Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

Classification Of Finger Movements Using Deep Learning

Yasin ALİ[1], Yücel KOÇYİĞİT[2*]

[1] Manisa Celal Bayar University, Graduate School, yasinmufadala99@gmail.com, Orcid No: 0009-0006-7897-7657
[2] Manisa Celal Bayar University, Electrical-Electronics Engineering Department, yucel.kocyigit@cbu.edu.tr, Orcid No: 0000-0003-1785-198X

ARTICLE INFO

Article history:

Received 12 September 2025
Received in revised form 13 December 2025
Accepted 8 January 2026
Available online 24 March 2026

Keywords:

Electromyography, finger movement classification, deep learning, convolutional neural networks, long short-term memory, independent component analysis

Doi: 10.24012/dumf.1782057

* Corresponding author

ABSTRACT

This study aims to classify finger movements using electromyography (EMG) signals and deep learning models. Leveraging the potential of EMG signals for prosthetic control, diagnosis, and biomedical applications, a dataset of 350 trials was created by recording five repetitions of seven different finger movements from ten healthy volunteers. During the data processing stage, signal separation was performed using the FastICA method to resolve conflicts in finger-channel mapping. Subsequently, features were extract-ed using Time-Domain Descriptors (TDD), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) modules. The Mutual Information (MI) method was applied for selecting the most discriminative features, and a Fully Connected Neural Network (FCNN) model was chosen for the classification task. The results demonstrated that the model could identify finger movements with high accuracy rates. These findings indicate that deep learning is an effective tool for movement classification in EMG-based systems and point to its applicability in fields such as prosthetic control and rehabilitation technologies.

Introduction

In recent years, rapid advancements in deep learning techniques have led to significant transformations in the field of biomedical signal processing, revealing strong potential especially for the analysis of complex time-series data such as electromyography (EMG). EMG signals are biological recordings that contain electrical information related to muscle activity, and they play a crucial role in applications such as prosthetic control, rehabilitation technologies, human machine interfaces, and movement analysis [1,2]. However, EMG signals are inherently difficult to classify reliably due to their high noise levels, inter-muscle cross-talk, inter-subject physiological variability, and complex time frequency characteristics.

Deep learning architectures provide substantial advantages in overcoming these challenges. Convolutional Neural Networks (CNN) are effective in learning spatial patterns within the signals, while Long Short-Term Memory (LSTM) networks demonstrate strong performance in modeling sequential dependencies in time-series data [3]. In this study, a dataset consisting of 350 EMG recordings obtained from seven finger movements performed five times by ten healthy participants was used. The processing pipeline included channel separation with FastICA, feature extraction using Time Domain Descriptors (TDD), CNN and LSTM architectures, feature selection through Mutual Information (MI), and final classification via a Fully Connected Neural Network (FCNN).

The purpose of this research is to develop an effective, robust, and deep learning-based model capable of accurately classifying finger movements from EMG signals. Accordingly, the central research problem is the lack of a method that can simultaneously extract both temporal and spatial information from the complex structure of EMG signals and integrate this information effectively for classification.

The hypothesis of the study proposes that combining TDD time-domain features with the spatial learning capability of CNN and the temporal modeling strength of LSTM will yield higher classification accuracy compared to using these methods individually. Furthermore, it is anticipated that this combined representation will become more distinguishable and discriminative when processed through an FCNN architecture.

This study introduces a novel hybrid approach that combines time-domain descriptors (TDD) with the spatial feature learning of CNN and temporal modeling of LSTM, integrated through an FCNN, to simultaneously capture both spatial and temporal characteristics of EMG signals for finger movement classification—an approach that, to the best of our knowledge, has not been reported in previous studies.

Deep learning models for EMG-based movement classification are increasingly adopted compared to traditional methods. Various studies have investigated the performance of different model types and feature extraction techniques.

Atzori et al. (2016) conducted a study using surface EMG (sEMG) data to control prosthetic limbs with a CNN-based classification model. The study involved 67 healthy and 11 amputee participants. The CNN model achieved an accuracy of 66% for healthy individuals and 38% for amputees. This work demonstrates the feasibility of deep learning approaches for prosthetic control.

Triwiyanto et al. (2024) employed a CNN model to classify hand movements using raw sEMG signals obtained from ten healthy participants. The convolutional neural network achieved approximately 97% accuracy. Notably, this study showed that high accuracy could be obtained without explicit feature extraction, highlighting CNN's capability to process raw sEMG data effectively.

Reza Azhiri et al. developed an RNN-based system for real-time EMG classification. Time-frequency features were extracted using discrete wavelet transform and then classified using RNNs. The system reported 96% accuracy within 600 milliseconds, demonstrating the suitability of RNN models for both speed and accuracy in real-time applications.

Lee et al. (2022) investigated the classification of ten different hand and finger movements from ten healthy participants using artificial neural networks (ANN), support vector machines (SVM), random forests (RF), and logistic regression (LR). The ANN achieved the highest accuracy at 94%, while SVM, RF, and LR achieved 87.6%, 83.1%, and 53.9%, respectively. This study emphasizes that ANN is less affected by inter-individual differences and can define more movements per channel, reducing computational load.

Sanjay Kumar et al. (2020) examined the nonlinear relationships between sEMG signals and finger movements and modeled them for practical applications. The collected data were classified using an MRD model, and performance was evaluated with metrics such as mean squared error, mean absolute error, and correlation coefficient. The results indicated that the model could predict finger movements with up to 91% correlation and effectively capture nonlinear synergies. This study contributes to modeling nonlinear relationships in EMG data and integrating them into prosthetic applications.

Ali Arı (2023) combined TDD, CNN, and LSTM models for feature extraction, selected meaningful features using the MRMR method, and performed classification with SVM. This approach achieved an accuracy of 98.34%, demonstrating the effectiveness of deep learning models in both feature extraction and classification tasks.

This literature review indicates that both deep learning and traditional machine learning models are used for EMG-based movement classification. Model selection depends on data size and feature extraction methods. CNN- and RNN-based models provide high accuracy and real-time application advantages, while models supported by feature extraction improve classification performance further.

Background

Surface Electromyography (sEMG) is a fundamental biomedical technique that measures the electrical activity of muscles using non-invasive methods. Electrodes are placed on the skin surface to record motor unit action potentials from the underlying muscles. These signals quantitatively reveal the timing, intensity, and coordination parameters of muscle activity [9]. The historical development of sEMG, widely used in fields like clinical diagnosis, rehabilitation, and prosthetic control, dates back to the late 19th century. With technological advances, the method gained precision, particularly in the mid-20th century, and has now become an indispensable component of digital signal processing and human-machine interfaces [9]. The basic principle relies on electrical signals emitted from motor neurons during muscle contraction reaching the skin surface. Electrode configuration, skin-muscle tissue properties, and signal-to-noise ratio play a critical role in recording these action potentials [10].

