Classifying EMG signals using LDA and SVM classifiers and determining their applicability for online training

Aaron Dylan Blanco

Faculty of Natural, Mathematical & Engineering Sciences
Robotics MSc

August 2023

Acknowledgments
First of all, I'd like to say many thanks to my supervisor Ernest Kamavuako and PhD student Bingbin Wang. You have been a great help to this project and I'm grateful for all the time and effort you put aside to guide me.

To my supervisor, thank you for pushing me as I needed the extra pressure to get myself into gear and really finish this project. Thank you again for organising meetings and putting time aside out of your busy schedule to keep track of my performance as well as giving great feedback. I really appreciate it.

Secondly, to Bingbin, thank you for answering all of my endless questions and giving me advice when you could. You were also a big help as whenever I ran into an issue or had a query you would always get back to me in a timely manner. Even all the way till the end when I asked about you the extra data, etc.

Lastly, thank you again for all the support you all gave. I really enjoyed this project to the end despite not getting to control the robotic arm.

Abstract
Electromyography (EMG) signal classification is the focus of this dissertation. This research is motivated by the pressing need to improve the lives of amputees and individuals born with limb deficiencies. The study seeks to contribute to the development of more advanced and responsive prosthetic devices by employing computational methods. To evaluate the efficacy of the classifiers, the research methodology consisted of altering the window and overlap sizes. In terms of accuracy, the SVM classifier consistently outperformed the LDA classifier, particularly within the critical 300ms real-time operation window. Nonetheless, as evidenced by their respective log loss values, both classifiers demonstrated a high degree of prediction accuracy. Further analysis, including the F1 score and a detailed confusion matrix, revealed the strengths and limitations of each classifier in greater detail. The SVM emerged as a robust instrument for the classification of real-time EMG signals, whereas the LDA, despite its challenges, demonstrated room for further refinement and optimisation. Comparative analysis of the existing literature revealed that while the performance of the SVM was comparable to that of other studies, the performance of the LDA had room for improvement. Future research should investigate alternative feature extraction techniques, data augmentation, and dimensionality reduction, according to the recommendations. This dissertation concludes by highlighting the transformative potential of machine learning in biomedical engineering, specifically prosthetic technology.

Table of Contents

Nomenclature

EMG	**Electromyography**
ECG	**Electrocardiogram**
EEG	**Electroencephalography**
SNR	**Signal-to-noise ratio**
MAV	**Mean Absolute Value**
WL	**Waveform Length**
ZC	**Zero Crossings**
SSC	**Slope Sign Changes**
LDA	**Linear Discriminant Analysis**
SVM	**Support Vector Machines**
KNN	**K-Nearest Neighbours**
ANN	**Artificial Neural Network**
LOOCV	**Leave-one-out**

Table of Figures

Table of Equations

1 Introduction
Electromyography (EMG) signals, the electrical manifestations of neuromuscular activities, have a long history of promise in biomedical engineering and rehabilitation. As the global population of amputees and individuals born without limbs continues to rise, the need for innovative prosthetic solutions has never been greater. Recent statistics highlight the magnitude of the problem, revealing a sizable population coping with the physical and psychological consequences of limb loss. Traditional prostheses, while beneficial, frequently fall short when it comes to providing natural and perceptive limb movement. The discipline of machine learning, which has revolutionised countless industries, is now poised to revolutionise prosthetic technology. By utilising machine learning to classify EMG signals, it is possible to develop prostheses that are more responsive, adaptive, and in line with the user's intent. This dissertation investigates the complexities of EMG signal classification and its transformative potential for improving the lives of individuals with limb deficiencies. As we navigate the complexities of this technology, we remain rooted in our primary motivation: to restore a semblance of normalcy and functionality to the lives of innumerable individuals awaiting a brighter, more autonomous future.

1.1 Amputation Statistics
To comprehend the significance of this undertaking, it is necessary to comprehend the number of lives it impacts. In 2020, 584,454 infants were born in England, and 13,065 of them had at least one congenital abnormality (such as being born without limbs) [1], which equates to one in every 45 births. In addition, the number of major lower limb amputations in the United Kingdom (UK) has increased consistently, with 7,545 cases recorded between 2015 and 2018, compared to 6,955 cases between 2012 and 2015 [2]. One in every 1,900 new-borns in the United States is estimated to be born with a limb reduction [3]. In addition, an estimated 185,000 amputations were caused by trauma, compared to approximately 1.6 million in 2005 [4] [5] [6]. The number of amputations in the United States could reach 3.2 million by 2050, according to projections [5]. In both the United States and the United Kingdom, a significant number of people are missing appendages, and as the number continues to rise, more and more people will be affected. In addition, it should be noted that the statistics of other nations have not been included, which suggests the possibility of a larger global population of limbless individuals. Therefore, both the emphasis on the need for this technology and the measures taken to resolve it must increase.

1.2 Prostheses
Due to technological advancements, prostheses have emerged as a remedy to the problem of limb loss. Prostheses are artificial substitutes designed to replace missing body elements [7]. There are numerous types of prostheses, including passive, body-powered, externally powered myoelectric, and others [8]. These technological advances have made it possible for amputees to resume their pre-amputation lifestyles or achieve independence in some fashion. However, the current status of this technology prevents it from replicating the function of a healthy limb.

2 Aim & Objectives
Keeping all of this in mind, the objective of this endeavour is to identify an appropriate classification model for EMG data. This offline model will be used to train an online model that will control the movement of a robotic arm during real-time operation. The following are the principal goals that must be attained in order to achieve this objective:

- Design a bandpass and notch filter
- Design an overlapping windowing method
- Design a feature extraction method
- Train data on a classifier
- Determine an optimal classifier

The following section will discuss the relevant background and theory that will assist in the understanding of the project, as well as provide analyses of cutting-edge technologies that will support the methodology that will be used in this report.

3 Background & Theory
3.1 Methods of Data Acquisition
Before beginning the endeavour, it is beneficial to be aware of other body signals. Thus, you can determine which method is optimal for controlling a robotic limb, given that one may provide more data than the other. EMG, also referred to as a muscle signal, is a biomedical signal that measures the electrical currents

generated in muscles during contraction and relaxation in neuromuscular activities [9]. Intramuscular EMG and surface EMG can both be used to record EMG signals [10]. Intramuscular EMG is an invasive technique that involves inserting needles or wires into the muscles, while surface EMG is a non-invasive technique that involves positioning electrodes on the surface of the skin. Electroencephalography (EEG), which records brain activity, and electrocardiography (ECG), which records cardiac activity, exist in addition to electromyography (EMG) [11]. These two methods are considered beneficial, but EMG has several advantages over ECG and EEG. ECG and EEG impulses typically fall below 100 Hz, whereas EMG signals range between 5 Hz and 2 kHz [12]. This allows for more precise measurements and a more comprehensive analysis of muscle activity.

