponds to the network with four Conv layers only. This analysis shows that the performance of the model with the Bi-LSTM layers is better than without Bi-LSTM layer (i.e., only Conv layers). Importantly, for the EMGHandNet, this analysis also supports the choice of using two Bi-LSTM layers following the CNN network.

The proposed model is also analysed with different type of RNN layers, such as LSTM, Gated Recurrent Unit (GRU), Bidirectional GRU (Bi-GRU), RNN, and Bidirectional RNN (Bi-RNN) on NinaPro DB1 dataset. The results in terms of the classification accuracy are shown in Fig. 7. From these results, it is evident that the Bi-LSTM is better suited in the proposed framework as it leads to the highest performance as compared to other type of RNN layers. The Bi-LSTM layer better captures the long-term temporal relations compared to other type of RNN layers as it learns both forward and backward temporal characteristics.

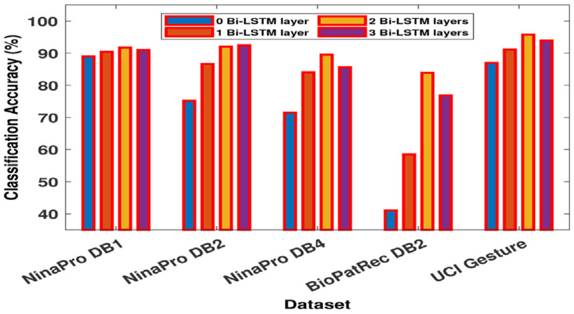


**Fig. 6 – Performance analysis with respect to No. of Bi-LSTM layers in the proposed EMGHandNet model over all datasets - Aggregate data scheme. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).**

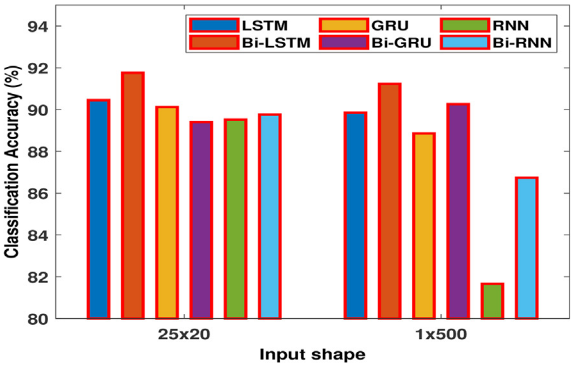


**Fig. 7 – NinaPro DB1 performance analysis with different type of recurrent layers such as LSTM, Bi-LSTM, GRU, Bi-GRU, RNN and Bi-RNN layers in the proposed model.**

### 5.3. *Analysis on Pre-processing Methods*

The role of preprocessing for the improvement of classification performance is analyzed through following experiments, see Table 6. In one experiment, the preprocessing method in the EvCNN [29] is replaced with our preprocessing approach. This is compared with the combination consisting of the preprocessing method from the EvCNN followed by the EMGHandNet. The EMGHandNet with our preprocessed data exhibits better classification accuracy compared to other combinations. This analysis shows that the preprocessing technique used in this work is better suited for sEMG data based hand activity classification as compared to the preprocessing and architecture from [29]. Further experiments are carried out by comparing the EMGHandNet model with and without any preprocessing of data. It is observed that, for BioPatRec DB2, there is an improvement of 17.69% in classification accuracy compared to that without preprocessing. Similarly, for NinaPro DB4, a comparable performance improvement of 10.58% is observed. Additionally, for the remaining datasets the improvement is at least 3%. This shows that there is an improvement in performance of the model when data is preprocessed compared to the case without preprocessing.

### 5.4. *Loss Curves for Individual Datasets*

Fig. 8 shows the loss curves from the EMGHandNet corresponding to each of the five datasets. The input shapes chosen are as follows, for NinaPro DB1: 25 × 20 × 10, for NinaPro DB2 and NinaPro DB4: 25 × 400 × 12, for BioPatRec DB2: 15 × 400 × 8 and for for UCI Gesture: 30 × 50 × 8. The EMGHandNet obtains a stable response after 70 epochs for each of the five datasets.