Datasets refer to organized collections of muscle activity signals in sEMG studies. Anatomically correct positioning of electrodes, device calibration, and subject selection are elements that directly affect data quality. Final datasets, created by processing raw data through procedures such as missing value imputation, normalization, and artifact filtering, are presented in tabular structures containing observations in rows and time series features in columns.

Classification of finger movements is of vital importance in the fields of prosthetic control, virtual reality, and neuroscience. This process typically consists of three basic stages: In the data collection phase, multimodal sensors such as EMG signals (muscle activity), accelerometers (position), and gyroscopes (angular velocity) are used. Then, in the feature extraction phase, discriminative features such as signal amplitude, frequency components, and time-frequency metrics are calculated. Finally, in the

classification algorithms phase, Support Vector Machines (SVM) are used for non-linear problems, Artificial Neural Networks (ANN) for learning complex movement patterns, and K-Nearest Neighbors (KNN) for classification based on similar movement clusters. This integrated approach provides highly accurate results in applications such as bionic hand control.

Materials and methods

Materials

In this study, we utilized the dataset developed by Sanjay Kumar Dwivedi et al. [10]. The dataset contains surface electromyography (sEMG) signals recorded from 10 healthy and intact participants (9 males, 1 female), aged between 26 and 31 years. Written informed consent was obtained from all participants prior to the experiments, and all procedures were conducted in accordance with the principles of the Declaration of Helsinki.

sEMG signals were recorded from eight extrinsic muscles of the hand and forearm, known to contribute to wrist and finger movements as indicated in Table 1, using bipolar active Ag-AgCl electrodes. The inter-electrode distance was set to 20 mm, and a reference/ground electrode was placed on the participant's olecranon.

Table 1. Determination of Surface Electrode Placement on Muscles

Channel	Target muscle	Hand/finger function
1	Abductor Pollicis Longus (APL)	Thumb abduction, extension
2	Flexor Carpi Radialis (FCR)	Wrist, hand flexion
3	Flexor Digitorum Superficialis (FDS)	PIP flexion of fingers 2-5
4	Flexor Digitorum Profundus (FDP)	DIP flexion of fingers 2-5
5	Extensor Digitorum (ED)	Extension of fingers 2-5
6	Extensor Indicis (EI)	Index finger
7	Extensor Carpi Ulnaris (ECU)	Wrist extension and abduction
8	Extensor Carpi Radialis (ECR)	Wrist and thumb

Signals were amplified using a BA1104 pre-amplifier and transmitted wirelessly via a TU-4 telemetry unit (Digitex Lab. Co. Ltd). The sEMG signals were measured within a frequency range of 1 Hz to 1 kHz, sampled at 2 kHz, and digitized with a 12-bit A/D converter.

Participants were asked to perform three different movement tasks:

Individual finger flexion-extension (IFFE)

Simultaneous flexion-extension of all fingers (AFFE)

Random flexion-extension of one or more fingers (RFFE)

In the first task, participants moved each finger sequentially in the flexion-extension plane. In the second task, all fingers were moved simultaneously, resembling hand opening and semi-closing. In the third task, participants moved any finger freely within the flexion-extension plane while keeping the arm and elbow in a neutral position. The first task included 5 sets (one per finger), while the remaining tasks consisted of one set each. Each set comprised 5 trials, with each trial lasting 20 seconds. All movements were limited to finger flexion and extension, with wrist and elbow maintained in a fixed position. Participants were instructed to perform movements at their perceived normal speed (~2 cycles per second) and to achieve maximum flexion and extension of each finger at least once in each trial.

The resulting dataset contains EMG signals from 10 participants across 7 tasks × 5 trials, with 8 muscle activation channels per trial. Raw sEMG signals were band-pass filtered between 10–500 Hz using a 4th-order Butterworth filter, rectified, normalized to the peak amplitude per channel, and low-pass filtered at 4 Hz. The signals were then downsampled to match the reference motion capture frequency, resulting in a final processed dataset of 4000 samples × 8 channels for each trial.

Kinematic data were not used in this study; this decision was made to focus exclusively on EMG-based classification and to evaluate the model performance solely based on muscle electrical activation, which was adopted as an academic and methodological choice.

Methods

Finger-Channel Mapping: To focus the analysis, channels corresponding to 5 basic finger movements were selected:

Table 2. Determination of Channels for Fingers

Finger/Channel	1	2	3	4	5	6	7	8
Thumb	✓							
Index						✓		
Middle			✓	✓	✓			
Ring			✓	✓	✓			
Little			✓	✓	✓			

To focus the analysis on individual finger movements, channels corresponding to the five basic finger movements were selected, specifically excluding channels 2 (Flexor Carpi Radialis), 7 (Extensor Carpi Ulnaris), and 8

(Extensor Carpi Radialis) as they are primarily associated with wrist and thumb movements rather than the target fingers. Among the seven recorded finger movements, only the first five tasks corresponding to the individual flexion extension of each finger were used for analysis. This choice was made because these tasks isolate the movement of each finger, providing clearer and more specific EMG signals for individual finger classification. In contrast, the remaining two tasks, which involve simultaneous or random finger movements, produce overlapping and less distinguishable EMG patterns, making them less suitable for accurately training and evaluating a finger-specific classification model.

Data Reduction and FastICA: Signals from the middle, ring, and little fingers (Channels 3-4-5) were combined into a single component and separated into independent components using the FastICA algorithm. This process was applied to address signal power imbalances across different fingers and to standardize the data to 4000×1 dimensions.

Mathematical Basis of the FastICA Algorithm: Mixed signals (X) are modeled in terms of a mixing matrix (A) and independent source signals (S) as follows:

$$X = AS \tag{1}$$

where:

X (4000×3): Raw EMG data from three channels

$A(3 \times 3)$: Unknown mixing matrix

$S(3 \times k)$: Independent components (k: number of components)

FastICA extracts the source signals by calculating the weight matrix W, which is the inverse of A:

$$\hat{S} = WX \tag{2}$$

The resulting output matrix $\hat{S}$ (4000×1) represents the combined signal of middle-ring-little finger movements. This process provided a consistent data structure of 4000×1 dimensions for all fingers [11,12].

Training-Test Split: The processed data were randomly split as 70% training and 30% test to evaluate the model's generalization ability. The training set was used for model optimization, and the test set was used for final performance evaluation.

