EMG Signal Classification Using Deep Learning and Time Domain Descriptors-Based Feature Extraction for Hand Grip Movement Recognition

Ali Ari

Department of Computer Engineering, Faculty of Engineering, Inonu University, Malatya 44280, Turkey

Corresponding Author Email: ali.ari@inonu.edu.tr

https://doi.org/10.18280/ts.400311

1. INTRODUCTION

People worldwide lose some of their limbs due to burns, diseases such as diabetes, and work and traffic accidents. A study conducted in France in 2013 stated that approximately 60% of these cases caused damage to the arm region [1]. This seriously restricts people's daily activities and significantly reduces their quality of life. These problems are increasingly interesting in upper extremity rehabilitation devices with recurrent and capable of performing daily activities of individuals and providing auxiliary power [2].

Electromyography (EMG) is an electrophysiology procedure used to evaluate and diagnose the functions of muscle and nerve cells. EMG, which provides detailed data on the state and functioning of skeletal muscles, is the signal generated during the contraction of skeletal muscles. EMG provides the acquisition of bio-electrical signals of Motor Unit Action Potentials (MUAPs) [2]. EMG is divided into two according to the reception of bio-electrical signals. Surface EMG (sEMG) is generally preferred in subjects with limb loss and is taken with electrodes placed above the skin, while invasive EMG is used for movement restrictions in human bodies and is taken with needle electrodes placed in the muscles [3]. The location of the electrodes used to record muscle activities may have more data than a motor unit, as the muscle fibers of the motor units in different muscle parts are interconnected. In this respect, it is very important for researchers to record the correct information of the EMG signal to recognize finger movements individually and in combination to control robotic and artificial organs. Because the human hand enables it to perform a wide variety of tasks, including precise control and rough actions. We use our hands to hold objects, pull, push, perform delicate tasks such as typing, and make gestures during communication. These actions are provided by precisely controlled muscles and sensors sensitive to muscle lengths and epidermal sensors such as touch, temperature, and pressure [4]. Therefore, since sEMG signals reveal human muscle activities and distinguish wrist, hand, and finger movements, they have been effectively used recently to control state-of-the-art prostheses and design skeletal prosthetic hands [4-8].

In the literature, statistical algorithms, wavelet transform, empirical mode decomposition (EMD), artificial intelligence-based feature extraction, and classifier methods are generally preferred to classify basic hand grasp movements. Lee et al. [9] proposed a machine learning-based model to classify hand and finger movements. EMG signals from the hand and finger motions of ten volunteers were employed in their investigation. Using an artificial neural network (ANN) and support vector machine (SVM), they were able to classify the EMG signals measured from three channels with an accuracy of 94% and 87.40%, respectively. Bhattachargee et al. [10] used Fourier and statistical features such as Root Mean Square (RMS), std, Variance, and Median to classify finger
movements with EMG signals. The obtained features reached 98.50% accuracy by applying them to the supervised learning-based Gradient Boosting (GB) classification algorithm.

Fang et al. [11] proposed a 2D EMG map to examine the relationship between hand movements and multichannel sEMG signals. For this purpose, they developed a special 16-channel electrode to collect sEMG signals on the forearm and used differential root mean square (DRMS) values as a feature, which references the relationship between adjacent EMG channels from sEMG signals from these multichannel electrodes. In distinguishing the four-hand movements, DRMS outperformed the traditional root mean square (RMS) approach with K-means and fuzzy C-means by 29.0% and 36.8%, respectively. Nishad et al. [12] developed the wavelet transform-based filter bank method to decompose the cross-covariance of sEMG signals to classify basic hand movements. They sequenced the features extracted with Kraskov entropy from the sEMG signals obtained from five subjects. They achieved 98.55% classification accuracy with these features, which were then applied to the k-nearest neighbor (k-NN) classifier.

Sapsanis et al. [13] used an EMD approach to classify baseline hand movement from EMG signals. This approach extracts different features in EMG signals’ time and frequency domain. Then, the features were reduced by principal component analysis and applied to a linear classifier for classification. Subaşı et al. [14] proposed a model for prosthetic hand control based on wavelet packet transform (WPT) from sEMG signals and multi-scale principal component analysis (MSPCA). They used the statistical values of the subbands of the sEMG signals as features. To provide better classification performance, the first reduced the feature vector size with MSPCA and then applied decision trees as input for the classification. They achieved 98.33% classification accuracy in experimental studies.

