Technical University of Denmark

Special course - NeuroRobotics Technology Lab
DTU

Design and neural control of a robotic prosthetic hand through electromyography analysis

Students :
Emmie Cluzel - s233243

Supervisor(s) :
Silvia Tolu
Júlia Rey Vilches

May 29, 2024

Contents

1 Introduction **4**
- 1.1 Context . . . 4
- 1.2 Objectives . . . 4
- 1.3 Resources available . . . 6

2 State-of-the-Art **7**
- 2.1 Physiology of hand movements : Muscles and mechanism . . . 7
- 2.2 Electromyography data . . . 8
- 2.3 Filtering, segmentation and features extraction . . . 9
- 2.4 Classification works and accuracy . . . 10
  - 2.4.1 The ANN classifier . . . 11
  - 2.4.2 The k-NN classifier . . . 11
  - 2.4.3 Bagging and boosting algorithms . . . 11

3 EMG data collection and signal pre-processing **12**
- 3.1 Protocols for EMG data collection . . . 12
- 3.2 Filtering of the EMG data and envelope detection . . . 12
- 3.3 Signal segmentation . . . 14

4 PCA and classification methods **16**
- 4.1 Features extraction . . . 16
- 4.2 Principal Components Analysis (PCA) for dimension reduction . . . 16
- 4.3 Finger movements classification with EMG signals . . . 19
  - 4.3.1 The ANN classifier . . . 19
  - 4.3.2 The k-NN classifier . . . 19
  - 4.3.3 Bagging and boosting algorithms . . . 19

5 Classification results and discussions **20**
- 5.1 Classification criteria . . . 20
- 5.2 Classification results for seven movements classification . . . 21
- 5.3 Evaluation of the classifier on a new succession of movement as input . . . 23
- 5.4 Results discussion . . . 24

6 Robotic hand control **25**

7 Conclusion and further prospects **26**
- 7.1 Conclusion . . . 26
- 7.2 Improvement and future work . . . 26

Nomenclature **27**

References **28**

List of figures **29**

List of tables **30**

A Appendix **31**
A.1 Materials and project management 31
A.2 Pre-processing results 32
A.3 PCA results 33
A.4 Classifiers and classification results 35
A.4.1 Classifier performances 35
A.4.2 Classifiers 36
A.4.3 Results 36

Abstract

The aim of this study is to classify movements using EMG recordings and to control a robotic hand via Arduino in response. Seven movements are analyzed, and the signals undergo the traditional steps of filtration, segmentation, feature extraction, and Principal Components Analysis (PCA) before being integrated into a classifier. Four different types of classifiers were tested, and their results are discussed. The best testing accuracy, 93%, was achieved using a Random Forest classifier with four principal components. To verify the generalizability of these results, the same process was repeated with a second subject. It was found that while the pre-processing steps are generalizable, the PCA and classifier are specific to each individual.

The Python and Arduino codes for this project are available on Github : Special Course Repository

1 Introduction

1.1 Context

The convergence of robotic technology and electromyography (EMG) hand movement recognition is revolutionizing multiple domains, from enhancing surgical precision to empowering amputees with regained mobility. Robotic hands, guided by intricate EMG signals, are setting new standards in robotic assistance across diverse fields such as surgery and industry, offering unprecedented dexterity and reliability. Moreover, these advancements are transformative for amputees, providing sophisticated prosthetic solutions that closely mimic natural hand movements. In the realm of rehabilitation, EMG-driven robotic hands are becoming invaluable, aiding patients in recovering motor functions with greater efficiency and effectiveness. This synergy between robotics and EMG improves quality of life and functional independence for many.

Figure 1: Bionic Hands: Empowering Amputees to Reclaim Mobility [1].

Surface Electromyography (EMG) signals, captured via surface electrodes, offer a window into the intricate activity of skeletal muscles. By collecting EMG data from forearm muscles, researchers can analyze and classify these signals to discern the user's intended movements. This sophisticated process allows algorithms to interpret muscle activity patterns and control robotic arms accordingly. As a result, EMG technology enables intuitive and responsive robotic assistance, seamlessly translating user intentions into precise robotic actions.

1.2 Objectives

The objective of this study is to control a robotic prosthetic hand through EMG analysis using only one single-channel EMG sensor. For this purpose, EMG sensors are placed on the user's forearm when wearing the Paragit sleeve (Figure 4). Seven specific movements are studied: hand opening, hand closing, thumb folding, index folding, middle finger folding, ring finger folding, and little finger folding (Figure 2). According to the literature, an accuracy ranging from 85% to 95% can be expected, depending on the number of sensors used.

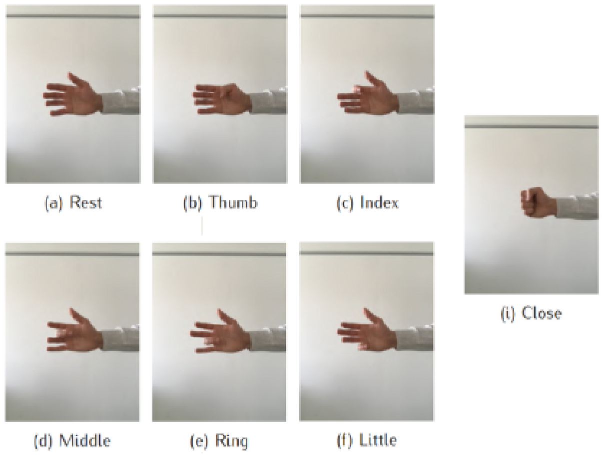


Figure 2: Seven movements represented the seven classes of the classification study.

The main steps of the project, as illustrated in Figure 3, include : collecting EMG data using a single sensor embedded in the Paragit sleeve, filtering the data, extracting relevant information through segmentation and feature extraction, ranking the features, applying a classifier to identify which signals correspond to specific movements, and finally, controlling a pre-built robotic hand via Arduino.
The project management timeline, depicted in the GANTT chart in Appendix Figure 17, provides a detailed overview of the project's schedule and milestones.

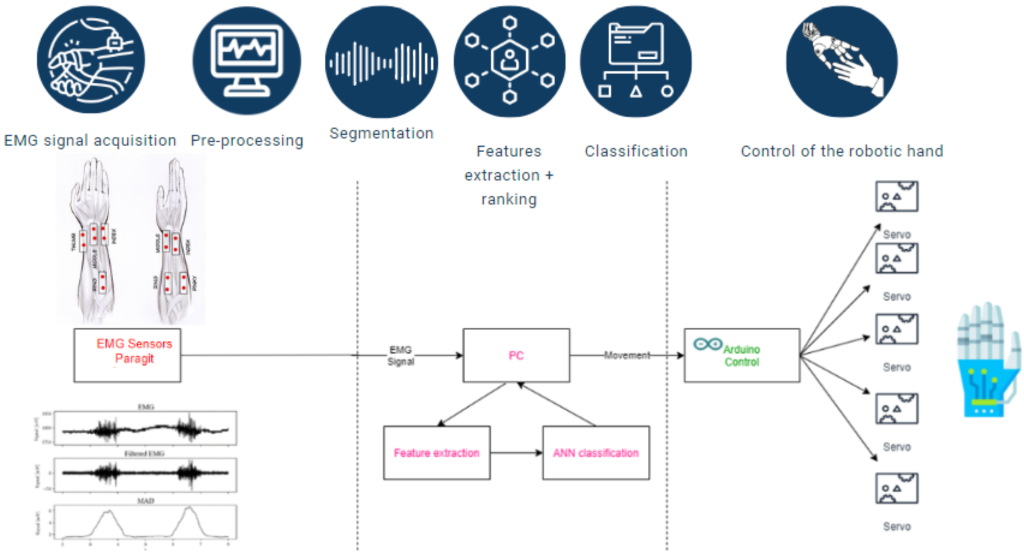


Figure 3: Main steps for hand movements classification : from data collection to robotic control.

To record the EMG signals, EMG sensors are positioned on the user's forearm when wearing the Paragit sleeve (Figure 4). In this project, only one Paragit sensor is utilized. Using one sensor simplifies the setup and reduces the complexity of the system. Moreover, it is less expensive compared to using multiple sensors, making it more accessible and affordable. Finally, a single sensor setup is more compact, which enhances user comfort and portability.

The main Challenges associated with the project include :

- Collecting high-quality signals from optimal forearm locations using only one single-channel EMG sensor.
- Efficiently segmenting the signal to detect events (movements) and identify corresponding features.
- Extracting and selecting the most relevant features to classify finger movements with an acceptable error rate.

1.3 Resources available

For this project, the materials available are :

- An already-built robotic hand equipped with 5 servo motors, previously used for an opening/closing study (Figure 4) [2].
- The Paragit sleeve [3], discussed in the previous section (Figure 4), which contains the EMG sensors and the monitoring device connected to them.
- Arduino board, breadboard, and wires.
- Various non-physical resources, such as research papers, scientific articles, discussion groups, and collaborative development platforms like Kaggle [4] [5] and Github [6].

(a) (b)

Figure 4: Already built robotic hand available from the NRT laboratory at DTU (b). Paragit sleeve composed of 1 EMG sensor to collect EMG data from forearm muscles (a).

2 State-of-the-Art

2.1 Physiology of hand movements : Muscles and mechanism

To design an effective prosthetic or robotic hand, it is essential to understand the anatomy of the forearm and hand [7] and identify the key elements involved in hand movements. First and foremost, five main nerves can be observed in the forearm (Figure 5) [8] :

- Ulnar nerve : when you hit your "funny bone," causing numbness and tingling, you are actually hitting the ulnar nerve. The ulnar nerve powers forearm muscles that bend the small and ring fingers and controls most small hand muscles.
- Radial nerve : the radial nerve powers muscles that straighten the wrist and all fingers.
- Median nerve : the median nerve controls most flexor and pronator muscles in the forearm such as the ones moving the thumb, index, middle, and half of the ring finger.
- Musculocutaneous nerve : the musculocutaneous nerve controls the coracobrachialis, biceps, and brachialis muscles, which bend the elbow and move the arm.
- Axillary nerve : the axillary nerve allows arm lifting and external rotation.

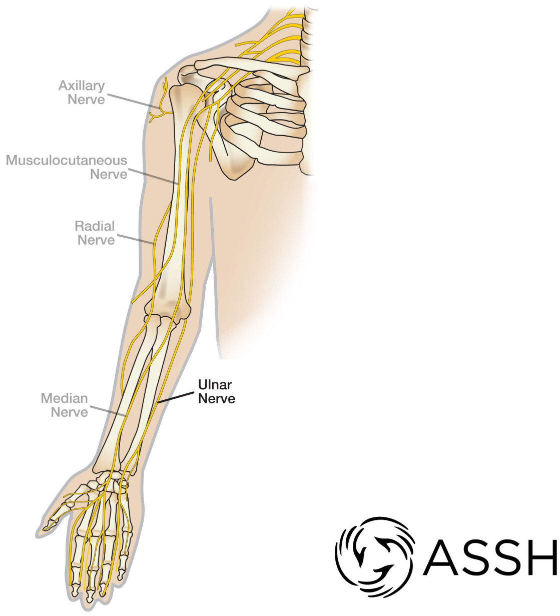


Figure 5: Main forearm peripheral nerves [8].

