Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 22, Iss. 2, pp. 48-61, 2024

COMPARISON OF SEMG-BASED HAND GESTURE CLASSIFIERS

Guilherme C. De Lello, Gabriel S. Chaves
PEE/COPPE, Universidade Federal do Rio de Janeiro – UFRJ
{ guilherme.delello, gabriel.chaves}@smt.ufrj.br

Juliano F. Caldeira
PEE/COPPE, Universidade Federal do Rio de Janeiro – UFRJ
julianoc@ufrj.br

Markus V. S. Lima
Poli & PEE/COPPE, Universidade Federal do Rio de Janeiro – UFRJ
markus.lima@dee.ufrj.br

Abstract – Machine learning techniques have shown success in classifying hand gestures. As the prevalence of prosthetic devices continues to rise, the adoption of non-invasive technologies, such as surface electromyography (sEMG), becomes paramount. This study systematically assesses the isolated influence of classification algorithms within hand gesture recognition (HGR) systems using sEMG data and dynamic time warping (DTW) based features. This approach effectively handles temporal variations in sEMG signals by leveraging DTW, ensuring input features are invariant to gesture speed. Six supervised learning classifiers were evaluated: the multilayer perceptron, support vector machine, logistic regression, linear discriminant analysis, *k*-nearest neighbors, and decision tree. Cross-validation was employed to fine-tune the segmentation hyperparameters, significantly improving results. To ensure reproducibility, the source code has been made available, the proposed system design has been detailed, and the evaluation protocols have been described. Our findings indicate that logistic regression outperformed other classifiers in this setup, achieving 95.2% accuracy in classifying six hand movements from ten healthy individuals, representing a 1.6% improvement over the best previously reported performance using the same publicly available dataset. Future research will assess the proposed HGR system's generalization capability on larger datasets suitable for training more complex classifiers, including deep learning models.

Keywords – Hand gesture classifier, segmentation, sEMG, dynamic time warping, artificial neural networks, support vector machines, logistic regression, linear discriminant analysis, *k*-nearest neighbors, decision tree.

1. INTRODUCTION

The problem of *hand gesture recognition* (HGR) involves detecting and classifying muscle contractions resulting from the intention of performing hand movements. Solving this problem can significantly improve human-machine interface applications, such as robotic hand control [1], game interfaces [2], augmented reality [3], sign language recognition [4, 5], and active prostheses [6]. The latter plays a vital role in rehabilitating upper limb amputees. Experiencing the loss of a limb can profoundly impact a person's body, emotions, relationships, career, and overall way of living [7]. In the case of transradial amputation, a prosthetic hand can be a helpful aid for individuals in adjusting to their new lifestyle. Therefore, myoelectric prostheses have become increasingly relevant [8, 9].

Myoelectric prostheses are devices controlled by the signals generated by muscle contractions, known as *electromyography* (EMG) signals. However, for such a prosthesis to work correctly, the residual limb must have measurable EMG signals [8, 10]. In most cases, the device is limited to opening and closing hands. Users can train to increase the complexity of the gestures through sequential control strategies. However, controlling this prosthesis requires significant endeavor, high skill levels, and extended training sessions [10]. Based on a study by Bowker et al. [11] in 1992, it was found that prostheses designed for upper extremities have higher rejection rates compared to those designed for lower extremities. To reduce rejection rates and improve ease of use, researchers recommend non-invasive techniques like *surface electromyography* (sEMG) for prosthetic devices [12].

The sEMG signal captures the spatial and temporal patterns of the electrical activities generated by motor neurons and muscle fibers [13]. These electrical pulses occur in response to movement commands from the brain [13–16]. When we move our hands, multiple muscle stimuli are triggered, which can be captured by an sEMG sensor placed on the skin. These sensors can control external devices, such as prostheses, that are simple to use with high reliability [17–19].

Due to their stochastic and non-stationary nature, automatically recognizing sEMG patterns can be challenging. Researchers usually divide sEMG-based HGR systems into six stages: data acquisition, segmentation, preprocessing, feature extraction, classification, and post-processing [20]. The first stage is the data acquisition, which involves capturing and reading data examples. The second stage, the segmentation, partitions the data for further processing. Segmentation usually employs two techniques: sliding windowing and gesture detection. The former splits the sEMG signal into multiple adjacent or overlapping time windows, while the latter extracts the muscle contraction pattern by identifying the beginning and end of muscle activity. The third stage is preprocessing, which conditions the signal for feature extraction. Feature extraction, the fourth stage, transforms sEMG data into relevant features for the gesture recognition problem. The fifth stage is classification. In this stage, the classifier matches the features to their respective labels. Finally, the post-processing stage refines the classification output, enhancing the interpretation and evaluation of the model results.

Recently, machine learning approaches have achieved excellent performance in designing natural gesture interfaces [12, 20–22]. These methods identify muscle contraction patterns in sEMG signals and match them with predefined gestures through supervised learning. Shi et al. [23] achieved 94.0% accuracy in identifying four hand postures by using k-nearest neighbors (KNN) and adjacent sliding windowing to segment the data. Similarly, Yang et al. [24] proposed a solution using Gaussian mixture models, hidden Markov models, and overlapped sliding windows to classify six hand movements, achieving 99.0% accuracy. In another study, Yang et al. [25] recognized nine hand gestures using linear discriminant analysis (LDA) and overlapped sliding windows, achieving 92.2% accuracy. Benatti et al. [26] used a support vector machine (SVM) and gesture detection to classify seven hand movements, achieving 89.2% accuracy. Sultana et al. [27] analyzed 33 research articles and found that the multilayer perceptron [1] (MLP) achieved the highest reported accuracy in HGR, while the SVM was the second most commonly used classifier. Yasen and Jusoh [31] confirmed MLP as the most applied classifier in their review of 71 studies.

Modern machine learning techniques have also shown excellent results in gesture classification [14, 27, 31]. For example, Redrovan and Kim [32] used a convolutional neural network (CNN) and adjacent sliding windows to classify eight hand gestures, achieving an accuracy of 93.0%. Similarly, Côté Allard et al. [33] recognized seven gestures using CNNs and overlapped sliding windows, achieving an accuracy of 97.2%. Conversely, Moslhi et al. [34] introduced an innovative approach to hand gesture classification by using Transformers, a network architecture originally designed for Natural Language Processing, outperforming most CNNs on NinaPro DB1, CapgMyo A, and CapgMyo B datasets.

Regardless the numerous machine learning techniques available to HGR researchers, few studies have compared the performance of different classifiers while keeping the other five stages fixed. In 1999, Englehart et al. [35] conducted experiments to compare the performance of LDA and MLP. Their findings showed that LDA performed similarly to MLP for time-frequency features when principal component analysis (PCA) was used to reduce the number of dimensions. Despite being a less complex model, LDA could replace MLP without degrading accuracy because PCA projects mutually uncorrelated features, thus allowing a linear model to effectively accomplish the discrimination task. In 2008, Oskoei et al. [36] demonstrated that SVMs could match or exceed the performance of LDA and MLP for both time and frequency-domain features. In 2015, Atzori et al. [37] reported that nonlinear classifiers such as SVM with Radial Basis Function (RBF) kernel and MLP achieved better performance than linear SVM, LDA and KNN. More recently, in 2020, Mendes et al. [38, 39] evaluated seven classifiers – Naïve Bayes, KNN, Extreme Learning Machine (ELM), Random Forest, LDA, Quadratic Discriminant Analysis, and SVM – while comparing two feature reduction approaches: feature selection and dimensionality reduction. Their feature selection experiment indicated that ELM and SVM with RBF are the classifiers that best combined high accuracies with a minimal number of features. Meanwhile, the dimensionality reduction experiment revealed that SVM with RBF achieved the highest overall accuracy.

Dynamic time warping (DTW) is another promising technique in HGR systems. DTW is a well-established method for handling temporal variations in time series comparisons and has been successfully applied in various fields, including speech processing [40] and computer graphics [41]. By using DTW as a similarity measure, HGR systems can ensure that input features remain invariant to the speed of gestures performed during sEMG acquisition. Despite its widespread use in other domains, the application of DTW in sEMG-based HGR systems have been explored by relatively few research groups [21, 42–50].