Khusaba et al. [15] used two EMG electrodes placed on the forearm to improve control of the prosthetic fingers. He used EMG signals from these two electrodes and individual and combined finger movements that included ten different hand movements. Miften et al. [16] proposed a method for recognizing hand movements using EMG signals, including logarithmic spectrogram-based graph signal (LSGS), AdaBoost k-means (AB-k-means), and feature selection. In the study, effective feature selection was made by extracting features from EMG signals with the LSGS model. Then, the selected EMG features were classified into the AB-k-model. Aslan et al. [17] performed feature extraction using Distribution Entropy (DisEn) and Normal Cumulative Distribution Function (NCDF) methods to detect hand movements from EMG signals. Then these features used classifiers such as SVM and extreme learning machine (ELM) to classify hand gestures. In experimental studies, it distinguished hand movements with 98% accuracy performance. Rabie et al. [18] obtained frequency and time characteristics from EMG signals by short-time Fourier transform (STFT). Their studies have reduced the size of the extracted features by two-dimensional reduction methods. Then, they analyzed the effect of size reduction techniques on classification performance. The experimental studies found that the diffusion map outperformed the principal component analysis method. Feature extraction was performed using the Correlation-Similarity Analysis (F-DFA) to classify the EMG signals and RMS methods with non-overlapping windows to measure statistical similarity. The extracted features were classified by machine learning methods such as decision trees and k-NN [19, 20]. On the other hand, Tuncer et al. [21] developed a new model for prosthetic hand control based on the discrete wavelet transform and the triplet model. They used a two-level feature selection method to select the most effective features obtained from this model and a k-NN classifier to classify the selected features.

As seen in the literature, many methods have been developed for detecting hand grasp movements from EMG signals and creating prosthetic hands. Except for a few recent studies, techniques such as wavelet transform, EMD, and principal component analysis for feature reduction have been preferred. Although significant achievements are achieved in current studies, there are disadvantages, such as the high number of features. Therefore, these methods are not preferred in real-time applications.

This study is that when the works in the literature are examined, as far as we know, the features obtained from the use of LSTM, CNN, and TDD approaches together as an attribute in the classification of hand grip movements have not been found. Because new and efficient feature extraction is adopted, which can process stationary and nonlinear biomedical signals. In this respect, the proposed hybrid model will significantly contribute to designing an effective and low-cost prosthetic hand that can support individuals with partial or complete hand loss using a combined LSTM, CNN, and TDD approach. The main contributions of this article are summarized as follows:

- This study proposes a new framework with TDD and deep learning-based framework to analyze EMG signals and also remove EMG properties that can recognize hand comparatives.
- The study focuses on the design of the Minimal-Radiancy-Maximal-Relevance selection to select the most effective features.
- The study has preferred SVM to classify the most effective feature of different handicrafts.
- Finally, the study also evaluates the performance of the proposed model by comparing it with state-of-the-art methodologies.

The rest of the paper is organized as follows. The Section 2, the dataset and the proposed methodology are described. Section 3 discusses the experimental setup, the experimental results, and our findings. Section 4 presents the results and future work.

2. METHODOLOGY

The main purpose of this study is to present a model that can automatically distinguish hand grip movements from EMG signals for people with partial or complete hand loss. The model, the general framework given in Figure 1, consists of three stages. In the first stage, TDD, CNN, and LSTM models are extracted features from EMG signals. Then, with the MRMR feature selection approach, the most effective features were selected among these features. Finally, SVM is used to classify hand grip movements from these most effective features.
**Figure 1.** General framework of proposed model.

**Figure 2.** Six hand grip movements and the EMG signals of each electrode for these movements.
2.1 EMG dataset

We used the EMG hand grasps signals dataset from the UC Irvine Machine Learning Repository, freely available for study [22]. It used two EMG channels to record EMG data. EMG signals in the dataset were collected from five healthy subjects two males and three females, aged 20 to 22. Flexor Carpi Ulnaris and Extensor Carpi Radialis, Longus, and Brevis surface electrodes attached to each subject’s hands with elastic bands on the two forearm surfaces were used to record information about muscle activities.

Figure 2 shows six hand grasp movements and the EMG signals of each electrode of this hand grasp. The national instrument LabVIEW was used to sample the EMG signals at 500 Hz. Then, a Butterworth bandpass filter with low and high cutoff values of 15 Hz and 500 Hz was used to de-noise from the EMG signals. Also, a Notch filter was used at 50 Hz to eliminate line noises. Each subject was asked to perform each hand grasp movement for 6 seconds with the free force: Cylindrical (CY), Hook (HO), Lateral (LA), Palmar (PA), Tip (TI) and Spherical (SP). Each subject repeated these six hand grasp movements 30 times with the free speed, totaling 180 times. There are EMG signals of 900 hand grasps in total in the dataset. Besides, one healthy subject (male, 22-year-old) conducted the six grasps 100 times each for 3 consecutive days. The measured time is 5 sec.

2.2 CNN

In recent years, Convolutional Neural Networks (CNN) is a typical deep learning method widely applied in fields such as image processing, pattern recognition, speech recognition, and medical engineering [23-27]. The biggest advantage of CNN structures is that there is no need for manual feature selection. CNN structures can be designed as one-dimensional (1D), two-dimensional (2D), or three-dimensional (3D) [28]. 1D-CNN structures are used in serial data processing, 2D-CNN in image and text recognition, and 3D-CNN in video and medical image recognition [24]. CNN generally consists of input, convolution, pooling, fully connected, and output layers.

