Electromyography Signal-based Gesture Recognition for Human-Machine Interaction in Real-Time through Model Calibration

Christos Dolopikos[1], Michael Pritchard[2] Jordan J. Bird[3], and Diego R. Faria[4]

ARVIS Lab (Aston Robotics, Vision, and Intelligent Systems Laboratory), Computer Science, Aston University, B4 7ET, Birmingham, United Kingdom
[1]`chdolopikos@gma il.com`; {[2]`pritcham`,[3]`birdj1`,[4]`d.faria`}`@aston.ac.uk`

Abstract. In this work we achieve up to 92% classification accuracy of electromyographic data between five gestures in pseudo-real-time. Most current state-of-the-art methods in electromyographical signal processing are unable to classify real-time data in a post-learning environment, that is, after the model is trained and results are analysed. In this work we show that a process of model calibration is able to lead models from 67.87% real-time classification accuracy to 91.93%, an increase of 24.06%. We also show that an ensemble of classical machine learning models can outperform a Deep Neural Network. An original dataset of EMG data is collected from 15 subjects for 4 gestures (Open-Fingers, Wave-Out, Wave-in, Close-fist) using a Myo Armband for measurement of forearm muscle activity. The dataset is cleaned between gesture performances on a per-subject basis and a sliding temporal window algorithm is used to perform statistical analysis of EMG signals and extract meaningful mathematical features as input to the learning paradigms. The classifiers used in this paper include a Random Forest, a Support Vector Machine, a Multilayer Perceptron, and a Deep Neural Network. The three classical classifiers are combined into a single model through an ensemble voting system which scores 91.93% compared to the Deep Neural Network which achieves a performance of 88.68%, both after calibrating to a subject and performing real-time classification (pre-calibration scores for the two being 67.87% and 74.27% respectively).

Keywords: Real-time Gesture Classification, Machine Learning, Deep Learning, Biosignal Processing, EMG

Acknowledgment

This work is partially supported by EPSRC-UK InDex project (EU CHIST-ERA programme), with reference EP/S032355/1 and by the Royal Society (UK) through the project "Sim2Real" with grant number RGS\R2\192498.

1 Introduction

Human-Machine interfaces have been relatively static in recent years, despite many technological advancements in both hardware and software. The efficiency and productivity of conventional devices for interacting with machines has peaked, and therefore alternative means of interaction have to be further explored. The realization that human-beings are the only source of input in this relation is important as it encourages us to shift the balance of Human-Machine interaction closer to the human aspect. As humans we use our hands to interact with devices and machines every day; in order to perform actions such as typing, bioelectrical signals are carried through the spinal cord from the brain to the muscle motor neurons to generate muscular activity [20]. Thalmic Labs' Myo Armband [2] features a set of Electromyography (EMG) sensors [21] and an inertial measurement unit (IMU) [27], and can provide an alternative interface by reading that forearm bioelectricity. In this work we classify between four distinct hand gestures (and a fifth class for the neutral, or resting, position) using electromyographic data obtained with a Myo band. Although human beings are biologically similar [10], there are small differences among them which can significantly affect the performance of biosignal sensing devices; we hence collected a novel EMG dataset from a diverse population of fifteen able-bodied adults.

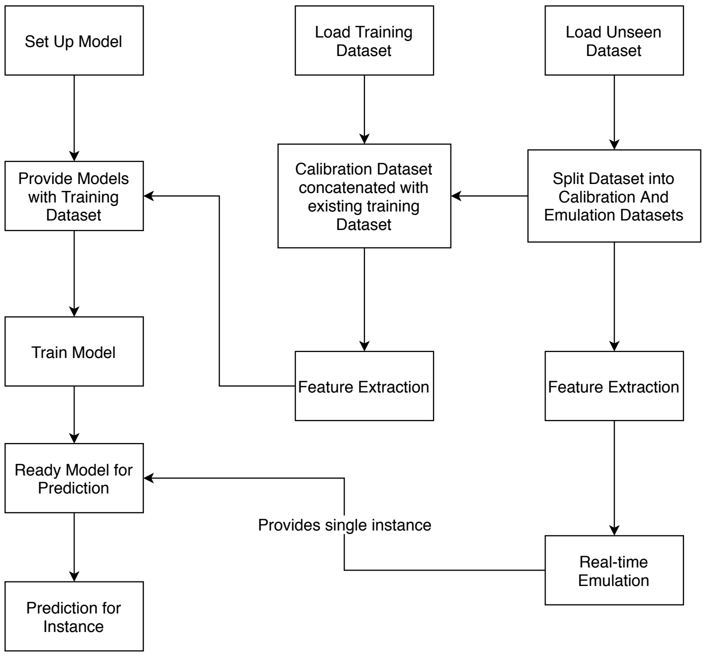
Set Up Model
Load Training Dataset
Load Unseen Dataset
Provide Models with Training Dataset
Calibration Dataset concatenated with existing training Dataset
Split Dataset into Calibration And Emulation Datasets
Train Model
Feature Extraction
Feature Extraction
Ready Model for Prediction
Provides single instance
Real-time Emulation
Prediction for Instance