Hence, this work conducted a systematic comparison of sEMG-based HGR systems using different classifiers, while maintaining the same dataset, preprocessing algorithms, segmentation strategy, features, post-processing, and overall experimental conditions. We compared the performance of six supervised learning classifiers: MLP, SVM, logistic regression (LR), LDA, KNN, and decision tree (DT). We designed our proposed system to identify six types of hand movements performed by ten volunteers. Due to its recent development, publicly available datasets, explicit protocols, and leverage of DTW-based features, we selected the HGR system proposed by Benalcázar et al. [21] as our baseline. This system serves as the reference point for comparing the performance of the different classifiers tested in our research. Our experiments followed the same architecture, features, and parameters as the baseline to ensure a fair comparison. However, we made two essential modifications: 1) we used cross-validation to fine-tune the segmentation hyperparameters, which improved the performance, as demonstrated in our previous research [51]; and 2) we analyzed the performance of various systems that retained the same characteristics as the baseline except for the classifier. Additionally, we created a public GitHub repository, named *HGR Lab* [52], providing the Python code to reproduce our experiments.

The dataset used in this study, while comprehensive for the evaluation of traditional classifiers, lacks the number of examples necessary for training more complex models like deep learning with convolutional layers [14, 53]. This limitation is a significant factor in our choice of classifiers. Despite these constraints, our systematic approach provides valuable insights into the performance of various traditional classifiers, contributing to the body of knowledge in HGR systems. Future work will aim to expand the dataset to explore the potential of deep learning techniques.

This text is organized as follows. In Section 2, we present the proposed HGR system for classifying hand gestures. In Section 3, we report the experimental results obtained using different classifiers. In Section 4, we examine the performance achieved by each classifier. Finally, in the last section, we draw our conclusions.

[1] The multilayer perceptron is also known as the feedforward neural network [28–30].

2. HAND GESTURE RECOGNITION SYSTEM

A standard structure for real-time HGR systems comprises six stages: data acquisition, segmentation, preprocessing, feature extraction, classification, and post-processing [20]. The HGR systems assessed in this work share the same algorithms for all stages, except the classification step. This section explains the algorithms performed in each phase, which are based on the HGR system presented by Benalcázar et al. [21]. Additionally, we implemented the modifications in the segmentation stage suggested by Chaves et al. [51], which consist in optimizing the segmentation threshold for each subject. This section presents an overview of the algorithms employed in each stage, while these two works mentioned above present their detailed mathematical description.

2.1. DATA ACQUISITION

The first stage reads signals measured by sEMG sensors. In this work, we utilized the publicly available datasets by Benalcázar et al. [21]. These datasets contain sEMG signal recordings collected using the Myo armband [54], a low-cost commercial device from Thalmic Labs equipped with eight sEMG sensors sampled at 200 Hz. Recent examples of research using this device can be found in [55, 56]. Data from 10 subjects were recorded, and each person contributed with a *training dataset* and a *testing dataset* performing five gestures: fist, wave-in, wave-out, fingers spread (aka open), and double tap (aka pinch).

Each *training dataset* $\mathcal{D}_{\text{train}}$ contains 30 trials, each trial being an 8-channel time series recorded by the same individual executing the following sequence of hand movements: rest, gesture of interest, and then rest. There are five examples for each gesture of interest, with the subject performing this sequence of movements in each example. Additionally, the training dataset includes five recordings of the person keeping his arm resting throughout the trial. The *testing dataset* $\mathcal{D}_{\text{test}}$ was created using the same protocol and contains 150 trials, 30 for each gesture of interest. These dataset sizes were determined by Benalcázar et al. [21] in their original dataset creation. The fact that both datasets include the gesture labels of each trial enables the use of the supervised learning classifiers employed in this work.

2.2. SEGMENTATION

The segmentation stage partitions EMG data into multiple analysis windows called segments. A segment is a time slot of myoelectric data used as input for the subsequent stages. Two standard segmentation techniques for EMG data are sliding windowing and gesture detection [20].

The sliding windowing technique splits the data into either disjoint or overlapping segments [13, 57–61]. In disjoint (or adjacent) segmentation, the position of the following window is incremented by an amount equal to its size. In overlapping segmentation, the new window slides over the current by a stride length less than the window size. Thus, overlapping segmentation allows the classifier to generate a denser stream of outputs [62].

There is a trade-off in choosing the segment length: more data produces features with lower variance, thus increasing classification accuracy. Conversely, the processing time to generate a decision also increases [63]. Therefore, the segment length is constrained by the desired response time of the system, which, for real-time prosthetic control systems, should not be greater than 300 ms [64]. All systems presented in this work use overlapping analysis windows of length 2.5 s and a stride of 50 ms. These parameters were originally determined by Benalcázar et al. [21] and were adopted in our study to maintain consistency with their work. The segment length of 2.5 s ensures that the analysis windows capture enough data to provide a reliable context for gesture classification, while the stride of 50 ms balances the need for timely responsiveness with the computational load, ensuring that the system remains efficient for real-time applications.

Gesture detection estimates each hand gesture's beginning and end, usually by setting a threshold to some signal characteristic. We applied the gesture detection algorithm detailed by Benalcázar et al. [21] for all systems. This algorithm estimates the power of the sum of the envelopes of all eight EMG channels in sub-windows of 125 ms and compares the result with a segmentation threshold τ with a fixed value of 10. Nevertheless, instead of using a fixed threshold for all subjects, in this work, we employed the k-fold cross-validation method to estimate the threshold that optimizes individual accuracy, as described by Chaves et al. [51].

2.3. PREPROCESSING

Preprocessing is an essential step in conditioning EMG signals before feature extraction. It is typically done through filtering or rectifying [21]. In all systems, we performed both operations during preprocessing. First, we rectified the signal. After that, we applied a low-pass Butterworth filter with a cutoff frequency of 20 Hz [65–68]. The resulting signal is the envelope of the sEMG. Figure 1 illustrates a typical sEMG signal from the dataset, demonstrating amplitude variations according to the hand's movement types. Amplitude remains low while resting and increases when the user executes the gesture. We aim to classify the portion of the signal that contains the gesture of interest, which is a random process that comprises transient and steady-state components [13, 14, 21, 69, 70].

2.4. FEATURE EXTRACTION

Feature extraction transforms the original data into a new space of variables where the pattern recognition problem is more straightforward to solve [28]. We used the "distance" between the preprocessed sEMG segment and the signal representing each trained gesture's center as feature vectors. Since there are examples of five distinct gestures plus examples of complete rest in the dataset $\mathcal{D}_{\text{train}}$, the feature vector has length six.