Data preprocessing

In the data preprocessing stage, three methods were combined to extract meaningful features from raw signals: Time-Domain Descriptors (TDD), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) [8]. The features extracted by these methods were then ranked using Mutual Information (MI), and the top 30 most discriminative features were selected.

Time-Domain Descriptors (TDD)

TDD consists of 22 different metrics that quantify the statistical properties of signals. Each reflects the kinematic behavior of finger movements (acceleration, regularity, energy distribution, etc.). The basic features are: Mean value of signal, Standard deviation of signal, Maximum value of signal, Minimum value of signal, Root mean square of signal, Mean absolute value of signal, Wavelength of signal, Number of zero crossings, Number of slope sign changes, Skewness value of signal, Kurtosis value of signal, Variance of signal, Energy value of signal, Entropy of signal, Mean absolute difference, Maximum absolute difference, Median value of signal, Range of signal (maximum - minimum), 25th percentile of signal, 75th percentile of signal, Interquartile range (P75 - P25), Zero crossing rate

Convolutional Neural Network (CNN)

The CNN architecture was designed to extract spatial patterns from 1D signals:

Input Layer: Signals of length 4000 samples are converted into a 3D tensor.

Convolutional Layer: Local features are extracted with 64 filters. The convolution operation:

$$S(t) = (X * W)(t) = \sum_{i=1}^{k} W(i) \cdot X(t + i - 1) \tag{3}$$

ReLU Activation: To add non-linearity

$$f(x) = \max(0, x) \tag{4}$$

Max Pooling: Dimensionality is reduced with max pooling using a window of 2 samples.

$$y(t) = max\{S(t + i) | i = 0, \dots, p - 1\} \tag{5}$$

Flattening: Multi-dimensional outputs are converted into a single vector.

Dense (FullyConnected) Layers: Two fully connected layers, Dense (128) and Dense (64) with ReLU activation are used to extract high-level features from the flattened outputs of previous layers. Each neuron in a Dense layer is connected to all inputs from the preceding layer, summing the weighted inputs and adding a bias term:

$$z = \sum_{i=1}^{n} w_i \cdot x_i + b_i \tag{6}$$

The result is then processed using the ReLU activation function:

$$a = ReLU(z) \quad (7)$$

This allows the network to extract advanced information from the input data and represent complex patterns [13].

Parameters:

Conv1D: $64 \times (3 \cdot 1 + 1) = 256$parameters

Dense (128): Flatten dimension × 128 + 128 bias

Dense (64): 128 × 64 + 64 bias

Training Process: The CNN is trained through forward and backward propagation. During forward propagation, the input signal passes through the layers to generate predictions. The loss function compares predicted and true labels, commonly using categorical crossentropy

$$\mathcal{L} = -\sum_{i} y_i \log(\hat{y}_i) \quad (8)$$

Backpropagation calculates the gradients:

$$\frac{\partial \mathcal{L}}{\partial W} \quad (9)$$

Weights are updated using the Adam optimizer:

$$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W} \quad (10)$$

LSTM

The Long Short-Term Memory (LSTM) network was employed to extract temporal features from standardized one-dimensional EMG signals. LSTM is particularly suitable for modeling sequential dependencies, enabling the network to capture the dynamic behavior of finger movements over time. The LSTM architecture consists of the following structures:

The standardized EMG signals $X \in \mathbb{R}^N$are reshaped into a three-dimensional tensor of shape (number of samples, $N, 1$), where Nrepresents the number of time steps and 1 denotes the single feature per time step.

First LSTM layer (64 cells, return_sequences=True): Processes the input sequence one time step at a time. Each memory cell contains four gate mechanisms: forget gate f_t, input gate i_t, candidate cell state $\tilde{C}_t$, and output gate o_t, computed as:

- *Cell State:* The cell state represents LSTM's memory, holding past information, updated at each step to preserve information.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (11)$$

- *Forget Gate*: Determines how much past information to keep:

$$f_t = \sigma(W_f \cdot [h_{\{t-1\}}, x_t] + b_f) \quad (12)$$

- *Input Gate*: Controls how much new information is added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{\{t-1\}}, x_t] + b_i) \quad (13)$$

$$\widehat{C_t} = \tanh(W_C \cdot [h_{\{t-1\}}, x_t] + b_C) \quad (14)$$

- *Cell State Update*:

$$C_t = f_t \cdot C_{\{t-1\}} + i_t \cdot \widehat{C_t} \quad (15)$$

- *Output Gate:* Controls how much information from the cell is used for prediction; the model output consists of the output gate, cell state, and hidden state.

$$o_t = \sigma(W_o \cdot [h_{\{t-1\}}, x_t] + b_o) \quad (16)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (17)$$

Second LSTM layer (32 cells, return_sequences= False): Receives the sequence of hidden states from the first layer. Outputs only the last hidden state $h_N \in \mathbb{R}^{32}$, summarizing temporal information across the entire sequence.

Dense layer (32 neurons, ReLU activation): Transforms the LSTM output into a high-level temporal feature vector:

$$z = \sum_{i=1}^{n} w_i \cdot x_i + b_i \quad (18)$$

$$a = ReLU(z) \quad (19)$$

Parameters:

First LSTM layer: 17,408 parameters

Second LSTM layer: 12,416 parameters

Dense layer: 1,056 parameters

Training Process: The LSTM network is trained using Backpropagation Through Time (BPTT). The process consists of the following steps:

Forward Pass: The input sequences are processed through both LSTM layers, updating hidden and cell states sequentially.

Loss Computation: The network output is compared to ground truth labels using categorical crossentropy:

$$\mathcal{L} = -\sum_{i} y_i \log(\hat{y}_i) \quad (20)$$

Backward Pass (BPTT): Gradients are computed across all time steps:

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{N} \frac{\partial \mathcal{L}}{\partial h_t} \frac{\partial h_t}{\partial W} \quad (21)$$

Weight Update: Parameters are updated with the Adam optimizer:

$$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W} \quad (22)$$

Batch and Epoch Iterations: Multiple epochs and batches are used to allow the network to learn both short-term and long-term temporal dependencies present in the EMG signals.

Feature Selection

The total of 118 features obtained from TDD (22), CNN (64), and LSTM (32) were ranked using MI based on their information content with the target classes:

$$I(X_i;Y) = \sum_{x_i \in X_i} \sum_{y \in Y} P(x_i, y) \log\left(\frac{P(x_i,y)}{P(x_i)P(y)}\right) \quad (20)$$

The top 30 features with the highest MI values were selected. This step reduced the computational load while increasing classification accuracy. Figure 1 and Figure 2 show the feature vectors selected first and second by MI. The feature vector represents the coefficients extracted using the feature extraction methods for 50 samples of the same movement.