Fig. 1. Real-time emulation System

The main scientific contributions of this work are as follows:

1. Collection of an original EMG dataset of 4 gestures from 15 individuals. (Available at Kaggle [12])[1]
2. Achievement of better performance of voting classifiers over deep learning
3. Results support the importance of model calibration for real-time classification through classical, deep, and transfer learning.
4. Success in real-time classification of unseen data

As part of this work we intended to verify our classification system in real-time, however due to the ongoing 2020 pandemic of the SARS-CoV-2 virus [13], we were unable to safely access the necessary equipment. We hence devised a system to emulate real-time reading of pre-recorded EMG data for classification.

The real-time classification process starts by loading the required models and the training dataset. Simultaneously we load the unseen dataset, from which a portion is extracted and incorporated into the training dataset for calibration purposes. This mimics the calibration period that would typically take place in a physical system [17]. This extended training dataset goes through the feature extraction process and is passed to the classification models to train upon. The remainder of the unseen data meanwhile undergoes the same feature extraction process as the training set, and awaits the completion of model training. When the models are ready, single instances of the unseen testing dataset are provided in turn for classification. Figure 1 depicts an overview of our real-time emulation approach for electromyographic gesture recognition.

2 Background

As the majority of movements of the hand and wrist [9] are controlled by muscles in the forearm, it is appropriate to measure electromyographic (EMG) activity in these muscles to identify hand movement. Thalmic Labs' Myo Armband [26] is a wearable human-electronics interface which combines muscle activity sensors utilizing surface electromyography (sEMG) and inertial sensor signals in order to provide a gesture recognition system for electronic device control. The bioelectric signals generated by muscle activity have potential differences in the order of µV – mV [20], which are recorded by the sEMG sensors and converted by Thalmic's software to a unitless measure of muscular "activation" [25]. Surface EMG sensors require direct contact with the skin in order to conduct the bioelectric signals of the underlying muscles; the quality of the acquired signals is affected by various factors including the condition of the skin and the fat and muscle density of the forearm [2]. To develop Machine Learning models which are more robust to variations in such conditions, it is beneficial to collect EMG data from a range of individuals. EMG recognition devices such as the Myo are widely used for prosthetic limbs [18], controlling robotic arms [22], wheelchairs [6], and have even been integrated as part of a smart home environment [29].

[1] Dataset available at https://www.kaggle.com/chrisdolopikos/eleectromyography-dataset

2.1 Related Work

There is much precedent in the literature for using Thalmic's Myo device to collect electromyographic data for offline classification. Benalcázar et al. [3] used a k-nearest-neighbours model to classify five different gestures from a group of ten individuals, reaching up to 98.5% accuracy when incorporating an activity detection system to remove any segments in the EMG signals which did not correspond to a movement of the hand. The attributes provided to their kNN were the levels of EMG activity detected at each of the Myo's eight sensors over a given period.

Conversely, Bird [4] and Kobaylarz [17] both extracted an ensemble of statistical features from the time-series EMG data provided by the Myo to be used as model attributes. Bird et al. demonstrated the effectiveness of a feature ensemble previously developed for use with electroencephalography (EEG) in the EMG domain, using an optimised deep neural network topology to classify four gestures from ten subjects at an accuracy of 85%. They also made use of transfer learning techniques, using network weights learned from one type of biological signal as a starting point for the other (as opposed to random starting weights). Kobylarz succeeded in using this feature ensemble to classify EMG data corresponding to three hand gestures, classifying unseen data with a Random Forest at up to 97% when incorporating some calibration of the model to the new participant. Similar accuracies were also reached with a voting system comprising both a Random Forest and a Support Vector Machine (SVM).

Fewer studies however have been able to achieve online, or real-time, gesture classification. Kim et al. [16] found success in real-time classification of four gestures from single-channel EMG measurement. They similarly extracted statistical features from the data, using an ensemble that shares some similarities with that of Bird and Kobylarz whilst also using some different features, notably the number of zero-crossings and the degree to which the signal was periodic. By combining a kNN and a Bayesian model they reached 94% accuracy, and were able in real-time to control a radio-controlled car through gesture.

One developing application of electromyographic gesture control is sign language recognition. The British Deaf Association reports that over 85,000 individuals in the United Kingdom use British Sign Language as their primary preferred form of communication [7]. Communicating with computer systems is an integral part of modern life; many devices are now "hands-free", relying on voice recognition, hence creating a need to ensure that such devices are still made accessible for the deaf. Deja et al. found that sign language users responded positively to an electromyographic human-computer-interface for signing using a pair of Myo devices [11], although further work was needed in constructing a system robust to syntax and grammar errors.