Moreover, the arm muscles are organized into four layers, with the superficial layer being the most visible and primarily responsible for the larger movements of the arm and forearm. The intermediate layer focuses on wrist and finer movements, the deep layer controls intricate finger and thumb actions, and the very deep layer is essential for fine motor control and dexterity [9] [10].

In this project, the robotic hand was previously built by another student with a focus on these aspects. The rigid parts of the hand were created using 3D printing. The joints were designed using a pivot mechanism coupled with a spring to mimic the role of the extension tendons. Five servo motors and cables are used to represent the abduction muscles responsible for finger movement.

Each finger's movement is controlled by various muscles, making precise control a complex task. Recording the activity of forearm muscles to identify specific movements is challenging because the signals from different muscles can interfere with each other. For example, for the flexion and extension, muscles like the FDS and ED control the bending and straightening of the fingers.

2.2 Electromyography data

EMG and especially sEMG detect activity of superfical muscles. The muscles involved in controlling finger and hand movements and commonly targeted for surface electromyography (sEMG) recording are primarily located in the superficial and intermediate layers of the forearm [11] :

- Superficial Layer of Forearm Muscles : Flexor Carpi Radialis (FCR) and Flexor Carpi Ulnaris (FCU) flexe and abduct the wrist. Palmaris Longus helps in flexing the wrist and increases the strength of grip by tightening the fascia in the palm. Extensor Digitorum (ED) that extends the fingers and the wrist.
- Main Muscles in the Intermediate Layer: Flexor Digitorum Superficialis (FDS), that flexes the middle phalanges of the fingers and play a crucial role in controlling the fingers independently. It enables activities such as writing, typing, and fine motor skills.

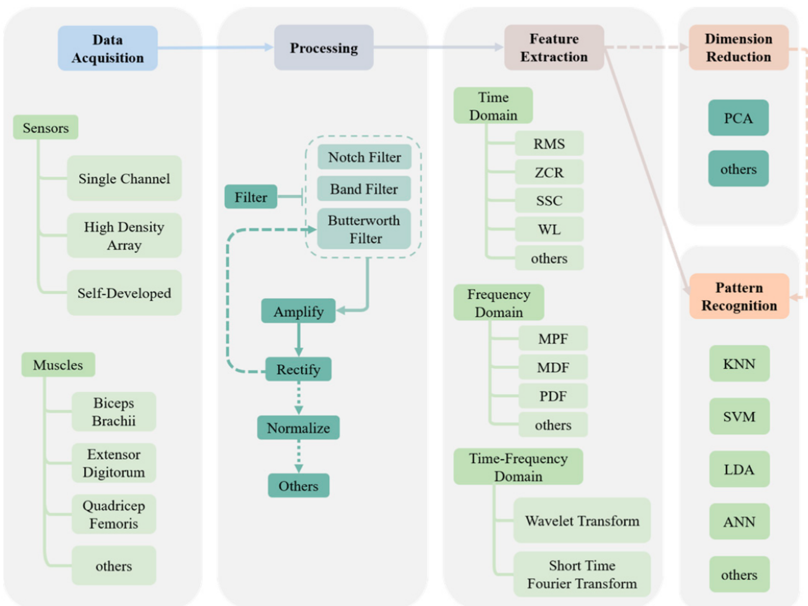


Figure 6: Processing and recognition methods of sEMG signals [11].

By focusing on these muscles within their respective layers, surface EMG can more effectively capture the electrical activity needed to characterize specific hand and finger movements, even with potential signal interference from overlapping muscle activities.
It is crucial to maintain the same sensor position in different trials. However, a slight deviation of the electrode will lead to a significant impact on the experimental results, but maintaining the exact location in every test is challenging.

Various sensors can be used for EMG monitoring, each offering unique advantages (Figure 6). Single-channel sensors are simple and cost-effective, providing basic muscle activity data. High-density arrays consist of multiple electrodes, offering detailed spatial information about muscle activity patterns. Other EMG monitoring methods include self-developed sensors, multi-channel systems, which record from multiple muscle sites simultaneously, and wireless sensors, which offer greater mobility and convenience for real-world applications.

In literature, 4 to 8 EMG sensors are most of the time used in order to achieve good accuracy for 4 to 7 movement classification. However, some researches accept the challenge to try it with only 2 sensors [12]. In this paper from Mohd Haris, B. Venkata Rao and Pavan Chakraborty, 2 EMG sensors are used and localised on the brachioradialis and flexor Cuprum ulnaris muscles. Nine movements are recorded with 6 repetitions of each movement. After applying ANN and k-NN (with 6 neighbors) classifiers, classification accuracies of approximatively 86% and 93% has been achieved for KNN and ANN respectively.

2.3 Filtering, segmentation and features extraction

For every signal, traditional filtering methods must be applied. These include removing the mean of the signal, eliminating the 50 Hz line noise from surrounding electrical devices, and reducing high-frequency noise. Rectifying the signal (taking the absolute value) is also common to facilitate analysis. Advanced filtration methods can then be used depending on the signal characteristics and the information to be extracted. One such advanced operation is envelope extraction, which identifies and extracts the slowly varying amplitude envelope of a signal, representing the overall shape or outline of the waveform's amplitude variations.

Then, segmentation is a crucial step in processing and analyzing EMG signals. It allows for the precise division of the continuous signal into meaningful segments that correspond to different phases of muscle activity. This step is essential because it significantly improves the accuracy and reliability of subsequent analysis and classification. Segmentation can be achieved through various methods, such as event-based segmentation, peak detection, time window segmentation, and more advanced techniques like the Teager Kaiser Energy Operator (TKEO) [13].

Both time domain and frequency domain features are popular in the literature, but time domain features are more favored. They offer good discrimination power and are less computationally intensive compared to frequency domain features [12]. The main relevant time-domain features include 2.3:

- Mean : average value of the signal. This feature is useful for understanding the overall level of muscle activation.
- Standard Deviation : amount of variation or dispersion in the signal.
- Variance : key indicator of the signal's spread and is useful for understanding the overall energy or intensity of the muscle activity.
- Skewness : measures the asymmetry of the signal's distribution.
- Kurtosis : indicates the "tailedness" of the signal's distribution, showing how much of the signal's variance is due to outliers. High kurtosis means more outliers, suggesting bursts of high muscle activity.
- Root Mean Square (RMS) : measure of the signal's magnitude, reflecting the power of the muscle contractions.
- Waveform Length : cumulative length of the waveform over the segment, indicating the complexity and variability of the signal.
- Mean Absolute Value (MAV) : average of the absolute values of the signal.
- Maximum Value (Max Value) :identifies the peak level of muscle activation, which can be important for detecting the strongest contractions.
- Zero Crossings : number of times the signal crosses the zero amplitude line, which indicates the frequency content of the signal.
- Slope Sign Changes : number of times the slope of the signal changes sign, indicating the variability and complexity of the signal.
- Activity : signal's energy, represented by the variance.
- Complexity : indicates how the frequency content of the signal changes over time, reflecting variations in the signal's structure.

2.4 Classification works and accuracy

In the context of hand movement classification, a classifier is a computational tool that takes input data related to hand movements, such as EMG signals, and assigns them to different categories representing specific hand gestures. The classifier learns patterns from training data, which includes examples of different hand movements along with their corresponding labels. Once trained, the classifier can then predict the movement associated with new input data, enabling the automated recognition and categorization of hand movements.
Numerous resources are accessible online, including platforms like Kaggle [14] and Github [6], offering a plethora of codes and implementations that enable the efficient utilization of classifiers.

The artificial neural network (ANN) is very popular in literature to classify the EMG signal as it has the good learning capability. However, other classification techniques such as k-nearest neighbor and SVM are also popular for EMG signal recognition. The focus will be on the ANN, k-NN classifiers and some boosting and bagging classifiers.

2.4.1 The ANN classifier

ANN are computational models inspired by the human brain's neural networks. They consist of layers of interconnected nodes (neurons), where each node processes input data and passes the results to the next layer. The primary strength of ANNs lies in their ability to learn complex, non-linear relationships through a process called training, which involves adjusting the weights of the connections based on the input data and corresponding output labels.

2.4.2 The k-NN classifier

k-Nearest Neighbors (k-NN) is a simple, non-parametric classification algorithm used in machine learning. It classifies a data point based on the majority class of its k nearest neighbors in the feature space, making decisions based on the closest training examples.

2.4.3 Bagging and boosting algorithms

Bagging and boosting are ensemble learning techniques that combine multiple models to improve classification performance [15]. Traditionally, bagging reduces variance by averaging results, while boosting reduces bias by focusing on the hardest cases.
Bagging involves training multiple instances of the same model on different subsets of the training data, created through random sampling with replacement. The final prediction is typically obtained by averaging the predictions of all models by majority voting. This approach helps prevent overfitting, making the model more robust.
Boosting, on the other hand, focuses on sequentially building models where each new model attempts to correct the errors of the previous ones. It assigns higher weights to misclassified instances, forcing subsequent models to pay more attention to these difficult cases. The final prediction is a weighted sum of the predictions from all models. Boosting may be prone to overfitting if not properly regulated.

3 EMG data collection and signal pre-processing

3.1 Protocols for EMG data collection

The Paragit sleeve, typically used for Parkinson's disease monitoring, collects EMG data, temperature, and IMU data to characterize Parkinson's symptoms. The sleeve should be worn as shown in Figure 4, with the cross positioned on the inner elbow to detect data from the primary hand muscle. The sensor is mainly positioned over the superficial Palmaris Longus muscle [16]. This setup consists of a single-channel sensor with three electrodes: one reference, one electrode, and one counter electrode.
It is important to note that collecting muscle signals for hand movement using only one sensor is particularly challenging. Traditionally, multiple electrodes are placed on the subject's forearm to detect signals from different muscles corresponding to specific movements. With only one sensor, this flexibility is not possible, and only one zone of the forearm is primarily targeted.

Once the sleeve is put on and the monitoring device is inserted, the subject proceeds with seven measurements, each representing 30 repetitions of different movements: hand opening, hand closing, and individual finger movements. This results in seven classes of movement to classify. The data can then be transferred from the device to the computer via a USB cable.
The Paragit device provides three .dat files (int16 format) containing EMG data, temperature data, and IMU data, along with one CSV metadata file detailing the files, including the sampling frequency (2504.495361 Hz). For this project, only the EMG.dat file is utilized.

Data were collected from two subjects, one male and one female, to compare results and determine if the pre-processing and classification methods effective for one subject would also be effective for the other.

Additionally, a new sequence consisting of the seven movements performed consecutively (as shown in Figure 2), repeated twice, is created to simulate real-life usage by a patient.