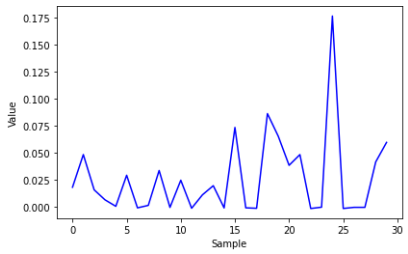


Figure 1. First Feature Selected by MI

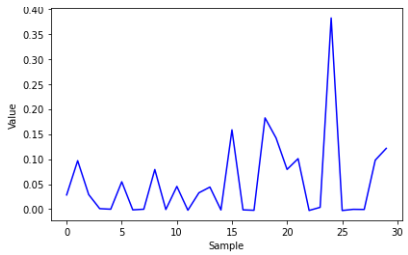


Figure 2. Second Feature by MI

Classification

Deep learning offers critical advantages in biomedical signal analysis due to its ability to model non-linear relationships in large and complex datasets. Unlike traditional machine learning methods, multilayer neural networks show superior performance in processing high-dimensional time series data like EMG. In this study, deep learning techniques were preferred for classifying finger movements; a combination of CNN and LSTM models was used for the feature extraction stage, while a Fully Connected Neural Network (FCNN) was chosen for classification.

Fully Connected Neural Network

Although Convolutional Neural Networks (CNN) are effective with spatial patterns, they are limited in capturing time-dependent dynamics in EMG signals. Recurrent Neural Networks (RNN) are suitable for time series but suffer from the "vanishing gradient" problem in long sequences. Although LSTM partially solves this problem, its computational cost is high. In contrast, FCNN learns non-linear patterns in EMG data at high speed thanks to its fully connected structure between all layers.

There are three main reasons for choosing FCNN in this study:

Computational Efficiency: Provides 5 times faster training time than LSTM with the 10-participant dataset.

Model Simplicity: Does not create an excessive parameter load for the complex but low-dimensional features in EMG signals (30 features selected by MI).

Accuracy/Performance Balance: It yields better results in learning due to its structure where each neuron is connected to all neurons in the previous layer and its ability to learn non-linear patterns in signals [14].

The fully connected layers of FCNN directly process the 30 features selected by Mutual Information to learn the discriminative characteristics of finger movements. This architecture allows it to establish a balance of high accuracy with low latency, which is critical for real-time applications such as prosthetic control. Consequently, FCNN was evaluated as the model that best combines computational efficiency, model simplicity, and performance for EMG-based movement classification.

Layer structure: The Fully Connected Neural Network (FCNN) model consists of three main layers for classifying finger movements: an input layer, two hidden layers, and an output layer. The model's architecture is designed to process the 30 features selected by Mutual Information (MI) to provide high-accuracy classification.

Input layer: A fully connected structure that receives 30 features and processes them with 128 neurons. Each neuron sums all input features with weights:

$$z_j = \sum_{i=1}^{30} w_{ji} x_i + b_j \quad (23)$$

Here w_{ji} is the weight between the i th feature and the j th neuron, x_i is the input data, and b_j is the bias term. The output is passed through the ReLU activation function:

$$f(z_j) = \max(0, z_j) \quad (24)$$

This function zeros negative values, allowing the model to learn non-linear relationships. Weight updates are performed using backpropagation with the derivative of the loss function:

$$w_{ji} \leftarrow w_{ji} - \eta \frac{\partial \mathrm{L}}{\partial w_{ji}} \quad (25)$$

$$b_j \leftarrow b_j - \eta \frac{\partial \mathrm{L}}{\partial b_j} \quad (26)$$

With the learning rate η, the model captures complex patterns in EMG features.

Hidden layers: The first hidden layer consists of 128 neurons, the second of 64 neurons. ReLU activation and 30% Dropout are applied in each layer. Dropout prevents overfitting by randomly disabling neurons during training:

$$a_j{}^l = \begin{cases} 0, & r_j \le p \\ f(z_j{}^l), & r_j > p \end{cases} \quad (27)$$

Inter-layer data flow is as follows:

$$z_j{}^l = \sum_{i=1}^{n_{l-1}} w_{ji}{}^l a_i{}^{l-1} + b_j{}^l \quad (28)$$

This structure learns hierarchical relationships between features, distinguishing patterns specific to thumb, index, middle, ring, and little finger movements.

Output layer: Each neuron in the output layer calculates probabilities for the 5 finger movements using information from the hidden layers and operates with the following formula:

$$o_j = \sum_{i=1}^{n_{L-1}} w_{ji}{}^L a_i{}^{L-1} + b_j{}^L \quad (29)$$

Softmax activation converts scores to probabilities:

$$P(y = j|x) = \frac{e^{o_j}}{\sum_{k=1}^{C} e^{o_k}} \quad (30)$$

The o_j values are calculated from the hidden layer outputs (Equation 23). Categorical Crossentropy is used as the loss function:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (31)$$

This function measures the alignment of the model's predictions with the true labels. Derivatives are used during backpropagation to update the weights.

Model Compilation

The model compilation stage is the step where three critical components defining the learning process behavior are specified: the loss function, optimization algorithm, and metrics. For this multiclass classification problem, Categorical Crossentropy was chosen as the loss function. This function measures the discrepancy between the model's predicted probabilities y_{ic} and the one-hot encoded form of the true labels y_{ic} :

$$L = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log(\hat{y}_{ic}) \quad (32)$$

Here N represents the number of samples, and C=5 represents the finger movement classes (thumb, index, middle, ring, little). Minimizing the loss value aims to bring the probability distributions for each class closer to the true labels.

The Adam algorithm was used for optimization. Adam operates with two moment estimates (m_t: first moment, v_t: second moment) that increase the speed and stability of gradient descent. The weight update steps are as follows:

Moment estimates:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (33)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t{}^2 \quad (34)$$

Here g_t is the current gradient, β_1=0.9 and β_2=0.999 are momentum parameters.

Bias Correction:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (35)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (36)$$

Weight Update:

$$\theta_t = \theta_{t-1} - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (37)$$

θ_t represents the model's weights.η is the learning rate. ϵ is a small constant used to prevent division by zero errors.

Adam's dynamic learning rate provides efficient convergence in learning the complex patterns of EMG signals, while its momentum mechanisms facilitate avoiding local minima. Accuracy was monitored as the performance metric, and the model's classification success was quantitatively evaluated.