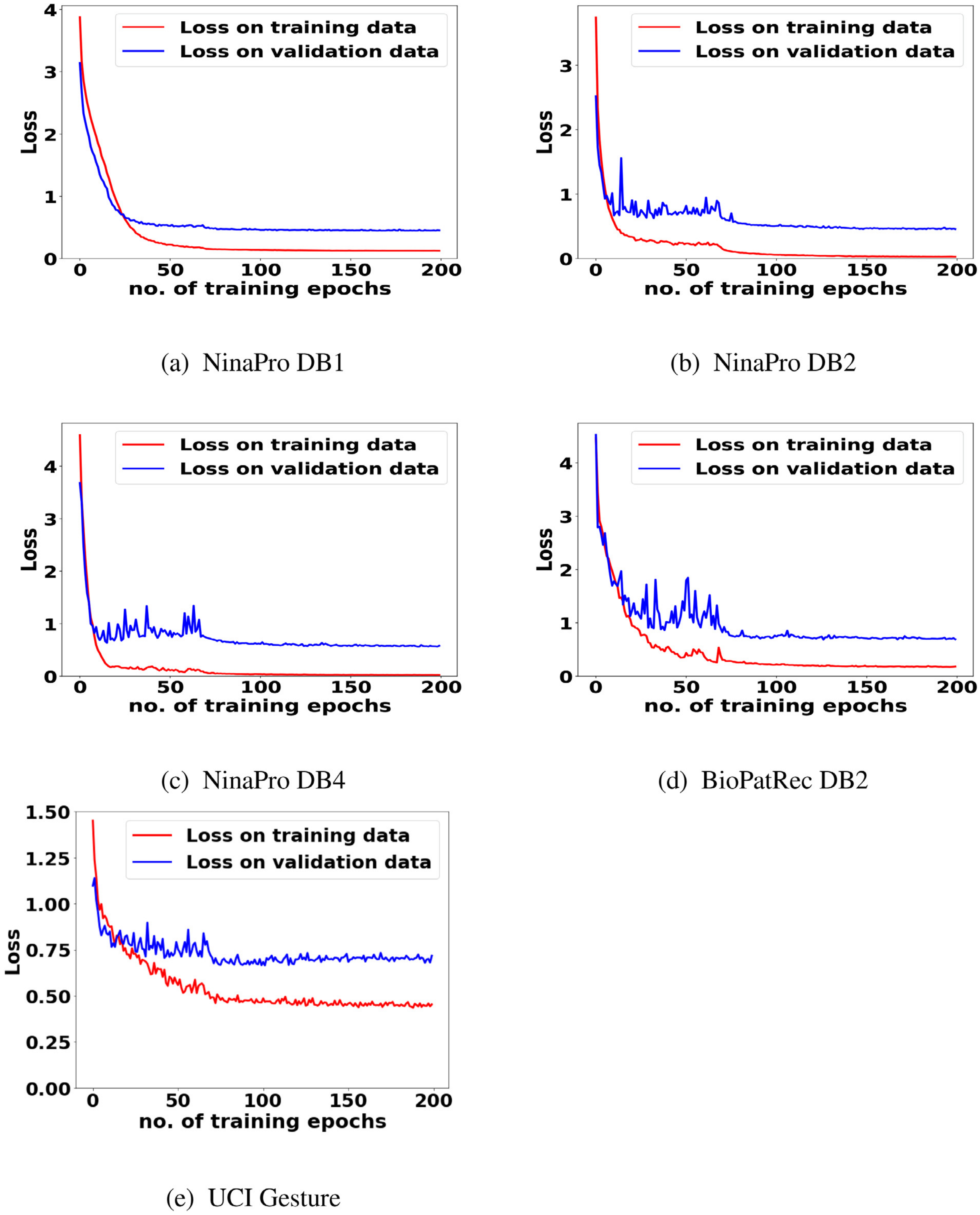
### 5.5. *Performance Comparison*

The EMGHandNet performance is compared with that of the MsCNN, EvCNN, CNNLM where the results are reproduced in this work and a few other models for which the results are reported from the literature (see Tables 7 and 8).

In the subject-wise analysis, as shown in Table 7, for the NinaPro datasets, the EMGHandNet shows higher average test accuracy compared to the state-of the art methods. The subject-wise average accuracy of the EMGHandNet model for NinaPro DB1 is 95.77%, which is an improvement of 7.57% over 88.2% from the Multi-view CNN [57]. For the NinaPro DB2 dataset, the EMGHandNet with an accuracy of 95.9% improved by 12.2% against 83.7% of the Multi-view CNN [57]. The EMGHandNet has an improvement of 8.73% over reproduced CNNLM [40]. For the Ninapro DB4, the EMGHandNet with an average test accuracy of 91.65%, outperforms the Attention sEMG model [58] by 18.65%. For the UCI Gesture, there is a small improvement of 1.33% over Deep Neural architecture [59] by attaining a test accuracy of 98.33%. Based on the existing approaches, the BioPatRec DB2 dataset is used without any data augmentation, hence for the EMGHandNet model, the performance of 91.29% is approaching that of the state of the art.

