Optimised EMG pipeline for gesture classification

Jarlath Warner[1], Richard Gault[1], and John McAllister[1]

***Abstract*— In the expanding field of robotic prosthetics, surface electromyography (sEMG) signals can be decoded to seamlessly control a robotic prosthesis to perform the desired gesture. It is essential to create a pipeline, which can acquire, process, and accurately classify sEMG signals in order to replicate the desired hand gesture in near real-time and in a reliable manner. In this study, an optimised pipeline is proposed. This pipeline encompasses the main stages of sEMG signal processing and hand gesture classification and implements a sliding window approach, which is the main focus of the optimisation. In this study, a range of different parameters and modelling approaches are evaluated. The main contributions of this work are a robust and extensive analysis of sliding window parameter selection and an optimised pipeline that could be implemented in practice with minimal overheads. The optimum pipeline is efficient and achieves accurate prediction of hand gestures with an uninterrupted processing pipeline.**

I. INTRODUCTION

Robotic prosthetics can help restore mobility to individuals who have suffered an amputation or limb impairment. With recent developments in computational models, complex, non-linear data such as those captured by surface electromyography (sEMG) can be accurately modelled to enable classification tasks. This requires an analysis pipeline which can efficiently conduct the various stages of processing required for the classification of EMG signals. This study aims to fine tune and build upon existing classification pipelines [1], [2], [3] for gesture recognition based on EMG signals captured from a person's forearm. Section II will review some of the key methodologies considered in recent studies before the proposed experimentation is outlined in Section III. Section IV details the findings of the experiments that are subsequently discussed in Section V. The paper concludes with a summary of the paper's findings and direction for future work (Section VI).

II. BACKGROUND

Previous studies [1], [2], [3] have achieved very accurate classification of EMG signals following 3 principle stages; signal pre-processing, feature extraction, and classification. The studies primarily follow the same pre-processing steps, however they slightly diverge at the feature extraction stage. There is a balance in the amount of time needed to take to process and classify the signal and the performance of the analysis pipeline.

A sliding window approach has been shown to be an effective way to handle the sampling of EMG data. A window of raw EMG signal is captured and processed at one time. Studies such as [1] and [3] utilise commercially available wireless MYO EMG Armband which is less reliant on the precise electrode placement compared with the wet sensor approach in [2]. [3] and [1] showed, using similar input features, that it is possible to classify hand gestures from MYO Armband recordings with high accuracy (98% and 98.7% respectively) with minimal processing time (9.75ms and 227.76ms respectively). However, both studies differed in their choice of window length and stride length (i.e. the number of time steps considered in a single window and the distance/time between consecutive windows). Given that [1] was, on average, able to perform more accurately but take longer time than [3] it motivates the question: is there an optimum window and stride length for sliding window approaches to EMG classification?

III. METHODS

A. Dataset

This study utilises an existing EMG dataset [4], [5] captured using the MYO Thalmic bracelet with 8 channels sampled at a fixed sampling rate of 200Hz [6], [7]. In total, 36 subjects performed six (or in some cases seven) hand gestures giving eight class labels that the signal could belong to, (1) hand at rest, (2) hand clenched in a fist, (3) wrist flexion, (4) wrist extension, (5) radial deviations, (6) ulnar deviations, (7) extended palm, and finally class label 0 representing "unmarked data". These gestures are illustrated in Figure 1. Note that Gesture 7 ("extended palm") was only performed by 2 subjects. The duration of each gesture was 3 seconds with a pause of 3 seconds between gestures in the sessions. It should be noted that the dataset includes a ground truth label at every time-point. Real world adoption of EMG controlled robotic prosthesis needs the inclusion of a null class (label 0) so that a trained model can still interpret the EMG signal and not insist on the occurrence of a gesture. However, the dataset contains vastly more instances of unmarked data (or null movements) than of the 7 prescribed gestures. To improve the balance between the classes a random selection of epochs labelled 0 are chosen equal to the average number of epochs available for gestures 1 - 7.