3.2 Functionality of Electromyography (EMG)

Understanding the relationship between muscle movement and EMG signals is crucial, as it may provide a clearer indication of where to position electrodes on the arm to obtain more accurate data readings. Muscle movement originates in the motor cortex of the brain, where neural activity causes the desired muscle action. This neurological signal is then transmitted from the brain to the spinal cord, where motor neurons deliver movement information to the appropriate muscle. The higher motor neurons then transmit this signal to the lower motor neurons, which directly innervate [13] the muscle at the neuromuscular junction, where nerves and muscle fibres meet [14]. Muscle innervation results in the release of calcium ions, which induces a change in muscle tension. This process entails depolarization, which is defined as a modification of the electrochemical gradient. We detect the EMG signal through this change in current caused by depolarization [15]. Consequently, the EMG signal can efficiently classify diverse inputs as unique actions. EMG is also measured in microvolts and has a linear relationship with both the intensity of muscular contraction and the number of active muscles. When muscle contraction becomes more intense and a larger number of muscles are stimulated, the measured voltage amplitude increases. In addition, EMG signals can be identified even when no visible actions are occurring. Therefore, EMG recordings provide an additional source of information about cognitive-behavioral functioning that would otherwise be concealed by observation [15]. However, despite the advantages of EMG, certain limitations must be taken into account. These include the possibility of noise being recorded by the electrode due to electronic equipment, the individual's physical condition, electromagnetic sources, cross talk (unwanted EMG signal from an unmonitored muscle group), and internal noise resulting from anatomical, biochemical, and physiological factors associated with muscle fibres and tissues. Moreover, the inherent instability of the EMG signal is problematic because it possesses quasi-random characteristics (frequency components between 0 and 20 Hz are typically unstable), resulting in unwanted noise [10].

3.3 Myoelectric Control

As briefly mentioned in section 1.2, myoelectric control is a type of prosthesis. This form of prosthesis is important to this project because it relates to the use of EMG signals by translating the classified data (which is a particular hand motion) into robotic arm movement. Using this technology, individuals who have lost limbs or have motor impairments can regain functional movement. Myoelectric control can be broken down into four basic steps: feature extraction, classification, motor controller, and motor [16]. Feature extraction begins with the selection of features from a data set for use in classification. During classification, a processing algorithm analyses the EMG signals to identify specific patterns and characteristics that are indicative of intended movements. For instance, machine learning techniques are frequently employed to train a system to recognise distinct patterns of muscle activation associated with particular actions. After determining the user's intent, the myoelectric control system translates it into commands that operate the external device. In a prosthetic limb, for example, the system can activate motors or actuators to initiate the desired movements, enabling the user to control the limb's motion, grasp, and other functions. Myoelectric control has substantially enhanced the functionality and usability of assistive devices, but ongoing research and development are focused on refining the technology. Efforts are being made to improve signal processing algorithms, electrode designs for improved signal acquisition, and sensory feedback for a more immersive and intuitive user experience.

3.4 EMG Pattern Recognition

3.4.1 Data Pre-processing

Preprocessing EMG signals is essential for ensuring the reliability and accuracy of data analysis by removing potential contaminants and enhancing the signal-to-noise ratio (SNR). Electrode shift, power-line interference, motion artefact, and electronic noise are contaminants that can significantly distort EMG signals and hinder their interpretation [17]. To combat power-line interference, notch filters are frequently used to eradicate harmonically related power grid frequencies. Using high-pass Butterworth filters with cutoff frequencies typically set to 10 or 20 Hz [18], motion artefacts caused by electrode-skin interface impedance changes or cable motion can be reduced. Electrical noise can be reduced by using high-performance components and low-pass filters with cutoff frequencies of 450 or 500 Hz, as EMG signal energy in frequencies beyond this range is typically negligible during muscle contraction. In a separate article [19], band-pass filters were used to eliminate high and low frequencies. The purpose of the low frequency cut-offs is to eradicate baseline drift, perspiration, and direct current offset. They adjust the frequency between 5 Hz and 20 Hz. As for reducing high frequencies, this eliminates high frequency noise and prevents aliasing. Typically, the high frequency cut-off is left quite high in order to capture rapid on-off pulses of the signal between 20 Hz and 450 Hz. In addition, the notch filter is set to 50 Hz or 60 Hz to eliminate power-line interferences that are inherent in EMG data; the frequency depends on the power grid specification of the country/region; in this project, the data is from the United States (US), so 60 Hz is required.

3.4.2 Data Segmentation

After preprocessing has been performed. We must segment the data in order to analyse it effectively. Segmentation is used to analyse vast amounts of data in parts rather than as a whole because doing so requires a great deal of computational power, and segmentation is required if real-time control is desired. Windowing is a segmentation technique [19]. Two varieties of windowing schemes, overlapping windowing and adjacent windowing, have been proposed to generate preprocessed signal segments. In adjacent window segmentation, successive windows are extracted from a time series by incrementally increasing an index by the window size. Overlapping window segmentation is used to increase the density of the decision stream by incrementing neighbouring windows with lesser durations than the window length. Consequently, contiguous windows share elements. Windowing operations are utilised to extract EMG data frames with a fixed duration, which are necessary for feature extraction. One article [20] suggests that windows should last between 100 and 300 milliseconds. In another, however, it is stated that less than 200 milliseconds do not contain sufficient information [21]. Therefore, these assertions will be tested in the method section, and the results will speak for themselves. To reiterate, the desired outcome of this project is the development of an offline classifier. Although this offline classifier will be used to train an online classifier, the project will be aware of the parameters that influence this. In real-time myoelectric control, for example, there is a maximum length limit on the window to ensure that the time interval between successive windows and the computation time (update rate) does not exceed 300 milliseconds to prevent any perceived latency [22]. Despite the real-time limitation, overlapping windowing enables the use of segments longer than 200 milliseconds; consequently, the size of the length and overlap of the window will be varied during the methodology phase of this report in order to identify the optimal solution.

3.4.3 Feature Extraction

Feature extraction is the process of transforming raw data into a set of features or attributes that effectively represent the underlying patterns in the data, thereby aiding in the differentiation between various states or classes. Time-domain features such as Mean Absolute Value (MAV), Variance, Waveform Length (WL), Zero Crossings (ZC), and Slope Sign Changes (SSC) are extracted from EMG data. Frequency-domain characteristics including the Median Frequency, the Mean Frequency, and the Power Spectral Density. In addition, time-frequency characteristics including Wavelet Transform coefficients. The objective is to extract from the raw EMG signals details which can be used for further analysis, such as classification or clustering. The features should be able to incorporate relevant data about the electrical activity of the muscle while ignoring noise or irrelevant details. When deciding which domain to select features from, it is essential to understand their functionality. Time-domain refers to procedures that derive properties from the unprocessed EMG time-series. Frequency-domain features extracted from the Fourier transform of the EMG signal are used to collect information regarding motor unit recruitment rates and muscle fatigue. Finally, time-frequency refers to any transformation of an EMG signal that combines time-domain and frequency-domain. Due to its high accuracy in low-noise environments and low computational complexity [23], the time-domain is frequently utilised the most among these three. Having high accuracy only in low-noise environments appears to be disadvantageous, but this issue can be resolved through signal processing. In addition, its low complexity is a desirable characteristic for real-time systems; consequently, the time-domain will be the focus of feature extraction for this project moving forward; and the features that will be used, which are cited and widely used in myoelectric control schemes, are Hudgin's Time-domain [24], which consists of the features MAV, WL, ZC, and SSC.

3.4.4 Classification

After feature extraction, classification is all that remains. Classification is essential for the operation of a myoelectric control system, as this component serves as the link between the EMG signal and the actuated robotic limb. A classifier's primary function is to identify patterns in data. It accomplishes this by first obtaining a feature vector (containing essential data features and the classes to which they belong) and a vector of labels. This data is used to train the classifier, which then produces a model that can be evaluated using performance metrics. When selecting a classifier, it is essential to understand that certain classifiers are better adapted for specific tasks. Numerous classifiers, such as Linear Discriminant Analysis (LDA) [25], Support Vector Machines (SVM) [26], K-Nearest Neighbours (KNN) [27], and Artificial Neural Network (ANN) [28], have been investigated in myoelectric control systems. Among these, LDA has been extensively adopted and utilised with a set of features proposed by Hudgins [24]. In addition, LDA has sufficient classification accuracy and computational usage, making it suitable for real-time applications.

