CLASSIFICATION OF PHYSICAL ACTIVITY USING EMG READINGS

FINAL REPORT

TEAM DATALOGY
BHAVYA SHARMA
ELISSA YE

0. Executive Summary

This project is an attempt by the authors to implement supervised machine learning models to classify EMG signals recorded from human subjects. EMG or Electromyography readings involve placing electrodes on the muscles and calculating the electrical activity of the muscle. The primary purpose of EMG readings is monitoring muscle functioning and assessing the neuromuscular health of a subject. It is also used in the field of kinesiology to study human movements and its impact on the signals.

EMG signal collected using surface-based (skin) electrodes are generally noisy and complicated. Several signal processing steps are required to attain the correct signal characteristics and classifying the same. The authors of the report specifically focused on classifying physical activities performed by four subjects in the controlled environment into 'normal' and 'aggressive.' The results shall help biomedical experts understand characteristics of aggressive activity better and apply the results in fields ranging from performance enhancement to anomaly detection.

The report follows the following approach. The authors first introduce the problem and explain the data followed by a section on data visualization and signal processing. The authors then explain the results of preliminary classification done using time-series features. Authors then proceed and perform feature selection to obtain the feature space that improves the results the most. The feature space selected is then used to conclude the results.

1. Introduction

Electromyography is the process of recording electrical signals from the skeletal muscles to monitor their health and assess neuromuscular connection. [1] The electromyography (EMG) readings are generally taken using an EMG apparatus which records muscle 'contraction' and 'relaxation' over a period using electrodes attached to the muscles of the subject/patient. The signal hence received from each electrode is a function of time, frequency, and amplitude.

According to Reaz et al. [2], 'the collected signal is complicated and is controlled by the nervous system. It is dependent on the anatomical and physiological properties of muscles.' These signals can be collected using both surface-based and intramuscular electrodes. The paper [2] also mentions that the signals collected from electrodes on the skin surface are generally noisy since they 'travel through different tissues' before being collected. Pre-processing and proper classification of signals are hence paramount for diagnosis and other industrial applications.

In our project, we are working on a very specific application of EMG signal processing using surface electrodes which has its roots in kinesiology. According to [3], kinesiology can be defined as "the study of the principles of mechanics and anatomy in relation to human movement." We aim to build a supervised learning model to classify EMG signals of subjects performing a particular action. A successful classification model will help biomedical engineers understand the time-series features that differentiate a particular activity from another. The results will have applications ranging from performance enhancement of an athlete to detection and diagnosis of anomalous behavior in muscles. E.g., neuromuscular problems like epileptic seizures.

We are specifically focusing on building a classifier that can successfully classify an 'aggressive' action against a 'normal' action. We hypothesize that the aggressive actions and normal actions vary a lot in terms of signal amplitude, frequency, and pattern. Hence, a model built on the processed signals will be able to classify between the activities successfully. Several studies have been conducted on the similar topic including the ones published by Reaz at al. [2], Sezgin. N [4], and Merlo et al. [5] that focus on using signal transformation methods like wavelet transform, empirical mode decomposition, spectral analysis and wigner-ville distribution to featurize the signals and use classification models.

We followed a slightly different approach by first filtering the data and then creating relevant time-series features from the signal using a pre-existing library in Python called *tsfresh* [6]. It calculates various features from the signal including absolute energy, entropy, Fourier transform coefficients, wavelet transform coefficients (Mexican hat wavelet) and it also calculates basic statistics like mean, variance, kurtosis, and skewness. [6]

We trained our models using the random forest, logistic regression, and naive bayes classifiers and assessed their performance by comparing to a default model. We also performed a sensitivity analysis by tuning the process of feature extraction, varying standardization techniques, tuning hyperparameters and by feature selection. We have concluded the report with the results and identifying the scope of future work.

2. Data collection and source

The source of our dataset is 'EMG Physical Action Dataset' found on UCI Machine Learning repository [7]. The dataset was provided by Theo Theodoridis from the School of Computer Science and Electronic Engineering at the University of Essex. The data is a result of an experimental study done at Essex involving three male and one female subject (Age 25 – 30) who have experienced aggression in scenarios such as physical fighting. The subjects were asked to perform the following ten normal and ten aggressive activities:

Normal: Bowing, Clapping, Handshaking, Hugging, Jumping, Running, Seating, Standing, Walking, Waving.
Aggressive: Elbowing, Front Kicking, Hammering, Headering, Kneeing, Pulling, Punching, Pushing, Side kicking, Slapping

The performance of each subject was recorded using an EMG apparatus which involved placing eight skin-surface electrodes on different muscles of the subjects:

R-Bic: right bicep, **R-Tri**: right tricep, **L-Bic:** left bicep, **L-Tri**: left tricep, **R-Thi:** right thigh, **R-Ham:** right hamstring, **L-Thi:** left thigh, **L-Ham:** left hamstring.