**Table 6 – Performance analysis using accuracy (%) of different preprocessing techniques with various architectures - Aggregate data scheme. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).**

| Dataset/ Methodology | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| Olsson Preprocessing - EvCNN [29] | 81.57 | 66.64 | 13 | 77.71 | 54.41 |
| Olsson Preprocessing - EMGHandNet | 86.98 | 81.11 | 21.6 | 67.8 | 91.16 |
| Proposed Preprocessing - EvCNN [29] | 84.54 | 60.52 | 38 | 52.71 | 77.2 |
| Proposed Preprocessing - EMGHandNet | **91.76** | **92.01** | **89.5** | **83.9** | **93.48** |
| Without Preprocessing - EMGHandNet | 89.5 | 91.62 | 78.92 | 66.21 | 93.02 |



(a) NinaPro DB1

(b) NinaPro DB2

(c) NinaPro DB4

(d) BioPatRec DB2

(e) UCI Gesture

**Fig. 8 – Loss curve observed during training of EMGHandNet over NinaPro DB1, NinaPro DB2, NinaPro DB4, BioPatRec DB2 and UCI Gesture datasets - Aggregate data scheme. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).**

In the aggregated scheme, as shown in the Table 8, for the NinaPro datasets, the accuracy of the EMGHandNet is again better compared to that of the existing models. Specifically, for NinaPro DB1, the EMGHandNet obtained an accuracy of 91.76%, which is an improvement of at least 3.56% as compared to 88.2% based on Multi-view CNN [57]. For NinaPro DB2, an accuracy of 92.01% is observed using the EMGHandNet, which is an improvement of 8.31% as compared to

**Table 7 – Classification accuracy (%) analysis with different architectures - Subject wise analysis. * are reproduced results. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).**

| Dataset/ Architecture | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| RMS, MAV, and DASDV by yang et al. [55] (Sub-wise) | 91.35 | - | - | - | - |
| Multi-sEMG-features image by Cheng et al. [56] | 82.54 | - | - | - | - |
| MsCNN [28] (Sub-wise) | 85 | - | - | - | - |
| MsCNN[28] (Sub-wise) * | 74.25 | 50.99 | 34.1 | 66.18 | 95.58 |
| EvCNN [29] (Sub-wise) | 81.4 | 71.6 | - | 91.4 | - |
| EvCNN [29] (Sub-wise) * | 68.06 | 80.54 | 22.3 | 84.19 | 45.55 |
| CNN-RNN by Hu et al. [36] | 87 | 82.2 | - | 94.1 | - |
| Multi-View CNN by Wei et al. [57] | 88.2 | 83.7 | 58 | 94.0 | - |
| Attention sEMG by Josephs et al. [58] | - | - | 73 | - | - |
| Deep Neural Network by Potekhin et al. [59] | - | - | - | - | 97 |
| CNNLM [40] (Sub-wise)* | 92.99 | 87.17 | 87.37 | 89.17 | 86.51 |
| **EMGHandNet (Sub-wise)** | **95.77** | **95.9** | **91.65** | **91.29** | **98.33** |

**Table 8 – Classification accuracy (%) analysis with different architectures - Aggregate data scheme. * are reproduced results. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).**

| Dataset/ Architecture | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| Benchmark classifier with handcrafted features [26] (Aggregate) | 75.3 | 75.3 | - | - | - |
| CNN by Atzori et al. [25] (Aggregate) | 66.6 | 60.3 | - | - | - |
| LDA by Nazemi et al. [60] | 84.23 | - | - | - | - |
| LS-SVM by Nazemi et al. [60] | 85.19 | - | - | - | - |
| Random Forest by Rubio et al. [61] | - | - | - | - | 95.39 |
| Energy features [23] (Aggregate) | 87.07 | 90.61* | 60.67* | 89.36* | 93.98* |
| MsCNN [28] (Aggregate) * | 60 | 31.41 | 33.9 | 62.4 | 89.5 |
| EvCNN [29] (Aggregate) * | 81.57 | 66.64 | 13 | 77.71 | 54.41 |
| CNN-RNN by Hu et al. [36] | 87 | 82.2 | - | 94.1 | - |
| Multi-View CNN by Wei et al. [57] | 88.2 | 83.7 | 58 | 94.0 | - |
| Attention sEMG by Josephs et al. [58] | - | - | 73 | - | - |
| CNNLM [40] (Aggregate)* | 79.26 | 78.71 | 83.92 | 71.65 | 86.51 |
| **EMGHandNet (Aggregate)** | **91.76** | **92.01** | **89.5** | **83.9** | **93.48** |

**Table 9 – Statistics from McNEMAR's test on Energy Features [23] and EMGHandNet for NinaPro DB2. Performance comparison based on correct proportions.**

| | Energy Features [23] | | | |
|---|---|---|---|---|
| | | Correct | Incorrect | Total |
| EMGHandNet | Correct | $f_{cc} = 3350$ | $f_{ic} = 200$ | 3550 |
| | Incorrect | $f_{ci} = 255$ | $f_{ii} = 113$ | 368 |
| | Total | 3605 | 313 | 3918 |