The input layer can process multidimensional data in standardized forms. Standardized input data increases the efficiency of CNN by helping to improve learning performance.

The convolution layer is an important layer for CNN and uses the convolution operation denoted by “∗” instead of general matrix multiplication. The learnable parameters of this layer consist of a set of learnable filters known as kernels. This layer creates a data matrix that scrolls through all the input data and multiplies it with the core data to extract some features of that data. In short, the convolution layer performs feature extraction. The neurons in this layer are called filters; they take the input data and turn it into output feature maps [29, 30]. Mathematically, the filter shifts from left to right until it reaches the maximum width of the image. Then the filter starts from the leftmost value of the next row. The process continues until all input data is complete. As a result, the output of each convolution layer is the convolution result of multiple input properties, and its mathematical relation is defined as follows [31]:

\[ y_{l}^{i+1}(j) = K_{l}^{i} * x_{l}(j) + b_{l}^{i} \]  

where, \( K_{l}^{i} \) describe the weight of the \( i \)th filter kernel at layer \( l \), \( x_{l}(j) \) describe the \( j \)th local region at layer \( l \), \( b_{l}^{i} \) describe the bias of the \( i \)th filter kernel layer \( l \), and \( y_{l}^{i+1}(j) \) describe the input of the \( j \)th neuron in frame \( l \) of layer \( l+1 \).

After the convolution operations, it is applied to the activation function to enable the network to obtain a nonlinear expression of the input signal to make the learned features more divisible. In this step, Rectified Linear Unit (RELU) is generally used in the activation process to accelerate the CNN. The property of this function is that the derivative of the function is always one as long as the input value is greater than 0. Thus, the gradient explosion and gradient vanishing problems are also solved. In general, RELU can be defined as follows [32]:

\[ a_{l}^{i+1}(j) = f \left( y_{l}^{i+1}(j) \right) = \max\{0, y_{l}^{i+1}(j)\} \]  

The pooling layer is used to reduce the parameters of the neural network. With this layer, data is down sampled, reducing the width and height of a large matrix, reducing computational cost, and preventing overfitting. While average and maximum pooling can be used in this process, maximum pooling is more commonly used in practice. The maximum pooling formula is as follows [32]:

\[ p_{l}^{i+1}(j) = \max_{(j-1)W+1\leq s\leq jW} \{q_{l}(t)\} \]  

where, \( t \in \{(1-1)W+1, jW\} \) represent the width pooling and \( p_{l}^{i+1}(j) \), represent the value of the neuron at layer \( l+1 \), and \( q_{l}(t) \) represents the weight of the neuron at layer \( l \).

The fully connected layer connects the neurons of the previous and subsequent layers. It usually transforms the output of the last pooling layer into a one-dimensional flattened matrix that also functions as the input of the fully connected layer. Also, this layer can integrate differentiated local information into the convolution or pooling layer. Fully connected layer general formula [32]:

\[ z^{l+1}(j) = f \left( \sum_{i=1}^{n} \sum_{t=1}^{m} W_{li}^{i} a_{l}^{i}(t) + b_{l}^{i} \right) \]  

where, \( f \) represents the RELU function, and \( z^{l+1}(j) \) represents the activation value of the layer \( l+1 \). The output layer is known as the layer where classification labels are created. In this layer, a softmax classifier is generally based on logistic regression and has linear classifier features with a multi-class classification format.

Figure 3. Details on the designed CNN architecture

The 1D-CNN architecture designed for this study is shown in Figure 3. The proposed 1D-CNN has five Convolution (Conv) layers with filter sizes of 1×1, 2×2, 4×4, 2×2 and 1×1 respectively. Also, the two release layers have a release rate of 0.051 and 0.1, respectively. fully connected layers contain 100 and 6 neurons, respectively.
LSTM, whose general structure is given in Figure 4, is a special Recurrent Neural Network (RNA) model developed by Hochreiter and Schmidhuber [32].

Figure 4. The architecture of the LSTM cell

LSTM structurally has a memory cell and gate mechanism to solve the gradient burst and gradient disappearance problems that mainly occur in RNA models [31-34]. Thanks to the memory cell's structure, it provides long-term recall of information [35]. The update gate determines this memory cell (gt), input gate (it), forget gate (ft), and output gate (ot) mechanisms [34], and this process is performed by the six equations given below [31].

\[ g_t = \tanh \left( W_g x_t + W_g h_{t-1} + b_g \right) \]  
\[ i_t = \sigma \left( W_i x_t + W_i h_{t-1} + b_i \right) \]  
\[ f_t = \sigma \left( W_f x_t + W_f h_{t-1} + b_f \right) \]

where, \( \tanh(\cdot) \) is the hyperbolic tangent function, \( W \) refers to the weights matrix, \( \sigma(\cdot) \) refers to the sigmoid function, \( c_t \) refers to the memory cell, \( h_t \) refers to the output of the LSTM, and \( b \) refers to the bias vector. The 1D-LSTM architecture designed for this study is shown in Figure 5. The proposed 1D-LSTM contains two fully connected (fc) layers of 100 and 6 neurons, respectively, along with two dropout layers with a 0.5 dropout rate. Softmax and Class layers are also present.