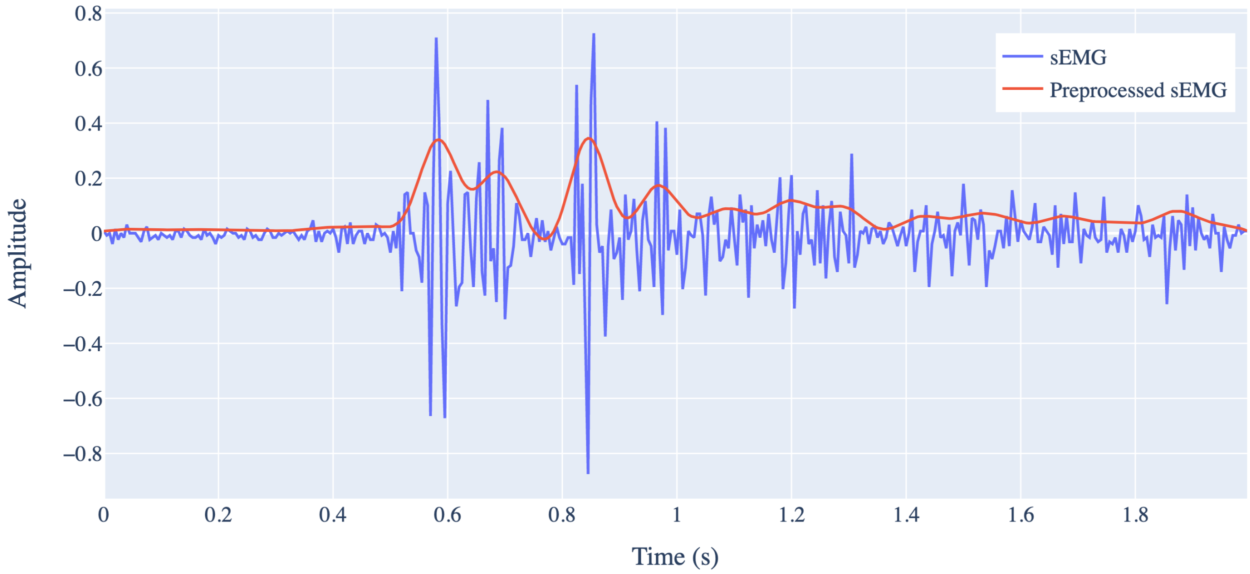


Figure 1: One channel of sEMG data corresponding to a double-tap gesture (blue) and the result of its preprocessing stage (red).

The similarity measure chosen was the *Dynamic Time Warping* (DTW) distance, which compares two time series by computing the cost to align them optimally. DTW originated from speech processing, but there are applications in fields such as data mining, bioinformatics, robotics, and computer graphics [41]. To define the center of each trained gesture for a person, the DTW cost was computed between every pair of examples of the same gesture in $\mathcal{D}_{\text{train}}$. The example with the lowest DTW cost among all others was then selected as the class center.

2.5. CLASSIFICATION

The goal of the classifier is to assign each input vector from the feature set to a trained gesture class. In this study, we evaluated the performance of five different classification algorithms: SVM, LR, LDA, *K*NN, and DT, whose implementations were drawn from the Scikit-learn project [71]. These models were chosen because they have been successfully used in classifying sEMG data in previous studies [20,27,38,39,48,49,63,72]. However, it is important to note that our dataset had limited training data, which hindered the use of more complex classifiers, such as deep learning models.

We compared the performance of these five classifiers with the results reported by Benalcázar et al. [21] and Chaves et al. [51]. Their studies used the MLP with three layers (input, hidden, and output) and six neurons in each layer. The hidden layer utilized the hyperbolic tangent activation function, while the output layer employed the softmax function. They used the cross-entropy cost function and the gradient descent optimization method [28, 30, 73]. The input layer received normalized feature values obtained by applying the formula $(X_i - \mu_i)/\sigma_i$, where X_i, μ_i and σ_i represent, respectively, the i^{th} input vector, its mean, and its standard deviation. We adopted this same feature normalization process in all our experiments.

The details of the classifiers evaluated in this work are as follows. The hyperparameters for the MLP were adopted from the baseline HGR system [21], while those for the SVM, LR, LDA, *K*NN, and DT were determined through an exhaustive search. However, to ensure a fair comparison, we did not optimize these hyperparameters for individual subjects, as the MLP parameters in [21] were not optimized in this manner.

i. **Support vector machines** are maximum margin classifiers that optimize the distance from the decision boundary to any of the samples [28]. We used the multiclass one-versus-one implementation with the RBF kernel from LIBSVM [74].

ii. **Logistic regression** is a generalization of linear regression for the classification problem that uses the sigmoid (or logistic) function as the activation function. We chose the liblinear solver [75] and ℓ^2-norm penalty.

iii. **Linear discriminant analysis** is a particular case of Gaussian discriminant analysis in which the covariance matrices are assumed to be shared by all classes [29]. We chose the singular value decomposition (SVD) solver.

iv. ***K*-nearest neighbors** is a non-parametric classifier that estimates the probability distribution of previously unseen examples belonging to a specific class by the relative frequency of their *k* closest neighbors [29]. We used $k = 5$, following the work by Benalcázar et al. [48] with a similar dataset.

v. **Decision Tree** is a non-parametric supervised learning method that recursively partitions the input space, defining a local classification model in each resulting segment [29]. For this study, we used the Gini index as the cost function.

It is essential to notice that determining the model parameters for the SVM, LR, and LDA classifiers corresponds to a convex optimization problem. So, any local solution must also be at a global minimum, independent of the model's initial conditions [28, 29]. On the other hand, the MLP is characterized by a multimodal cost function. This fact implies that, depending on how the model parameters of the MLP are initialized, a different solution is achieved [30]. Finally, despite *K*NN and DT being non-parametric algorithms, the former achieves a unique solution if the tie criterion is not random [28]. In contrast, Scikit-learn's DT classifier implementation could converge to different solutions [71].

2.6. POST-PROCESSING

The dense stream of decisions produced by the overlapped segmentation technique described in Section 2.2 could overload the prosthesis controller. Thus, those decisions might be combined in a post-processing algorithm to reduce the number of instructions sent to the controller and improve system performance. For this purpose, we used the same procedure as Benalcázar et al. [21], eliminating consecutive gesture label repetitions.

3. EXPERIMENTAL RESULTS

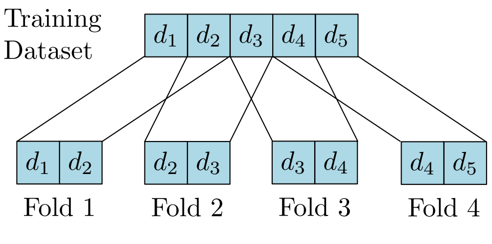
In this section, we present the results of the experiments designed to compare the performance of the HGR system described in Section 2 using different classifiers, *ceteris paribus*. Additionally, we compare these new results with those reported by Benalcázar et al. [21] and Chaves et al. [51], who followed the same method but employed the MLP as the classifier. As described in Section 2.2, the key difference between these approaches is that Benalcázar et al. [21] used a fixed segmentation threshold τ, whereas Chaves et al. [51] demonstrated that optimizing τ using *4*-fold cross-validation can enhance model performance.

We have established a public GitHub repository named *HGR Lab* [52], which contains the Python code needed to reproduce the experiments presented in this study.

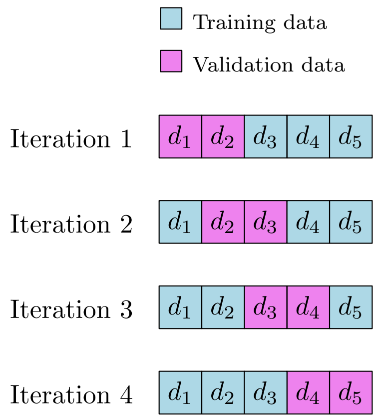
3.1. HYPERPARAMETERS TUNING

We developed individual systems for each subject, meaning the system's training and evaluation relied solely on data from a single person. For each classifier, we optimized the gesture detection threshold τ defined in Section 2.2 for each of the ten subjects. This individual segmentation optimization allows the model to account for variations of the sEMG patterns produced by different people while executing hand movements, thus improving classification performance. To illustrate, consider two people closing their hands. The first person does so slowly with little force, while the second performs the same gesture rapidly with more force. Clearly, the sEMG patterns of the second person will differ in duration and intensity from those of the first person.

Chaves et al. [51] proposed three cross-validation methods to choose the optimum threshold: 3-fold, 4-fold, and 5-fold. In this work, we utilized the 4-fold cross-validation, which produced the best results in [51]. To perform the 4-fold cross-validation, we divided the training dataset, consisting of five data examples for each gesture, into four folds, as depicted in Figure 2a. In this figure, d_i for $i = 1, ..., 5$ represents the data examples from a single gesture. During each iteration, as illustrated in Figure 2b, one fold was designated as validation data, whereas the remaining folds served as training data.



(a) Training dataset splitting for the 4-fold cross-validation.



(b) Iterations of the 4-fold cross-validation procedure.

Figure 2: 4-fold cross-validation procedure.