83.7% of Multi-view CNN [57] but an improvement of 1.4% against the Energy features [23]. For the NinaPro DB4, the achieved accuracy is 89.5%, which is an 5.58% improvement over 83.92% of the CNNLM [40]. For the UCI Gesture, the achieved accuracy is 93.48%, which is an improvement of 3.98% over 89.5% of the MsCNN [28]. Finally, for the BioPatRec, the EMGHandnet improved the peformance by 6.19% against the accuracy of the EvCNN [29]. The EMGHandnet has outperfomed the machine learning classifiers such as [26,23]. Among the existing methods for the five datasets, the CNNLM [40] and Energy features [23] comes closest to the accuracy of the EMGHandNet. Additionally, for the 4 datasets other than UCI Gesture, the MsCNN [28] has the least performance among the competing methods. The McNemar's test [62] is implemented to analyze the statistical significance of the observed improvement in classification performance. The terms in Table 9 are defined as follows: $f_{cc}$ is the number of patterns for which both classifiers result in correct prediction, $f_{ci}$ when EMGHandNet is correct and the state of the art classifier is incorrect, $f_{ic}$ is Vice versa, and $f_{ii}$ is the number of patterns for which both classifiers predictions are incorrect. From these metrics, the chi-square statistic is calculated [62] as $6.64 > 3.84$ (the threshold for one degree of freedom). Hence the improvement is significant at 95% confidence level. For the NinaPro DB1, the chi square value evaluated against the Energy features [23] is 47.2 and for the NinaPro DB4, the chi square value against CNNLM [40] is 10.07. For the UCI Gestures, the chi-square value against the Energy features [23] is

0.529, which indicates that both the classifiers provide similar results.

### 5.6. Time and Space Complexity Analysis

The time and space complexity of the ML methods are calculated based on the analysis provided in [29]. The time complexity of each Conv 1D, Bi-LSTM, and Fully Connected (FC) layer are computed based on the number of Floating Point Operations Per Second (FLOPS) and provided in Table 10. The time complexity for UCI Gesture can also be computed similar to the methods used for the first four datasets, see Table 10. The total training time, for NinaPro DB1, NinaPro DB2, NinaPro DB4, BioPatRec DB2 and UCI Gesture datasets are 48 min, 48.3 min, 16 min, 6.6 min and 6.6 min respectively. The space complexity of Conv 1D, Bi-LSTM, FC and Batch Normalization (BN) layer are given as 4HC [29] based on the output shape $H \times C$ and given in Table 11. Additional memory is allocated based on No. of learning parameters (weights and biases).

The time complexity and space complexity of Bi-LSTM is same [43]. The space complexity of Conv 1D [63], FC and BN layers are theoretically computed considering 32 bit operating system. The computational complexity depends on the number of trainable parameters. Based on this, the complexity of different architectures are compared as provided in Table 12. For most of the datasets, it is observed that the computational complexity of the EMGHandNet is higher compared to that of the EvCNN and CNNLM but less compared to the MsCNN.

### 5.7. Discussion

The performance of the proposed EMGHandNet model has improved due to the following reasons:

- The proposed hybrid CNN and Bi-LSTM architecture is able to extract both cross-channel and temporal features. The 1-D convolution encodes the cross-channel and short-term temporal information and the Bi-LSTM encodes the long-term temporal information in both forward and backward directions.
- The whole trial data of input samples is provided to the model as compared to segmented data in the existing methods. Instead of considering segmented data as an input pattern, the whole trial data, as a pattern, provides

**Table 10 – Time complexity of layers in the network ($n$ = No.of filters, $v$ = vertical stride, $y$ = kernel height, $N$ = No. of patterns, $N_e$ = No. of epochs and $N_A$ = No. of activities).**