Figure 5. Details on the designed LSTM architecture

2.4 Time-domain descriptors

Time-Domain Descriptors (TDD), whose block diagram is given in Figure 6, are one method used to classify hand movements from EMG signals [36]. This algorithm can be represented as a function of \( X[k] \) Discrete Fourier transform (DFT) frequency, using the DFT first from the EMG signal defined as \( X(j) \) with \( j = 1, 2, \ldots, N \) length, and \( f_s \) sampling frequency. The basis of TDD method is based on Parseval's theorem, whose principal derivatives state that the sum of the squares of the function is equal to the sum of the squares of its transformation.

\[ a_t = \sigma(W_o x_t + W_o h_{t-1} + b_o) \]  
\[ c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \]  
\[ h_t = a_t \cdot \tanh(c_t) \]

where, \( \tanh(\cdot) \) is the hyperbolic tangent function, \( W \) refers to the weights matrix, \( \sigma(\cdot) \) refers to the sigmoid function, \( c_t \) refers to the memory cell, \( h_t \) refers to the output of the LSTM, and \( b \) refers to the bias vector.

Figure 6. Block diagram of the time-domain descriptors [36]
\[
\sum_{j=0}^{N-1} |x[j]|^2 = \frac{1}{N} \sum_{j=0}^{N-1} |X[k]| X^*[k] = \sum_{j=0}^{N-1} P[k]
\]  

(11)

where, \(P[k]\) is the power spectrum with phase omitted. \(k\) is the frequency index, and the result is the product of multiplying \(X[k]\) by its conjugate \(X[k]\) divided by \(N\).

As a result of the fact that the whole frequency definition of the Fourier transform is symmetric to the zero frequency, direct access to the power spectral density in the time domain is challenging. Therefore, we must deal with the entire spectrum, including positive and negative frequencies. However, this situation can be solved by using Parseval's theorem and the time derivative property of the Fourier transform for nonzero values. As seen in Figure 6, TDD consists of six features (\(f_1, f_2, \ldots, f_6\)), the details of which are given below [36, 37].

**Root squared zero order moment:** This property shows the total power in the frequency domain and can be described by the following equation:

\[
\bar{m}_0 = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} x[j]^2}
\]  

(12)

**Root squared second and fourth order moments:** Hjorth argues that the second moment can be seen as a power, but as a power with a varying spectrum, denoted by the notation \(k^2 P[k]\), which corresponds to the frequency function \(kX[k]\). The first derivative of the time function is the time equivalent of a frequency function multiplied by \(k\) [38].

\[
\bar{m}_2 = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} k^2 P[k]} = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (kX[k])^2}
\]  

(13)

A repetition of this procedure gives the moment.

\[
\bar{m}_4 = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} k^4 P[k]} = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (\Delta x[j])^2}
\]  

(14)

Similarly, like the 4th order moment in Eq. (14), the 8th order moment can be defined as follows:

\[
\bar{m}_8 = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} k^8 P[k]} = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (\Delta^4 x[j])^2}
\]  

(15)

Taking the 4th and 8th derivatives allows us to calculate the spectral moments in the time domain while reducing the signal's overall energy. Therefore, we apply a power transformation as follows to normalize the \(m_0, m_4,\) and \(m_8\) range and reduce the effect of noise on all moment-based features.

\[
m_0 = \frac{\bar{m}_0}{\lambda}, m_4 = \frac{\bar{m}_4}{\lambda}, m_8 = \frac{\bar{m}_8}{\lambda}
\]  

(16)

With \(\lambda\) is empirically set to 0.1. Following that, the first three features derived from these variables are defined as follows:

\[
f_1 = \log(m_0), f_2 = \log(m_0 - m_4), f_3 = \log(m_0 - m_8)
\]  

(17)

**Sparness (SP):** The sparness feature measures how much energy is contained in a vector’s few constituent parts. It is provided as.

\[
f_4 = \log \left( \frac{m_4}{\sqrt{m_0 m_8}} \right)
\]  

(18)

**Irregularity Factor (IF):** a measurement that reflects the proportion of upward zero crossings to peak crossings. In terms of their spectral moments can the number of upward zero crossings (ZC) and the number of peaks (NP) in a random signal be represented. The equivalent characteristic can be expressed as:

\[
f_5 = \log \left( \frac{ZC}{NP} \right) = \log \left( \frac{m_4}{m_0} \right) = \log \left( \frac{m_4}{m_0 m_8} \right)
\]  

(19)

**Waveform Length Ratio (WL):** given that the waveform length feature is defined as the addition of the absolute values of the signal derivatives, we describe our WL feature as the ratio of the first derivative’s waveform length to the second derivative's waveform length.

\[
f_6 = \log \left( \frac{\sum_{j=0}^{N-1} |\Delta^2 x[j]|}{\sum_{j=0}^{N-1} |\Delta x[j]|} \right)
\]  

(20)

With the approach of TDD detailed above, 19 features are extracted from EMG signals, each channel represented by 1x3000 data points and 19x12 features for two channels. A 19x12 feature matrix is obtained from the EMG signals of each person’s six basic hand movements.