Model Training and Validation

Model training is the process in which the FCNN learns finger movements using the 30 features selected by Mutual Information. Accuracy measures the model's generalization ability on unseen test data. In this process, labels for the 5-class (thumb, index, middle, ring, little) classification are converted into vector form using one-hot encoding:

$$y = [1,0,0,0,0] \quad (38)$$

This encoding allows the model to make probability predictions for each movement. To monitor the model's performance and prevent overfitting, the training data was split into two sets using hold-out validation method: 80% for training and 20% for validation. The validation set consists of data that the model has never seen during training and is used to evaluate the model's loss and accuracy at the end of each epoch. This approach is critical to determine whether the model merely memorizes the training data or can generalize effectively to new data. If the validation performance is low, it indicates potential overfitting, which can be mitigated by adjusting hyperparameters, such as the number of layers, number of neurons, dropout rate, or learning rate.

After training, the model's test set performance is evaluated. The test set is completely excluded from both the training and validation phases and serves to objectively measure the model's generalization capability on entirely new, unseen data. This separation ensures reliable assessment of the model's real-world classification performance.

Model performance is evaluated using four main metrics: accuracy, precision, recall, and F1-score. These metrics are calculated on both validation and test sets to provide a comprehensive analysis of the model's classification ability. While the validation set allows monitoring of generalization during training, the test set reveals the final, objective performance of the model. This methodology offers a reliable and academically accepted approach, especially for limited datasets and multi-class classification problems.

Accuracy: Measures overall prediction success.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (39)$$

Here TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative) values are used.

Precision: Shows how many of the positive predictions are actually correct.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (40)$$

Recall: Reflects how many of the true positives were correctly predicted.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (41)$$

F1-Score: The harmonic mean of precision and recall; critical in class imbalances.

$$\text{F1} - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (42)$$

In the result evaluation, training and test accuracy curves were compared. High training accuracy but low test accuracy is a sign of overfitting. In this study, using F1-score and class-based precision metrics, the model was ensured to show consistent performance across all finger movements.

Research findings and discussion

Within the scope of preprocessing results, TDD features successfully reflected the kinematic dynamics of finger movements (Fig. 3). For example, in thumb movements, RMS and waveform length measurements consistently revealed the intensity changes of muscle activation.

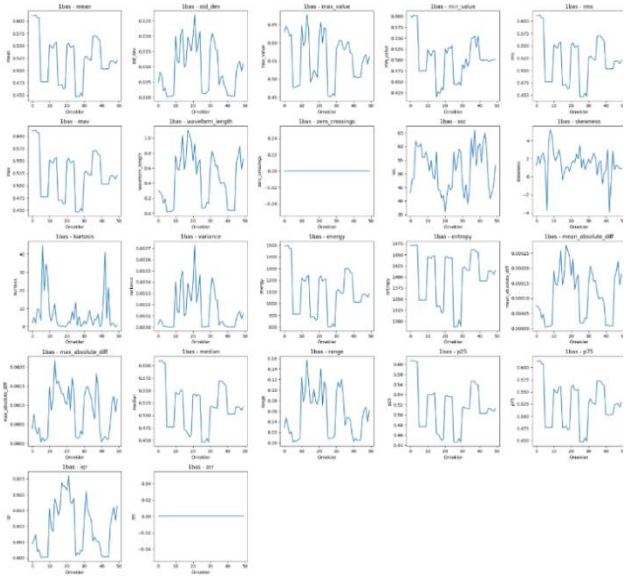


Figure 3. TDD Features

CNN features (Fig. 4, Fig. 5) encoded the spatial-frequency patterns of the signals with 64-dimensional vectors, while LSTM features (Fig. 6, Fig. 7) modeled the dependencies in the time series with 32 features.

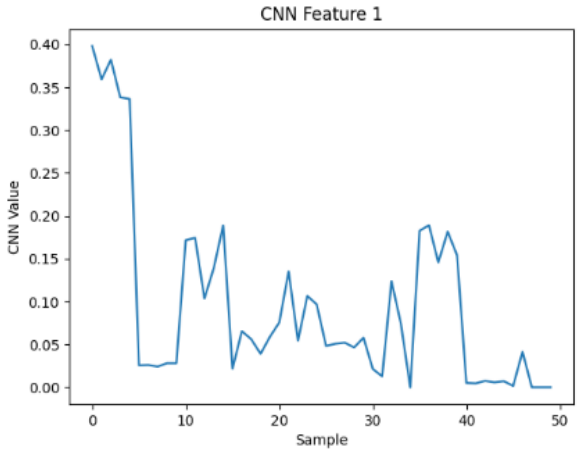


Figure 4. First Feature by CNN

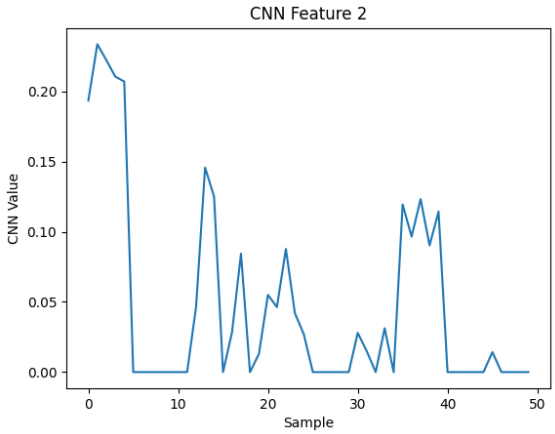


Figure 5. Second Feature by CNN

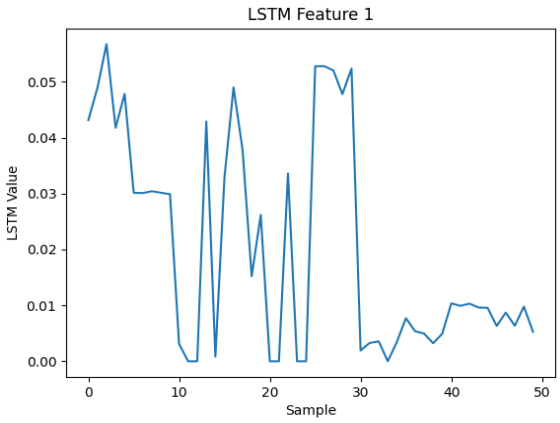


Figure 6. First Feature by LSTM

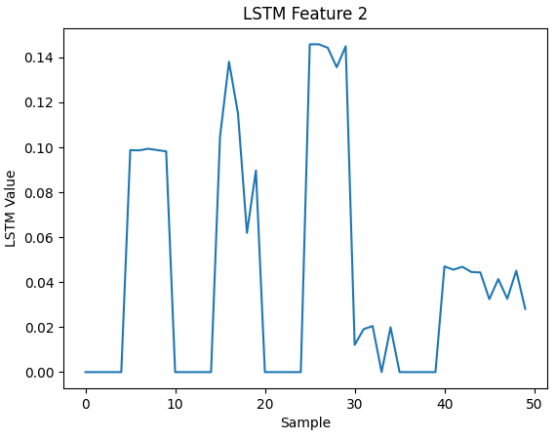


Figure 7. Second Feature by LSTM

Feature selection performed with Mutual Information (MI) reduced the 118 features to the top 30 most discriminative features (Fig. 8).