Kaya & Kumbasar used data from a Myo to classify gestures representing the digits 0 - 9 in Turkish sign language [15]. In a similar fashion to Bird, Kobylarz, and Kim they used a sliding time-series window to extract statistical features from the EMG data, successfully classifying the ten gestures with kNN, SVM, and Artificial Neural Network models. Their SVM was able to reach a F-score of

0.866, with 10-fold cross validation, though it remained to be seen whether the model would be capable of classifying unseen data. Abreu et al. [1] developed a 20-class system based on the Brazilian sign language alphabet Libras. They rectified and averaged the electromyographic signals from the Myo device, and used these averaged signals as the input to an ensemble of twenty SVMs, one per letter, each trained using a one-vs-all approach. While very high cross-validation accuracies were reached for all letters, these accuracies decreased significantly when the system was presented with real-time, unseen data.

Whilst our study does not set out to classify sign language gestures specifically, it is hoped that with our successful exploration of real-time methods gesture classification we can help lay the groundwork for furthering development in this area.

3 Implementation: EMG Data Processing and Classification

3.1 Data Acquisition

The experimental setup for data collection consists of a Myo Armband (Figure 2) connected to a computer via Bluetooth Low Energy (BLE). The connection is achieved using Myo Hub which enables the device's SDK and allows Python to gain access to the device as well. A Python script is used to record forearm EMG data produced by muscle activity when executing a gesture for 60 seconds [17]. Each of the Myo's eight EMG sensors produces an EMG reading and the IMU produces a single reading; the resulting data file is hence ten columns wide: one stating unix timestamps, eight for the EMG sensors and one for the IMU readings. The EMG data streams at 200Hz whilst the IMU data (which was not utilised in this study) streams at 50Hz. Each single measurement session results in a single .csv file.

A single subject would wear the armband on their right forearm (data was also collected from the participants' left arms, which we did not use in our study but still intend to make available for the research community). The measuring procedure started after a brief demonstration of one of the required defined gestures: a closed fist (finger flexion [14]), spread open fingers (finger abduction [14]), waving inwards (wrist flexion [14]), and waving outwards (wrist extension [14]). The subject was then instructed to perform this gesture repeatedly for 60 seconds. To minimize the impact of random errors, 5 different repetitions per gesture were executed. A single subject hence performed four different gestures for five minutes each.

One factor contributing to random errors is the participant themselves. Continuous repetition of gestures is exhaustive for the muscles. As time passes, they start becoming stiffer and as a result the subject struggles to perform the gestures correctly, directly affecting the quality of EMG readings. To mitigate this each subject was given one to two minutes for muscle relaxation after the completion of each measurement task. Different stress levels might also contribute to

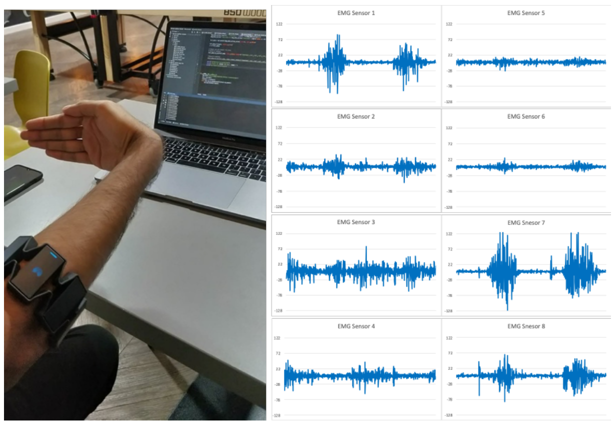

Fig. 2. Subject performing "Wave-in" gesture (left), and the resulting elecromyographic waveforms (right).

different or random bioelectrical activity resulting in outliers. To mitigate this and lower stress levels the data collection sessions were conducted in an environment that was familiar to the subjects. Fifteen able-bodied subjects (9 male, 6 female) participated in this experiment. The mean age of the subjects was 26, with the youngest participant being 20 and the oldest 52. Whilst a diverse population was sought, there would be merit in future work expanding the dataset even further, both in number and to individuals from a more diverse range of ages.

3.2 Data Preprocessing

To ensure the integrity of the collected data all files were initially checked for missing values, NaNs, and other discrepancies of a similar nature. Any individual measurement instances containing such errors were removed.