3.4.5 Performance Evaluation

After training a classifier, you must assess the model's performance on unobserved data. Classification accuracy is one method for evaluating the performance of a classifier:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions\ made} \quad (1)$$

The closer the accuracy is to 100%, the better; however, just because the accuracy is high does not mean the model is excellent, as it may have issues such as overfitting, which prevents the classifier from simply memorising the data. This is undesirable, as you want your model to be as generalised as possible so that it can make accurate predictions with data outside of the training set. Consequently, cross validation is implemented to prevent this. Cross-validation is a statistical technique used in machine learning to evaluate the performance of a model on unobserved data and to determine a model's robustness. Common implementation techniques include K-Fold and Leave-One-Out (LOOCV). K-Fold randomly divides a dataset into 'k' equal-sized subsets or folds, with one subset retained as the validation set and the remaining 'k-1' subsets serving as the training set. Cross-validation is repeated 'k' times, with each of the 'k' subsets serving as the validation set exactly once. The results of 'k' are then averaged to generate a single estimate. LOOCV is a special case of k-fold cross-validation where 'k' equals the number of observations in the dataset, meaning that each observation is used as the validation set precisely once. Due to its balance between computational efficiency and performance estimation accuracy, K-Fold is the superior implementation of the two; and for the majority of problems, LOOCV is too computationally costly, particularly for large datasets. There is more to evaluating the performance of a model than cross validation. These other metrics include the confusion matrix, F1 score, precision, recall, and log loss. The F1 score, which is the Harmonic Mean of precision and recall, is used to determine the accuracy of a test. The F1 score, with a range of [0, 1], indicates the accuracy and robustness of your classifier. Precision and recall are metrics used to calculate the F1 score, and each metric influences it in its own manner. For instance, if precision is high but recall is low, this indicates that the classifier is extremely accurate but overlooks many difficult-to-classify instances.

$$F1 = 2 \times \frac{1}{\frac{1}{precision}+\frac{1}{recall}} \quad (2)$$

Precision is the ratio between the number of correct positive outcomes and the number of positive outcomes predicted by the classifier:

$$Precision = \frac{True\ Positives}{True\ Positives+Fals\ \ Positives} \quad (3)$$

Recall is the number of accurate positive results divided by the total number of samples that should have been identified as positive:

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Negatives} \quad (4)$$

The log loss metric penalises incorrect classifications (works well with multi-class classification). Given that N samples belong to M classes, the log loss is computed as follows:

$$Log\ Loss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \times log(p_{ij}) \qquad (5)$$

Where y_{ij}, indicates whether the sample i belongs to class j or not. p_{ij}, indicates the probability of sample i belonging to class j. Log loss has no upper bound and is with the range [0, ∞], where closer to 0 means indicates higher accuracy.
Confusion matrix is a matrix that describes the complete performance of the model. In a multiclass problem, the matrix is larger than a 2x2 where each row represents the instances of the predicted class, and each column represents the instances of the actual class. The diagonal elements in the matrix represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabelled.

4 Methods

This section will discuss the pre-processing, segmentation, feature extraction, and classification techniques that utilise the following toolboxes: Machine Learning and Statistics Toolbox (MST), Signal Processing Toolbox (SPT), and EMG Feature Extraction Toolbox (EFET) [29]. A LDA and SVM classifier was trained using the Hudgins Time Domain features. In section 6, where the results are discussed, a comparison of their performance is illustrated. In addition, the data is classified into seven distinct motions, including rest, elbow flexion, elbow extension, pronation, supination, hand open, and gripping.

4.1 Properties of the Data

The data used in this project consists of 39 files of raw recorded EMG signals, with each file representing a different form of hand motion. These files were obtained from a study designed to test a prosthetic limb controlled by a pattern recognition system [30]. Section 7 provides additional information on the legal, social, ethical, and professional issues associated with the data.

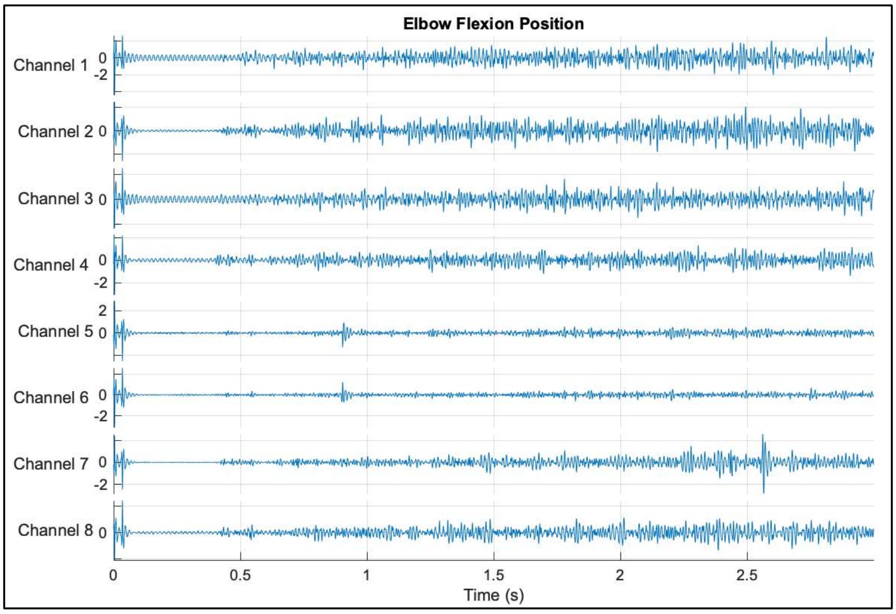

Figure 1) Raw data of Elbow Flexion position

The data was sampled at a rate of 1000 Hz, consists of eight channels, and the duration of the movement is approximately three seconds. Each data file is contained in a .mat file that contains a daq file. The daq file is divided into two sections: a 3000x8 double array containing the data, and a 3000x1 double array containing the recorded timestamps.

4.2 Pre-processing

Before sending the data into a classifier and expecting it to train, several stages are required, the first of which is data filtering. As stated in section 3.1, unprocessed EMG data contains a great deal of noise that can be caused by interference from power lines, electrode shift, etc. A bandpass and notch filter is employed to ensure that only valuable data is analysed. The bandpass filter has low-cut and high-cut frequencies of 10 Hz and 499 Hz, respectively, while the notch filter has a notch frequency of 60 Hz. The following code demonstrates the implementation of this.

```
function final_filtered_data = process_data(data, fs, lowcut, highcut)

% Extract the DAQ_DATA
emg_data = data.daq.DAQ_DATA;

% Cutoff frequency matrix
cutoff = [lowcut highcut]/(fs/2);

[b,a] = butter(2, cutoff, 'bandpass'); % Butterworth filter
bandpass_filtered_data = filter(b, a, emg_data);

% Notch filter
notch_freq = 60; % Notch frequency
notch_width = 30; % Notch width
wo = notch_freq/(fs/2); % Normalized frequency
bw = wo/notch_width; % Bandwidth
[bn,an] = iirnotch(wo, bw); % IIR notch filter design
notch_filtered_data = filter(bn, an, bandpass_filtered_data);

% The final filtered data
final_filtered_data = notch_filtered_data;
end
```

Figure 2) Function to apply a bandpass and notch filter. (fs is the sampled frequency at 1000 Hz)

4.3 Segmentation

In order to further prepare the data, it is subdivided into small chunks so that processing the data requires less computational effort. This section employs the overlapping windowing technique, as this approach is commonly employed in online training. In order to ascertain the performance of the offline model, it is essential that this technique replicates certain conditions of online training. In addition to the EMG data, the function below accepts the window_size and overlap_size parameters. The tested window sizes are 200, 300, and 450, and the overlap sizes are 50%, 65%, and 80%. To segment the data, however, it is necessary to determine how to segment each of the EMG data's eight channels. Consequently, the function iterates through each channel, divides it according to its window size and overlap size, and appends the data to a cell that contains each segment.

```
function frame_cell = segment_data(data, window_size, overlap_size)
% Get the size of the data
[r, c] = size(data);

% Initialise an empty cell array to store the windows of data
frame_cell = {};

% Start index for the first window
n_start = 1;

% Loop over the data in windows
while (n_start + window_size <= r)
% Extract the current window of data
window = data(n_start:n_start+window_size-1,:);

% Store the window of data in the cell array
frame_cell{end+1} = window;

% Update the start index for the next window
n_start = n_start + window_size - overlap_size;
end
```

Figure 3) Function to apply overlapping windowing.