### 2.5 Minimal-redundancy-maximal-relevance

MRMR is used to improve the performance of the relationship between the features obtained in feature extraction and the target classes [39]. The redundancy of some features in the feature extraction process can negatively affect the learning accuracy [40]. The MRMR feature selection algorithm eliminates this negative situation and reduces the computational cost of the proposed models by reducing the feature redundancy according to the level of relevance. In particular, the maximization of the relevance of a feature subset \(S\) to the class label \(c\) is referred to as max relevance, denoted as max \(D(S, c)\). According to the study [5], a feature subset’s importance is indicated by the following:

\[
D(S, c) = \frac{1}{|S|} \sum_{f_i \in S} \Phi(f_i, c)
\]  

(21)

where, the relevance of a feature \(f_i\) to \(c\) is indicated by \(\Phi(f_i, c)\). Based on pair-wise feature reliance, the term “feature redundancy” is used. If two important characteristics are strongly correlated, losing one of them would not significantly
affect the other’s ability to discriminate based on class. Min-redundancy. A feature subset of mutually incompatible features, is chosen using \(\text{min} \ R(S)\). According to the following definition, feature subset \(R(S)\) redundancy:

\[
R(S) = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} \Phi(f_i, f_j)
\]  

(22)

Accordingly, given a subset of features, the \(m\)-1 selected feature of \(Sm\)-1 should select the \(m\)th feature that best meets the criteria in Eq. (23):

\[
\max_{f_j \in Sm-1} \left[ \Phi(f_i, c) - \frac{1}{m-1} \sum_{f_i \in S} \Phi(f_i, f_j) \right]
\]  

(23)

2.6 Support vector machine

In this section, SVM, one of the popular classification methods, is used to extract features from medical signals and classify features consisting of a combination of features obtained from 1D-CNN and LSTM methods [41].

SVM is one of the widely used machine learning methods based on Vapnik’s statistical learning theory, which classifies data by separating them with linear and nonlinear boundaries [42]. The main function of this method is based on finding the maximum margin hyperplane that will provide optimal separation of the data by transforming the input data into a higher dimensional space [35, 43]. The hyperplane where \(w\) represents the normal vector and \(b\) means bias can be defined by Eq. (24):

\[
f(x) = w^T \cdot x + b
\]  

(24)

The relation to optimizing the hyperplane can be defined by Eq. (25):

\[
\begin{bmatrix}
\min \left\{ \frac{1}{2} \|w\|^2 \right\} \\
y_i - w^T x_i - b \leq \alpha \\
w^T x_i + b - y_i \leq \alpha
\end{bmatrix}
\]  

(25)

where, \(x\) defines the deviation between \(f(x)\) and the target \(y\). Eq. (25) can be expressed as follows, considering the regularization parameter \(C\), and the slack variable \(\xi_i\):

\[
\begin{bmatrix}
\min \left\{ \frac{1}{2} \|w\|^2 + C \times \sum_{i=1}^{n} (\xi_i + \xi_i^*) \right\} \\
y_i - w^T x_i - b \leq \alpha + \xi_i \\
w^T x_i + b - y_i \leq \alpha + \xi_i^*
\end{bmatrix}
\]  

(26)

Since Eq. (26) is in a non-convex form, it can be optimized in Eq. (27) by adding the Lagrange multiplier \(\delta_i\):

\[
f(x) = \sum_{i=1}^{n} (\delta_i - \delta_i^*)K(x_i, x) + b
\]  

(27)

where, \(K(x_i, x)\) is the kernel function. Although many kernel functions exist, the Gaussian radial basis function (RBF) is generally used.

3. EVALUATION OF PERFORMANCE

Based on deep learning and TDD feature classification, the proposed hybrid model is evaluated for Accuracy (Acc), recall, Specificity (Spc), Precision (Pre), F1-score, and receiver operating characteristic (ROC) analysis. The analysis makes use of a confusion matrix defined by the following metrics:

True-Positive (TP) is the number of outcomes in which the proposed model correctly classifies the positive class.

False-Positive (FP) is the number of incorrect outcomes in which the proposed model incorrectly classifies the positive class.

True-Negative (TN) is the number of outcomes in which the proposed model correctly classifies the negative class.