We used the baseline threshold value of $\tau = 10$ [21] as a reference to create the search space, varying this hyperparameter from 10 to 20 in the optimization process. This range was selected by analyzing the spectrum of the preprocessed sEMG signal at the transition from the relaxed position to the gesture of interest. Our analysis concluded that values below 10 reflect fluctuations in the relaxed position, while values above 20 indicate that the gesture has already reached its steady state. The optimal τ for each individual and classifier is listed in Table 1, which also includes the values utilized by Benalcázar et al. [21] and Chaves et al. [51]. In the event of a tie, we chose the highest threshold that provided maximum accuracy. As expected, the optimal threshold varied among individuals.

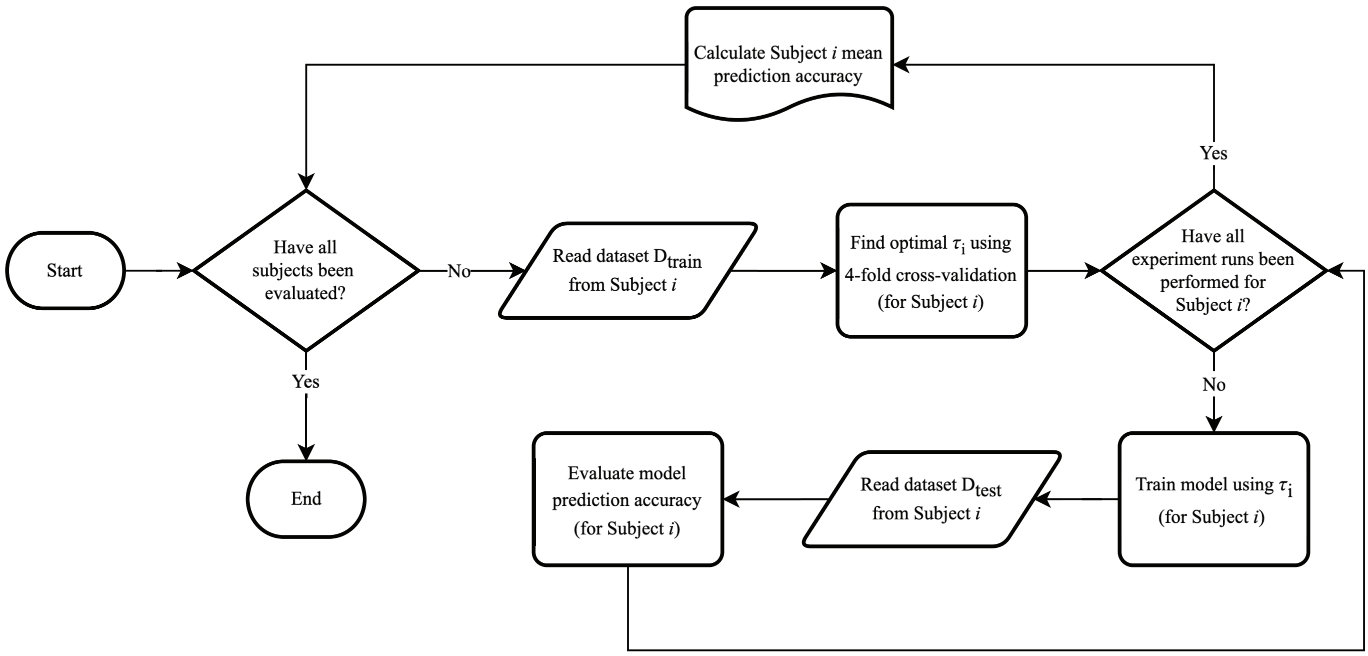


Figure 3: Experiment flowchart for a given classifier.

3.2. COMPARISON OF CLASSIFIERS' PERFORMANCE

To evaluate the effectiveness of the SVM, LR, LDA, *K*NN, and DT classifiers in solving the gesture recognition problem using the datasets described in Section 2.1, we measured the accuracy of five different HGR systems. Each system employed the same algorithms outlined in Section 2 for all stages except the classifier, which varied across the systems. The hyperparameters for each classifier were selected as detailed in Section 2.5.

We followed a standardized approach for each classifier, as illustrated in Figure 3, which included the following steps. Firstly, we used the 4-fold cross-validation method to find the optimal threshold τ for each subject. This process is explained in detail in Section 3.1. Secondly, we trained a separate model for each of the ten subjects using their own dataset D_{train} and evaluated the prediction accuracy [76] (aka test accuracy) of that model using their own dataset D_{test}. The prediction accuracy for a given classifier and subject was calculated according to:

$$\text{Prediction accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}\,.$$

This metric provides the ratio of correct classifications, giving an overall measure of the classifier's performance. Finally, we repeated this procedure 100 times to determine the mean prediction accuracy and its standard deviation for each classifier.

The results for each classifier are presented in Table 2, which includes the results of the experiments conducted by Benalcázar et al. [21][2] and Chaves et al. [51] [3]. These results are also visually represented in Figure 4. The logistic regression classifier demonstrated the best performance, achieving a mean prediction accuracy of 95.2% with zero standard deviation. This is due its convex cost function, which ensures the classifier reaches the same final solution as long as the model converges, as explained in Section 2.5. The mean prediction accuracy and standard deviation by subject are shown in Table 3.

4. DISCUSSION

4.1. STRENGTHS AND LIMITATIONS OF THE DATASET AND BASELINE HGR SYSTEM

Our experiments compared in a fair way the performance of the machine learning classifiers MLP, SVM, LR, LDA, *K*NN, and DT in solving the HGR problem. We achieved this goal by keeping the datasets, data acquisition, segmentation, preprocessing, feature extraction, and post-processing modules unaltered throughout the experiments. By keeping all but the classification stage precisely as prescribed by the authors of the dataset and the baseline HGR system [21], we inherited its strengths and weaknesses.

Strengths: The dataset [21] offers a comprehensive and relevant data collection for analysis and research purposes. The baseline HGR system [21] is a reliable tool that researchers can use as a foundation for designing new HGR systems. The careful selection of features has enabled a wide range of classification algorithms to label hand gestures successfully.

[2] Baseline HGR system using MLP and fixed τ.
[3] HGR system using MLP and optimum individual τ.

Table 1: Optimum individual segmentation thresholds using 4-fold cross-validation*

* Except $\tau_{baseline}$, which was chosen accordingly to Benalcázar et al. [21]

	Subject									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
$\tau_{\mathbf{baseline}}$ [21]	10	10	10	10	10	10	10	10	10	10
$\tau_{\mathbf{MLP}}$ [51]	10	13	15	15	17	10	19	10	13	19
$\tau_{\mathbf{SVM}}$	19	20	12	19	19	19	19	19	19	19
$\tau_{\mathbf{LR}}$	19	20	19	19	19	19	19	19	19	19
$\tau_{\mathbf{LDA}}$	19	20	15	19	17	16	19	19	19	19
$\tau_{\mathbf{KNN}}$	19	20	19	19	14	16	19	19	19	19
$\tau_{\mathbf{DT}}$	18	18	20	16	11	18	14	19	19	20

Table 2: Mean prediction accuracy and standard deviation of the HGR systems using different classifiers

Classifier	**Prediction Accuracy (%)**
Baseline (MLP with fixed τ) [21]	90.1 ± 0.8
Multilayer Perceptron (MLP) [51]	93.6 ± 0.7
Support Vector Machine (SVM)	90.1 ± 0.0
Logistic Regression (LR)	$\mathbf{95.2 \pm 0.0}$
Linear Discriminant Analysis (LDA)	94.3 ± 0.0
K-Nearest Neighbors (KNN)	91.3 ± 0.0
Decision Tree (DT)	84.8 ± 2.8

Table 3: Mean prediction accuracy and standard deviation by subject for different classifiers