4.4 Feature Extraction

Once the data has been segmented, the dimensionality of the original data must be reduced to a set of informative and significant features. These features not only reduce the complexity of the data, but they can also enhance a classifier's performance and accuracy. To accomplish this, ZC, SSC, MAC, and WL from the Hudgins Time Domain [24] are utilised. The function below iterates through the cell containing the segmented data and stores the data in an array of features in order to apply these features to the data. The result of the feature array is a nx32 matrix, where n is the number of segments into which the original EMG data has been divided.

```
function features = extract_features(data)
% Initialise an empty matrix to store the features
features = [];
% Loop over each window
for i = 1:length(data)
% Get the current window
window = data{i};
window_features = [];
% Loop over each column (channel) of the window
for j = 1:size(window,2)
% Get the data for the current channel
channel_data = transpose(window(:,j));
% Compute features
mav_fe = jfemg('mav',channel_data);
zc_fe = jfemg('zc',channel_data);
ssc_fe = jfemg('ssc',channel_data);
wl_fe = jfemg('ewl',channel_data); % Enhanced WL
% Append features to the window's feature vector
window_features = [window_features mav_fe zc_fe ssc_fe wl_fe];
end
% Append the window's feature vector to the matrix of features
features = [features; window_features];
end
end
```

Figure 4) Function to extract features from data.

4.5 Classification

Two sets of data, one for training and one for testing, were used to construct the classifier. There was a 70% training to 30% testing ratio. The training set was also subjected to 10 K-Fold cross validation. After dividing the data, the 10-fold classification error was calculated to determine whether the model overfit or underfit the data. Changes were made to the window and overlap sizes of the segmentation method (sizes are defined in section 4.3) and the features used in the feature extraction technique in order to reduce the

```
% Extracts and stores the features and labels seperately to be used in training
features = feature_table{:, 1:32};
labels = feature_table.Movement;
% Split the data into training and test sets
rng('default'); % For reproducibility
cvPartition = cvpartition(labels, 'Holdout', 0.3);
trainingData = features(cvPartition.training,:);
trainingLabels = labels(cvPartition.training);
testData = features(cvPartition.test,:);
testLabels = labels(cvPartition.test);
% Perform 10-fold cross-validation
lda = fitcdiscr(trainingData, trainingLabels, "DiscrimType", "linear");
cvmodel = crossval(lda, 'KFold', 10);
classError = kfoldLoss(cvmodel); % Compute the classification error
fprintf('10-fold cross-validated classification error: %.2f%%\n', classError*100);
% Training Over Entire Dataset
Mdl = fitcdiscr(trainingData, trainingLabels, "DiscrimType", "linear");
```

Figure 5) Segment of the main code where an LDA classifier is trained (Training a SVM is similar to this implementation).

error. These modifications did not involve a feature set change, such as switching from WL (a Time Domain Feature) to a Fourier Domain feature. Simply put, the EMG Feature Extraction Toolbox provided enhanced variants of features like Enhanced WL and Enhanced MAV. Once the classification error was minimised to the greatest extent feasible. Training on the entire data set commences. As previously stated, it was intended to train an LDA and SVM classifier, and the following section demonstrates how their performance was evaluated.

4.6 Performance Evaluation

After training the classifier, the performance of the model was evaluated. Using the 'predict' function, the model was tested against the testing set and obtained results indicating how well the classifier performed on unobserved data. In section 3.4.5, accuracy, confusion matrix, classification error, precision, recall, F1-score, and Log Loss are listed as classifier evaluation techniques. In contrast to LDA classifiers, MATLAB parameter functions determine SVM log loss differently than LDA classifiers. Below is the code for its implementation:

```
% Evaluation
predictedLabels = predict(Mdl, testData);

% Confusion matrix
confusionMat = confusionmat(testLabels, predictedLabels);
confusionchart(testLabels, predictedLabels, "Title", 'LDA Model', ...
"ColumnSummary","column-normalized","RowSummary","row-normalized");

% Accuracy
accuracy = sum(diag(confusionMat)) / sum(confusionMat(:));
fprintf('Accuracy on the test set: %.2f%%\n', accuracy*100);

% Calculate Individual Class Errors
numClasses = size(confusionMat, 1); % Number of classes
classError = zeros(numClasses, 1);
for i = 1:numClasses % Compute class error for each class
correctClassifications = confusionMat(i, i);
totalInstances = sum(confusionMat(i, :));
classError(i) = (totalInstances - correctClassifications) / totalInstances;
end
for i = 1:numClasses % Display class error
fprintf('Class %d Error: %.2f%%\n', i, classError(i)*100);
end

% Overall Classification Error
classificationError = loss(Mdl, testData, testLabels);
fprintf('Overall Class Error: %.2f%%\n', classificationError*100);

% Precision, Recall, F1-Score
numClasses = numel(unique(testLabels));
precision = zeros(numClasses, 1);
recall = zeros(numClasses, 1);
f1 = zeros(numClasses, 1);
for i = 1:numClasses
precision(i) = confusionMat(i,i) / sum(confusionMat(:,i));
recall(i) = confusionMat(i,i) / sum(confusionMat(i,:));
f1(i) = 2 * (precision(i) * recall(i)) / (precision(i) + recall(i));
end
avgPrecision = mean(precision, 'omitnan');
avgRecall = mean(recall, 'omitnan');
avgF1 = mean(f1, 'omitnan');
fprintf('Precision: %.3f\n', avgPrecision);
fprintf('Recall: %.3f\n', avgRecall);
fprintf('F1 Score: %.3f\n', avgF1);

% Log Loss
logLoss = loss(Mdl, testData, testLabels, 'LossFun', 'crossentropy');
fprintf('Log Loss: %.3f\n', logLoss);
```

Figure 7) Implementation of machine learning evaluating metrics.

```
% Log Loss
logLoss = loss(Mdl, testData, testLabels, 'BinaryLoss', 'logit');
```

Figure 6) Implementation of log loss for SVM classifier

5 Results & Discussion

This section evaluates the performance of the LDA and SVM classifiers. Window size and overlap size were varied throughout the tests to determine which parameters produced the best results.