3.2.1 Data Cleaning As previously described the participants of the data collection are themselves a source of possible errors. Observation of the produced datasets lead to the recognition of systematic errors; it was found that due to variation in the participants' reaction times in initiating muscle movement when instructed, a portion at the start of each recording did not have any significant EMG activity. It would not be appropriate to consider this "dead" time part of the gesture as no physical muscular activity was taking place. To address this, all EMG readings were divided into two sub files, one containing the actual gesture and one containing the null data . This processing was done in MATLAB [19], with the assessment of when EMG activity started being performed by observation of graphical plots of the signals. The null data was used to form a rejection class, resulting in a total of five defined gestures.

3.2.2 Rectification Raw EMG readings typically range from $-128mV$ to $+127mV$ for each timestamp and are highly oscillatory. This means that in some cases, simple statistical measures such as the mean will result in values close to zero, regardless of the intensity of electromyographic activity. The ensemble of statistical features used in this study includes more complex measures which are less liable to be influenced by this, however to maximise the informativity of the featureset the EMG waveforms were rectified (i.e. the modulus of the values was taken) before feature extraction was performed. Both the rectified and raw datasets were analysed to investigate the impact of this process.

3.2.3 Feature Extraction Raw usage of the collected EMG data is not beneficial for machine learning due to the data's stochastic nature. EMG data, as previously mentioned, are bioelectric signals that are non-stationary, non-linear and of random nature. This means that even when a subject replicates the exact same gesture, at a given point in time during the gesture's execution it is highly likely that the activated muscles do not have identical bioelectric activity. Therefore, some statistical analysis on the raw data is required to produce a meaningful dataset for machine learning.

Electroencephalographic (EEG) signals have been reported to bear similarities with EMG signals [4]; both are relatively low frequency, very low amplitude electrical waveforms. Previous research has performed statistical feature analysis for EEG and EMG by introducing sliding windows of length of 1 second at an overlap of 0.5 seconds to segment the data [4, 5, 17]; we make use of the same feature extraction approach here. Firstly, a sliding 1-second window is divided into two windows of 0.5 seconds decreasing the size of the initial wave to equally sized segments of the wave. The feature extraction process then comprises of 3 stages; each stage is related to the size of the window that is being processed:

- 1-second window
  - Mean and Standard deviation of the wave are computed
  - Skewness and Kurtosis of each wave
  - Maximum and minimum values
  - Sample variances of each wave, plus the sample covariances of all pairs of the waves
  - The eigenvalues of the covariance matrix
  - The upper triangular elements of the matrix logarithm of the covariance matrix
  - The magnitude of frequency components of each signal, obtained using a Fast Fourier Transform (FFT)
- 0.5-second windows
  - The change of (between the first and second sliding window):
    - The sample mean and the sample standard deviation
    - Maximum and minimum values
- 0.25-second windows produced due to offset
  - The mean of each 0.25-second window
  - All paired differences of means between the windows

- Maximum and minimum values and their paired differences

The input to the feature extraction process is an 8-signal wide array (with one column for each of the Myo's 8 EMG sensors), from which a total of 1771 attributes are generated. The resulting featureset is then shuffled, and before being passed to a classifier is standardised such that all attributes are in the range [-1 to $+1$].

3.3 Classifier Model Optimisation & Tuning

With data collection and preprocessing stages being completed, the resultant dataset is in a form suitable for use in the classifiers. Before the training of the classifiers starts, the dataset is split into two chunks; one for training and one for testing purposes. Details of machine learning tuning and results are presented in the following section. The utilization of these sub-datasets allows for training of a model classifier and testing of its performance against a subset of data. This is repeated as part of the parameter tuning process. Classifiers have sets of parameters that determine their learning ability that need to be tuned.

3.3.1 Implementation Equipment Experiments were developed in Python 3.7 with the Scikit Learn library [23]. Deep Learning weights that were produced using the training dataset were produced on an Intel Core i7@3.7GHz and a GTX980Ti GPU in the Keras library [8]. The produced weights using the calibration dataset and models were built using an Intel Core i7@2.6 GHz. The same configuration was used for the real-time emulation system described in 3.4.2.

3.3.2 Support Vector Machine Support Vector Machines are kernelised models, meaning that their learning abilities depend on the kernel utilized, alongside C and gamma parameters of a given model. For this classifier ScikitLearn was used [23]. Parameter tuning results suggested that in this case a higher C parameter could enable the SVM to achieve higher levels of accuracy, whereas for the gamma parameter the reverse was true; a lower gamma resulted in higher accuracy. The best performing kernel for the datasets was the Radial Basis Function (RBF). Based on experimentation, the chosen set of parameters used for our SVM is: kernel"RBF", C=100, Gamma=0.00001.