B. Preprocessing

The EMG signals are pre-processed in keeping with previous studies [3], [1], [2], first normalising all 8 channels so that signal values are in the range [-1, 1] (using `sklearn.preprocessing.MinMaxScaler`) and then applying positive rectification. A 4th order low-pass digital

[1]All authors are from the School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, UK ({jwarner02, richard.gault, jp.mcallister}@qub.ac.uk).

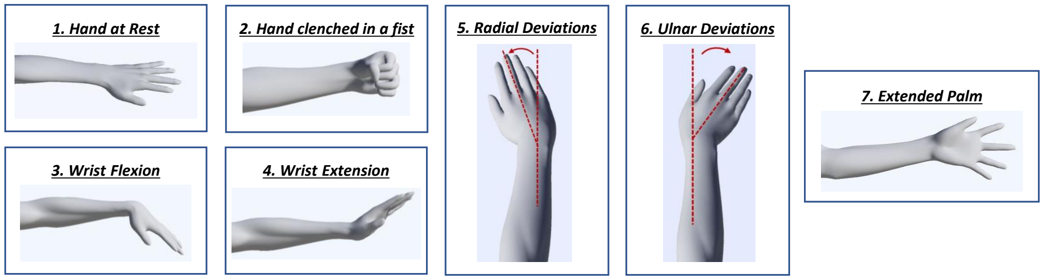

Fig. 1. Visual representation of the gestures performed by participants in the experimental paradigm.

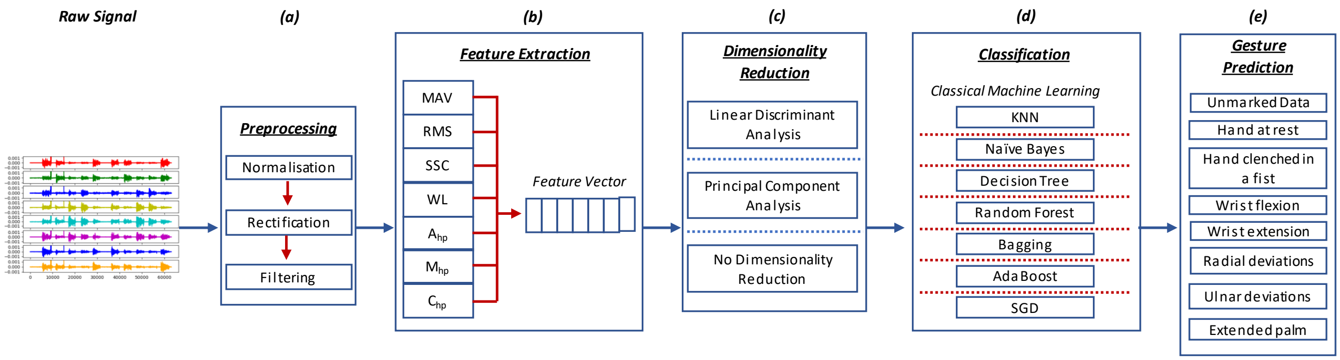

Fig. 2. Figure illustrating the overall signal processing pipeline considered in this study.

Butterworth filter with a 5Hz cut-off frequency is then applied. Next, a window of predetermined length is extracted from a signal buffer across all 8 channels and various features (described in Section III-C) are calculated. Once the features have been calculated for this window, the window then slides across the signal buffer, at a predetermined stride length and the feature extraction process is repeated. In this study a number of different window lengths from 100-500ms are investigated in steps of 50ms and stride sizes ranging from 25-150ms in steps of 25ms to identify the optimum parameters for the sliding window. If the scenario arises where the window contains more than 1 class label, for example during the transition from one gesture to the next gesture, then the majority class label is taken for that window.

C. Feature Extraction

This section will outline the 7 features extracted for each window in the signal buffer (Figure 2c) based on features used in [1], [2], [3]. These features are defined as follows:

1) MAV (Mean Absolute Value): The Mean Absolute Value (MAV) is the average of the absolute values of the amplitude of the signal in the window and captures muscle contractions. It is calculated by

$$MAV = \frac{1}{N}\sum_{k=1}^{N}\left|s(k)\right| \tag{1}$$

where N is the total number of time points and s is a single channel of the EMG signal.