3.2 Filtering of the EMG data and envelope detection

The EMG.dat file is decoded to extract the EMG signals, which are inherently sensitive and noisy due to various factors such as electrical interference and muscle movement artifacts. Therefore, these signals undergo several common and necessary filtering operations using Paragit parameters specific to this monitoring device and EMG analysis :

- Removing the DC component : The mean is removed from the signal to eliminate the direct current (DC) offset, which is crucial because it can skew the analysis and interpretation of the EMG data.
- Removing 50Hz line noise : A Butterworth bandstop filter is applied to remove the 50Hz line noise caused by surrounding electrical devices. This step is important to reduce interference and ensure the signal's integrity.

- Removing high frequency noise: To further reduce noise, a low-pass Butterworth filter with a cutoff frequency of 700Hz is used. The cutoff frequency is determined by dividing the desired cutoff frequency by the Nyquist frequency, which helps in eliminating unwanted high-frequency components from the signal.

The signal after pre-processing are more clear, the noise-to-signal ratio is lower and the movements executed are easily visible (Figure 7). The same filtering operations have been done on the 7-movements succession (Appendix Figure 18).

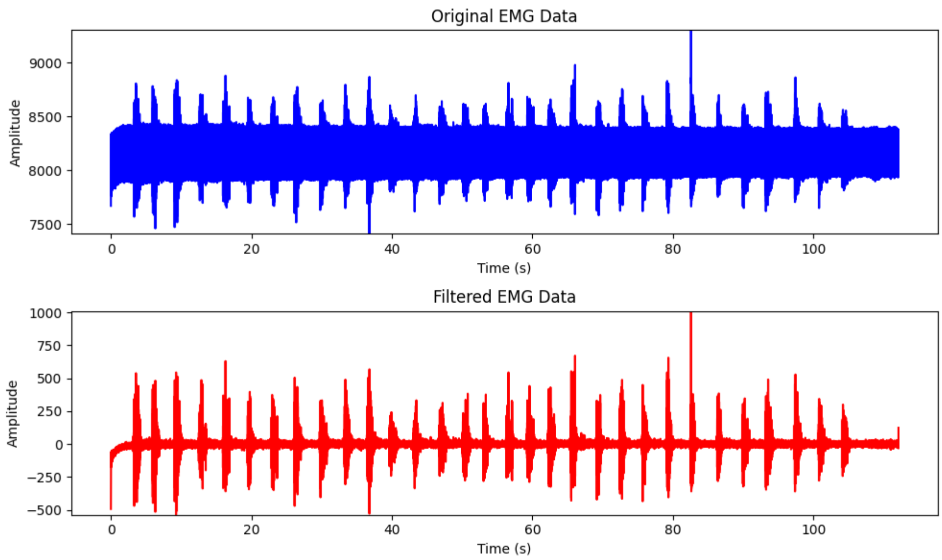


Figure 7: EMG signal collected from the Paragit device before (top) and after (down) filtering steps (30 repetitions of the closing movement).

After filtering, the next step is envelope detection of the signal to identify and extract the slowly varying amplitude envelope of a signal, representing the overall shape or outline of the waveform's amplitude variations. This is particularly useful for detecting events, such as movements, within the EMG signal.
The process begins with applying a new band-pass filter to the signal. Next, the signal is rectified by taking its absolute value. Finally, a Butterworth low-pass filter is applied with a low pass parameter set to 0.5, which is then normalized by dividing by half the sampling frequency. This results in a smoother and more easily interpretable signal. As a result, the 30 movements are now clearly visible and distinct in the signal, as shown in Figure 8.
At this stage, it's already evident that distinguishing the closing signal from the thumb movement signal is feasible (refer to Appendix Figure 19). This distinction arises from the lower magnitude of the signal associated with thumb activation, likely due to its weaker intensity.
However, certain "problematic" observations also emerge. Primarily, it's apparent that even for the same movement, the magnitude can vary significantly based on the subject's exertion level. This variability introduces complexity into the classification process.

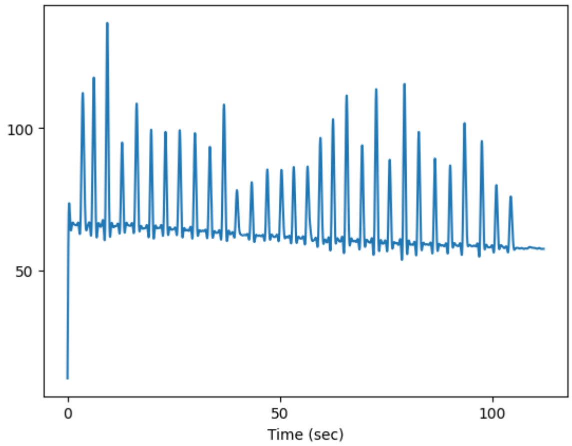


Figure 8: Envelop EMG signal from the filtered signal. On this picture, the distinct movements executed by the subject are easily visible.

3.3 Signal segmentation

After completing envelope detection, the signal needs to be segmented. This segmentation aids in identifying patterns, detecting events, or extracting relevant features from each segment for more detailed and accurate signal analysis. In this study, four different segmentation methods were explored :

– Peak Detection Segmentation : Peak detection segmentation identifies and segments the signal around local maxima, which represent significant events. For this method, peak detection based on threshold or on the prominence of the signal [17]. However, for this method, the time window was symmetric and fixed while an EMG event can sometimes be asymmetric or longer/shorter depending on the movement length. Thus, this method was not adapted for the EMG segmentation wished.

– Teager Kaiser Energy Operator (TKEO) : as explained in the 2 section, this advanced method is very interesting for event based segmentation but the results obtained with the signals from this study were not consistent.

– Window-Based Segmentation : Window-based segmentation divides the signal into fixed-size time windows, which can overlap or not. It is useful for steady-state signals and time-series analysis. It is typically frequently used for EMG analysis. This method applied to our signal is observable in Appendix figure 20.

– Event-Based Segmentation Event-based segmentation divides the signal based on detected events, such as sudden amplitude changes as visible on Figure 9. Here the threshold used was determined empirically as *threshold=(2*mean+1*minimum)/3*, with the mean and the signal minimum defined on a rolling time window in order to fit as best as possible the signal variation.

After testing these two segmentation methods with various classifiers, it became evident that event-based segmentation yields higher testing accuracy. Additionally, this segmentation method simplifies the labeling process, as each segment corresponds to a specific movement. Conversely, assigning accurate labels to time window segments would have been more challenging. Therefore, event-based segmentation will be retained for the classification study.

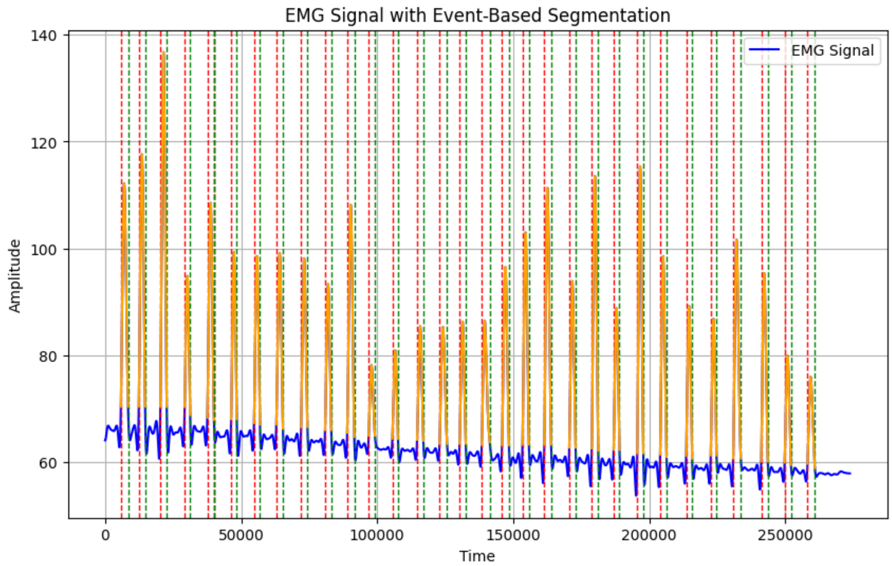


Figure 9: Event-based segmentation for the closing movement signal

The same segmentation method has been applied to the 7 movements succession dataset (Figure 10). On this graph, the problem underlined previously is fully realised : the movements 9-10 and 11-12 are supposed to be the major and ring fingers, those signal being usually smaller such as the thumb one. However, on this figure those signals have a higher amplitude than the opening and closing of the hand, probably because the subject didn't insisted a lot on the closing and opening movements but a lot on the major and ring finger ones. The classifier that will be trained to classify those movements will have to deal with this kind of complexity.

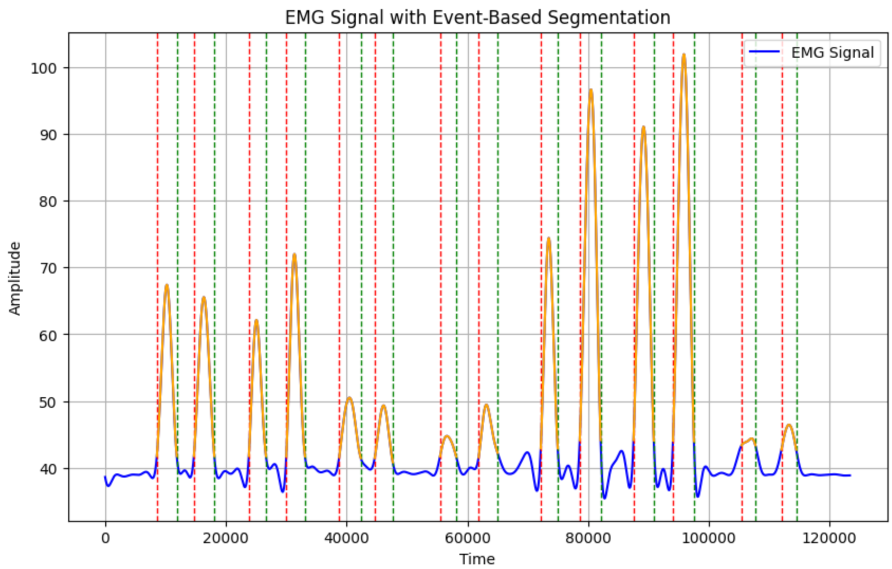


Figure 10: Event-based segmentation for the 7-movements succession signal

4 PCA and classification methods

When the pre-processing and segmentation are complete, the segmented signal is ready for the next steps: feature extraction, ranking, and classification analysis. Initial classification results using raw EMG signals and features extraction were mostly unsatisfactory, with test accuracy falling below 50%.

4.1 Features extraction

First, features need to be extracted from the segmented EMG signal. Extracting features is essential to capture the relevant information that characterizes muscle activity. These features are crucial for accurate classification, analysis, and interpretation of the signal in applications like prosthetic control or medical diagnostics.