3.3.3 Random Forest The implementation of this classifier is also achieved with the ScikitLearn library. The first parameter that can be optimised in this classifier is the number of features per split (n_estimators). The search for this parameter started from 200 and had a maximum value of 2000. The next optimised parameter is the maximum number of features when looking for a split (max_features), which can be square root, log2, or auto. The parameter related to the length of the constructed trees is max_depth. This parameter's search started from 100 with a maximum of 500. Utilizing the RandomizedSearchCV method provided by Scikit Learn library, the cross-validation parameter is set to 3 and the number of iterations to 100 so as to ensure the validity of the findings. The optimized Random Forest model has n_estimator, max_features and max_depth set to 1600, auto and 220 respectively.

3.3.4 Artificial Neural Network The model used consists of a single hidden layer. Based on experimental tuning results, 20 neurons for a single layer provided a competitive advantage over a model with 10 neurons, however performance was not significantly affected by an increase to 30 neurons. Therefore, to minimize memory usage the 20 neurons configuration was the preferred choice. Despite the fact that learning rate affects the performance of the models both positively and negatively, it also affects the memory usage. While the model is training, high learning rates result in error messages indicating possible memory overflow. The set of optimised parameters used for the Multilayer Perceptron was hence an L-BFGS solver, 20 neurons, 0.1 learning rate.

3.3.5 Deep Neural Network The DNN used for the on-the-fly real-time emulation consists of five hidden layers with the ReLU activation function. The five layers consisted of 206, 226, 298, 167, and 36 neurons respectively. As the classification problem of this work is a multiclass one a softmax activation function is used, taking place at the last layer of the network with 5 neurons.

Another DNN is constructed for use with a set of weights found using EEG data in [4]; in that study these weights were found to be a suitable starting point for using with EMG data. The number of layers and neurons are the same as those described above. Similarly, the output layer utilizes a softmax activation function with 3 neurons.

The weights produced above are input to the first DNN with EMG data to create another set of weights, to be used for the emulated real-time classification. The first model is also used to randomly produce some weights using the same dataset and save them for later use.

3.4 Classifier Implementation

3.4.1 Voting Ensemble An ensemble voting mechanism was constructed from the SVM, Random Forest, and ANN models. As the models are provided with an instance i they each predict the class of i, outputting an array of 5 probabilities, one per class, that total to 1. The arrays produced by the SVM and ANN are scaled by a factor of 1.5 whilst the output array of the Random Forest is multiplied by a scale of 2. Consistency on the performance of Random Forest, due to its tree structure, resulted in arbitrary biases based on the classifier's performance throughout the experiments. This provides a confidence vote to Random forest in case of disagreement. The scaled arrays are then summed to provide a single array of similar length. The index with the higher score represents the prediction of class of i.

3.4.2 Real-time Classification A user is typically expected to use the Myo Armband to classify their gestures in real-time. Machine learning algorithms are often expected to perform poorly given real-time data and sometimes require the use of a small additional training dataset to recalibrate their models. Due to the ongoing 2020 SARS-CoV-2 pandemic, we were unable to access the Myo armband for genuine live classification with a new subject. However, in anticipation of this, data had been collected from an additional sixteenth participant (male, aged 21) who was left out of the training set entirely. This unseen raw EMG was cleansed and preprocessed as described above, including the formation of a rectified version of the data. The calibration process starts simultaneously with the emulation procedure, by splitting this previously unseen data of the sixteenth participant into thirds. One third is used for calibration and the remaining two thirds for emulating real-time data. The calibration process implements Platt's Scaling [24], a technique that calculates a score to probability calibration curve using the training set. Calibration data are then combined with the existing and previously seen training data for retraining the classifiers. The emulation portion of the data also underwent the same feature extraction as the training dataset. Each instance was then passed in turn in near-real-time to the classifiers, thus emulating live human input (this process is depicted in Figure 1).

The Random Forest, SVM and ANN are combined with the voting mechanism described above to provide a single prediction in order to compete with the DNN. The process for DNNs is slightly different for the real-time data; two of them were required to load the previously produced weights. One DNN was left to randomly produce weights and use those for the on the fly predictions, whereas the other DNN model uses Fine-tune Learning, by getting the feature extracted calibration data to generate some weights that are going to be loaded in the same model for the emulation process.

4 Results

This section presents our experimental results. In all cases both the rectified dataset produced by section 3.2.2 and the dataset without rectification as in 3.2.1 were trialled to investigate the impact of the rectification. In all tables the reported accuracy is the arithmetic mean of 5 separate runs, whilst the values in parentheses indicate 95% confidence intervals on that accuracy (using the Wilson method [28]). To enable further analysis of classification errors, we also present confusion matrices, precision, and recall scores for the best-performing model in each experiment.

4.1 Model Benchmarking

We here set a benchmark by using a randomised 66% of the offline dataset for training and the remaining 33% for testing, providing us a goal for comparison with our real-time results.