False-Negative (FN) is the number of outcomes in which the proposed model incorrectly classified the negative class.

The relations for these metrics are given in Eqns. (28)-(32).

\[
\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \times 10
\]  

(28)

\[
\text{Recall} = \frac{TP}{TP + FN} \times 100
\]  

(29)

\[
\text{Spc} = \frac{TN}{TN + FP} \times 100
\]  

(30)

\[
\text{Pre} = \frac{TP}{TP + FP} \times 100
\]  

(31)

\[
\text{F1 Skor} = \frac{2 \times \text{Recall} \times \text{Pre}}{\text{Recall} + \text{Pre}} \times 100
\]  

(32)

4. EXPERIMENTAL RESULTS AND DISCUSSION

Experimental works of the proposed model were carried out in Matlab with 32 GB RAM, NVIDIA P2200 Quadro GPU card, and Intel Xeon Silver 3.3 GHz processor installed on the 10 Pro Operating System.

In experimental studies, EMG signals of 1×3000 size for each channel of 5 different volunteers were used, details given in Section 2.1. Two different experimental studies were conducted to detect basic hand grip movements, utilizing the 1D-CNN, LSTM, and TDD+SVM models separately and using the three models together (hybrid).

Firstly, for the training of the proposed 1D- CNN and LSTM models, the data set is randomly divided into two parts, 86.67% training and 13.33% testing. The performances of 1D-CNN and LSTM models, whose details are given in sections 2.2 and 2.3, respectively, depend on many hyperparameters such as learning rate, batch size, number of training periods, dropout rates, and number of hidden neurons. In this study, after empirical evaluation that considers various combinations of hyperparameters to determine the appropriate value of the parameters, the training process was completed with the CNN model through the SGDM optimizer with a fixed learning rate of 0.01, 32 mini-batch size arrays, and iterations at 60 epochs.

Likewise, the Adam optimizer trained the LSTM model with an initial learning rate of 0.006, 64 mini-batch size arrays, and iterations at 70 epochs. The training of both models has been completed with loss rates of 0.12 and 0.10, respectively. It is desired that deep learning models have minimum loss values during training, which shows that they provide a more
robust fit performance to the training data. After completing the training of both deep learning models according to the above parameters, the classification results of the test data are given in Table 1.

From the same data set with TDD, feature extraction from EMG signals and classification with SVM is as follows.

Nineteen features are extracted from 1×3000 data points in each channel of a subject’s handgrip movement and represented by 19×2 features for the two channels. A 95×12 feature matrix is obtained from the dataset’s EMG signals of 1 subject. These features are then applied to the SVM to classify hand gestures. Classification performances of each person’s basic hand movements in experimental studies of each model at this stage are shown in Table 1.

The maximum performance was attained in subject 2 with 100% of all evaluation criteria, as shown in Table 1 results using 1-D CNN. Subject 1, subject 3 in LSTM classification results, and subject 2, subject 4 in TDD+SVM classification results achieved the highest performance. In both methods, accuracy, recall, Spc, pre, and F1 scores were 95.833%, 95.833%, 99.167%, 96.667%, and 95.767% success rates, respectively.

In the second experiment, features to distinguish basic hand movements from EMG signals were extracted separately from 1D-CNN, LSTM and TDD models. One hundred features are extracted from the 1D-CNN FC-2 layer, 100 from the FC layer of LSTM, and 228 from the TDD. Then, the most effective features of each model are selected and combined with the MrMr feature selection approach. In this process step, the 13 most effective features for 1D-CNN, 15 for LSTM and 150 for TDD are extracted. When these features are combined, a total of 178 most effective features are obtained. The SVM classification performances of each subject’s combined most effective features are shown in Table 2.

According to the results in Table 2, accuracy and recall performance evaluation criteria of 95.83%, specificity of 99.16%, precision of 96.67% and F1 score of 95.76% were achieved in subjects 1 and 2. In addition, 100% performance was achieved in all performance evaluation criteria in peer 3, subject 4, and subject 5.

Figures 7-9 show the confusion matrices of the proposed hybrid model for the classification of basic hand grip movements of 5 subjects in the dataset. The purpose of confusion matrices is generally to provide an opportunity to see the classification sensitivity of each basic hand grip movement and its ratio to precision. In confusion matrices, rows represent actual class values, and columns represent the proposed model’s predicted class. As seen in the confusion matrix of subject 1 in Figure 7, the actual cylindrical (CY) hand grip was incorrectly identified as the proposed hybrid model hook (HO). It is classified with 75% accuracy in CY hand grip movement and 100% accuracy in all other hand grip movements. In total, a 95.83% accuracy rate was reached in the classification of all hand grip movements. The proposed model predicted all other hand grip movements correctly.

A similar situation is also seen in the confusion matrix of subject 2 in Figure 8. The proposed hybrid model incorrectly detected the hand grasp, which is a hook (HO), as cylindrical (CY). The proposed model predicted all other hand grip movements correctly. In other words, the proposed model is classified with 75% accuracy in HO hand grip movement and 100% accuracy in all other hand grip movements. In total, a 95.83% accuracy rate was reached in the classification of all hand grip movements.