Subject	**Classifier**					
	MLP [51]	**SVM**	**LR**	**LDA**	***K*NN**	**DT**
#1	97.8 ± 0.7	97.3 ± 0.0	98.0 ± 0.0	98.0 ± 0.0	97.3 ± 0.0	84.4 ± 8.4
#2	97.2 ± 0.8	98.0 ± 0.0	98.0 ± 0.0	97.3 ± 0.0	98.0 ± 0.0	83.3 ± 7.6
#3	78.1 ± 3.6	66.0 ± 0.0	90.0 ± 0.0	88.0 ± 0.0	76.7 ± 0.0	73.8 ± 7.6
#4	96.0 ± 0.2	94.7 ± 0.0	96.0 ± 0.0	96.0 ± 0.0	95.3 ± 0.0	92.1 ± 6.5
#5	88.5 ± 4.8	70.7 ± 0.0	85.3 ± 0.0	80.7 ± 0.0	71.3 ± 0.0	65.9 ± 13.4
#6	88.0 ± 1.6	80.0 ± 0.0	87.3 ± 0.0	88.0 ± 0.0	84.7 ± 0.0	90.3 ± 7.2
#7	92.3 ± 2.7	95.3 ± 0.0	98.0 ± 0.0	98.0 ± 0.0	94.7 ± 0.0	83.8 ± 9.2
#8	99.9 ± 0.3	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	97.3 ± 0.0	92.2 ± 8.3
#9	98.8 ± 0.5	98.7 ± 0.0	99.3 ± 0.0	97.3 ± 0.0	98.0 ± 0.0	87.1 ± 9.3
#10	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	94.8 ± 3.9
Average	93.6 ± 7.0	90.1 ± 12.9	95.2 ± 5.5	94.3 ± 6.5	91.3 ± 10.1	84.8 ± 9.0

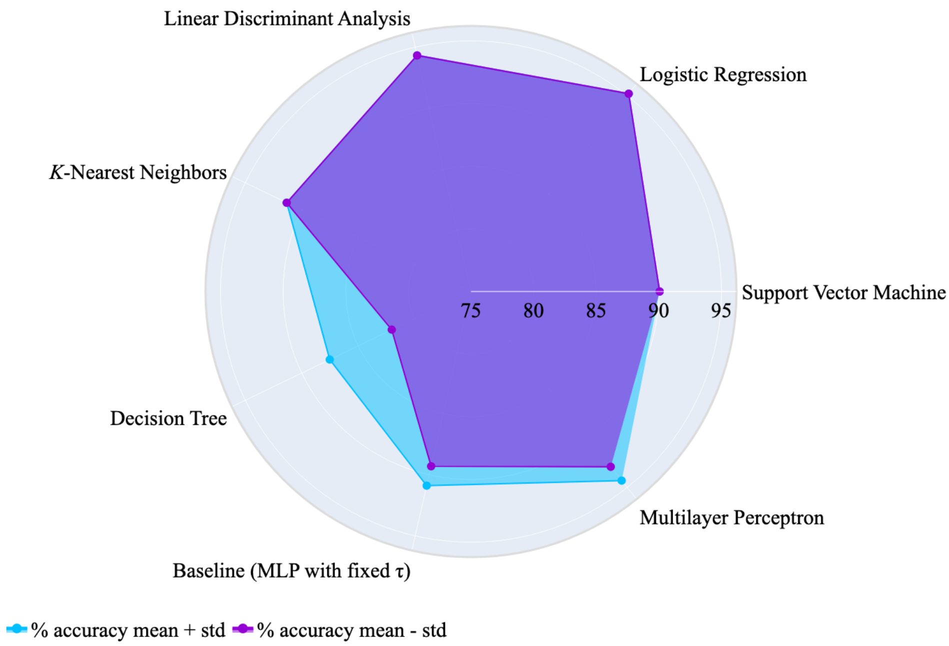


Figure 4: Prediction accuracy and standard deviation of the HGR systems using different classifiers. The blue area indicates the mean prediction accuracy plus one standard deviation for each classifier. The purple area indicates the mean prediction accuracy minus one standard deviation for each classifier. These results are also represented in Table 2.

Limitations: However, the dataset [21] may have limitations or biases that could affect the accuracy of the analysis, and the baseline HGR system [21] may have shortcomings that need to be considered and addressed in further research as we enumerate below.

1. The dataset would need more examples to train a complex classifier, like a deep learning model with convolutional layers.
2. The dataset lacks annotations for the segmentation ground truth, which makes it difficult to measure the prediction accuracy of the segmentation stage.
3. The post-processing step of the baseline HGR system [21] could be improved by adopting other criteria, such as majority voting, as suggested by Englehart et al. [62].
4. According to the literature [27], we expected the MLP to achieve the highest accuracy among the classifiers, which was not the case in our study. It is important to highlight that the training of the baseline HGR system using MLP [21] did not involve fine-tuning hyperparameters such as the regularization factor and the number of epochs. These hyperparameters are crucial for preventing model overfitting, specially given that the number of MLP coefficients exceeds the number of training examples [76]. This imbalance may explain why the MLP achieved lower prediction accuracy compared to the LR: the LR classifier in this system has 42 coefficients, whereas the MLP has 84, with only 30 training examples available. Therefore, by fine-tuning the regularization factor (we used the Euclidean norm of the coefficients as the regularizer function) and/or employing the *early stopping* technique, it might be possible to achieve better results than those reported in this work.

4.2. THRESHOLD OPTIMIZATION AND TIEBREAKING

Although our main objective was to compare classifiers, we also fine-tuned each classifier's segmentation threshold τ using 4-fold cross-validation. We included the result from our previous research [51] in this work because using the same thresholds for all classifiers could lead to biased comparisons. It's important to consider this when evaluating accuracy since some classifiers may be more suitable for specific values of τ than others.

The cross-validation method can be helpful when working with small datasets [28]. However, it presented challenges when choosing the best threshold value (τ) for the segmentation module. Different threshold values resulted in the same cross-validation error during the experiments, creating a tie among possible thresholds. To address this issue, we chose the highest threshold value with the lowest cross-validation error, as described in Section 3.1. If multiple values of τ yielded the same cross-validation error, we selected the maximum among those thresholds. Consequently, the segmentation stage extracted more relevant information about the sEMG signal, resulting in a higher signal-to-noise ratio (SNR) and improving the HGR system's performance. This approach could presumably be more impactful when dealing with very noisy data.

4.3. HYPERPARAMETERS TUNING AND GENERALIZATION

As mentioned in Section 4.1, the baseline HGR system [21] did not use cross-validation to choose the optimum values of hyperparameters, such as the regularization weight. Consequently, classifiers trained in this way are more likely to overfit, i.e., to memorize training data, and thus build models with less generalization capacity.

In our particular set of experiments, the lack of fine-tuning could explain why MLP and SVM achieved poor performance compared to linear classifiers such as LR and LDA: training models with more parameters than the number of examples in the training dataset using arbitrarily chosen hyperparameters could have caused them to suffer from the curse of dimensionality.

If this hypothesis is correct, cross-validation for other important hyperparameters could improve the performance of all classifier models, especially those with greater Vapnik-Chervonenkis dimension [77].

4.4. SUBJECTS WITH LOW ACCURACY SCORES

Upon examining Table 3, which displays the performance of the classifiers sorted by subject, it becomes apparent that the worst performing subjects are subjects $\#3$, $\#5$, and $\#6$. Across almost all classifiers, these subjects exhibit a mean prediction accuracy of less than 90%.

The HGR system misclassified some of these three subjects' gestures for two possible reasons. The first is the measurement noise in their datasets, and the second is that the training trials might not have accurately represented the probabilistic density function of their hand movements in the test trials. Both factors could harm the system's performance as the classifiers are more likely to confuse gestures under such conditions. We can gain insight into this issue by examining the confusion matrix of subject $\#3$ in the test dataset using the LR classifier, as shown in Figure 5. The LR model labeled correctly all *fist* gestures but classified four of the *wave in* movements as *fist* and four of the *wave out* movements as *wave in*. Finally, some of the *double tap* examples were labeled as *fingers spread*, *relax*, and *wave in*.