Table 1. Model benchmark scores using dataset without rectification

Classifiers	Accuracy (%)
Base Classifiers	
Multilayer Perceptron	**92.16** [89.02, 94.58]
Support Vector Machine	87.75 [84.03, 90.79]
Random Forest	91.12 [87.75, 93.65]
Deep Networks	
Random Weights	86.84 [82.81, 89.82]
EEG Weights	86.84 [82.81, 89.82]
EMG Weights	85.78 [81.90, 89.09]

Table 2. Confusion Matrix of Table 1's best performing model, MLP

		Predicted class				
		fist	open	wave-in	wave-out	null
Actual class	fist	1.00	0.00	0.00	0.00	0.00
	open	0.02	0.80	0.04	0.00	0.16
	wave-in	0.00	0.01	0.90	0.09	0.00
	wave-out	0.04	0.02	0.09	0.85	0.00
	null	0.01	0.06	0.01	0.00	0.91

Table 3. Model benchmark scores using dataset with rectification

Classifiers	Accuracy (%)
Base Classifiers	
Multilayer Perceptron	**92.16** [89.02, 94.58]
Support Vector Machine	88.61[84.95, 91.52]
Random Forest	90.08 [86.50, 92.71]
Deep Networks	
Random Weights	85.26 [81.30, 88.60]
EEG Weights	82.63 [78.30, 86.12]
EMG Weights	83.68 [79.49, 87.11]

Table 4. Confusion Matrix for Table 3's best performing model, MLP

		Predicted class				
		fist	open	wave-in	wave-out	null
Actual class	fist	1.00	0.00	0.00	0.00	0.00
	open	0.06	0.79	0.04	0.00	0.10
	wave-in	0.00	0.01	0.89	0.08	0.02
	wave-out	0.03	0.04	0.09	0.83	0.01
	null	0.01	0.05	0.02	0.02	0.89

Table 5. Accuracy, Precision, and Recall scores for the best performing benchmark models

Multilayer Perceptron	Accuracy (%)	Precision (%)	Recall (%)
Without rectified Data	92.16	92.08	80.83
With rectified Data	92.16	96.66	79.51

Comparison of Tables 1, 3 & 5 suggests that the rectification process did not provide a significant competitive advantage to the models' performances. They also suggest that base classifiers individually are capable of outperforming the developed deep neural networks, albeit by a relatively small degree.

4.2 Real-time Classification of Unseen Data

In emulated real-time we were able to classify unseen data with accuracies competitive with the above benchmark when the models were calibrated, though even uncalibrated performance was strong for the Fist, Wave In, and Wave Out gestures as seen in Tables 7 & 9. We again found base classifiers to individually outperform the deep neural networks as seen in Tables 11, 13 & 15, though the DNN achieved the best performance when uncalibrated (Tables 6, 8 & 10).

Table 6. Real-time model performance without calibration using unrectified dataset (* denotes the use of calibration dataset for the production of weights)

Classifiers	Accuracy (%)
Voting Mechanism	67.87 [62.88, 72.47]
Deep Networks	
Random Weights	66.80 [61.75, 71.42]
EEG Weights	66.59 [61.46, 71.16]
EMG Weights	69.96 [65.17, 74.57]
EMG Calibration Weights*	**74.27** [69.49, 78.48]

Table 7. Confusion Matrix for Table 6's best model, EMG calibrated-weights DNN

		Predicted class				
		fist	open	wave-in	wave-out	null
Actual class	fist	0.77	0.00	0.04	0.01	0.21
	open	0.00	0.11	0.23	0.27	0.40
	wave-in	0.00	0.00	0.95	0.00	0.05
	wave-out	0.00	0.10	0	0.81	0.10
	null	0.00	0.24	0.33	0.00	0.44

Table 8. Real-time model performance without calibration using rectified dataset (* denotes the use of calibration dataset for the production of weights)

Classifiers	Accuracy (%)
Voting Mechanism	64.84 [59.76, 69.57]
Deep Networks	
Random Weights	67.00[62.03, 71.68]
EEG Weights	68.01 [63.17, 72.74]
EMG Weights	68.61 [63.45, 73.00]
EMG Calibration Weights*	**70.63** [65.74, 75.10]

Table 9. Confusion Matrix of Table 8's best performing model, DNN with calibrated weights

		Predicted class				
		fist	open	wave-in	wave-out	null
Actual class	fist	0.77	0.01	0.02	0.00	0.21
	open	0.00	0.14	0.29	0.25	0.32
	wave-in	0.00	0.00	0.88	0.00	0.12
	wave-out	0.00	0.09	0.01	0.87	0.04
	null	0.00	0.25	0.38	0.05	0.31

Table 10. Accuracy, Precision, and Recall scores for the best performing uncalibrated models