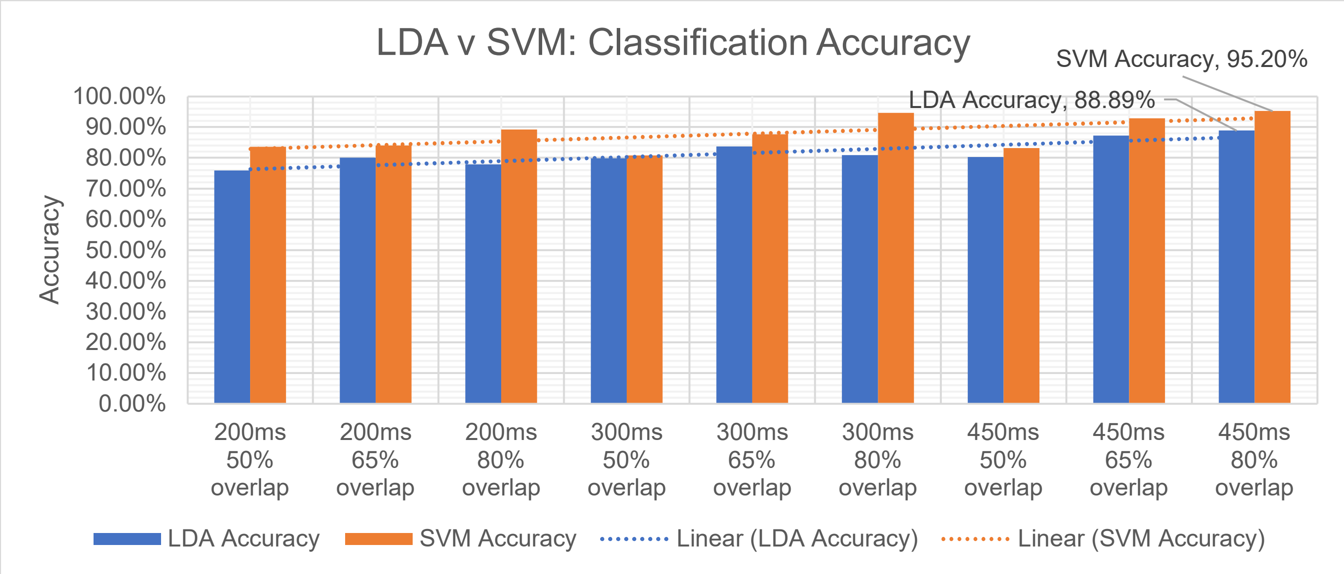

Figure 8) Classification accuracy of classifiers vs the specified window and overlap size.

With larger window size and overlap size, accuracy increases. Referring back to the 300ms limit for real-time operation, the SVM classifier performs well with an accuracy of 94.61% at 80% overlap, while the LDA classifier performs poorly with an accuracy of 80.92%. This variance in precision is present throughout the data. With a maximum accuracy of 95.20% at 450ms and 80% overlap; SVM appears to be a promising classifier. In addition, SVM performs better with higher percentages of overlap, whereas LDA's performance varies between percentages of overlap, indicating that overlap is not a significant factor effecting the performance of an LDA classifier, whereas data indicates that window size can be.

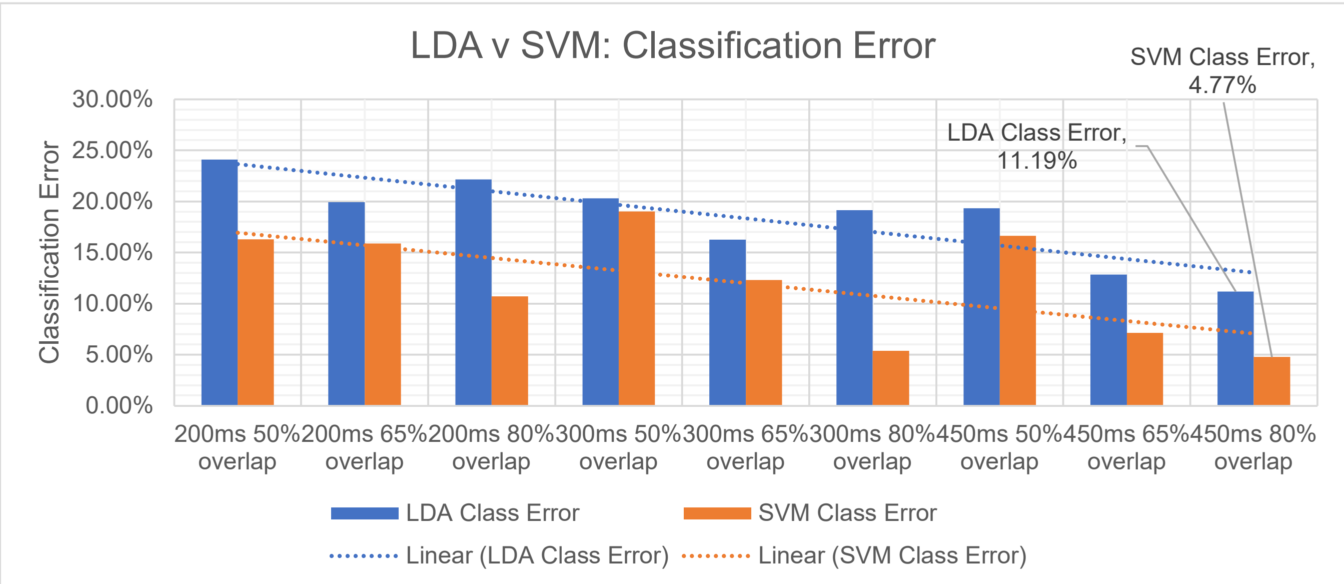

Figure 9) Classification error of classifiers vs specified window and overlap size.

Once more, the SVM classifier appears to be the most effective by significant margins. At 300ms and 50% overlap, the performance of the LDA and SVM classifiers are comparable; thereafter, there is a distinct winner in this category. The SVM performs well at 300ms and 80% overlap with a classification error of 5.37 percent, suggesting that it is suitable for real-time applications. However, 450ms with an overlap of 80% indicates promise as it yields a maximum reading of 4.77 percent. The classification accuracy and error results suggest that if your window size exceeds the prescribed limit of 300ms, you can compensate for the longer duration by increasing the overlap size. This is supported by the fact that the LDA classifier's performance is also improving, but this assertion must be tested to be confirmed. In addition, the percentage of overlap appears to play a significant role in the performance of SVMs, such that the larger the overlap, the lower the class error. On the other hand, this does not apply to an LDA classifier, as 65% overlaps perform well for window sizes of 200 and 300ms, but 80% overlaps perform marginally better for window sizes of 450ms. To restate, it is evident that a larger window size improves the performance of both the LDA classifier and the SVM classifier.