One way to compare the behavior of different classifiers is to analyze their confusion matrices side by side. Figure 6 displays the confusion matrix for subject $\#3$, as classified by SVM using the test dataset. Four common patterns emerge from comparing those two confusion matrices: 1) the SVM, like the LR, correctly classified all 30 test examples of the *fist* gesture; 2) both models misclassified some of the *wave in* gestures as *fist*; 3) both models misclassified some of the *wave out* gestures as *wave in*; and 4) both models confused some of the *double tap* gestures with *fingers spread*, *relax*, and *wave in*. These similarities suggest that certain gesture patterns, such as *fist*, are easier to learn, while others are more challenging to distinguish in the feature space.

According to our analysis, the sEMG pattern of the *fist* gesture is easier to learn because the signal of interest is usually concentrated in a well-defined time range and has a high magnitude. On the other hand, the sEMG data from gestures like *fingers spread*, *wave in*, and *wave out* are usually distributed over time. Combining this characteristic with measurement noise or poor data recording can result in misclassifications of these gestures. It's worth noting that even the LR classifier achieved a mean prediction accuracy equal to or less than 90% for subjects $\#3$, $\#5$, and $\#6$, as shown in Table 3. Although the LR was the best classifier, achieving a mean prediction accuracy of 95.2%, these three subjects presented a significant discrepancy since other subjects achieved a mean prediction accuracy equal to or greater than 96%, resulting in a standard deviation of 5.5%.

5. CONCLUSIONS AND FUTURE WORK

In this article, we proposed a comparison among different classifiers for HGR applications. Our HGR system comprises six stages: data acquisition, segmentation, preprocessing, feature extraction, classification, and post-processing. To ensure a fair comparison, we altered only the classification stage. This methodology had its advantages and disadvantages. On the one hand, it allowed for a fair comparison among classifiers; on the other hand, the limited training data hindered the use of more complex classifiers. We designed experiments that classified six hand gestures from ten healthy subjects using six supervised learning classifiers, including MLP, SVM, LR, LDA, *K*NN, and DT. We used cross-validation to fine-tune the segmentation hyperparameter. When the cross-validation error is the same for several different values of the segmentation threshold, we proposed choosing the highest value to break ties. We combined these approaches with DTW-based features to achieve good-to-excellent performance for all the classifiers we tested, especially for the logistic regression. The results showed that all classifiers, with the exception of the DT, outperformed the baseline HGR system [21]. The best classifier was the LR, achieving 95.2% of mean prediction accuracy, surpassing the second-best classifier by 0.9%, which was the LDA. It is worth mentioning that we remained faithful to the baseline system [21] by not fine-tuning any of the tested classifiers, which may explain why the MLP and SVM classifiers performed worse than the LR and LDA.

In our future work, we aim to enhance the performance of the HGR system by modifying stages beyond classification and tuning key hyperparameters. For example, the window length parameter, which controls the number of samples the system analyzes during preprocessing, must be chosen carefully to balance performance and system delay. Additionally, we will explore alternative validation methods instead of cross-validation to prevent ties in the search for optimal hyperparameters. Furthermore, we intend to experiment with computationally efficient models, such as Weightless Neural Networks (WNNs). Also known as RAM-based Neural Networks or N-Tuple Classifiers, WNNs differ from models that adjust weights during training. Instead, WNN neurons update their memory contents as they learn, enabling rapid software execution and cost-effective hardware implementations [78–81]. We also plan to evaluate the generalization capability of our approach on other datasets. Extending our evaluation to larger datasets will enable the assessment of our method with different techniques, including deep learning.

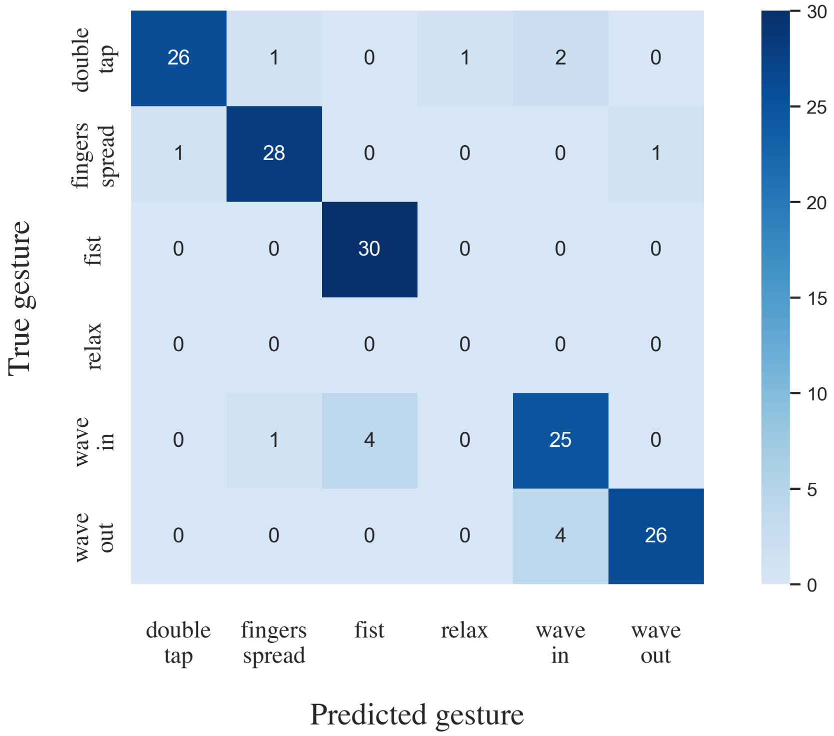


Figure 5: Confusion matrix for subject #3 using LR classifier.

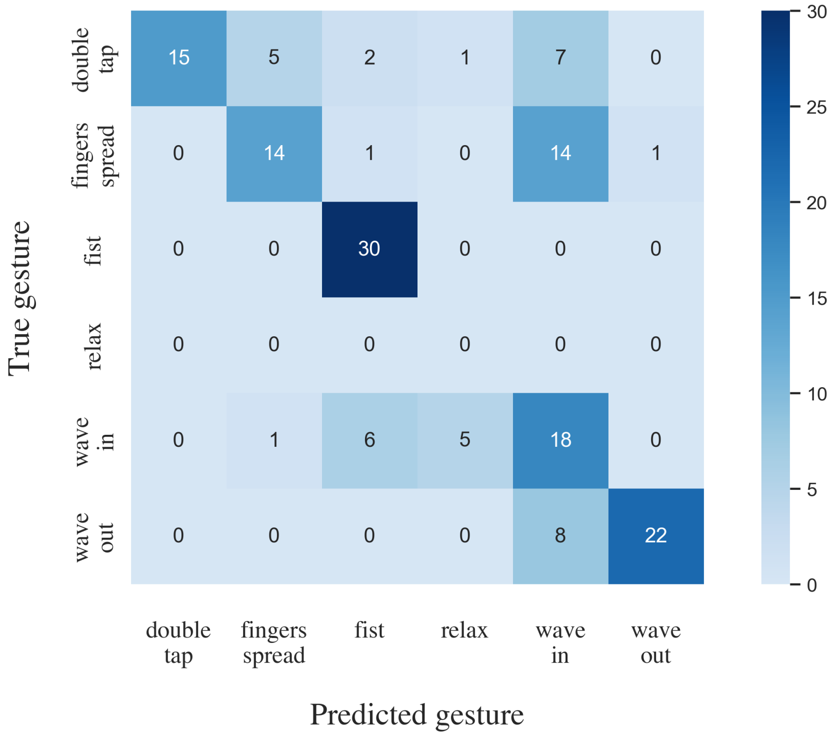


Figure 6: Confusion matrix for subject #3 using SVM classifier.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. It was also partially financed by CNPq and FAPERJ.

REFERENCES

[1] G.-C. Luh, Y.-H. Ma, C.-J. Yen and H.-A. Lin. "Muscle-gesture robot hand control based on sEMG signals with wavelet transform features and neural network classifier". In *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 627–632. IEEE, Jul. 2016.