DNN	Accuracy (%)	Precision (%)	Recall (%)
Without rectified Data	74.27	67.11	67.21
With rectified Data	70.63	64.38	61.34

Table 11. Real-time model performance with calibration using unrectified dataset

Classifiers	Accuracy (%)
Voting Mechanism	**90.24** [86.81, 92.95]
Deep Networks	
Random Weights	86.06 [82.20, 89.33]
EEG Weights	83.36 [80.99, 88.35]
EMG Weights	85.58 [81.60, 88.84]
EMG Calibration Weights	83.23 [78.89, 86.61]

Table 12. Confusion Matrix of Table 11's best performing model, Voting Mechanism

Actual class \ Predicted class	fist	open	wave-in	wave-out	null
fist	0.97	0.01	0.01	0.00	0.01
open	0.00	0.63	0.12	0.07	0.18
wave-in	0.00	0.00	1.00	0.00	0.00
wave-out	0.00	0.00	0.00	0.99	0.01
null	0.00	0.27	0.67	0.00	0.07

Table 13. Real-time model performance with calibration using rectified dataset

Classifiers	Accuracy (%)
Voting Mechanism	**91.93** [88.70, 94.35]
Deep Networks	
Random Weights	87.23 [83.42, 90.31]
EEG Weights	81.54 [77.40, 85.36]
EMG Weights	88.68 [84.95, 91.52]
EMG Calibration Weights	85.58 [81.60, 88.84]

Table 14. Confusion Matrix of Table 13's best performing model, Voting Mechanism

Actual class \ Predicted class	fist	open	wave-in	wave-out	null
fist	0.96	0.00	0.02	0.00	0.02
open	0.00	0.23	0.34	0.15	0.28
wave-in	0.00	0.00	1.00	0.00	0.00
wave-out	0.00	0.01	0.00	0.99	0.00
null	0.00	0.00	0.27	0.00	0.72

Table 15. Accuracy, Precison, and Recall scores for the best performing real-time models with calibration

Voting Mechanism	Accuracy (%)	Precision (%)	Recall (%)
Without rectified Data	90.24	83.83	77.18
With rectified Data	91.93	88.04	78.13

4.3 Discussion

The confusion matrices presented in subsections 4.1 & 4.2 allow us to infer that some of the classification errors were in part affected by human anatomy. The "open fingers" gesture was the most likely to be misclassified in our benchmark experiment (Tables 2 & 4, with 21.56% & 21.27% misclassification respectively). For the real-time experiments presented in section 4.2, "open fingers" misclassification percentages reach 89.06%, 85.62%, 36.56% & 77.18% in tables 7, 9, 12 & 14, respectively. In all these cases the misclassification percentages mentioned are substantially higher than those of the other gestures' classes.

It can be observed that the "open-fingers" gesture was most frequently erroneously recognized as the "wave-in" & "wave-out" gestures. The "open-fingers" gesture comprises primarily an abduction of the fingers, involving the *extensor carpi radialis* (*brevis* & *longus*), *extensor digitorium communis*, and *flexor carpi radialis* muscles [14]. The *flexor carpi radialis* is also utilised in combination with the *flexor digitorium superficialis* and *flexor digitorum profundus* [14] in flexion of the wrist, i.e. the "wave-in" gesture. Similarly, wrist extension and hyperextension, as performed in the "wave-out" gesture, also requires activation of the *flexor digitorum profundus* [14]. This shared employment of certain forearm muscles provides some insight as to why gestures which outwardly appear very distinct in fact have electromyographic similarities, hence leading to misclassification between these gestures.

5 Conclusion & Future Work

Future extensions to this study could consider a larger number of participants for collecting the base dataset, since this would allow the algorithms to generalise a greater number of people, which may minimise the amount of data required for calibration purposes. Whilst in this work the results were based on processing of EMG data collected from the right forearm, the process could be replicated for the left and would be likely to achieve similar results. Future work may even be able to develop an ambidextrous gesture classification system. Additionally, the relationship between a larger number of gestures and the required size of calibration datasets could also be explored, to move beyond the four classes (plus one neutral class) studied in this work.

To finally conclude, Human-Computer Interaction has been somewhat static in the state-of-the-art for many years compared to other computer science fields since real-time classification is often forgotten, with research focusing on maximising the accuracy of models on static training and testing datasets. In this work, we contributed towards methods for real-time gesture recognition, developing approaches that would allow the technology to be used in the real world. Connecting users with technology would allow for higher level of multi-tasking and productivity without physically discriminating its users, that is, setting the basis of bringing humans in unison with technology. It is evident from these experiments that calibration processes are important for real-time classification of hand gestures, improving the best model performance from 67.87% to 91.93% accuracy when a short calibration exercise is performed. Additionally, this study shows the combination of multiple classical models can outperform a Deep Learning neural network, providing motivation for future investigations into the optimal choice of classifiers for this kind of problem. The product of the study is a process of successfully classifying hand gestures in real-time.