For each segment, the thirteen features described in section 2.3 are extracted. It is important to choose the correct features to train the classifier because using too many or irrelevant features can lead to overfitting, increased computational complexity, and reduced generalization ability of the model. Conversely, selecting the most informative and discriminative features enhances the model's ability to capture essential patterns and variations in the signal, leading to more accurate and reliable outcomes.

4.2 Principal Components Analysis (PCA) for dimension reduction

Principal Component Analysis (PCA) serves as a powerful tool in classification studies for several reasons, including dimensionality reduction, feature extraction, and insights into the explained variance and feature contributions within different principal components.
In many classification problems, datasets often contain a large number of features, leading to increased computational complexity and the risk of overfitting. PCA addresses this issue by transforming the original feature space into a new coordinate system defined by principal components. These components, ordered by the amount of variance they explain, capture the most significant variability in the data while reducing its dimensionality. Additionally, PCA extracts meaningful features by identifying the most informative combinations of the original features [18].

In the top graph of Figure 11, the EMG features dataset is projected onto the first two principal components (PCs), with each color representing a distinct class movement. Initially, notable patterns emerge regarding the distribution density of data points along the PC1 and PC2 axes for each movement. This suggests the potential feasibility of movement classification. However, it's important to note that high accuracy cannot be guaranteed, as there is no clear indication that the data points corresponding to each movement are entirely distinct from those of other movements. The 3D projection on the three first PCs is also represented in Appendix (Figure 21).

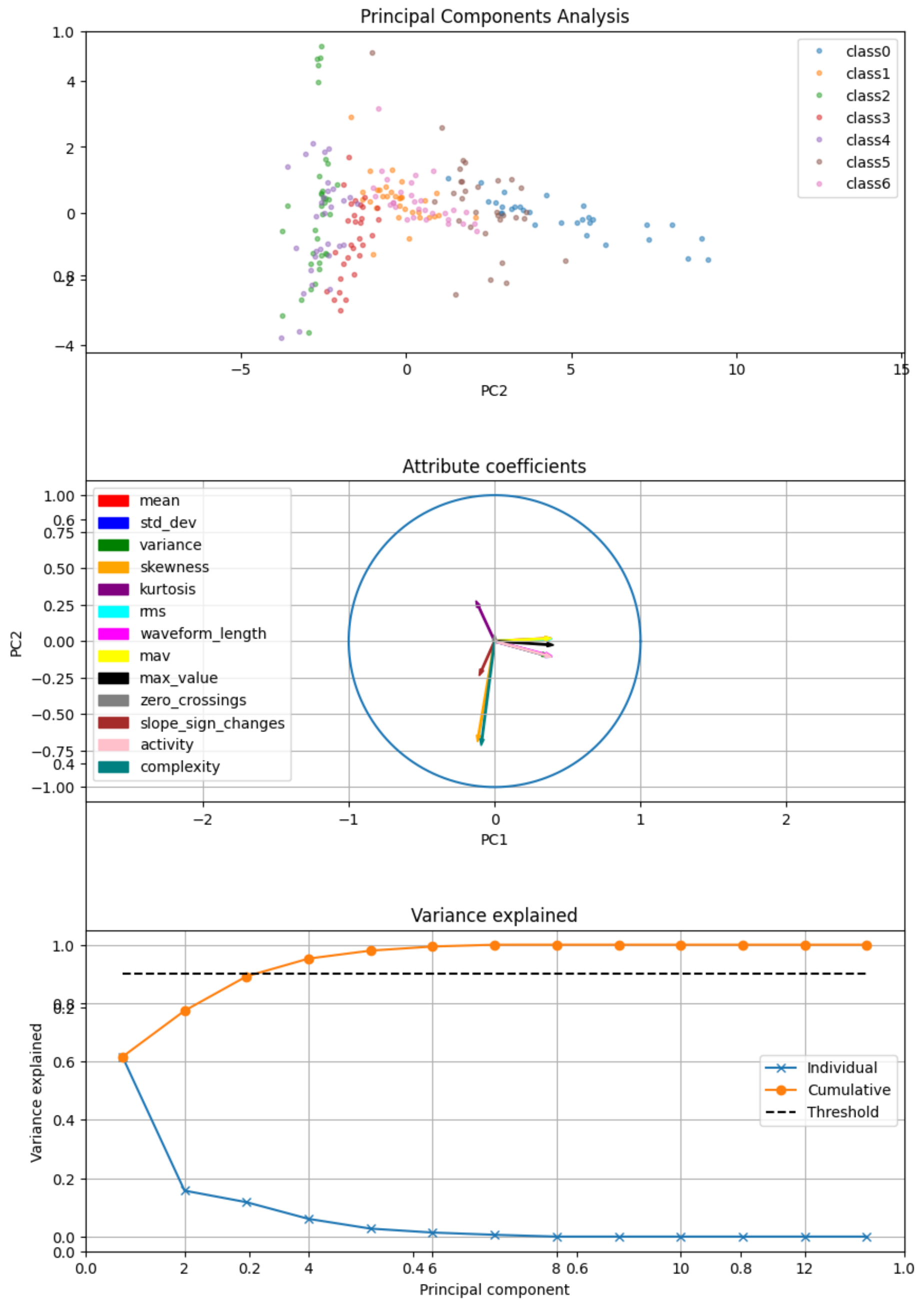


Figure 11: Principal Component Analysis study underlying the feature dataset projection on the PCs (top), their contribution to the two first PCs (middle) and the variance explained by those PCs (bottom).

Each principal component is a linear combination of the original features, with coefficients representing the contribution of each feature to the component. These loadings indicate the strength and direction of the relationship between the features and the principal component (as representing on Figure 11 on the middle graph). By examining feature contributions, researchers gain insights into which features have the greatest influence on the variability captured by each principal component. This knowledge facilitates the interpretation of the underlying structure of the data and assists in selecting relevant features for subsequent analysis or modeling.
In this analysis, the variance and skewness features exhibit significant amplitudes when

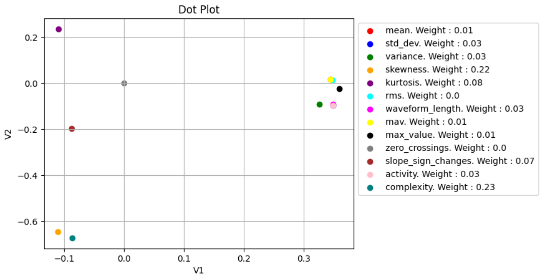
projected onto both principal components (PCs), indicating their strong influence on the PCs. This observation is further elucidated in Figure 12, depicted in the left graph. Specifically, the variance and skewness features appear to exert a notable negative impact on PC2, while the kurtosis feature shows a slight positive influence. Additionally, features such as RMS, maximum, waveform length, and activity are likely to have a positive impact on PC1. Interestingly, the zero-crossing feature demonstrates no discernible impact on either of the two principal components, as depicted in the graph.

Furthermore, PCA quantifies the amount of variance captured by each principal component (Figure 11, bottom figure), providing insights into how much of the original variability in the data is retained in the lower-dimensional space. This measure of explained variance aids researchers in determining the appropriate number of principal components to retain based on the desired level of retained information. By understanding the explained variance, researchers can make informed decisions regarding the dimensionality reduction process and its impact on classification performance.
Within the context of the 7-movement features, it appears that four principal components (PCs) are required to account for over 90% of the variance. Consequently, retaining the first four PCs is essential to attain higher accuracy in classification. This hypothesis will be further validated in the subsequent section 5.

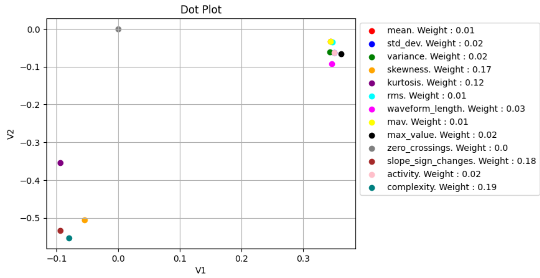
The entire PCA study has been conducted on the second subject as well, with the results presented in Appendix 22. The 3D projection on the three first PCs is also represented in Appendix (Figure 23). While the interpretation method remains consistent, notable differences emerge between the two subjects' results. Specifically, the dataset projections onto the first two principal components vary, as do the contributions of features to each signal (Figure 12). The number of principal components required to explain more than 90% of the variance remains the same in that specific case (4). These disparities underscore the individual specificity of EMG data, highlighting the unique physiological characteristics and muscle activation patterns exhibited by different individuals.
These individual-specific variations in EMG data have significant implications for the classification work:

- Model Generalization: Classifiers trained on data from one individual may not generalize well to others due to the unique characteristics of EMG signals.
- Feature Selection: The features contributing most significantly to signal differentiation can vary between individuals. Therefore, feature selection strategies must be tailored to each subject to ensure optimal classification performance.
- Model Complexity: The optimal number of principal components needed to capture relevant information and achieve satisfactory classification results may differ between individuals.
- Performance Evaluation: Evaluation metrics for classification models should account for individual variability in EMG data.

Overall, recognizing and accommodating individual-specific characteristics in EMG data is essential for developing accurate and reliable classification models, ultimately enhancing their effectiveness in real-world applications



(a)



(b)

Figure 12: Features contribution to the 2 first PCs for subject 1 (a) and subject 2 (b).

4.3 Finger movements classification with EMG signals

The objective of the classification conducted in this study is to categorize signal events among seven potential movements (labeled from class 0 to class 6), necessitating a seven-class classification task.
Four distinct types of classifiers are employed: two artificial neural networks (ANNs) with different architectures [5] [19], a k-NN and a bagging algorithm, Random Forest (RF) [4]. Additionally, CatBoost, a boosting algorithm, is utilized.

4.3.1 The ANN classifier

In this study, 2 different ANN are used, inspired from the literature reviewed underlined in section 2.4. The first one (ANN1) is composed of 2 hidden layers (RELU fucntions) and 1 softmax output layer [20]. The second one (ANN2) is composed of 3 hidden layers (2 RELU and 1 sigmoid) and 1 softmax output layer.

4.3.2 The k-NN classifier

As explained in the 2 section, k-Nearest Neighbors (k-NN) classifies a data point based on the majority class of its k nearest neighbors in the feature space. For optimal performance, this k-NN model has been trained using a range of neighbors from 1 to 22 (Appendiw Figure 25.

4.3.3 Bagging and boosting algorithms

For that study, a bagging algorithm, the Random Forest (RF) [21] (mechanism explained Appendix Figure 24) and the boosting algorithm, the CatBoost were explored.

5 Classification results and discussions

Before presenting the classification results and the achieved performance, it is important to highlight the characteristics of a good classifier and the factors that can influence its effectiveness.

5.1 Classification criteria

First of all, it is crucial to distinguish between training error and testing error to evaluate a model's performance accurately. Training error is the error a model makes on the training dataset, while testing error is the error on an unseen test dataset. A low training error indicates that the model has learned well from the training data, but a low testing error is more important as it signifies the model's ability to generalize to new data.