As seen in Figure 9 confusion matrix of subject 3, subject 4, and subject 5, all hand grip movements are predicted correctly by the proposed hybrid model. In other words, the proposed model is classified as 100% accuracy in all (CY, HO, PA, SP, TI) hand grasping movements.

### Table 1. Classification performances of 1-D CNN, LSTM, and TDD+SVM models (%)

<table>
<thead>
<tr>
<th>Subject No</th>
<th>1-D CNN</th>
<th>LSTM</th>
<th>TDD+SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td>Subject 1</td>
<td>95.833</td>
<td>95.833</td>
<td>95.833</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Recall</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td>95.833</td>
<td>95.833</td>
<td>95.833</td>
</tr>
<tr>
<td></td>
<td>Spc</td>
<td>Spc</td>
<td>Spc</td>
</tr>
<tr>
<td></td>
<td>Pre</td>
<td>Pre</td>
<td>Pre</td>
</tr>
<tr>
<td></td>
<td>96.667</td>
<td>96.667</td>
<td>96.667</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td>95.767</td>
<td>95.767</td>
<td>95.767</td>
</tr>
<tr>
<td>Subject 2</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Recall</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Spc</td>
<td>Spc</td>
<td>Spc</td>
</tr>
<tr>
<td></td>
<td>98.333</td>
<td>98.333</td>
<td>98.333</td>
</tr>
<tr>
<td></td>
<td>Pre</td>
<td>Pre</td>
<td>Pre</td>
</tr>
<tr>
<td></td>
<td>92.500</td>
<td>96.667</td>
<td>96.667</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td>91.601</td>
<td>95.767</td>
<td>95.767</td>
</tr>
<tr>
<td>Subject 3</td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Recall</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Spc</td>
<td>Spc</td>
<td>Spc</td>
</tr>
<tr>
<td></td>
<td>98.333</td>
<td>98.333</td>
<td>98.333</td>
</tr>
<tr>
<td></td>
<td>Pre</td>
<td>Pre</td>
<td>Pre</td>
</tr>
<tr>
<td></td>
<td>93.333</td>
<td>94.444</td>
<td>94.444</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td>91.534</td>
<td>91.111</td>
<td>91.111</td>
</tr>
<tr>
<td>Subject 4</td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Recall</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Spc</td>
<td>Spc</td>
<td>Spc</td>
</tr>
<tr>
<td></td>
<td>98.333</td>
<td>97.500</td>
<td>97.500</td>
</tr>
<tr>
<td></td>
<td>Pre</td>
<td>Pre</td>
<td>Pre</td>
</tr>
<tr>
<td></td>
<td>93.333</td>
<td>92.857</td>
<td>92.857</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td>91.534</td>
<td>88.312</td>
<td>88.312</td>
</tr>
<tr>
<td>Subject 5</td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Recall</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td>91.667</td>
<td>91.667</td>
<td>91.667</td>
</tr>
<tr>
<td></td>
<td>Spc</td>
<td>Spc</td>
<td>Spc</td>
</tr>
<tr>
<td></td>
<td>98.333</td>
<td>97.500</td>
<td>97.500</td>
</tr>
<tr>
<td></td>
<td>Pre</td>
<td>Pre</td>
<td>Pre</td>
</tr>
<tr>
<td></td>
<td>93.333</td>
<td>92.857</td>
<td>92.857</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td>91.534</td>
<td>88.312</td>
<td>88.312</td>
</tr>
</tbody>
</table>
Table 2. Proposed hybrid model and classification results (%)

<table>
<thead>
<tr>
<th></th>
<th>Acc</th>
<th>Recall</th>
<th>Spc</th>
<th>Pre</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject 1</td>
<td>95.83</td>
<td>95.83</td>
<td>99.16</td>
<td>96.67</td>
<td>95.76</td>
</tr>
<tr>
<td>Subject 2</td>
<td>95.83</td>
<td>95.83</td>
<td>99.16</td>
<td>96.67</td>
<td>95.76</td>
</tr>
<tr>
<td>Subject 3</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Subject 4</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Subject 5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Table 3. Comparison of the performance of the proposed model and the state-of-the-art models