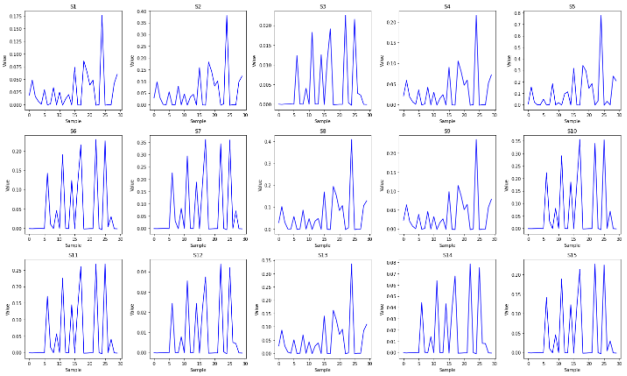


Figure 8. Top 30 Features Selected by MI

In the performance evaluation of traditional models, a linear Support Vector Machine (SVM) model was used to classify five different finger movements. The feature files were loaded, the labels were converted into numerical form, and all inputs were standardized using StandardScaler to help the SVM learn a stable and accurate decision boundary. The model was trained with kernel='linear' and C=1.0; the linear kernel separates classes with a hyperplane, while the C parameter controls the balance between margin width and error tolerance. During training, the SVM identified the optimal separating hyperplane and selected the most influential samples as support vectors.

During training, the model achieved an overall accuracy of 95%, with the thumb reaching an F1-score of 91%, indicating that the patterns in the training data were largely learned correctly. In the testing phase, the overall accuracy remained at 95%, and the index finger achieved the highest-class performance with a 100% F1-score. However, the ring finger's F1-score decreased to 93%, showing slightly lower performance compared to the other fingers. Class-based accuracy analysis revealed that the

thumb and index finger were predicted with 100% accuracy, while the ring finger had the lowest accuracy at 87%.

These results indicate that the SVM model can reliably classify some finger movements more accurately than others, and that class similarities or differences in data quantity can influence performance. Feature scaling and the selection of support vectors enabled the model to achieve high overall accuracy during both training and testing phases.

The K-Nearest Neighbors (KNN) algorithm with k = 5 was employed to classify finger movements. KNN is a non-parametric, instance-based learning method that classifies a new sample by examining the labels of its nearest neighbors in the feature space. Before classification, all features were standardized using StandardScaler, ensuring equal contribution of each feature to the distance calculations—an essential step for distance-based models such as KNN.

During the training phase, the KNN model achieved an overall accuracy of 97%, showing that the training samples were correctly clustered in the feature space. In the testing phase, the model reached 93% accuracy, slightly lower due to the sensitivity of KNN to class overlap and distance variations. Class-level accuracy results matched the general distribution observed in the SVM model, with most fingers classified reliably; however, the middle finger showed the lowest accuracy at 87%. In terms of F1-score, the index and little fingers achieved the highest performance at 97%, while the ring finger obtained the lowest value at 90%, indicating a higher degree of class similarity or feature overlap for that class.

FCNN classification results demonstrated remarkable performance. While 100% accuracy, precision, recall, and F1-score were achieved for all finger movements in the training phase, ≃99% overall accuracy was achieved on the test data. In the performance analysis, thumb, index, and little fingers yielded perfect results with 100% F1-score, while one misclassified sample in the middle finger led to a 93% F1-score. For the ring finger, 97% precision and 94% recall were measured. The confusion matrix (Fig. 9) confirms that only one middle finger movement was misclassified out of test samples.

Three key results stand out in the Discussion section: Firstly, the dataset reduced to 30 features by MI achieving ≃99% test accuracy in FCNN confirms the critical role of feature selection. Secondly, FCNN's 4-6% increase in test accuracy compared to SVM and KNN reflects its superior discriminative power, especially for movements with similar muscle activation. Finally, the single error in the middle finger may stem from the complexity of movement dynamics; this indicates the need for data augmentation and cross-participant validation in future studies. In light of these findings, the FCNN architecture, with ≃99% test accuracy, holds potential for practical applicability in prosthetic control systems

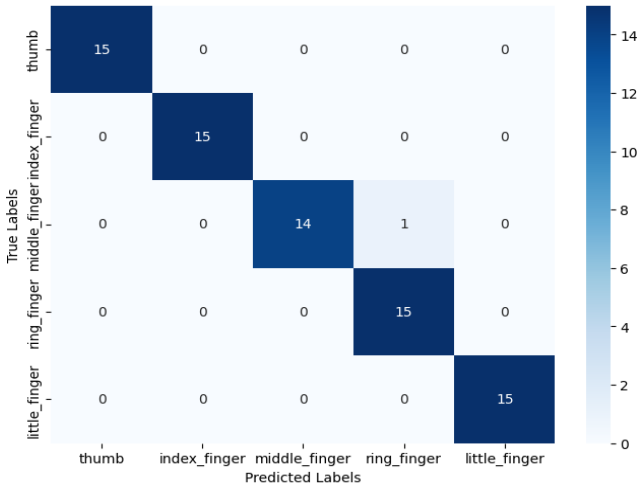


Figure 9. Confusion Matrix

.

Model Comparison and Literature Evaluation

In this study, the developed FCNN model outperformed both classical machine learning methods and widely used deep learning approaches in the literature. The results of the model comparison are clearly shown in Figure 10, where the accuracy comparison indicates that the FCNN produced higher and more consistent performance across all finger classes. Similarly, Figure 11 illustrates the distribution of performance metrics, confirming that the FCNN exhibited more stable and higher performance compared to other models. The SVM model achieved 95% validation accuracy, KNN reached 93%, while the FCNN achieved approximately 99% accuracy. This difference demonstrates that the FCNN can effectively learn the complex biological patterns contained in EMG signals.