2) RMS (Root Mean Square): Given a signal, x, over N time steps, the RMS of the signal is calculated as

$$RMS = \sqrt{\frac{1}{N}\sum_{k=1}^{N}x(k)^2} \tag{2}$$

3) SSC (Slope Sign Change): The SSC is used to describe the frequency information within the data.

$$SSC = \sum_{k=2}^{N-1} f(s(k)-s(k-1))\cdot(s(k)-s(k+1))) \tag{3a}$$

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3b}$$

4) WL (Waveform Length): The Waveform Length (WL) is the cumulative amount of change in the EMG signal defined as

$$WL = \sum_{k=2}^{N}\left|s(k) - s(k-1)\right| \tag{4}$$

5) Hjorth Parameter (Activity, Complexity and Morbidity): Hjorth Parameter comprises three components representing Activity (A_{hp}), Complexity (C_{hp}) and Mobility (M_{hp}) and has been used in previous EMG studies with great success. Equations eqs. (5) to (7) respectively define the Activity, Mobility and Com-

plexity.

$$A_{hp} = \sigma^2(s(k)) = \frac{1}{N-1}\sum_{k-1}^{N} s(k)^2 \quad (5)$$

$$M_{hp} = \sqrt{\frac{\sigma^2(\frac{ds(k)}{dt})}{\sigma^2(s(k))}} \quad (6)$$

$$C_{hp} = \frac{M_{hp}(\frac{ds(k)}{dt})}{M_{hp}(s(k))} \quad (7)$$

Where σ^2 is the variance of the signal and $\frac{ds(k)}{dt}$ represents the signal derivative. Note that the complexity function, C_{hp}, uses the Mobility function defined by Equation 6.

An additional step before modelling will investigate whether or not common dimensionality reduction approaches, namely Linear Discriminant Analysis (LDA) and Principal Components Analysis (PCA), are beneficial in classifying hand gestures. The features (with or without dimensionality reduction) are used as inputs to the models described in Section III-D.

D. Modelling Approaches

The models selected for this study present classical and time-tested approaches to act as a basis for the overall analysis of the signal processing pipelines and provide a general overview of how classical machine learning algorithms will perform. For this study, k-Nearest Neighbours (KNN), Naive Bayes, Decision Tree, Random Forest, Bagging, AdaBoost and Stochastic Gradient Descent (SGD) were investigated.

The model hyperparameters were arbitrarily chosen and may be sub-optimal for classification, however, this investigations primary focus is to investigate changes in window length and stride length rather than the optimal modeling approach and model configuration. The KNN was implemented using a neighbourhood of 5. Naive Bayes used default arguments from `Scikit-learn` with a variable smoothing value of 1e-9. The Decision Tree used a max depth of 30. Random Forest and Bagging used 5 estimators. AdaBoost used 15 estimators and SGD used 4 estimators and 4 max features. Note that for model parameters which are not explicitly mentioned above, it can be assumed that the default arguments were used, as specified by the model implementations defined in `scikit-learn` [13]. A train-test-validation split of 50%-20%-30% was used with data samples randomly chosen across all subjects.

E. Experiments

Experiments were implemented in Python (v3.9) and executed on a PC with an Intel Core i7 9700K, 16GB DDR4 RAM and an NVIDIA GeForce RTX 2070 SUPER. Data preprocessing and feature extraction was completed utilising both `EEGLib` (v0.4) [11] and `Pysiology` (v0.0.9.4) [12]. Matrix representation of the EMG signals and subsequent matrix calculations utilised `Pandas` (v1.2.5) and `Numpy` (v1.21.0). Classical Machine Learning models were created using `Scikit-learn` (v0.24.2) [13].

The aim of this experiment is to determine the optimum combination of window length and stride length for making gesture predictions. Window lengths from 50ms to 500ms in 50ms increments were considered with stride lengths ranging from 5ms to 150ms in 5ms increments. Furthermore, the computational cost of executing the pipeline using a trained model was calculated to determine the feasibility of using the analysis pipeline in a real world situation. Ideally the analysis pipeline will have minimal latency to maximise the time a control system might have to operate a prosthetic device without noticeable latency.