[2] N. Nasri, S. Orts-Escolano and M. Cazorla. "An sEMG-Controlled 3D Game for Rehabilitation Therapies: Real-Time Time Hand Gesture Recognition Using Deep Learning Techniques". *Sensors*, vol. 20, no. 22, Nov. 2020.

[3] C. L. Toledo-Peral, G. Vega-Martínez, J. A. Mercado-Gutiérrez, G. Rodríguez-Reyes, A. Vera-Hernández, L. Leija-Salas and J. Gutiérrez-Martínez. "Virtual/Augmented Reality for Rehabilitation Applications Using Electromyography as Control/Biofeedback: Systematic Literature Review". *Electronics*, vol. 11, no. 14, Jul. 2022.

[4] J. J. A. Mendes Junior, M. L. B. Freitas, D. P. Campos, F. A. Farinelli, S. L. Stevan and S. F. Pichorim. "Analysis of Influence of Segmentation, Features, and Classification in sEMG Processing: A Case Study of Recognition of Brazilian Sign Language Alphabet". *Sensors*, vol. 20, no. 16, pp. 4359, 2020.

[5] L. F. Brunialti, S. M. Peres, C. A. M. Lima and C. Boscarioli. "Aprendizado por Transferência para Aplicações Orientadas a Usuário: Uma experiência em Língua de Sinais". *Learning and Nonlinear Models*, vol. 11, no. 2, pp. 56–73, 2013.

[6] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl and O. C. Aszmann. "The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797–809, Jul. 2014.

[7] M. Gonzalez-Fernandez. "Amputation: Recovery and Rehabilitation". Available: `https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/amputation/amputation-recovery-and-rehabilitation` (visited on 12/06/2023).

[8] C. Behrend, W. Reizner, J. A. Marchessault and W. C. Hammert. "Update on Advances in Upper Extremity Prosthetics". *The Journal of Hand Surgery*, vol. 36, no. 10, pp. 1711–1717, Oct. 2011.

[9] M. E. Huang, C. E. Levy and J. B. Webster. "Acquired limb deficiencies. 3. Prosthetic components, prescriptions, and indications". *Archives of Physical Medicine and Rehabilitation*, vol. 82, no. 3, pp. S17–S24, Mar. 2001.

[10] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto and H. Müller. "Effect of clinical parameters on the control of myoelectric robotic prosthetic hands". *Journal of Rehabilitation Research and Development*, vol. 53, no. 3, pp. 345–358, 2016.

[11] J. H. Bowker, J. W. Michael and American Academy of Orthopaedic Surgeons, editors. *Atlas of limb prosthetics: surgical, prosthetic, and rehabilitation principles*. Mosby Year Book, second edition, 1992.

[12] M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti and L. Benini. "Robust Real-Time Embedded EMG Recognition Framework Using Temporal Convolutional Networks on a Multicore IoT Processor". *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 244–256, Apr. 2020.

[13] E. Clancy, E. Morin and R. Merletti. "Sampling, noise-reduction and amplitude estimation issues in surface electromyography". *Journal of Electromyography and Kinesiology*, vol. 12, no. 1, pp. 1–16, Feb. 2002.

[14] W. Li, P. Shi and H. Yu. "Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future". *Frontiers in Neuroscience*, vol. 15, Apr. 2021.

[15] D. Farina and O. Aszmann. "Bionic Limbs: Clinical Reality and Academic Promises". *Science Translational Medicine*, vol. 6, no. 257, Oct. 2014.

[16] E. J. Scheme, K. B. Englehart and B. S. Hudgins. "Selective Classification for Improved Robustness of Myoelectric Control Under Nonideal Conditions". *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1698–1705, Jun. 2011.

[17] C. Nissler, N. Mouriki and C. Castellini. "Optical Myography: Detecting Finger Movements by Looking at the Forearm". *Frontiers in Neurorobotics*, vol. 10, Apr. 2016.

[18] S. Amsuess, I. Vujaklija, P. Goebel, A. D. Roche, B. Graimann, O. C. Aszmann and D. Farina. "Context-Dependent Upper Limb Prosthesis Control for Natural and Robust Use". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 7, pp. 744–753, Jul. 2016.

[19] N. Jiang, H. Rehbaum, I. Vujaklija, B. Graimann and D. Farina. "Intuitive, Online, Simultaneous, and Proportional Myoelectric Control Over Two Degrees-of-Freedom in Upper Limb Amputees". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 501–510, May 2014.

[20] A. Jaramillo-Yánez, M. E. Benalcázar and E. Mena-Maldonado. "Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review". *Sensors*, vol. 20, no. 9, Apr. 2020.

[21] M. E. Benalcazar, C. E. Anchundia, J. A. Zea, P. Zambrano, A. G. Jaramillo and M. Segura. "Real-Time Hand Gesture Recognition Based on Artificial Feed-Forward Neural Networks and EMG". In *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1492–1496. IEEE, Sep. 2018.

[22] X. Chen and Z. J. Wang. "Pattern recognition of number gestures based on a wireless surface EMG system". *Biomedical Signal Processing and Control*, vol. 8, no. 2, pp. 184–192, Mar. 2013.

[23] W.-T. Shi, Z.-J. Lyu, S.-T. Tang, T.-L. Chia and C.-Y. Yang. "A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study". *Biocybernetics and Biomedical Engineering*, vol. 38, no. 1, pp. 126–135, 2018.

[24] J. Yang, J. Pan and J. Li. "sEMG-based continuous hand gesture recognition using GMM-HMM and threshold model". In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1509–1514. IEEE, Dec. 2017.

[25] C. Yang, J. Long, M. A. Urbin, Y. Feng, G. Song, J. Weng and Z. Li. "Real-Time Myocontrol of a Human–Computer Interface by Paretic Muscles After Stroke". *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1126–1132, Dec. 2018.

[26] S. Benatti, F. Casamassima, B. Milosevic, E. Farella, P. Schonle, S. Fateh, T. Burger, Q. Huang and L. Benini. "A Versatile Embedded Platform for EMG Acquisition and Gesture Recognition". *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 5, pp. 620–630, Oct. 2015.

[27] A. Sultana, F. Ahmed and M. S. Alam. "A systematic review on surface electromyography-based classification system for identifying hand and finger movements". *Healthcare Analytics*, vol. 3, pp. 100126, 2023.

[28] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006.

[29] K. P. Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012.

[30] S. S. Haykin. *Neural networks and learning machines*. Prentice Hall, third edition, 2009.

[31] M. Yasen and S. Jusoh. "A systematic review on hand gesture recognition techniques, challenges and applications". *PeerJ Computer Science*, vol. 5, pp. e218, 2019.

[32] D. V. Redrovan and D. Kim. "Hand gestures recognition using machine learning for control of multiple quadrotors". In *2018 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6. IEEE, Mar. 2018.

[33] U. Cote Allard, F. Nougarou, C. L. Fall, P. Giguere, C. Gosselin, F. Laviolette and B. Gosselin. "A convolutional neural network for robotic arm guidance using sEMG based frequency-features". In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2464–2470. IEEE, Oct. 2016.

[34] A. M. Moslhi, H. H. Aly and M. ElMessiery. "The Impact of Feature Extraction on Classification Accuracy Examined by Employing a Signal Transformer to Classify Hand Gestures Using Surface Electromyography Signals". *Sensors*, vol. 24, no. 4, pp. 1259, 2024.

[35] K. Englehart, B. Hudgins, P. Parker and M. Stevenson. "Classification of the myoelectric signal using time-frequency based representations". *Medical Engineering & Physics*, vol. 21, no. 6, pp. 431–438, 1999.

[36] M. Oskoei and Huosheng Hu. "Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb". *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1956–1965, 2008.

[37] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A.-G. Mittaz Hager, O. Deriaz, C. Castellini, H. Muller and B. Caputo. "Characterization of a Benchmark Database for Myoelectric Movement Classification". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 73–83, 2015.

[38] J. J. A. Mendes Junior, M. L. Freitas, H. V. Siqueira, A. E. Lazzaretti, S. F. Pichorim and S. L. Stevan. "Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach". *Biomedical Signal Processing and Control*, vol. 59, pp. 101920, 2020.