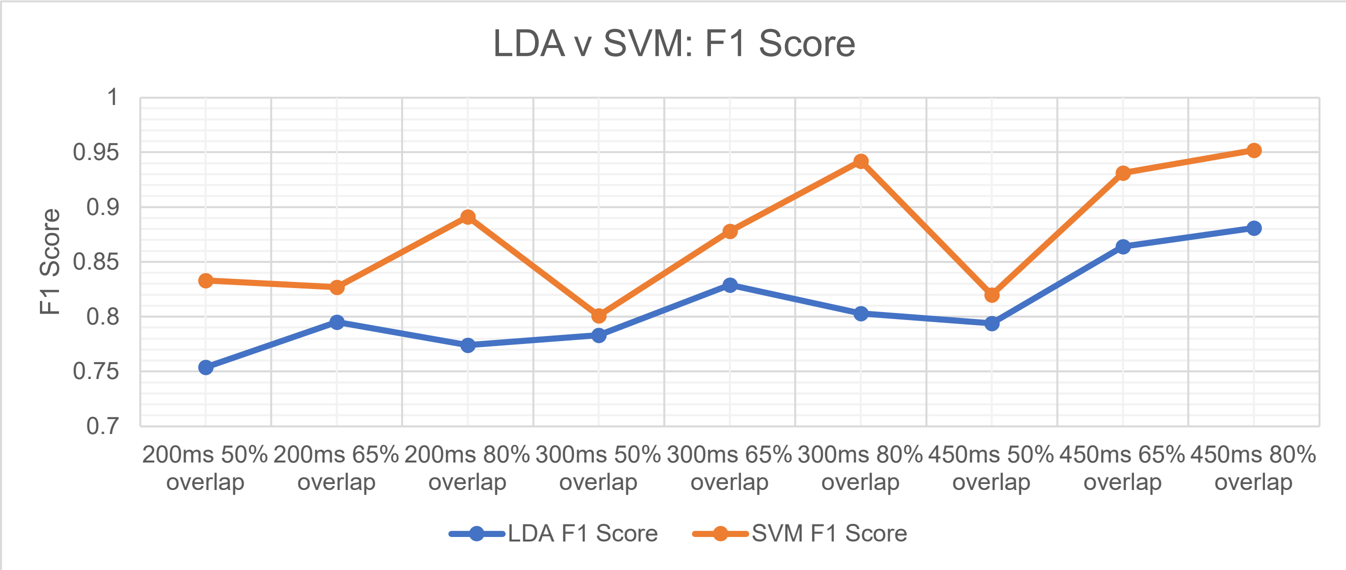

Figure 11) F1 score vs specified window and overlap size.

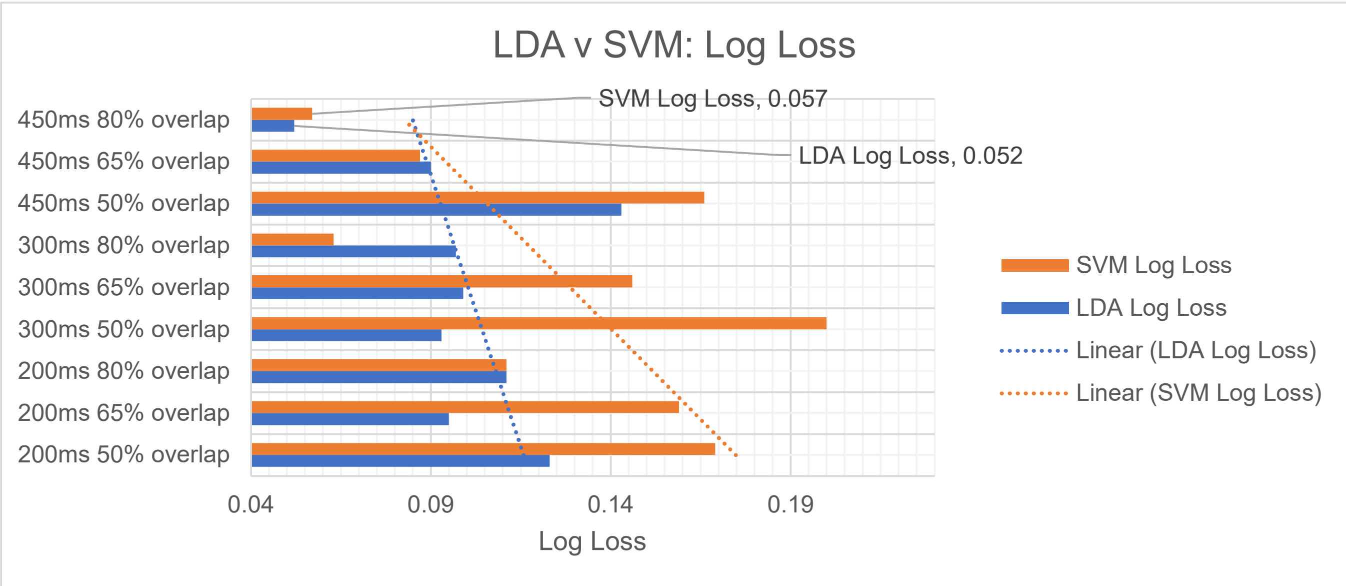

Figure 10) Log Loss vs specified window and overlap size.

In terms of the F1 score, the SVM classifier outperforms the LDA, indicating that it is more precise and robust. This characteristic is also highly desired in online training, making the SVM classifier preferable to the LDA classifier. In addition, the F1 score demonstrates that a higher overlap percentage leads to improved performance for an SVM classifier but not for an LDA classifier. Nevertheless, despite outperforming the LDA in three distinct categories, the SVM classifier is not the strongest in log loss. SVM's maximum log loss is 0.2 and its minimum is 0.057, which is near to LDA's score of 0.052. This is essential to note because it indicates that even though LDA has a lower accuracy than SVM, the likelihood of a true positive prediction is high. Although the SVM is outperformed on this metric, its values are still low and close to zero, making it a viable alternative. When evaluating a model, it's helpful to comprehend its overall performance, but it's also important to observe it in greater detail. This is where the confusion matrix comes in; it can provide information on where and to what extent things are being misclassified.