Several factors influence the quality of classification:

- The classifier chosen and its parameters
- The dataset itself: the quality, size, and representativeness of the dataset are critical. Data preprocessing, such as handling missing values, normalizing features, and balancing classes, also plays a vital role.
- The segmentation used
- The features extracted: techniques such as PCA can reduce dimensionality, keeping only the most informative features.
- The number of principal components used: using too few components might lose essential information, while too many can reintroduce noise.
- Ensemble Methods: combining multiple models (e.g., using bagging, boosting, or stacking) can improve classification accuracy by leveraging the strengths of different algorithms.
- Other factors can also impact the performances of the classification such as the feature scaling, outlier handling, cross-validation techniques (stratified k-fold or leave-one-out cross-validation), or model complexity.

In classification tasks, evaluating the performance of models using metrics such as accuracy, precision, recall, and F1 score is important. These metrics provide insights into the effectiveness of the classification model in correctly identifying instances of different classes. A reminder of how to calculate them is presented in Appendix section A.4.1.

Thus, in the context of hand gesture classification, where the goal is to distinguish between seven different movements, the distinction between false negatives (FN) and false positives (FP) may not be as critical as it might be in other applications. Instead, we can focus on overall accuracy, which directly relates to the true positives (TP).
When a false positive (FP) occurs, it means that a movement was incorrectly predicted as a certain gesture, resulting in two errors: one unintended movement being executed

and one intended movement not being executed. The result is similar for a false negative (FN). In both cases, the outcome is undesirable.
Given this scenario, improving the overall accuracy is paramount. Accuracy takes into account the total number of correct predictions (TP) out of all predictions made (Equation) 1, providing a comprehensive measure of the classifier's performance across all classes.

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} (\text{TP}_i + \text{FP}_i + \text{FN}_i)} \tag{1}$$

5.2 Classification results for seven movements classification

The chosen training/testing ratio is 80/20. The classifiers were trained on the training dataset and then evaluated on the testing dataset. The confusion matrices (Figure 13) present the false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) from this testing.

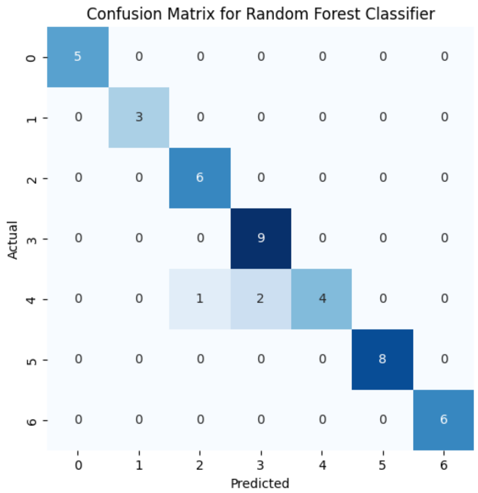
Four datasets were trained and tested for each classifier. These datasets were the principal components derived from feature extraction and subsequent PCA ranking, with 2, 3, 4, and 5 principal components. Table 1 summarizes the results of this study with the event-based segmentation used. As anticipated in section 4.2, the expected number of PCs to explain more than 90% of the variance for Subject 1 is 4 PCs. Indeed, accuracy increases from 55-66% for 3 PCs to 80-93% for 4 PCs. The accuracy does not improve significantly from 4 PCs to 5 PCs and even decreases for ANN1 and k-NN. Thus, the dataset with 4 PCs is the focus.
The best accuracies were observed for ANN2 (94%) and the Random Forest (93%). However, the accuracy of ANN2 fluctuates significantly between experiments, and the accuracy with 2 PCs and 3 PCs is better with the Random Forest. Therefore, the Random Forest classifier will be retained for further study.

Table 1: Accuracy for each classifier chosen for different number of PCs selected

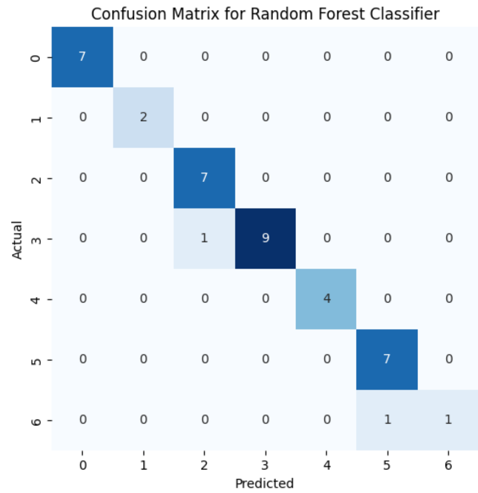
Classifiers	ANN1	ANN2	k-NN	RF
2 PCs	0.48	0.45	0.55	0.66
3 PCs	0.56	0.55	0.57	0.66
4 PCs	0.91	0.94	0.80	0.93
5 PCs	0.89	0.93	0.73	0.91

The same analysis was conducted for the time-window segmentation. The accuracy ranged from 60% with 2 PCs to 95% with 5 PCs. Therefore, this segmentation method can be considered similarly efficient. However, with this method, 5 PCs are needed to explain more than 90% of the variance.

Moreover, as explained previously, those analysis have also been conducted on a second subject. For this one, the same observations can be done and the confusion matrix and accuracy results for the RF classifier with 4 PCs are presented on Figure 13 and Figure 14.



(a)



(b)

Figure 13: Confusion Matrix resulting from the Random Forest classifier applied on the testing set with 4 PCs for subject 1 (a) and subject 2 (b).

The precision, recall and F1 score values for each movement class and the overall accuracy obtained with the RF classifier are presented on Figure 14 for the two subjects studied. The line "macro avg" refers to the unweighted average of the precision, recall, and F1-score for each class. It calculates the metric independently for each class and then takes the average across all classes. "Weighted avg," on the other hand, calculates the weighted average of precision, recall, and F1-score, where each metric is weighted by the number of true instances for each label. This means that labels with more instances contribute more to the average.

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5
1.0	1.00	1.00	1.00	3
2.0	0.86	1.00	0.92	6
3.0	0.82	1.00	0.90	9
4.0	1.00	0.57	0.73	7
5.0	1.00	1.00	1.00	8
6.0	1.00	1.00	1.00	6
accuracy			0.93	44
macro avg	0.95	0.94	0.94	44
weighted avg	0.94	0.93	0.93	44

(a)

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	7
1.0	1.00	1.00	1.00	2
2.0	0.88	1.00	0.93	7
3.0	1.00	0.90	0.95	10
4.0	1.00	1.00	1.00	4
5.0	0.88	1.00	0.93	7
6.0	1.00	0.50	0.67	2
accuracy			0.95	39
macro avg	0.96	0.91	0.93	39
weighted avg	0.96	0.95	0.95	39

(b)

Figure 14: Precision, recall, f-1 score and accuracy resulting from the Random Forest classifier with 4 PCs for subject 1 (a) and subject 2 (b).

The confusion matrices (Figure 13) and the performances tables (Figure 14) reveal that prediction errors frequently occur between the thumb, middle, and index fingers. Two main reasons can explain this. First, more testing data have been sampled for these fingers, leading to a higher chance of incorrect predictions. Additionally, this implies that the classifier was trained on fewer data points for these specific fingers. Second, these three movements are similar in terms of signal amplitude compared to the more distinct signals of hand opening or closing. They also originate from the same muscles, unlike the

ring and little fingers which use different muscles [9].
It is also evident in Figure 13(b) that the ring and little fingers are occasionally misclassified for each other. This occurs because they are controlled by the same muscle but not the one from which signals are primarily recorded. As a result, the signals have smaller amplitudes and are noisier.

5.3 Evaluation of the classifier on a new succession of movement as input

The aim of this part of the project is to simulate the real-world use of a robotic hand controlled by a user equipped with an EMG device. The user executes random movements among the seven predefined movements, and the goal for the classifier is to correctly identify each movement. Three different types of new datasets are tested to achieve this:

- Test 1: 30 repetitions of each movement from the subject 1 (already present in the training dataset, so the results are expected to be similar to the testing accuracy). This one is used to verify if the pre-processing treatment is well applied to the new dataset.
- Test 2: 30 repetitions of each movement from the subject 2 (not present in the training dataset).
- Test 3: A sequence of 7 movements repeated twice (14 movements in total) from the subject 1, simulating a possible real use of the robotic hand.

The same segmentation, feature extraction, standardization, and PCA techniques are applied, and the classifier's predict function is used to obtain the accuracy results. The classifier, which was trained and tested on the seven movements, has a testing accuracy of approximately 0.84. The accuracy results for these tests are as follows:

- Test 1, the accuracy is 89%
- Test 2, the accuracy is 9.1%
- Test 3, the accuracy is 25%.

For the test 1, the accuracy is slightly better than 0.84 since some data contained in the dataset have been used to train the classifier.
For the test 2, the accuracy is very low. In fact, as underlined in section 4.2, the features extracted and thus the PCA from one subject can't be used for another subject : feature values are specific to each movement of course, but also to each subject. While trying to predict the movement of a subject 2 with the classifier trained on the subject 1, the comparison can't be correctly done and the classifier can only misclassify.
For the test 3, back to the subject 1, the accuracy is lower than 0.84. Different reasons can explain this decrease in accuracy. First of all, the dataset for this test is smaller (14 movements instead of 30) so one error has a large impact on the accuracy. Moreover, the segmentation may have its part of responsibilities because it is the only pre-processing factor that vary between the test 1 and test 3. In fact, the mean calculated in the segmentation algorithm fluctuates more in a signal with different types of movements

than in a signal with the high repetitions of the same movement. Thus, the threshold for event-based classification is modified and not the same sections of the signal are selected. As a result, the features extracted differ.

5.4 Results discussion

To conclude on those classification results, a Bagging example, Random Forest, was explored and demonstrated the highest accuracy among the models tested. Additionally, a boosting algorithm, CatBoost, was tested but results were comparable to those of the Random Forest and indicating less robust predictions for new movement sequences.
Several observations made during the study suggest opportunities for enhancing and modifying specific aspects of the classification process. Those are presented in the following part.

First of all, it is primordial to underline that, when using only one sensor for classification, the accuracy of the classification heavily relies on two main factors: the amplitude and velocity of the movement. Indeed, the strength or intensity of the muscle contraction, represented by the magnitude of the EMG signal, influences the classification outcome. A stronger contraction typically generates a higher EMG signal amplitude, which can be easier to classify as a distinct movement. Also, the speed at which the movement is performed also plays a crucial role. Different movements may exhibit distinct velocity profiles, resulting in variations in the shape and duration of the EMG signal. The classifier must effectively interpret both the amplitude and velocity of the EMG signals to classify the movement correctly. Variations in these factors can lead to differences in classification accuracy. Moreover, this task becomes even more challenging when utilizing just one sensor.