| Layer Name | Input shape | Theoretical values for layers (FLOPS) | Theoretical values for NinaPro DB1 (FLOPS) | Theoretical values for NinaPro DB2 and NinaPro DB4 (FLOPS) | Theoretical values for BioPatRec DB2 (FLOPS) |
|---|---|---|---|---|---|
| $Conv1d$ 64@9 | $T_S \times N_C$ | $(T_S/v)n(2yN_C + 1)$ | 115840 | 2777600 | 185600 |
| $Conv1d$ 64@5 | $T_2 \times F_1$ | $(T_2/v)n(2yF_1 + 1)$ | 41024 | 1989664 | 1989664 |
| $Conv1d$ 64@5 | $T_3 \times F_1$ | $(T_3/v)n(2yF_1 + 1)$ | 20512 | 1005088 | 1005088 |
| $Conv1d$ 64@3 | $T_4 \times F_1$ | $(T_4/v)n(2yF_1 + 1)$ | 12320 | 308000 | 308000 |
| $Bi - LSTM1$ | $S_L \times F_2$ | $2 \times 4 \times (F_2 + N_L + 1)N_L$ | 424000 | 1652800 | 1652800 |
| $Bi - LSTM2$ | $S_L \times 2 \times N_L$ | $2 \times 4 \times (2N_L + N_L + 1)N_L$ | 1281600 | 1281600 | 1281600 |
| $FC1$ | $1 \times S_L \times 2 \times N_L$ | $S_L \times 2 \times N_L \times 512$ | $10000 \times 512$ | $10000 \times 512$ | $6000 \times 512$ |
| $FC2$ | $1 \times 512$ | $512 \times N_A$ | $512 \times 52$ | $512 \times 49/512 \times 52$ | $512 \times 26$ |
| $Total$ | – | – | 5760320 | 14159840/14161376 | 6743264 |
| $O(f)$ | – | $Total \times N \times N_e$ | $1.6 \times 10^{13}$ | $3.3 \times 10^{13}/8.8 \times 10^{12}$ | $1.7 \times 10^{12}$ |

**Table 11 – Space complexity of layers in the network ($n$ = No. of filters, $v$ = vertical stride, $y$ = kernel height, and $N_A$ = No. of activities).**

| Layer Name | Input shape | Output shape | Based on output shape (bytes) | Based on No. of learning parameters (bytes) |
|---|---|---|---|---|
| $Conv1d$ 64@9 | $T_S \times N_C$ | $T_1 \times F_1$ | $4T_1F_1$ | $4n(yN_C + 1)$ |
| $Conv1d$ 64@5 | $T_2 \times F_1$ | $T_3 \times F_1$ | $4T_3F_1$ | $4n(yF_1 + 1)$ |
| $Conv1d$ 64@5 | $T_3 \times F_1$ | $T_4 \times F_1$ | $4T_4F_1$ | $4n(yF_1 + 1)$ |
| $Conv1d$ 64@3 | $T_4 \times F_1$ | $T_5 \times F_1$ | $4T_5F_1$ | $4n(yF_1 + 1)$ |
| $Bi - LSTM1$ | $S_L \times F_2$ | $S_L \times (2 \times N_L)$ | $4 \times 2S_LN_L$ | $4 \times 2 \times 4 \times (F_2 + N_L + 1)N_L$ |
| $Bi - LSTM2$ | $S_L \times (2 \times N_L)$ | $S_L \times (2 \times N_L)$ | $4 \times 2S_LN_L$ | $4 \times 2 \times 4 \times (2N_L + N_L + 1)N_L$ |
| FC1 | $1 \times S_L \times 2 \times N_L$ | $1 \times 512$ | $4 \times 512$ | $4 \times 512(S_L \times 2 \times N_L + 1)$ |
| FC2 | $1 \times 512$ | $1 \times N_A$ | $4 \times N_A$ | $4 \times N_A(512 + 1)$ |
| BN1 | $T_1 \times F_1$ | $T_1 \times F_1$ | $4T_1F_1$ | $16 \times F_1$ |
| BN2 | $T_3 \times F_1$ | $T_3 \times F_1$ | $4T_3F_1$ | $16 \times F_1$ |
| BN3 | $T_4 \times F_1$ | $T_4 \times F_1$ | $4T_4F_1$ | $16 \times F_1$ |
| BN4 | $T_5 \times F_1$ | $T_5 \times F_1$ | $4T_5F_1$ | $16 \times F_1$ |
| BN5 | $1 \times 512$ | $1 \times 512$ | $4 \times 512$ | $16 \times 512$ |

Table 12 – Computational complexity comparison based on trainable parameters of various architectures - Aggregate data scheme. For NinaPro DB1, $T_S = 20$, for NinaPro DB2, NinaPro DB4 and BioPatRec DB2, $T_S = 400$ and for UCI Gesture, $T_S = 50$ (Time segment of 50 ms).

| Trainable parameters/ Methodology | NinaPro DB1 | NinaPro DB2 | NinaPro DB4 | BioPatRec DB2 | UCI Gesture |
|---|---|---|---|---|---|
| MsCNN [28] | $8.69 \times 10^6$ | $3.12 \times 10^6$ | $1.97 \times 10^8$ | $9.90 \times 10^7$ | $6.61 \times 10^7$ |
| EvCNN [29] | $3.77 \times 10^5$ | $1.31 \times 10^7$ | $2.80 \times 10^6$ | $2.92 \times 10^6$ | $1.66 \times 10^6$ |
| CNNLM [40] | $1.52 \times 10^5$ | $5.41 \times 10^5$ | $5.42 \times 10^5$ | $4.15 \times 10^5$ | $3.06 \times 10^5$ |
| EMGHandNet | $6.59 \times 10^6$ | $7.82 \times 10^6$ | $7.82 \times 10^6$ | $5.76 \times 10^6$ | $7.69 \times 10^6$ |

better sequential information for the model. This is because we do not know in which time segment the subject has performed the activity within a trial.