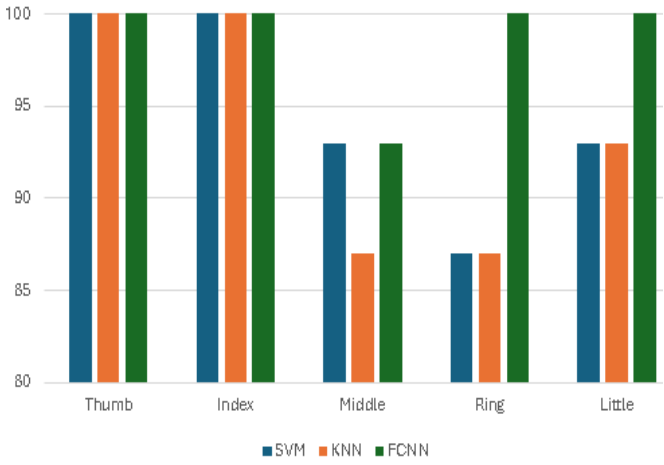


Figure 10. Model Accuracy Comparison

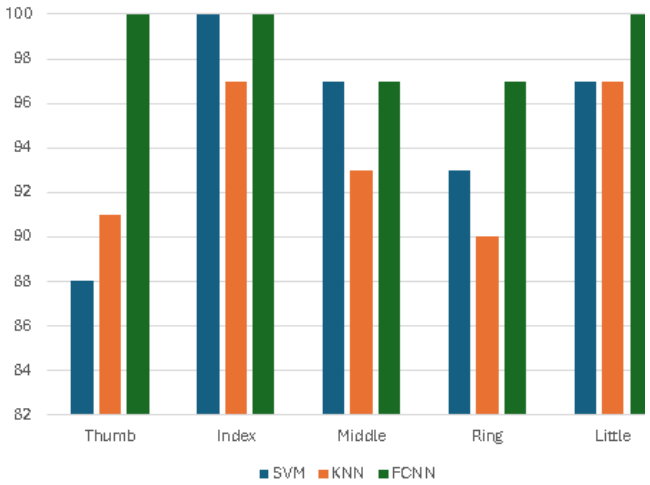


Figure 11. Performance Comparison

Analysis of the training and validation process further supports the model's success. As shown in Figure 12, the parallel progression of training and validation losses indicates that the model does not suffer from overfitting. Additionally, Figure 13 presents the convergence of training and validation accuracies, demonstrating high generalization capability and confirming the significant contribution of data partitioning strategies and Dropout layers to this stable behavior.

The findings of this study provide a notable performance advantage when compared with similar EMG-based classification models in the literature (Table 3). Atzori applied a CNN-based approach on 67 healthy and 11 amputee participants, reporting 66% accuracy for healthy individuals and 38% for amputees [4]. The large participant variation increased inter-subject variability and limited the CNN's ability to discriminate fine patterns in EMG signals.

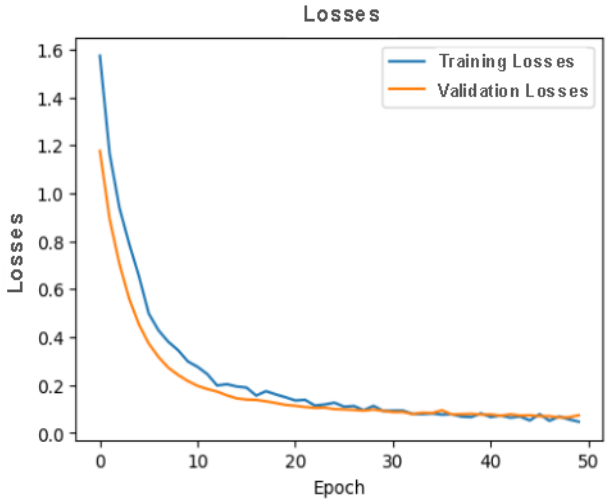


Figure 12. Training and Validation Losses

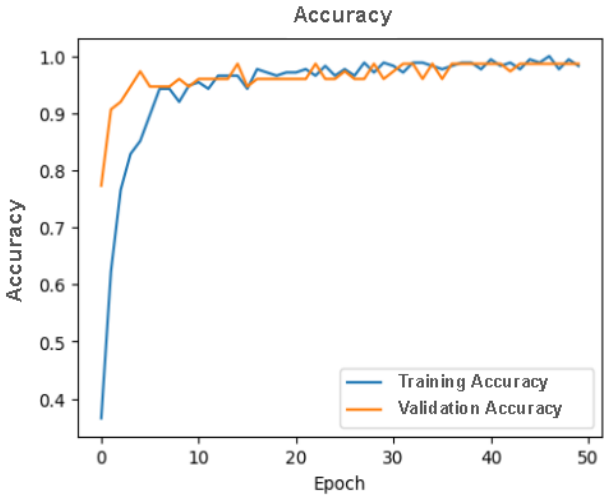


Figure 13. Training and Validation Accuracies

The CNN-based study conducted by Triwiyanto et al. used raw EMG signals directly as input and achieved 97% accuracy, demonstrating the spatial filtering advantages of CNN [5]. However, the amplitude-based variations in the raw signals were not fully captured, preventing results from reaching the level achieved by the FCNN in our study.

Azhiri's RNN-based study achieved 96% accuracy with time-frequency features and a prediction time of 600 ms, providing a noteworthy alternative for real-time control applications [6]. Nevertheless, RNN models are known to occasionally experience instability when capturing long sequential dependencies.

In machine learning-based studies, Lee et al. extracted only time-domain features and achieved 94% accuracy with ANN [7]. While the ANN approach is effective for small datasets, it cannot provide as powerful a representation as deep models for multi-channel EMG signals with high variance.

Another important study in the literature is Dwivedi's MRD model, developed using the EMG + finger kinematics dataset [10]. This model achieved 91% accuracy . Our study uses the same dataset. Dwivedi's approach modeled nonlinear relationships between EMG signals and kinematic data but was limited in representing complex patterns. In contrast, in our study, using only EMG signals, the FCNN model achieved approximately 99% accuracy, demonstrating that the same dataset can be utilized more efficiently. This difference can be attributed to feature extraction methods, model depth, and the number of hidden layers representing nonlinear relationships.