Moreover, the localization of EMG sensors with the Paragit sleeve introduces significant fluctuations due to their less precise positioning requirements. Thos fluctuations can influence the classification results, especially if two datasets are monitored at two different moments so with different sleeve placement.

Another point to underline is that, while k-fold cross-validation could have been explored, the hold-out method was employed in this study. However, it's worth noting that k-fold cross-validation offers several advantages such as allowing for more efficient data utilization, reduces bias, and enhances generalization performance. Therefore, in future studies, incorporating k-fold cross-validation could provide valuable insights into model performance and should be considered.

Finally, if this project was to be pursued, it would be crucial to train the models on more training data and also to separate training and testing data by respecting the class proportions, which is not happening here. In fact, on Figure 13(a), we can see that the testing data is, for example, 9 for the major finger and 6 for the little finger. In the Figure 13(b), it is event 10 data for the major finger and only 2 for the little one.

6 Robotic hand control

As the robotic hand already built is composed of 5 servo motors, the robotic language chosen for this project is Arduino. Indeed, Arduino offers several advantages in terms of simplicity, accessibility, and hardware compatibility. However, Arduino's processing power and memory constraints may limit the complexity and sophistication of control algorithms that can be implemented. This could be a drawback for applications requiring advanced motion planning or real-time feedback control as it could be the case for robotic hand control.
In this study, each of the five servo motors is connected to the Arduino through a breadboard, linking to the corresponding ground, 5V, and command pins, as shown in Figures 15 and 16. The Arduino UNO is then connected to the computer via a USB cable.

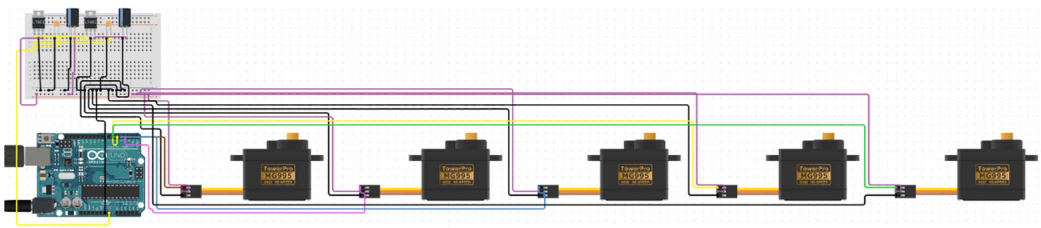
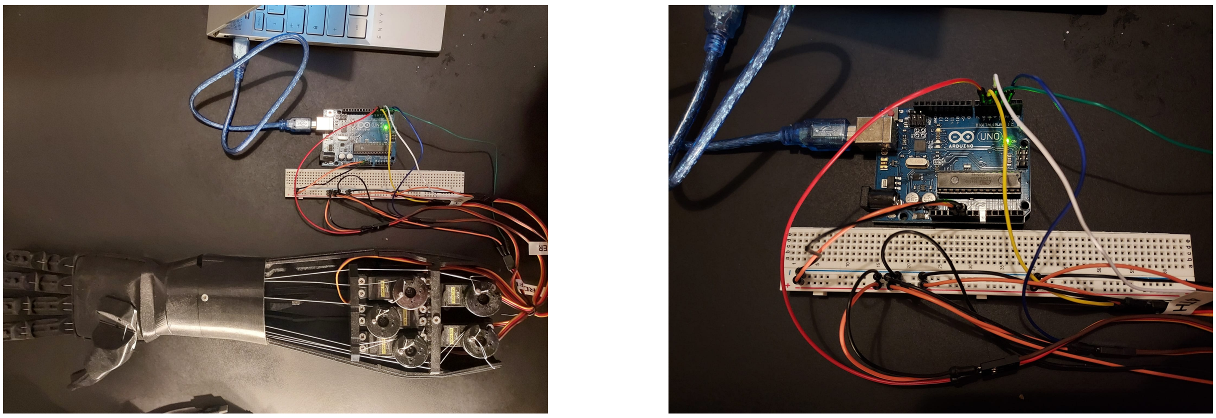


Figure 15: Arduino architecture [2].



(a) (b)

Figure 16: Arduino structure with Arduino UNO and breadboard.

The control code for the robotic hand requires the creation of a Python-Arduino communication interface. Since Arduino cannot directly access local files, Python reads the necessary data from a CSV file stored in the local repository on the computer, containing the movement classifications. Subsequently, Python transmits this information to Arduino [22]. Both the corresponding Python and Arduino codes are accessible on the project's GitHub repository (Special Course Repository) as well as a demonstration video of the robotic hand control. Once received, Arduino processes each element of the list and executes the corresponding movement. For example, for movement class 1 (closing), all servo motors activate to fold the fingers. Else, if class 6 is detected, the servo motor controlling the little finger folds accordingly.

7 Conclusion and further prospects

7.1 Conclusion

In conclusion, the pre-processing methods, PCA, and classifiers used in this project yielded satisfactory results. The training and testing outcomes from the initial experiment (see section 5.2) using the Random Forest classifier were notably impressive, achieving a testing accuracy of 93% with four principal components (PCs). However, the predictions made on a new dataset (as discussed in section 5.3) were less satisfactory, resulting in an accuracy of approximately 25%. This discrepancy is likely due to the low number of data points in the dataset or the segmentation method adopted.
Furthermore, while the pre-processing methods proved effective across two subjects, it is important to note that the features and trained classifier are specific to each subject.

Lastly, the robotic hand control via Arduino functions correctly. The outcomes from the classifier (predicted classes) are accurately received and interpreted by the algorithm, successfully controlling the corresponding fingers.

7.2 Improvement and future work

This project lasted four months, but with more time, numerous improvements could have been made. Firstly, real-time control was not achievable due to the Paragit device's lack of wireless capability to communicate EMG data. If real-time control were possible, reducing decision-making time would be crucial, which could involve choosing faster classifiers, utilizing more efficient segmentation methods, and extracting fewer features.

Additionally, it would be beneficial to detect the desired duration of the user's movement. Further studies on the effects of velocity and magnitude on the classification process could provide valuable insights. It would be possible to use methods such as Dynamic Time Warping (DTW) or Long Short-Term Memory (LSTM) networks, capable of learning temporal dependencies, can be relevant to predict the duration of the movement.

Another area of interest would be conducting similar experiments with two sensors instead of one to assess how much this could improve classification accuracy, especially on a new dataset with a sequence of 7 movements mixed in one recording.

Finally, as far as the discussion on the impact of FN/FP on the classification (section 5) and the hand gesture, an improvement could be implement to mitigate the impact of FN and FP and enhance the system's performance : introducing a "Resting" Movement when no specific movement is detected or when the classification confidence is low. An "unsure" movement is defined by a threshold of certainty based on the classifier's probability outputs. Then a probability-Based Classification needs to be implemented. Many classifiers, especially those used for multi-class classification, provide probability estimates for each class. Thus, if the highest probability among the predicted classes falls below the threshold, the movement is classified as "resting" class. By implementing this strategy, the system can ensure that when the classifier's confidence is low, the hand defaults to a safe, neutral position.

Nomenclature

ANN	Artifical Neural Network
EMG	Electromyography
k-NN	k-Nearest Neighbors
PC	Principal Components
PCA	Principal Components Analysis
RF	Random forest

References

[1] Jean-Pierre Mestanza. Bionic arm gives amputees a hand. https://www.thephuketnews.com/phuket-tech-bionic-arm-gives-amputees-a-hand-39616.php, 24 May 2013.

[2] Marin Holi. Prosthetic control with biomechanical sensors, March 12, 2021.

[3] Paragit website. Developed for patients - built for data. https://www.paragit.com/wearable.

[4] MUHAMMAD AHMAD SALEEM. Data manipulation classification of myo gestures. python · classify gestures by reading muscle activity. https://www.kaggle.com/code/muhammadahmadsaleem/data-manipulation-classification-of-myo-gestures.

[5] ABHIRUKTH CHAKRAVARTHY. Hand_gestures. python · classify gestures by reading muscle activity. https://www.kaggle.com/code/ikarosalpha/hand-gestures.

[6] KyungHyun Jovinus Lee Jiyoung Min. Electromyogram-based classification of hand and finger gestures using artificial neural networks. https://github.com/wnet500/EMG_hand_finger_gestures_classification.

[7] Randale Sechrest. Hand anatomy animated tutorial. https://www.youtube.com/watch?v=zyl6eoU-3Rg.

[8] American Society for Surgery of the Hand (ASSH). Nerves. https://www.assh.org/handcare/safety/nerves.

[9] Tim Taylor. The nerves of the arm and hand. https://www.innerbody.com/anatomy/nervous/arm-hand, Jul 16, 2019.

[10] Taim Talks Med. Muscles of the forearm (division, origin, insertion, function). https://www.youtube.com/watch?v=SvFajvk2qbc.

[11] ORCID Shuai Guo Yuwen Li Xianhua Li Tao Song, Zhe Yan and Fengfeng Xi. Review of semg for robot control: Techniques and applications. https://www.mdpi.com/2076-3417/13/17/9546, 23 August 2023.

[12] B. Venkata Rao Mohd Haris and Pavan Chakraborty. Emg signal based finger movement recognition for prosthetic hand control. https://ieeexplore-ieee-org.proxy.findit.cvt.dk/stamp/stamp.jsp?tp=&arnumber=7437907, 2015 International Conference on Communication.

[13] Ken Steinweg Paul DeVita Tibor Hortobágyi Stanislaw Solnik, Patrick Rider. Teager–kaiser energy operator signal conditioning improves emg onset detection. https://link.springer.com/article/10.1007/s00421-010-1521-8, 05 June 2010.

[14] KIRILL YASHUK. Classify gestures by reading muscle activity. https://www.kaggle.com/datasets/kyr7plus/emg-4.

[15] Emine Yamancorresponding author and Abdulhamit Subasi. Comparison of bagging and boosting ensemble machine learning methods for automated emg signal classification. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6885261/, 2019 Oct 31.

[16] Oliver Jones. Muscles in the anterior compartment of the forearm. https://teachmeanatomy.info/upper-limb/muscles/anterior-forearm/, October 29, 2023.

[17] The SciPy community. Scipy signal find peaks. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html/, Copyright 2008-2024.

[18] Aditya Sharma. Principal component analysis (pca) in python tutorial. https://www.datacamp.com/tutorial/principal-component-analysis-in-python, Jan 2020.

[19] GABRIEL ATKIN. Hand gesture prediction. python · classify gestures by reading muscle activity. https://www.kaggle.com/code/gcdatkin/hand-gesture-prediction/notebook.

[20] SAGAR SHARMA. Activation functions in neural networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, Sep 6, 2017.

[21] Bagging and random forest for imbalanced classification. https://www.geeksforgeeks.org/bagging-and-random-forest-for-imbalanced-classification/, 11 Mar, 2024.

[22] ansh2919. Serial communication between python and arduino. https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756, Nov 6, 2020.