It is observed that subject wise performance analysis is better compared to aggregated data scheme because the model understands hand actions better due to low intra-subject signal covariance in the former case against the higher inter-subject signal covariance in the latter case. A trend or correlation is observed with respect to the duration of the segment being classified and the performance of the model. For example, a time segment of 0.2s has shown better performance for the datasets NinaPro DB1, NinaPro DB2, NinaPro DB4, and BioPatRec DB2 and a time segment of 0.05s has shown better performance for UCI Gesture. It is inferred that the performance of the model is affected by the input shape provided to the CNN. Some of the limitations of the proposed model are: (a) The model may need further innovation to improve the performance for the BioPatRec DB2 data. (b) For each dataset, the model requires separate training and testing. To reduce re-training requirements a transfer learning technique can be explored.

## 6. Conclusion & Future Scope

### 6.1. Conclusion

In this paper, we have proposed a hybrid CNN and Bi-LSTM architecture for the classification of human hand activities by using the sEMG signals. The EMGHandNet is analyzed for five benchmark sEMG datasets for the hand activity classification. The EMGHandNet outperforms (by at least 4% and up to 18.65%) the state-of-the-art models in terms of the average classification accuracy for the NinaPro DB1, NinaPro DB2 and NinaPro DB4 datasets. A superior performance is also observed in the aggregated data analysis scheme. It is also observed that the preprocessing performed in this paper is better than the existing ones for the said problem. Moreover, it is found that the performance of the EMGHandNet model can be tuned based on the shape of input data to the time distributed wrapper. Finally, the two layer Bi-LSTM is determined to be a better choice as compared to other RNN types.

### 6.2. Future Scope

As the sEMG signal is considered in time domain only, there is scope for performance improvement by converting the given signals to the wavelet domain or the Empirical Mode decomposition (EMD) domain before further processing by the deep learning models. The model can also be implemented on a hardware device to understand issues during practical implementation. When deploying on hardware, some of the issues are time latency and portability. The time latency to test a sample is $1.8\mu s$ in the best case and $518\mu s$ in the worst case which is a major factor to be considered for real time implementation. We further plan to build an sEMG signal dataset for Indian population to analyse geography related variations among the datasets. Finally, the model can be further improved to learn from the signals corresponding to highly dynamic and high speed movements involved in activities such as sports.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This research is funded by SERB, Govt. of India under Project Grant No. CRG/2019/003801.

REFERENCES

[1] Guo W, Yao P, Sheng X, Zhang D, Zhu X. An enhanced human-computer interface based on simultaneous sEMG and NIRS for prostheses control. In: 2014 IEEE International Conference on Information and Automation (ICIA). p. 204–7.

[2] Fan Y, Yin Y. Active and progressive exoskeleton rehabilitation using multisource information fusion from EMG and force-position EPP. IEEE Trans Biomed Eng. 2013;60 (12):3314–21.

[3] Li Y, Chen X, Zhang X, Wang K, Wang ZJ. A sign-component-based framework for chinese sign language recognition using accelerometer and sEMG data. IEEE Trans Biomed Eng 2012;59 (10):2695–704.

[4] Cheng J, Chen X, Lu Z, Wang K, Shen M. Key-press gestures recognition and interaction based on sEMG signals. In: International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction. ICMI-MLMI '10. New York, NY, USA: Association for Computing Machinery; 2010. p. 1–4.

[5] Qi W, Su H, Aliverti A. A smartphone-based adaptive recognition and real-time monitoring system for human activities. IEEE Trans Hum Mach Syst 2020;50(5):414–23.

[6] Wen R, Tay W, Nguyen BP, Chng CB, Chui CK. Hand gesture guided robot-assisted surgery based on a direct augmented

reality interface. Comput Methods Programs Biomed 2014;116 (2):68–80.

[7] Wachs JP, Kölsch M, Stern H, Edan Y. Vision-based hand-gesture applications. Commun ACM 2011;54(2):60–71.

[8] Lu Z, Chen X, Li Q, Zhang X, Zhou P. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. IEEE Trans Hum Mach Syst 2014;44(2):293–9.

[9] Lu Z, Tong K, Shin H, Li S, Zhou P. Advanced myoelectric control for robotic hand-assisted training: outcome from a stroke patient. Front Neurol 2017;8:107.