IV. Results

Firstly we consider all the model pipelines where no LDA or PCA are used to reduce the input size. Random Forest had the overall highest accuracy of 98.2% which occurred when using a window length and stride length of 500ms and 5ms respectively. Table I shows the performance of the Random Forest model with different configurations of window length and stride (where there is a tie in the model performance across multiple configurations more decimal places were considered but condensed in the interest of space).It is not feasible to show the results for all configurations over all models in the space available within the current manuscript, however, it was observed that models configuration tended to favour large window lengths and small step sizes (Table II). A summary of the optimum window length and stride length for each model is outlined in Table II.

TABLE I
PERFORMANCE OF RANDOM FOREST UNDER DIFFERENT CONFIGURATIONS. HIGHEST ACCURACY PER WINDOW LENGTH IS IN BOLD. HIGHEST ACCURACY PER STRIDE LENGTH (SL) IS UNDERLINED.

SL	Window length								
	100	150	200	250	300	350	400	450	500
25	**97.1**	**97.3**	**97.3**	**97.4**	**97.7**	**97.8**	**97.9**	**98.1**	<u>**98.2**</u>
50	96.9	97.0	97.1	97.1	97.1	97.2	97.4	97.5	<u>97.6</u>
75	97.0	96.9	97.1	97.0	97.1	97.2	97.2	97.5	<u>97.6</u>
100	97.0	96.9	97.1	97.2	97.1	97.2	<u>97.2</u>	97.2	97.2
125	97.1	97.1	97.0	97.1	96.9	97.0	97.1	97.2	<u>97.3</u>
150	97.0	97.2	96.9	97.0	96.9	97.1	97.0	97.0	<u>97.3</u>

TABLE II
OPTIMUM SLIDING WINDOW CONFIGURATION WITH NO DIMENSIONALITY REDUCTION APPLIED.

Model	(Window length, Stride length)(ms)	Accuracy (%)
KNN	(500, 25)	87.3
Naive Bayes	(500, 125)	29.38
Decision Tree	(500, 150)	95.8
Random Forest	(500, 25)	98.2
Bagging	(500, 25)	98.0
AdaBoost	(350, 100)	31.7
SGD	(500, 125)	28.8

When dimensionality reduction was applied, model performance generally increased except in the case of SDG which performed better without any dimensionality reduction. Table III outlines the best performance achieved by each model and the dimensionality reduction method that was used (if any) to obtain this performance. LDA tended to improve performance more than PCA however the advantages were minimal (Table III), however, the highest overall accuracy was achieved when PCA was used.

All pipelines took less than 11ms to execute once the models were trained (Table IV). Whilst the Random Forest was one of the slower models to include in the pipeline the computational cost is minimal compared to the high performance in accuracy that it obtained. The inclusion of dimensionality reduction techniques had minimal or no computational overheads with average response time (+/-1ms) but generally enabled slightly higher performance.

TABLE III
MODEL ACCURACY FOR EACH CONFIGURATION (GIVEN AS (WINDOW LENGTH, STRIDE LENGTH) IN MS).

Model	LDA (Configuration) (%)	PCA (Configuration) (%)
KNN	(500, 25) **91.6**	(500,25) 87.3
Naive Bayes	(500, 150) **31.6**	(500,150) 30.7
Decision Tree	(500, 25) 96.3	(450,25) **96.6**
Random Forest	(500, 25) 98.1	(500, 25) **98.2**
Bagging	(500, 25) **98.0**	(500, 25) 97.95
AdaBoost	(350, 150) **32.3**	(350, 100) 31.1
SGD	(350, 50) 27.2	(500, 75) 27.4

TABLE IV
OPTIMUM PERFORMANCE FOR EACH MODEL WITH AND WITHOUT DIMENSIONALITY REDUCTION (DR) APPLIED.