<table>
<thead>
<tr>
<th>Authors</th>
<th>Feature extraction</th>
<th>Classification models</th>
<th>Classification accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sapsanis et al. [44]</td>
<td>EMD</td>
<td>Linear classifier</td>
<td>Subject #1: 87.25 Subject #2: 88.05 Subject #3: 85.53 Subject #4: 90.42 Subject #5: 94.80</td>
</tr>
<tr>
<td>Ruangpaisan et al. [45]</td>
<td>SVD</td>
<td>SMO</td>
<td>Subject #1: 96.67 Subject #2: 98.89 Subject #3: 96.67 Subject #4: 98.89 Subject #5: 100.00</td>
</tr>
<tr>
<td>Iqbal et al. [46]</td>
<td>SDV-PAC</td>
<td>k-NN</td>
<td>Subject #1: 82.78 Subject #2: 87.67 Subject #3: 83.11 Subject #4: 90.00 Subject #5: 90.00</td>
</tr>
<tr>
<td>Akben [47]</td>
<td>Histogram</td>
<td>k-means</td>
<td>Subject #1: 93.04 Subject #2: 86.66 Subject #3: 97.00 Subject #4: 99.23 Subject #5: 97.66</td>
</tr>
<tr>
<td>Subasi et al. [14]</td>
<td>WPD</td>
<td>Rotation forest</td>
<td>Subject #1: 95.56 Subject #2: 88.88 Subject #3: 92.22 Subject #4: 92.22 Subject #5: 98.33</td>
</tr>
<tr>
<td>Nishad et al. [12]</td>
<td>TQ WT</td>
<td>k-NN</td>
<td>Subject #1: 98.33 Subject #2: 97.78 Subject #3: 99.44 Subject #4: 98.89 Subject #5: 98.33</td>
</tr>
<tr>
<td>Yavuz and Eyupoglu [48]</td>
<td>Cepstral coefficients</td>
<td>GRNN</td>
<td>Subject #1: 98.33 Subject #2: 99.48 Subject #3: 99.44 Subject #4: 99.78 Subject #5: 98.69</td>
</tr>
<tr>
<td>Miften et al. [16]</td>
<td>LSGS</td>
<td>AB-k-means</td>
<td>Subject #1: 98.00 Subject #2: 99.00 Subject #3: 99.00 Subject #4: 99.00 Subject #5: 99.00</td>
</tr>
<tr>
<td>The Proposed model</td>
<td>CNN+LSTM+TDD</td>
<td>SVM</td>
<td>Subject #1: 95.83 Subject #2: 95.83 Subject #3: 100.00 Subject #4: 100.00 Subject #5: 100.00</td>
</tr>
</tbody>
</table>

The proposed model was compared in terms of classification accuracy with some state-of-the-art methods using the same data set, and the results are given in Table 3. Sapsanis et al. [44] created the dataset used in all these studies in 2013 and proposed a method based on feature extraction with EMD and a linear classifier. The proposed method recorded the lowest accuracy rate for subject 3, with 85.43%, and the highest accuracy rate for subject number 5, with 94.80%.

Figure 7. Confusion matrix for the proposed model using Subject 1

Figure 8. Confusion matrix for the proposed model using Subject 2

Figure 9. Confusion matrix for the proposed model using Subjects 3, 4, and 5

Receiver Operating Characteristic (ROC) curves are shown in Figure 10 and Figure 11, respectively, to more effectively show the performance of each hand grip movement of Subjects 1 and 2, where 100% performance was not achieved in the proposed hybrid model. The Area Under the Curve (AUC) value in the curves of the cylindrical (CY) grip movement in Subject 1 is 0.97, and the AUC in the hook (HO) hand grip in Subject 2 is 0.98.

Figure 10. ROC curve of the proposed model for subject 1
model has three stages: feature extraction, selection, and classification. In the first stage, feature extraction is done from EMG signals with reference to 1-D CNN and LSTM deep learning models and time domain descriptive approach. In the second stage, in the MRMR feature selection approach, the most effective features of each model are selected and concatenated. In the final stage, the most effective features combined are used to feed the SVM to classify basic hand gestures.

The proposed model was evaluated using an EMG dataset consisting of five subjects public and including their basic movements such as Cylindrical (CY), Hook (HO), Lateral (LA), Palmar (PA), Tip (TI) and Spherical (SP). In the use of only 1-D CNN in the proposed hybrid model, the hand-grasp movements reached 95.83% in subject 1, 100% in subject 2 and 91.66% accuracy rate in other subjects (3, 4 and 5). In the use of the LSTM model alone, (1, 2 and 3) reached 95.83% in subjects, 91.66% in subject 4 and 87.50% accuracy rate in subject 5. Similarly, in the use of TDD+SVM alone, the subject reached 91.66% in subject 1, 95.83% in subjects 2 and 4, 87.5% in subjects 3 and 83.33% accuracy rate in subject 5. However, in the proposed hybrid model achieved an accuracy of 95.83% in the first two of the five subjects and 100% in the other three subjects. It also achieved an average accuracy of 98.34% and a specificity of 99.66% for five subjects.

Experimental results show that the proposed method outperforms subjects 3, 4, and 5 with 100% accuracy in all state-of-the-art studies using the same dataset. In the future, the proposed model, with remarkable results, could be applied as a control unit to improve the quality of life of hand-amputated patients and a robotic and prosthetic exoskeleton to mobilize the hand.

REFERENCES