[10] Li Z, Xia Y, Su C. Intelligent networked teleoperation control. Springer; 2015.

[11] Su H, Qi W, Li Z, Chen Z, Ferrigno G, De Momi E. Deep neural network approach in EMG-based force estimation for human-robot interaction. IEEE Trans Artif Intell 2021;2 (5):404–12.

[12] Schabron B, Desai J, Yihun Y. Wheelchair-mounted upper limb robotic exoskeleton with adaptive controller for activities of daily living. Sensors 2021;21(17).

[13] Kim K, Park S, Lim T, Lee SJ. Upper-limb electromyogram classification of reaching-to-grasping tasks based on convolutional neural networks for control of a prosthetic hand. Front Neurosci 2021;15.

[14] Su H, Ovur SE, Zhou X, Qi W, Ferrigno G, De Momi E. Depth vision guided hand gesture recognition using electromyographic signals. Adv Robot 2020;34(15):985–97.

[15] Phinyomark A, Scheme E. An investigation of temporally inspired time domain features for electromyographic pattern recognition. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2018. p. 5236–5240.

[16] Phinyomark A, Phukpattaranont P, Limsakul C. Feature reduction and selection for EMG signal classification. Expert Syst Appl 2012;39(8):7420–31.

[17] Shenoy P, Miller KJ, Crawford B, Rao RPN. Online electromyographic control of a robotic prosthesis. IEEE Trans Biomed Eng 2008;55(3):1128–35.

[18] Al-Timemy AH, Khushaba RN, Bugmann G, Escudero J. Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees. IEEE Trans Neural Syst Rehabil Eng 2016;24(6):650–61.

[19] Shi WT, Lyu ZJ, Tang ST, Chia TL, Yang CY. A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study. Biocybern Biomed Eng 2018;38 (1):126–35.

[20] Waris A, Niazi IK, Jamil M, Englehart K, Jensen W, Kamavuako EN. Multiday evaluation of techniques for EMG-based classification of hand motions. IEEE J Biomed Health Inform 2019;23(4):1526–34.

[21] Tuncer T, Dogan S, Subasi A. Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition. Biomed Signal Process Control 2020;58 101872.

[22] Fatimah B, Singh P, Singhal A, Pachori RB. Hand movement recognition from sEMG signals using fourier decomposition method. Biocybern Biomed Eng 2021;41(2):690–703.

[23] Karnam NK, Turlapaty AC, Dubey SR, Gokaraju B. Classification of sEMG signals of hand gestures based on energy features. Biomed Signal Process Control 2021;70 102948.

[24] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521 (7553):436–44.

[25] Atzori M, Cognolato M, Müller H. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. Front Neurorobot 2016;10:9.

[26] Atzori M, Gijsberts A, Castellini C, Caputo B, Hager AGM, Elsig S, et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Sci Data 2014;1(1) 140053.

[27] Geng W, Du Y, Jin W, Wei W, Hu Y, Li J. Gesture recognition by instantaneous surface EMG images. Sci Rep 2016;6(1):36571.

[28] Wei W, Wong Y, Du Y, Hu Y, Kankanhalli M, Geng W. A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface. Pattern Recognit Lett 2019;119:131–8.

[29] Olsson AE, Björkman A, Antfolk C. Automatic discovery of resource-restricted convolutional neural network topologies for myoelectric pattern recognition. Comput Biol Med 2020;120 103723.

[30] Côté-Allard U, Fall CL, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, et al. Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE Trans Neural Syst Rehabil Eng 2019;27 (4):760–71.

[31] Qi S, Wu X, Chen WH, Liu J, Zhang J, Wang J. sEMG-based recognition of composite motion with convolutional neural network. Sens Actuators A Phys 2020;311 112046.

[32] Betthauser JL, Krall JT, Bannowsky SG, Lévay G, Kaliki RR, Fifer MS, et al. Stable responsive EMG sequence prediction and adaptive reinforcement with temporal convolutional networks. IEEE Trans Biomed Eng 2020;67(6):1707–17.

[33] Gautam A, Panwar M, Wankhede A, Arjunan SP, Naik GR, Acharyya A, et al. Locomo-Net: A low-complex deep learning framework for sEMG-based hand movement recognition for prosthetic control. IEEE J Transl Eng Health Med 2020;8:1–12.

[34] Koch P, Dreier M, Maass M, Phan H, Mertins A. RNN with stacked architecture for sEMG based sequence-to-sequence hand gesture recognition. In: 2020 28th European Signal Processing Conference (EUSIPCO). p. 1600–4.

[35] Ketykó I, Kovács F, Varga KZ. Domain adaptation for sEMG-based gesture recognition with recurrent neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). p. 1–7.