List of Figures

1 Bionic Hands: Empowering Amputees to Reclaim Mobility [1]. 4
2 Seven movements represented the seven classes of the classification study. . 5
3 Main steps for hand movements classification : from data collection to robotic control. 5
4 Already built robotic hand available from the NRT laboratory at DTU (b). Paragit sleeve composed of 1 EMG sensor to collect EMG data from forearm muscles (a). 6
5 Main forearm peripheral nerves [8]. 7
6 Processing and recognition methods of sEMG signals [11]. 8
7 EMG signal collected from the Paragit device before (top) and after (down) filtering steps (30 repetitions of the closing movement). 13
8 Envelop EMG signal from the filtered signal. On this picture, the distinct movements executed by the subject are easily visible. 14

9 Event-based segmentation for the closing movement signal 15
10 Event-based segmentation for the 7-movements succession signal 15
11 Principal Component Analysis study underlying the feature dataset projection on the PCs (top), their contribution to the two first PCs (middle) and the variance explained by those PCs (bottom). 17
12 Features contribution to the 2 first PCs for subject 1 (a) and subject 2 (b). 19
13 Confusion Matrix resulting from the Random Forest classifier applied on the testing set with 4 PCs for subject 1 (a) and subject 2 (b). 22
14 Precision, recall, f-1 score and accuracy resulting from the Random Forest classifier with 4 PCs for subject 1 (a) and subject 2 (b). 22
15 Arduino architecture [2]. 25
16 Arduino structure with Arduino UNO and breadboard. 25
17 Graphical chart for project management (GANTT) of the movement recognition and robot control project. 31
18 EMG signal collected from Paragit monitoring device before (top) and after (down) filtering steps. This signal was obtained during the 7-movements succession (twice of each movement). 32
19 Signal envelope for different movements monitored with the Paragit sleeve. 32
20 Time-window segmentation for the closing movement signal with 50% overlapping and 300ms as window lenght. 33
21 Graphical visualization of the features dataset projection on the 2 first PCs (left) and 3 first PCs (right) for the subject 1. 33
22 Principal Component Analysis study underlying the feature dataset projection on the PCs (top), their contribution to the two first PCs (middle) and the variance explained by those PCs (bottom), for subject 2. 34
23 Graphical visualization of the features dataset projection on the 2 first PCs (left) and 3 first PCs (right) for the subject 2. 35
24 Random Forest Bagging classifier principle [21]. 36
25 k-NN accuracy depending on the number of neighbors K selected. 36

List of Tables

1 Accuracy for each classifier chosen for different number of PCs selected . . 21

A Appendix

A.1 Materials and project management

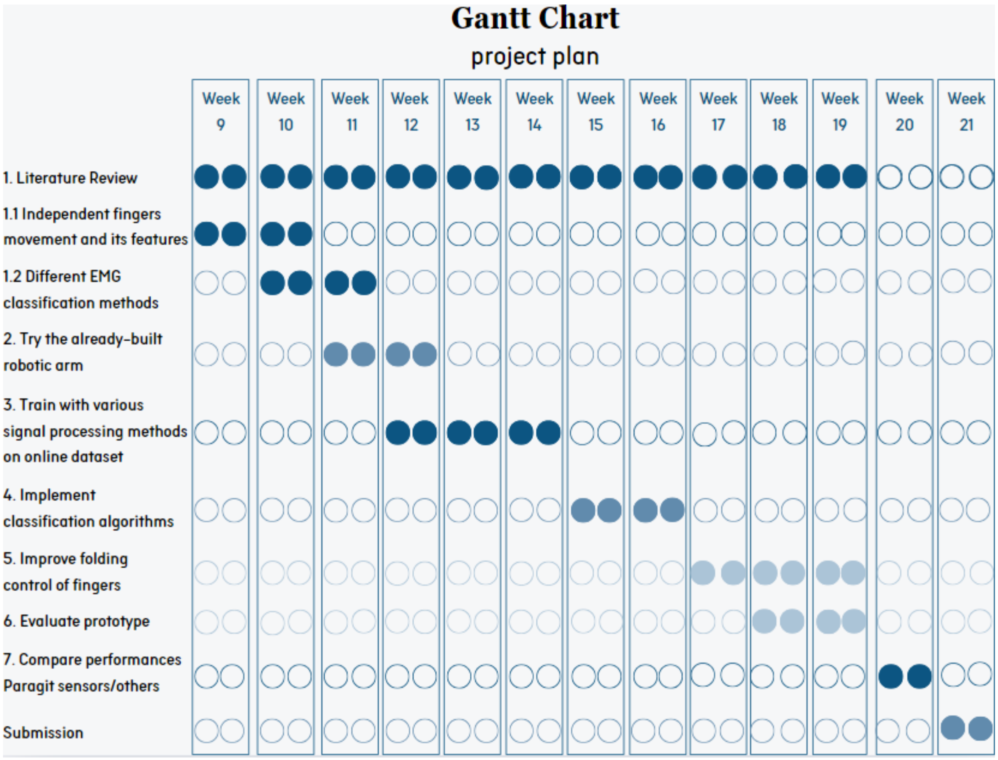


Figure 17: Graphical chart for project management (GANTT) of the movement recognition and robot control project.

A.2 Pre-processing results

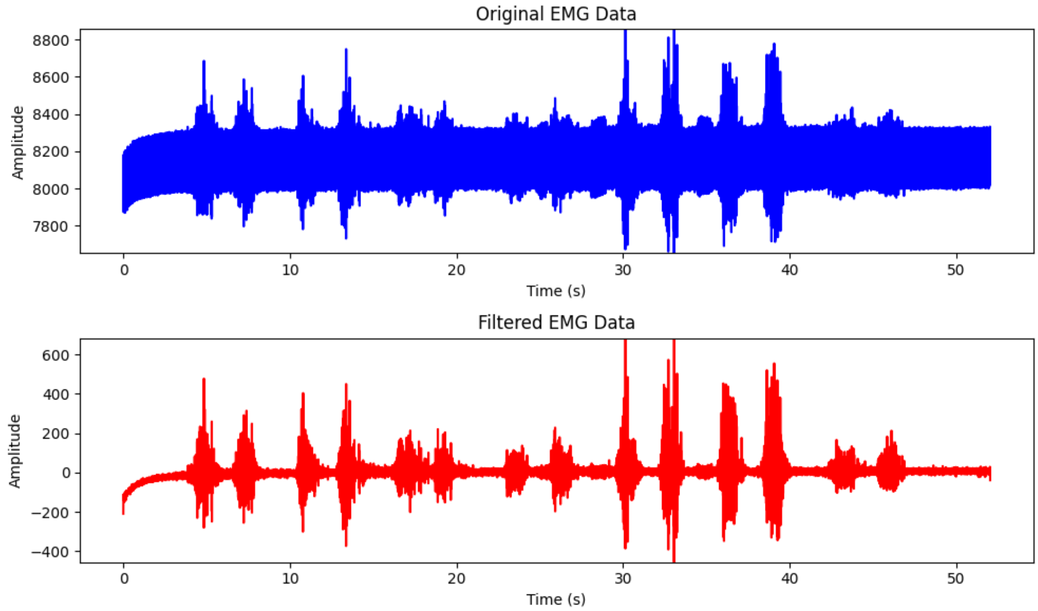
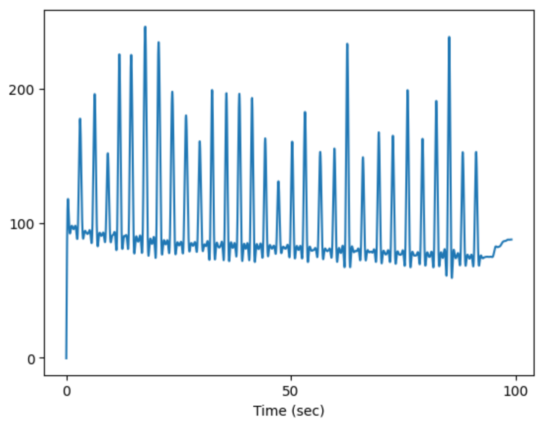
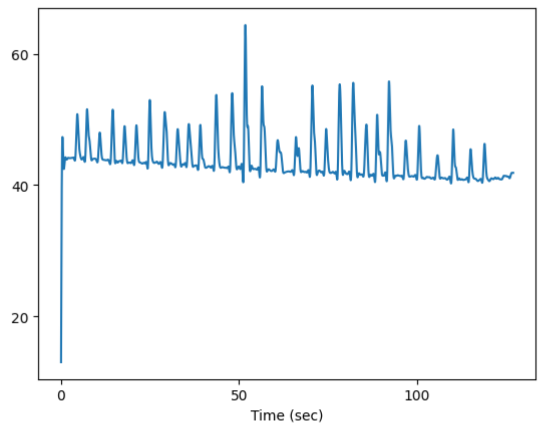


Figure 18: EMG signal collected from Paragit monitoring device before (top) and after (down) filtering steps. This signal was obtained during the 7-movements succession (twice of each movement).



(a) Signal envelope for the hand opening movement



(b) Signal envelope for the thumb movement

Figure 19: Signal envelope for different movements monitored with the Paragit sleeve.