References

1. J. G. Abreu, J. M. Teixeira, L. S. Figueiredo, and V. Teichrieb. Evaluating sign language recognition using the myo armband. In *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pages 64–70, 2016.
2. I. S. Aleem, P. Ataee, and S. Lake. Systems, devices, and methods for wearable electronic devices as state machines, Feb. 5 2019. US Patent 10,199,008.
3. M. E. Benalcázar, C. Motoche, J. A. Zea, A. G. Jaramillo, C. E. Anchundia, P. Zambrano, M. Segura, F. Benalcázar Palacios, and M. Pérez. Real-time hand gesture recognition using the myo armband and muscle activity detection. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6, 2017.
4. J. J. Bird, J. Kobylarz, D. R. Faria, A. Ekárt, and E. P. Ribeiro. Cross-domain mlp and cnn transfer learning for biological signal processing: Eeg and emg. *IEEE Access*, 8:54789–54801, 2020.
5. J. J. Bird, L. Manso, E. Ribeiro, A. Ekart, and D. Faria. A study on mental state classification using eeg-based brain-machine interface. In *2018 International Conference on Intelligent Systems (IS)*, Funchal, Madeira Island, Portugal, September 2018.
6. A. Boyali, N. Hashimoto, and O. Matsumoto. Hand posture and gesture recognition using myo armband and spectral collaborative representation based classification. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pages 200–201, 2015.
7. British Deaf Association. Help & resources, March 2017.
8. F. Chollet et al. Keras. https://keras.io, 2015.
9. S. W. Cummings, C. Tangen, B. Wood, and R. H. Crompton. Encyclopædia britannica, Apr 2018.
10. C. Darwin. *On the origin of species: A facsimile of the first edition.* Harvard University Press, 1964.
11. J. A. Deja, P. Arceo, D. G. David, P. L. Gan, and R. C. Roque. MyoSL: A framework for measuring usability of two-arm gestural electromyography for sign language. In M. Antona and C. Stephanidis, editors, *Universal Access in Human-Computer Interaction. Methods, Technologies, and Users*, pages 146–159, Cham, 2018. Springer International Publishing.
12. C. Dolopikos, M. Pritchard, J. J. Bird, and D. R. Faria. Collection of original emg dataset, 2020.
13. T. A. Ghebreyesus. WHO director-general's opening remarks at the media briefing on COVID-19 - 11 March 2020, Mar 2020.
14. H. Gray. *Anatomy of the human body.* Lea & Febiger, Philadelphia, Pennsylvania, USA, 20 edition, 1918.
15. E. Kaya and T. Kumbasar. Hand gesture recognition systems with the wearable myo armband. In *2018 6th International Conference on Control Engineering Information Technology (CEIT)*, 10 2018.
16. J. Kim, S. Mastnik, and E. André. Emg-based hand gesture recognition for real-time biosignal interfacing. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 30–39, 2008.
17. J. Kobylarz, J. J. Bird, D. R. Faria, E. P. Ribeiro, and A. Ekárt. Thumbs up, thumbs down: non-verbal human-robot interaction through real-time emg classification via inductive and supervised transductive transfer learning. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2020.

18. S. Masson, F. Fortuna, F. Moura, and D. Soriano. Integrating myo armband for the control of myoelectric upper limb prosthesis. In *XXV Brazillian Congress on Biomedial Engineering*, 10 2016.
19. The Mathworks, Inc., Natick, Massachusetts, USA. *MATLAB version 9.7.0.1296695 (R2019b) Update 4*, 2019.
20. R. Merletti and H. Hermens. Detection and conditioning of the surface emg signal. *Electromyography: physiology, engineering, and noninvasive applications*, pages 107–31, 2004.
21. R. Merletti and P. J. Parker. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.
22. G. Morais, L. Neves, A. Masiero, and M. C. Castro. Application of myo armband system to control a robot interface. In *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 227–231, 01 2016.
23. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
24. J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.
25. Thalmic Labs, Inc. How do i access the raw emg data from the myo armband?, 2015.
26. Thalmic Labs, Inc. Tech specs: Myo battery life, dimensions, compatibility, and more. Web Archive, 2016.
27. M. Victorino, X. Jiang, and C. Menon. *Wearable Technologies and Force Myography for Healthcare*, pages 135–152. Academic Press, 01 2018.
28. X. Yan and X. G. Su. Stratified wilson and newcombe confidence intervals for multiple binomial proportions. *Statistics in Biopharmaceutical Research*, 2(3):329–335, 2010.
29. O. K. Zheng and C. Cheng. Interactive lighting performance system with myo gesture control armband. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pages 381–383, 2018.