[36] Hu Y, Wong Y, Wei W, Du Y, Kankanhalli M, Geng W. A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition. PLoS One 2018;13(10):1–18.

[37] Wang Y, Wu Q, Dey N, Fong S, Ashour AS. Deep back propagation–long short-term memory network based upper-limb sEMG signal classification for automated rehabilitation. Biocybern Biomed Eng 2020;40(3):987–1001.

[38] Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). p. 2625–34.

[39] Bao T, Zaidi SAR, Xie S, Yang P, Zhang ZQ. A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography. IEEE Trans Instrum Meas 2021;70:1–9.

[40] Chen X, Li Y, Hu R, Zhang X, Chen X. Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method. IEEE J Biomed Health Inform 2021;25(4):1292–304.

[41] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012;25:1097–105.

[42] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5(2):157–66.

[43] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.

[44] Scheme E, Englehart K. On the robustness of EMG features for pattern recognition based myoelectric control; A multi-dataset comparison. In: 2014 36th Annual International

Conference of the IEEE Engineering in Medicine and Biology Society; 2014. p. 650–653.

[45] Scheme EJ, Hudgins BS, Englehart KB. Confidence-based rejection for improved pattern recognition myoelectric control. IEEE Trans Biomed Eng 2013;60(6):1563–70.

[46] Amsüss S, Goebel PM, Jiang N, Graimann B, Paredes L, Farina D. Self-correcting pattern recognition system of surface EMG signals for upper limb prosthesis control. IEEE Trans Biomed Eng 2014;61(4):1167–76.

[47] Lu Z, Tong K, Zhang X, Li S, Zhou P. Myoelectric pattern recognition for controlling a robotic hand: A feasibility study in stroke. IEEE Trans Biomed Eng 2019;66(2):365–72.

[48] Criswell E. Cram's introduction to surface electromyography. Burlington: Jones & Bartlett Publishers; 2010.

[49] Reifinger S, Wallhoff F, Ablassmeier M, Poitschke T, Rigoll G. Static and dynamic hand-gesture recognition for augmented reality applications. In: International Conference on Human-Computer Interaction. Springer; 2007. p. 728–37.

[50] Leonardis D, Barsotti M, Loconsole C, Solazzi M, Troncossi M, Mazzotti C, et al. An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation. IEEE Trans Haptics 2015;8(2):140–51.

[51] Dunai L, Novak M, García Espert C. Human hand anatomy-based prosthetic hand. Sensors 2021;21(1):137.

[52] Pizzolato S, Tagliapietra L, Cognolato M, Reggiani M, Müller H, Atzori M. Comparison of six electromyography acquisition setups on hand movement classification tasks. PLoS One 2017;12(10):1–17.

[53] Ortiz-Catalan M, Brånemark R, Håkansson B. BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. Source Code Biol Med 2013;8(1):11.

[54] Lobov S, Krilova N, Kastalskiy I, Kazantsev V, Makarov VA. Latent factors limiting the performance of sEMG-interfaces. Sensors 2018;18(4):1122.

[55] Yang K, Xu M, Yang X, Yang R, Chen Y. A novel EMG-based hand gesture recognition framework based on multivariate variational mode decomposition. Sensors 2021;21(21):7002.

[56] Cheng Y, Li G, Yu M, Jiang D, Yun J, Liu Y, et al. Gesture recognition based on surface electromyography-feature image. Concurr Comput 2021;33(6) e6051.

[57] Wei W, Dai Q, Wong Y, Hu Y, Kankanhalli M, Geng W. Surface-electromyography-based gesture recognition by multi-view deep learning. IEEE Trans Biomed Eng 2019;66(10):2964–73.

[58] Josephs D, Drake C, Heroy A, Santerre J. sEMG gesture recognition with a simple model of attention. In: Alsentzer E, McDermott MBA, Falck F, Sarkar SK, Roy S, Hyland SL, editors. Proceedings of the Machine Learning for Health NeurIPS Workshop. vol. 136 of Proceedings of Machine Learning Research. PMLR. p. 126–38.

[59] Potekhin VV, Unal O. Development of machine learning models to determine hand gestures using EMG signals. Ann DAAAM Proceedings 2020;7(1).

[60] Nazemi A, Maleki A. Artificial neural network classifier in comparison with LDA and LS-SVM classifiers to recognize 52 hand postures and movements. In: 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE; 2014. p. 18–22.

[61] Rubio AM, Grisales JAA, Tabares-Soto R, Orozco-Arias S, Varón CFJ, Buriticá JIP. Identification of hand movements from electromyographic signals using machine learning; 2020.

[62] Foody GM. Thematic map comparison. Photogramm Eng Remote Sensing 2004;70(5):627–33.

[63] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning; 2018.