Table 3. Comparison of Literature Studies

Study	Method	Accuracy	Data Type

Atzori et al.[4]	CNN	75%	EMG
Triwiyanto et al.[5]	CNN	97%	EMG
Azhiri et al.[6]	RNN	96%	EMG
Lee et al. [7]	ANN	94%	EMG
Dwivedi et. Al. [10]	MRD	91%	EMG+ Kinematic
This Study (SVM, KNN)	SVM, KNN	95%, 93%	EMG
This Study (FCNN)	FCNN	≃99%	EMG

Considering all findings from the literature together, the reasons for the superior performance of FCNN are clear. Firstly, the multi-layered architecture of FCNN is highly suitable for directly learning the rich biometric information contained in EMG signals. Although CNN models are optimal for spatial filtering, they may not always perfectly represent the time–amplitude structure of EMG. RNN models are strong in capturing sequential dependencies but have disadvantages in training stability and maintaining long dependencies. MRD lacks deep representational power. In contrast, FCNN can learn high-dimensional representations suited to the structure of EMG, resulting naturally in higher classification performance.

This study achieved approximately 99% accuracy with FCNN, surpassing the 66–97% accuracy range observed in the literature for both deep learning and classical machine learning methods. All visualizations presented in Figures 10–13 systematically validate the model's performance superiority and generalization ability. The success of FCNN represents an important finding that may contribute to the development of more reliable and high-accuracy EMG-based human–machine interface systems in the future.

Dataset Limitations

The dataset used in this study has several limitations that should be noted. First, the sample size is small, with only 10 healthy participants (9 male, 1 female) aged 26–31, which limits the generalizability of the results. Second, the participant diversity is low, as no amputees or individuals with different physiological conditions were included. Third, the movement types are restricted to finger flexion and extension under controlled laboratory conditions, limiting the representation of natural daily movements. Finally, only eight EMG channels covering extrinsic forearm muscles were used, excluding intrinsic hand muscles and other sensor modalities. These limitations indicate that while the FCNN model performs well on this dataset, its applicability to broader populations and more diverse movement scenarios may require further data collection and validation.

Conclusion and recommendations

In this study, feature selection and classification operations performed using the Python programming language in the Spyder environment provided high performance without overfitting. The FCNN model achieved 100% accuracy in the training phase and ≃99% overall accuracy rate in the test phase. When evaluated on a per-finger basis; thumb, index, ring, and little finger movements were classified with 100% accuracy, while the middle finger showed the lowest performance with 93% accuracy (Table 4).

Table 4. FCNN Test Classification Performance

Finger	Precision	Recall	F1-score	Accuracy	Total Accuracy
Thumb	100	100	100	100	
Index	100	100	100	100	
Middle	100	93	97	93	≃ 99
Ring	94	100	97	100	
Little	100	100	100	100	

In the feature extraction process, 22 features were obtained simply and effectively from time series data using TDD. To complement this, the CNN model extracted 64 spatial-frequency features, and LSTM extracted 32 temporal dependency features. The total of 118 features was reduced to the top 30 most discriminative features using the Mutual Information (MI) method, and this process enabled the FCNN classifier to operate with high accuracy.

In light of the challenges encountered during the study and the experiences gained, the following recommendations can be offered for researchers working in the same field:

In the data set selection phase, the collection methodology (devices, electrode configuration) of existing data sets and their use in previous studies should be carefully examined. This review is a critical step that may take several weeks to determine the appropriate data set. For *feature extraction*, trying methods widely accepted in the literature (TDD, CNN, LSTM) primarily will provide time efficiency. In the model selection process, the results of all trials (including failed ones) should be recorded systematically; because the cause of low performance may be data quality or feature selection rather than the model. Finally, additional techniques such as Dropout, data partitioning, and regularization must be applied to improve model performance.

These recommendations will help researchers optimize resource usage and achieve high-accuracy results in

biomedical signal processing studies such as finger movement classification.

Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared.

There is no conflict of interest with any person / institution in the article prepared.

Acknowledgement

In this study, the software provided by the project numbered 2024-047, supported by the Manisa Celal Bayar University Scientific Research Projects Coordination Unit, was used.

References

[1] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," 2023. doi: 10.3390/computers1205009.

[2] A. Ibrahim, V. R. Gannapathy, L. W. Chong, and I. S. M. Isa, "Analysis of electromyography (EMG) signal for human arm muscle: A review," in *Lecture Notes in Electrical Engineering*, 2016. doi: 10.1007/978-3-319-24584-3_49.

[3] M. Benzyane, M. Azrour, I. Zeroual, and S. Agoujil, "Exploring the Impact of Convolutions on LSTM Networks for Video Classification," in *Lecture Notes in Networks and Systems*, 2024. doi: 10.1007/978-3-031-48573-2_4.

[4] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks: a resource for the control of robotic prosthetic hands via electromyography," *Front Neurorobot*, vol. 10, 2016.

[5] T. Triwiyanto, V. Abdullayev, and A. A. Ahmed, "Deep Convolution Neural Network to Improve Hand Motion Classification Performance Against Varying Orientation Using Electromyography Signal," *International Journal of Precision Engineering and Manufacturing*, vol. 25, no. 6, 2024, doi: 10.1007/s12541-024-00985-x.

[6] B. Azhiri, M. Esmaeili, and M. Nourani, "Real-Time EMG Signal Classification via Recurrent Neural Networks," in *Proceedings- 2021 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2021*, 2021. doi: 10.1109/BIBM52615.2021.9669872.

[7] K. H. Lee, J. Y. Min, and S. Byun, "Electromyogram-based classification of hand and finger gestures using artificial neural networks," *Sensors*, vol. 22, no. 1, 2022, doi: 10.3390/s22010225.

[8] Ari, A. (2023). EMG signal classification using deep learning and time domain descriptorsbased feature extraction for hand grip movement recognition. Traitement du Signal, Vol. 40, No. 3, pp. 949-960. https://doi.org/10.18280/ts.400311.

[9] V. Medved, S. Medved, and I. Kovač, "Critical Appraisal of Surface Electromyography (sEMG) as a Taught Subject and Clinical Tool in Medicine and Kinesiology," 2020. doi: 10.3389/fneur.2020.560363.

[10] S. K. Dwivedi, J. Ngeo, and T. Shibata, "Extraction of Nonlinear Synergies for Proportional and Simultaneous Estimation of Finger Kinematics," *IEEE Trans Biomed Eng*, vol. 67, no. 9, 2020, doi: 10.1109/TBME.2020.2967154.

[11] A. Tharwat, "Independent component analysis: An introduction," 2018. doi: 10.1016/j.aci.2018.08.006.

[12] M. Chen and P. Zhou, "A novel framework based on FastICA for high density surface EMG decomposition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 1, 2016, doi: 10.1109/TNSRE.2015.2412038.

[13] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," 2018. doi: 10.1007/s13244-018-0639-9.

[14] P. Mahajan, "Fully connected vs Convolutional Neural Networks," *Medium*. [Online].Available:https://medium.com/swlh/fully-connected-vs-convolutional-neural-networks-813ca7bc6ee5.