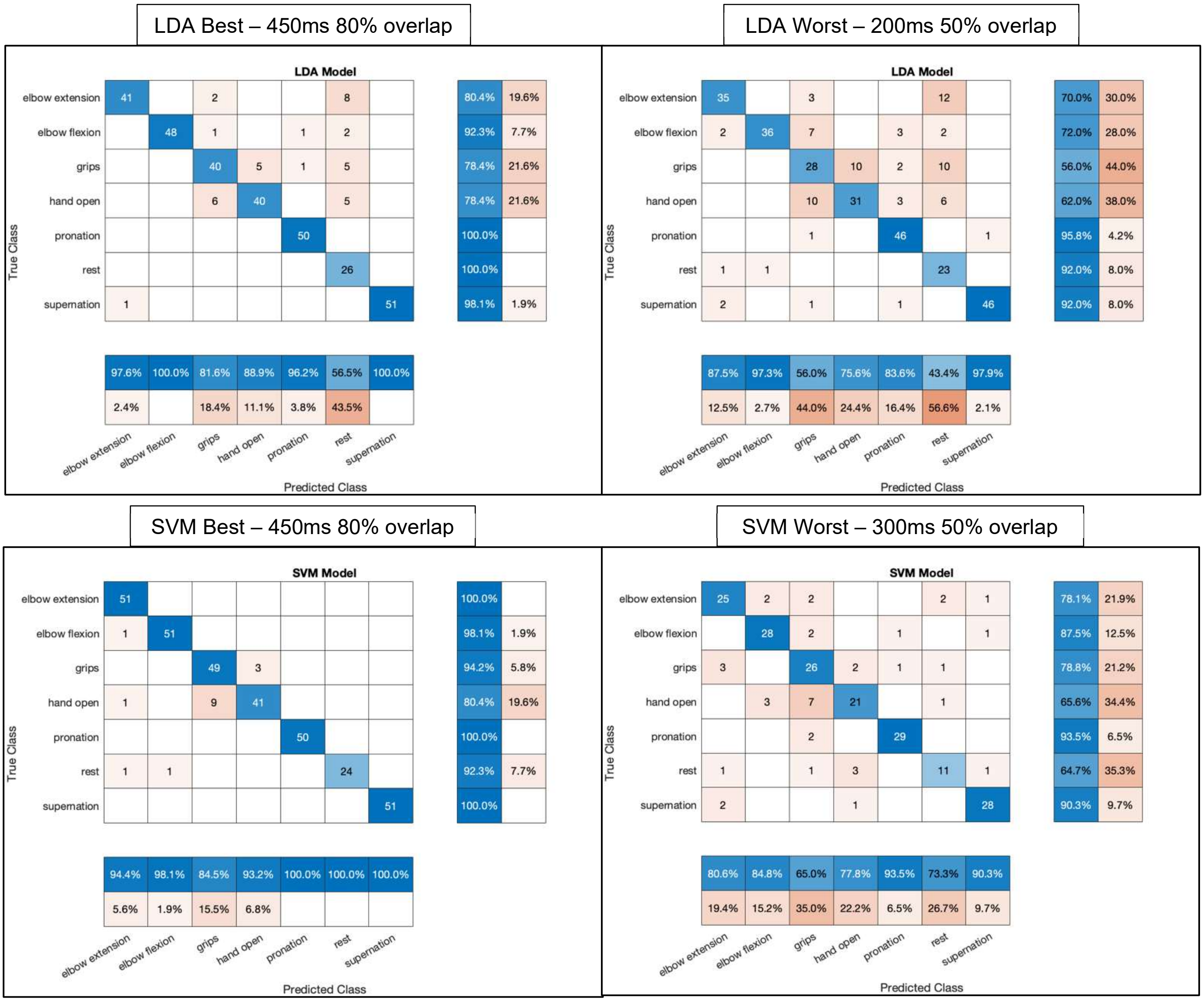

Figure 12) Collection of confusion matrices showing the best and worst classifications of the LDA and SVM classifier. Classification accuracy of LDA best is 88.98%, LDA worst is 75.85%, SVM best is 95.20% and SVM worst is 80.77%.

Analysing Figure 12 reveals that both predictors perform best with supernation, elbow flexion, and pronation, but poorly with the remaining classes. In general, the LDA classifier performs poorly in the rest position, whereas SVM performs relatively well. SVM misclassifies four classes at most, but three of those classes are under 7% for the lower column, with the highest class under 16%. On the other hand, LDA best misclassifies five items, but two of these classes are below 12% and the maximum is above 40%. This difference in performance demonstrates that when an SVM prediction is incorrect, it occurs at a lower frequency than when an LDA prediction is incorrect. The SVM worst misclassifies all seven classes with only four below 20% and nothing above 35%, while the LDA worst misclassifies all seven classes with only four below 20% and two above 40%. Similarly, SVM performs better in comparison to the LDA worst model. When an SVM classifier commits an error, it is still adequate, but can be enhanced. Nevertheless, an LDA classifier is insufficient. Comparing these results to those of other studies, the LDA classifier that was trained performed worse than anticipated, which may be the result of methodological errors. Other studies achieved 95.4% [31], 95.64% [32], and 97.5% [33] compared to this project's maximum accuracy of 88.98%. However, the trained SVM classifier performed as expected. For example, the accuracy of three articles was 98% and 96% [26] from one article, 73% [34] from another, and 95.8% [35] from the third. To improve the LDA classifier, it may be necessary to modify the feature extraction method, train the classifier on more data, or introduce a feature selection method to further reduce the data's dimensionality. Despite the claim that an LDA classifier is insufficient, it should not be ruled out because the average classification accuracy and error are 81.62 and 18.35 percent, respectively. This indicates that the classifier performs admirably, but not as well as the SVM classifier.

6 Professional Matters

6.1 Engineering Professional and Ethical Responsibilities

Engineering, as a field, is more than just technical knowledge. It represents a dedication to societal values, environmental conservation, and ethical principles. The Engineering Council of the United Kingdom, the principal organisation supervising the engineering profession, emphasises the need of engineers being not just technically competent but also genuinely devoted to sustaining societal and ethical ideals.

6.2 Maintaining Professional Standards

The data produced from the clinical study NCT03097978 [30] for EMG signal classification is a testament to the delicate balance between innovation and responsibility in the field of biomedical engineering. While the potential for detecting trends and making predictions is considerable, it is critical to treat this data with care, safeguarding the participants' privacy and confidentiality. The Engineering Council's data protection and ethical information usage principles serve as a foundational pillar, ensuring that engineering applications are transparent and ethical.

6.3 Social and Environmental Responsibility

This project's bigger objective includes not just technological improvements but also improving the quality of life for amputees or those born with missing limbs. This endeavour has significant societal ramifications. Prosthetic technology advancement has the potential to help not just individuals but also society, such as lowering healthcare expenses and increasing inclusion for people with disabilities. Environmentally, the emphasis on developing efficient solutions translates to energy-efficient gadgets, decreasing the overall environmental imprint and encouraging sustainability.

6.4 Ethics

The ethical implications of exploiting clinical trial data are critical. It's nice to know that the project's approach is consistent with the engineering community's professional standards and ethics. Every piece of data was obtained with the informed consent of participants, demonstrating the dedication to their rights and autonomy. Furthermore, the research's possible good consequences are consistent with the ethical ideal of maximising society benefits.

6.5 Summary

This project embodies the core of engineering: a well-balanced mix of technical innovation, societal effect, and ethical responsibility. It stands as a tribute to the principles that the engineering profession promotes by following to the Engineering Council's guidelines and pondering on the broader consequences.

7 Conclusion

The classification of EMG signals using machine learning techniques, specifically the LDA and SVM classifiers, has been a thorough investigation into biomedical engineering. The overarching goal was to utilise computational power to enhance the lives of amputees and those born with limb deficiencies, a noble endeavour rooted in engineering ethics and societal advancement. The thorough evaluation of classifiers with varying window and overlap diameters has revealed their strengths and limitations. The SVM classifier emerged as the victor, consistently outperforming the LDA classifier across the majority of metrics. Its adaptability to various overlap sizes and commendable precision, particularly within the critical 300ms real-time operation window, demonstrate its viability for real-world applications. While the LDA classifier showed promise, its performance was inconsistent, suggesting areas for further development. A deeper examination of performance metrics, such as the F1 score and log loss, highlighted the SVM's robustness even further. Even though their accuracy rates differ, the close log loss values between the two classifiers indicate that both have a high degree of confidence in their predictions. Particularly illuminating was the confusion matrix analysis, which revealed specific areas of strength and vulnerability for each classifier. While both classifiers excelled at distinguishing certain movements, their misclassification rates for others varied significantly. Such knowledge is indispensable for future optimisation efforts. Comparing our results to the existing research reveals that while the performance of the SVM is comparable to that of other studies, the performance of the LDA was somewhat subpar. This proposes possible improvement opportunities, such as feature extraction, data augmentation, or dimensionality reduction. The project's results are favourable in the grand scheme of things. With its superior accuracy and adaptability, the SVM classifier stands out as a potential instrument for real-time EMG signal classification. Despite its difficulties, the LDA classifier should not be disregarded. Its average performance metrics indicate that, with additional refinement, it, too, can be a valuable asset in the field of prosthetic technology. This research has established a firm foundation for future efforts in EMG signal classification. The insights gained, coupled with the ethical and professional standards maintained throughout the project, pave the way for innovations that have the potential to substantially improve the quality of life for many. As the field of biomedical engineering continues to evolve, initiatives such as this highlight the significance of continuous learning, adaptation, and the pursuit of excellence.

8 References

[1] N. Digital, “NCARDRS Congenital Anomaly Official Statistics Report,” National Disease Registration Service (NDRS), 2022.

[2] S. Whelan, “Preventing amputations major concern as diabetes numbers rise,” Public Health England, London, 2019.

[3] CDC, “Facts about Upper and Lower Limb Reduction Defects,” CDC, 2022.

[4] H. C. &. U. Project, “NIS Database Documentation,” Agency for Healthcare Research and Quality, [Online]. Available: https://www.hcup-us.ahrq.gov/db/nation/nis/nisdbdocumentation.jsp. [Accessed 18 1 2023].