[39] J. J. A. Mendes Júnior, M. Freitas, H. Siqueira, A. E. Lazzaretti, S. Stevan and S. F. Pichorim. "Comparative analysis among feature selection of sEMG signal for hand gesture classification by armband". *IEEE Latin America Transactions*, vol. 18, no. 6, pp. 1135–1143, 2020.

[40] L. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. Prentice Hall signal processing series. PTR Prentice-Hall, 1993.

[41] M. Müller. *Information retrieval for music and motion*. Springer, 2007.

[42] P. K. Koppolu and K. Chemmangat. "Classification of Hand Gestures with Real Time Muscle Activity Detection for Myoelectric Control of Upper Limb Prosthesis". In *2023 IEEE 20th India Council International Conference (INDICON)*, pp. 963–966. IEEE, 2023.

[43] P. Kumar, A. Phinyomark and E. Scheme. "Verification-Based Design of a Robust EMG Wake Word". In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 638–642. IEEE, 2021.

[44] M. Jabbari, R. N. Khushaba and K. Nazarpour. "Combined Dynamic Time Warping and Spatiotemporal Attention for Myoelectric Control". In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 5940–5943. IEEE, 2021.

[45] O. S. Powar and K. Chemmangat. "Reducing the effect of wrist variation on pattern recognition of Myoelectric Hand Prostheses Control through Dynamic Time Warping". *Biomedical Signal Processing and Control*, vol. 55, pp. 101626, 2020.

[46] O. S. Powar and K. Chemmangat. "Dynamic time warping for reducing the effect of force variation on myoelectric control of hand prostheses". *Journal of Electromyography and Kinesiology*, vol. 48, pp. 152–160, 2019.

[47] A. B. H. Amor, O. E. Ghoul and M. Jemni. "Sign language handshape recognition using Myo Armband". In *2019 7th International conference on ICT & Accessibility (ICTA)*, pp. 1–5. IEEE, 2019.

[48] M. E. Benalcazar, A. G. Jaramillo, Jonathan, A. Zea, A. Paez and V. H. Andaluz. "Hand gesture recognition using machine learning and the Myo armband". In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 1040–1044. IEEE, 2017.

[49] M. E. Benalcazar, C. Motoche, J. A. Zea, A. G. Jaramillo, C. E. Anchundia, P. Zambrano, M. Segura, F. Benalcazar Palacios and M. Perez. "Real-time hand gesture recognition using the Myo armband and muscle activity detection". In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6. IEEE, 2017.

[50] M. AbdelMaseeh, T.-W. Chen and D. W. Stashuk. "Extraction and Classification of Multichannel Electromyographic Activation Trajectories for Hand Movement Recognition". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 6, pp. 662–673, 2016.

[51] G. S. Chaves, A. S. Vieira and M. V. S. Lima. "Hand Gesture Classification using sEMG Data: Combining Gesture Detection and Cross-Validation". In *Anais do XVI Congresso Brasileiro de Inteligência Computacional*, pp. 1–8. SBIC, 2023.

[52] G. C. De Lello, G. S. Chaves, J. F. Caldeira and M. V. S. Lima. "Source code repository of the HGR Lab project". Available: `https://github.com/gclello/hgrlab/` (visited on 04/02/2024).

[53] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar and A. Mohammadi. "FS-HGR: Few-Shot Learning for Hand Gesture Recognition via Electromyography". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1004–1015, 2021.

[54] P. Visconti, F. Gaetani, G. Zappatore and P. Primiceri. "Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses". *International Journal on Smart Sensing and Intelligent Systems*, vol. 11, no. 1, pp. 1–25, 2018.

[55] Z. Zhang, Q. Shen and Y. Wang. "Electromyographic hand gesture recognition using convolutional neural network with multi-attention". *Biomedical Signal Processing and Control*, vol. 91, pp. 105935, 2024.

[56] H. A. Javaid, M. I. Tiwana, A. Alsanad, J. Iqbal, M. T. Riaz, S. Ahmad and F. A. Almisned. "Classification of Hand Movements Using MYO Armband on an Embedded Platform". *Electronics*, vol. 10, no. 11, pp. 1322, 2021.

[57] J. Fan, M. Jiang, C. Lin, G. Li, J. Fiaidhi, C. Ma and W. Wu. "Improving sEMG-based motion intention recognition for upper-limb amputees using transfer learning". *Neural Computing and Applications*, vol. 35, no. 22, pp. 16101–16111, Jul. 2021.

[58] S. Thusneyapan and G. Zahalak. "A practical electrode-array myoprocessor for surface electromyography". *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 2, pp. 295–299, Feb. 1989.

[59] V. T. Inman, H. Ralston, J. De C.M. Saunders, M. Bertram Feinstein and E. W. Wright. "Relation of human electromyogram to muscular tension". *Electroencephalography and Clinical Neurophysiology*, vol. 4, no. 2, pp. 187–194, May 1952.

[60] J. G. Kreifeldt. "Signal Versus Noise Characteristics of Filtered EMG Used as a Control Source". *IEEE Transactions on Biomedical Engineering*, vol. BME-18, no. 1, pp. 16–22, Jan. 1971.

[61] Y. St-Amant, D. Rancourt and E. Clancy. "Effect of smoothing window length on RMS EMG amplitude estimates". In *Proceedings of the IEEE 22nd Annual Northeast Bioengineering Conference*, pp. 93–94. IEEE, 1996.

[62] K. Englehart and B. Hudgins. "A robust, real-time control scheme for multifunction myoelectric control". *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, 2003.

[63] M. Asghari Oskoei and H. Hu. "Myoelectric control systems—A survey". *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.

[64] K. Englehart, B. Hudgin and P. Parker. "A wavelet-based continuous classification scheme for multifunction myoelectric control". *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.

[65] P. S. R. Diniz, E. A. B. Da Silva and S. L. Netto. *Digital signal processing: system analysis and design*. Cambridge University Press, second edition, 2010.

[66] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing*. Pearson, third edition, 2010.

[67] S. K. Mitra. *Digital signal processing: a computer-based approach*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 1998.

[68] M. H. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.

[69] B. Hudgins, P. Parker and R. Scott. "A new strategy for multifunction myoelectric control". *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, Jan. 1993.

[70] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto and H. Müller. "Electromyography data for non-invasive naturally-controlled robotic hand prostheses". *Scientific Data*, vol. 1, no. 1, Dec. 2014.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[72] G. Purushothaman and K. Ray. "EMG based man–machine interaction—A pattern recognition research platform". *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 864–870, 2014.

[73] R. O. Duda, P. E. Hart and D. G. Stork. *Pattern classification*. Wiley, second edition, 2001.

[74] C.-C. Chang and C.-J. Lin. "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[75] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin. "LIBLINEAR: A library for large linear classification". *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008. Software available at `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

[76] G. James, D. Witten, T. Hastie and R. Tibshirani. *An introduction to statistical learning: with applications in R*. Springer texts in statistics. Springer, second edition edition, 2021.

[77] Y. S. Abu-Mostafa, M. Magdon-Ismail and H.-T. Lin. *Learning from data: a short course*. AMLbook.com, 2012.

[78] W. W. Bledsoe and I. Browning. "Pattern recognition and reading by machine". In *Proceedings of the Eastern Joint Computer Conference*, IRE-AIEE-ACM 1959, pp. 225–232. Association for Computing Machinery, Dec. 1959.

[79] I. Aleksander and H. Morton. *An introduction to neural computing*. Chapman & Hall, 1990.

[80] I. Aleksander, M. D. Gregorio, F. M. G. França, P. M. V. Lima and H. Morton. "A brief introduction to Weightless Neural Systems". In *17th European Symposium on Artificial Neural Networks (ESANN)*, pp. 299–305, Apr. 2009.

[81] R. M. Haralick and A. C. Yuksel. "The N-Tuple Subspace Classifier: Extensions and Survey". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 22–39, 2021.