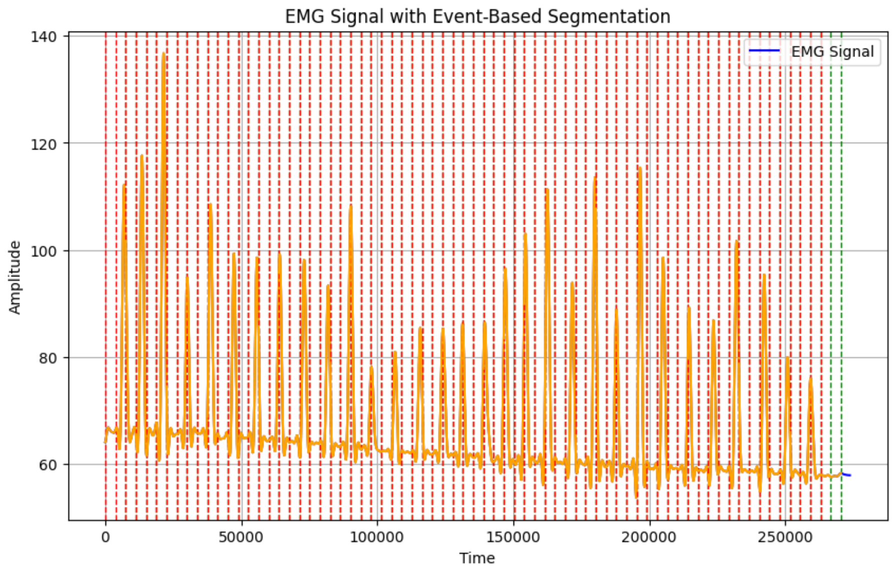
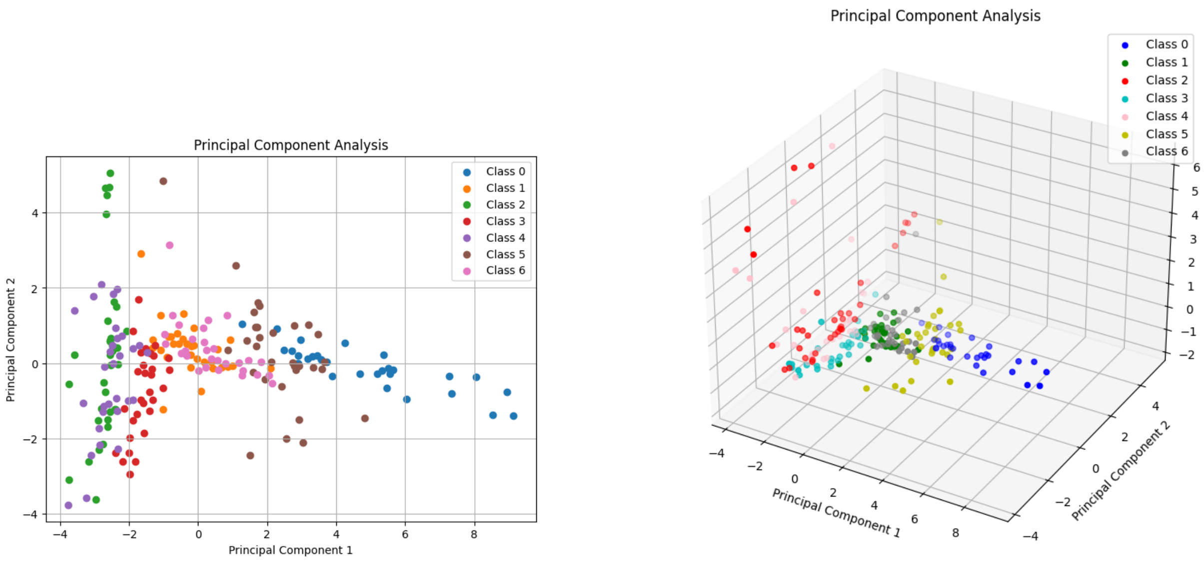


Figure 20: Time-window segmentation for the closing movement signal with 50% overlapping and 300ms as window lenght.

A.3 PCA results



(a) (b)

Figure 21: Graphical visualization of the features dataset projection on the 2 first PCs (left) and 3 first PCs (right) for the subject 1.

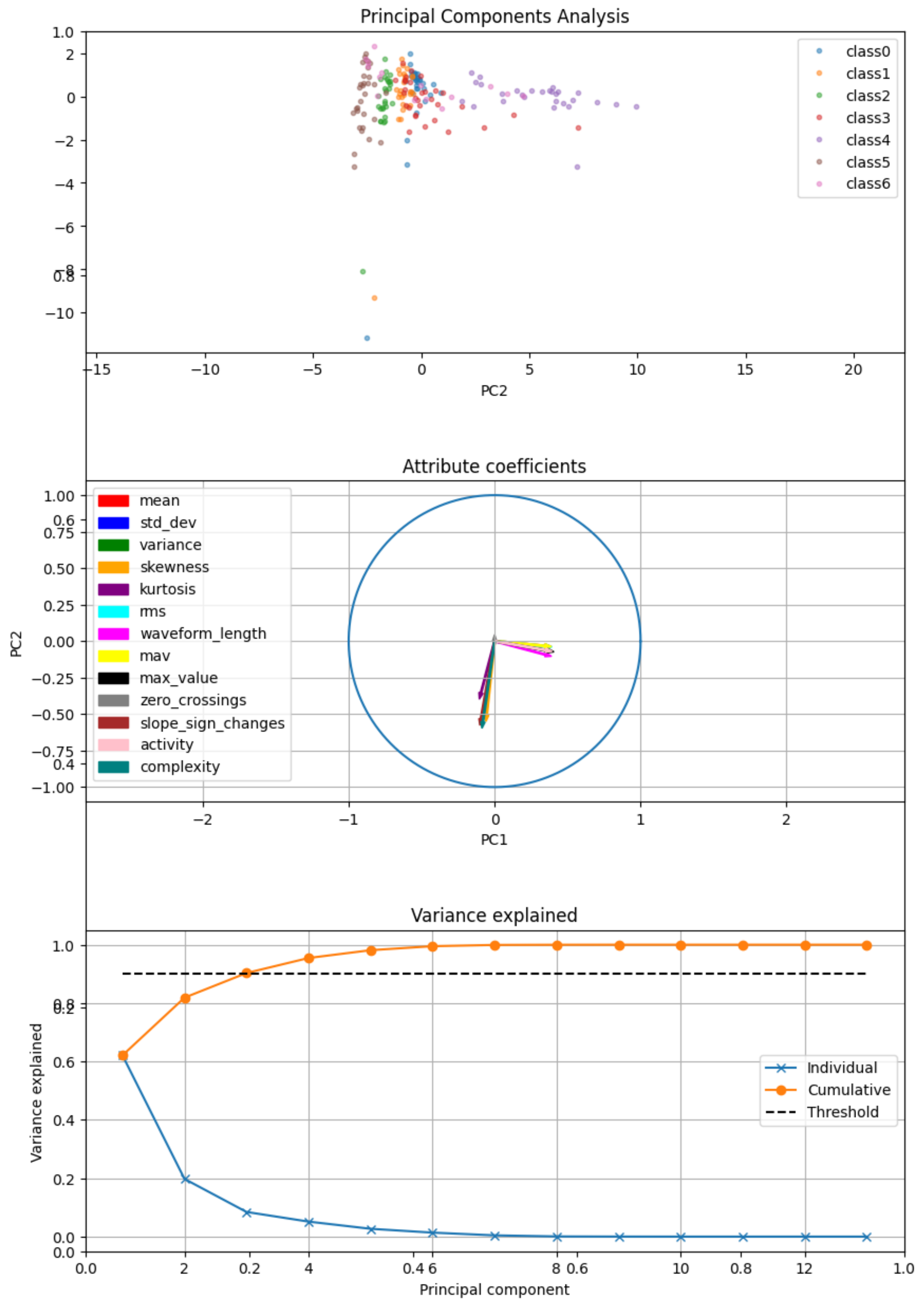
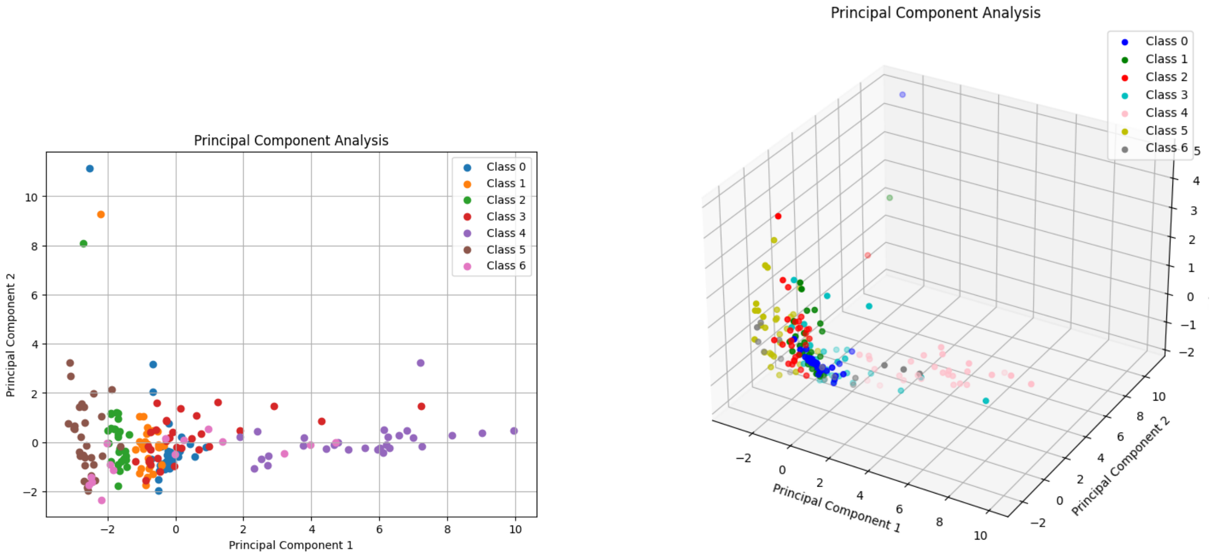


Figure 22: Principal Component Analysis study underlying the feature dataset projection on the PCs (top), their contribution to the two first PCs (middle) and the variance explained by those PCs (bottom), for subject 2.



(a) (b)

Figure 23: Graphical visualization of the features dataset projection on the 2 first PCs (left) and 3 first PCs (right) for the subject 2.

A.4 Classifiers and classification results

A.4.1 Classifier performances

1. Accuracy: measures the proportion of correctly classified instances out of the total instances in the dataset. It is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2}$$

2. Precision: measures the proportion of true positive predictions among all positive predictions made by the classifier. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3}$$

3. Recall (Sensitivity): measures the proportion of true positive predictions among all actual positive instances in the dataset. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

4. F1 Score: harmonic mean of precision and recall, providing a single metric that balances both precision and recall. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

These metrics are essential for evaluating the performance of classification models, particularly in scenarios with imbalanced classes or differing costs associated with false positives and false negatives.

A.4.2 Classifiers

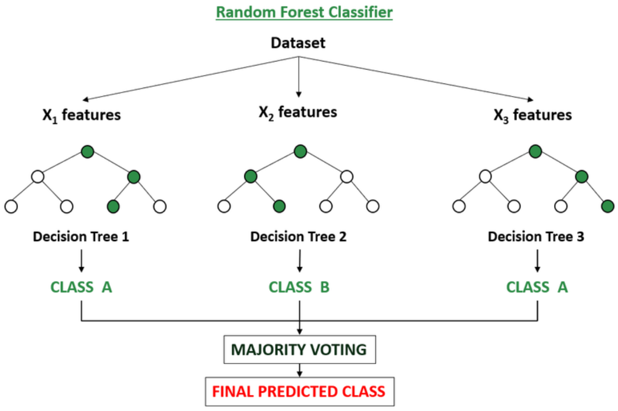


Figure 24: Random Forest Bagging classifier principle [21].

A.4.3 Results

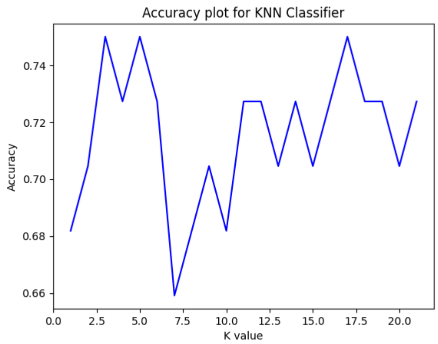


Figure 25: k-NN accuracy depending on the number of neighbors K selected.