[5] K. Ziegler-Graham, E. MacKenzie, P. Ephraim, T. Travison and R. Brookmeyer, “Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050,” *Archives of Physical Medicine and Rehabilitation,* vol. 89, no. 3, pp. 422-429, 2008.

[6] L. N. Herzog, C. A. Johnson and C. E. Gross, “A database study on incidence and trends in traumatic lower-extremity amputations in the United States,” *Current Orthopaedic Practice,* vol. 33, no. 4, pp. 329-333, 2022.

[7] healthdirect, “Prostheses,” [Online]. Available: https://www.healthdirect.gov.au/prostheses. [Accessed 20 January 2023].

[8] J. J. Stokosa, “Limb Prostheses Options,” American Prosthetics Institute, [Online]. Available: https://www.msdmanuals.com/en-gb/home/special-subjects/limb-prosthetics/limb-prostheses-options. [Accessed 19 January 2023].

[9] M. Raez, M. Hussain and F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications,” *Biol Proced Online,* pp. 11-35, 2006.

[10] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan and T. Chang, “Surface electromyography signal processing and classification techniques,” *Sensors (Basel),* p. 12431–12466, 2013.

[11] “Electrogram,” [Online]. Available: https://en.wikipedia.org/wiki/Electrogram. [Accessed February 2023].

[12] V. Gohel and N. Mehendale, “Review on electromyography signal acquisition and processing,” *Biophys Rev,* p. 1361–1367, 2020.

[13] “"Innervate",” Merriam-Webster.com Dictionary, [Online]. Available: https://www.merriam-webster.com/dictionary/innervate. [Accessed February 2023].

[14] “Neuromuscular junction, how’s that function?,” National Institute of Biomedical Imaging and Bioengineering (NIBIB), 2022. [Online]. Available: https://www.nibib.nih.gov/news-events/newsroom/neuromuscular-junction-how%E2%80%99s-function#:~:text=The%20neuromuscular%20junction%E2%80%94where%20nerves,(like%20Lou%20Gehrig's%20disease). [Accessed February 2023].

[15] B. Farnsworth, “What Is EMG (Electromyography) and How Does It Work?,” iMotions, 24 July 2018. [Online]. Available: https://imotions.com/blog/learning/research-fundamentals/electromyography-101/. [Accessed February 2023].

[16] P. Geethanjali, “Myoelectric control of prosthetic hands: state-of-the-art review,” *Med Devices ,* vol. 9, pp. 247-255, 2016.

[17] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan and T. Chang, “Surface Electromyography Signal Processing and Classification Techniques,” *Sensors 2013,* no. 13, pp. 12431-12466, 2013.

[18] C. De Luca, L. Donald Gilmore, M. Kuznetsov and S. Roy, “Filtering the surface EMG signal: Movement artifact and baseline noise contamination,” *Journal of Biomechanics,* vol. 43, no. 8, pp. 1573-1579, 2010.

[19] O. Samuel, M. Asogbon, Y. Geng, S. Pirbhulal, N. Ji, S. Chen, P. Fang and G. Li, “Intelligent EMG Pattern Recognition Control Method for Upper-Limb Multifunctional Prostheses: Advances, Current Challenges, and Future Prospects,” *IEEE Access,* vol. 7, pp. 10150-10165, 2019.

[20] F. Farfán, J. Politti and C. Felice, “Evaluation of EMG processing techniques using Information Theory,” *BioMed Eng OnLine 9,* no. 72, 2010.

[21] O.-C. M, B. R and H. B, “BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms,” *Source Code Biol Med,* vol. 8, no. 11, 2013.

[22] M. Oskoei and H. Hu, “Myoelectric control systems—A survey,” *Biomedical Signal Processing and Control,* vol. 2, no. 4, pp. 275-294, 2007.

[23] B. R and M. MH, “Evaluation of the forearm EMG signal features for the control of a prosthetic hand,” *Physiol Meas,* vol. 24, pp. 309-19, 2003.

[24] B. Hudgins, P. Parker and R. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Trans Biomed Eng,* vol. 40, pp. 82-94, 1993.

[25] 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.

[26] A. Ahmet and G. Mücahid, “Identification of EMG signals using discriminant analysis and SVM classifier,” *Expert Systems with Applications,* vol. 39, no. 1, pp. 44-47, 2012.

[27] M. Zardoshti-Kermani, B. C. Wheeler, K. Badie and R. M. Hashemi, “EMG feature evaluation for movement control of upper extremity prostheses,” *IEEE Transactions on Rehabilitation Engineering,* vol. 3, no. 4, pp. 324-333, 1995.

[28] F. Mattioli, E. J. Lamounier, A. Cardoso, A. Soares and A. Andrade, “Classification of EMG signals using artificial neural networks for virtual hand prosthesis control,” *Annu Int Conf IEEE Eng Med Biol Soc,* pp. 7254-7, 2011.

[29] J. Too, A. Abdullah, N. Saad, M and W. Tee, “EMG Feature Selection and Classification Using a Pbest-Guide Binary Particle Swarm Optimization,” *MDPI AG,* vol. 7, no. 12, 2019.

[30] S. R. AbilityLab, “Comparison of Pattern Recognition Control and Direct Control in TMR,” ClinicalTrials.gov, Chicago, 2018.

[31] X. Zhang, H. Huang and Q. Yang, “Real-time implementation of a self-recovery EMG pattern recognition interface for artificial arms,” *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Bio. Soc.,* pp. 5926-5929, 2013.

[32] D. C. Tkach, A. J. Young, L. H. R. E. J. Smith and L. J. Hargrove, “Real-time and offline performance pf pattern recognition myoelectric control using a generic electrode grid with targeted muscle reinnervation patients,” *IEEE Trans. Neural Syst. Rehabil. Eng.,* vol. 22, no. 4, pp. 727-734, 2014.

[33] L. Pan, D. Zhang, J. Liu, X. Sheng and X. Zhu, “Continuous estimation of finger joint angles under different static wrist motions from surface EMG signals,” *Biomed. Signal Process. Control,* vol. 14, pp. 265-271, 2014.

[34] T. Mastubara and J. Morimoto, “Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface,” *IEEE Trans. Biomed. Eng.,* vol. 60, no. 8, pp. 2205-2213, 2013.

[35] A. Stango, F. Negro and D. Farina, “Spatial correlation of high density EMG signals provides features robust to electrode number and shift in pattern recognition for myocontrol,” *IEEE Trans. Neural Syst. Rehabil. Eng.,* vol. 23, no. 2, pp. 189-198, 2015.

[36] J. Dailami, “The Myoelectric Arm: It’s Electrifying,” Illumin Magazine, May 2002. [Online]. Available: https://illumin.usc.edu/the-myoelectric-arm-its-electrifying/. [Accessed 21 January 2023].

[37] J. T. Shores, “Amputation,” John Hopkins Medicine, [Online]. Available: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/amputation#:~:text=Amputation%20can%20be%20traumatic%20(due,before%20birth%2C%20called%20congenital%20amputation. [Accessed February 2023].

[38] E. Campbell, A. Phinyomark and E. Scheme, “Current Trends and Confounding Factors in Myoelectric Control: Limb Position and Contraction Intensity,” *Sensors,* 2020.

[39] L. Hargrove, P. Zhou, K. Englehart and T. Kuiken, “The Effect of ECG Interference on Pattern-Recognition-Based Myoelectric Control for Targeted Muscle Reinnervated Patients,” *IEEE Transactions on Biomedical Engineering,* vol. 56, no. 9, pp. 2197-2201, 2009.

9 Appendix

9.1 Appendix A

Link to GitHub repository: https://github.com/kakashim0t0/Classifying-EMG-signals.git