Model	Time no DR (ms)	Time with DR (ms)
KNN	10.08	10.87
Naive Bayes	9.48	10.97
Decision Tree	9.18	8.98
Random Forest	10.57	10.28
Bagging	10.17	10.07
AdaBoost	8.97	10.48
SGD	9.28	9.58

V. DISCUSSION

The key finding of this work is that a larger sliding window lengths and smaller stride lengths will lead to improved performance with further benefits gained from applying dimensionality reduction. This could be due to the larger window length being able to capture more identifiable features of the EMG signal within the signal buffer, while also capturing any missed features due to the step size being smaller than the window size therefore, the next window would contain overlapping signals. In the cases of AdaBoost and SGD where this finding does not hold the classification performance is extremely low suggesting these modelling approaches should not be used in the current pipeline.

This study offers a defined pipeline covering the key stages of EMG data analysis that performs with near perfect accuracy in a very short time period which can be easily extended for other work. This study has also identified a feasible modelling approach that could be used in real world applications and be fine tuned for specific hardware implementations.

A limitation of this study is that it does not consider hardware implementation and the time costs associated with moving a prosthetic hand in real life. Although 8 discrete classes are considered in this study, they are categorical classes and this study does not consider the intermediary and incremental movements which would be realistic to those made by a user.

VI. CONCLUSION

This study presents an investigation into the optimum configuration for data analysis of an 8 channel EMG system for gesture classification, where such a system can be used in a real-time feedback solution. The results determined that a sliding window approach with sufficient window size and small step size can provide very high classification accuracy even when simple machine learning models are used. Future studies should consider more complex, intermediate stages of hand movements that are not captured by this categorical dataset. This research can be further extended to different scenarios of hardware implementation, offering a use case to areas other than just a prosthetic hand, such as robotic hand used in remote locations as well as other key EMG problems such as leg movements and prosthetic limbs.

REFERENCES

[1] Zhen Zhang et al. Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network. *Sensors*, 19:3170, 2019.

[2] Yang Zhou et al. EMG Signal Processing for Hand Motion Pattern Recognition Using Machine Learning Algorithms. *Archives of Orthopaedics*, 1:17–26, 2020.

[3] Raurale S., McAllister J., and del Rincon, JM. Real-time embedded EMG signal analysis for wrist-hand pose identification. *IEEE Transactions on Signal Processing*, 2713–2723, (68), 2020.

[4] Sergey Lobov et al.Latent Factors Limiting the Performance of sEMG-Interfaces. *Sensors*, 18:1122, 2018.

[5] Dua, Dheeru and Graff, Casey. UCI Machine Learning Repository. [https://archive.ics.uci.edu/ml/datasets/EMG+data+for+gestures] *"University of California, Irvine, School of Information and Computer Sciences"*, 2017.

[6] Phinyomark, Angkoon and Scheme, Erik. A feature extraction issue for myoelectric control based on wearable EMG sensors. *2018 IEEE Sensors Applications Symposium (SAS)*, 1-6, 2018.

[7] Mendez, I. and Hansen, B. W. and Grabow, C. M. and Smedegaard, E. J. L. and Skogberg, N. B. and Uth, X. J. and Bruhn, A. and Geng, B. and Kamavuako, E. N. Evaluation of the Myo armband for the classification of hand motions. *2017 International Conference on Rehabilitation Robotics (ICORR)*, 1211-1214, 2017.

[8] Thomas J., Maszczyk T., Sinha N., Kluge T., and Dauwels J. Deep learning-based classification for brain-computer interfaces. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 234–239, 2017.

[9] Yanzheng Lu, Hong Wang, Yangyang Qi, and Hailong Xi. Evaluation of classification performance in human lower limb jump phases of signal correlation information and LSTM models. *Biomedical Signal Processing and Control*, 64:102279, 2021.

[10] Dingbang Liu, Hao Yu, and Chai, Yang Low-Power Computing with Neuromorphic Engineering *Advanced Intelligent Systems*, 3:2000150, 2021.

[11] Cabañero-Gomez L., Hervas R., Gonzalez I., and Luis Rodriguez-Benitez eeglib: A Python module for EEG feature extraction *SoftwareX*, 15:100745, 2021.

[12] Gabrieli G., Azhari A., Esposito G. PySiology: A Python Package for Physiological Feature Extraction. *Neural Approaches to Dynamics of Signal Exchanges. Smart Innovation, Systems and Technologies*, 151, 2020.

[13] Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.