According to [7], the eight electrodes take continuous records of muscles for each activity. There are about 10,000 samples per activity, with a sampling frequency of 200Hz and about 15 actions in each experimental session for each subject. One data frame records activity for one action. Since there are 20 actions per subject, and there are four subjects, we have a total of 80 data frames.

3. Data processing

As explained in the previous section, a single data frame represents an action which contains approximately 10,000 rows and eight columns representing each channel. We hypothesize that aggressive actions vary from normal actions in terms of frequency, amplitude and signal pattern. To test the hypotheses, the very first step was to visualize the raw data.

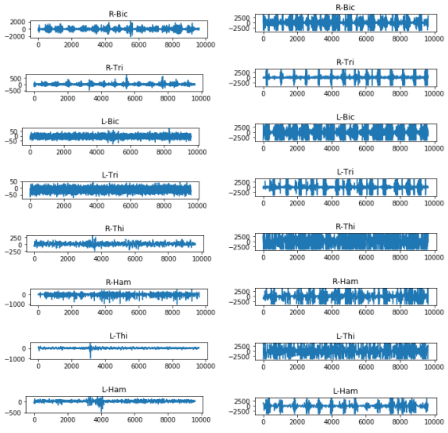

Figure 1: Raw data for Subject_1 Handshaking vs. Subject_1 Punching

We can clearly see the difference between the muscle signals of normal and aggressive activities regarding both pattern and amplitude scale.. The data, however, is noisy and unclean. According to Reaz et al. [2], there are two issues when collecting signals from electrodes attached to skin surface: signal-to-noise ratio and distortion of signals. Noise is added to the data since the electrical signal interacts with blood vessels and tissues between the skin

and muscle. To improve the signal-to-noise ratio, we detrended the raw data to filter out the DC offset [8]. Next, we rectified the data by converting the signal to a single polarity, to ensure that signals don't average to 0 during analysis. The Savitzky-Golay filter, set to an order of 5 and framelength of 49, was then applied to smoothen the data and further eliminate noise. [8]

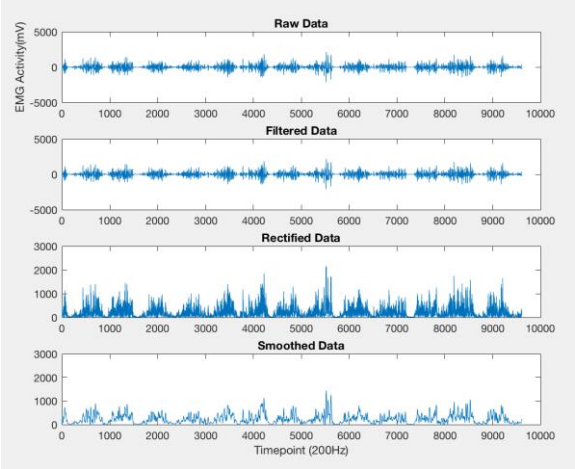

Figure 3: Stages of signal filtering

The data is now clean and processed. However, we need to extract time-series features from each of our action dataframes which we could feed to our classification models. Since each action dataframe contains the particular action performed about 15 times, we can observe approx. 15 crest and troughs of the wave in the data. We hence divided individual time-series actions into several smaller windows and extracted features for each window using sliding/moving window approach.

The sliding-window approach will ensure that features are extracted after every fixed interval of records. For example, we can create action windows of size 1000 records at an interval of 100 records. Thus, we will have the first action window from record 0 to 1000, the second window from 100 to 1100, third from 200 to 1200 and so on. Hence, the resulting features extracted from the windows will represent the original data well.

To extract these features, we used the *tsfresh* (Time Series Feature extraction based on scalable hypothesis tests) library available for python. Tsfresh library essentially creates aggregated features for every window of action for each channel. Hence, we end up with one vector representing each window in our dataset containing features for all the channels.
Since we have approximately 10,000 records in each action dataframe, we considered the following three combinations of extracting features:

1. Action window of size 500 records sliding after every 50 records resulting in a total of 119 windows.
2. Action window of size 1000 records sliding after every 100 records resulting in a total of 91 windows.
3. Action window of size 2000 records sliding after every 200 records resulting in a total of 41 windows.

For all three combinations, we extracted all possible features time-series features *tsfresh* could calculate (6305 features). Thus, we converted each action dataframe into a new featurized dataframe with 6305 columns and 119 rows (combination 1) or 91 rows (combination 2) or 41 rows (combination 3).

4. Modeling

4.1 Choosing a model

The main aim of our project is to classify actions into normal and aggressive. Since there are 10 normal and 10 aggressive actions for each subject, we assigned the label '0' to normal and '1' to aggressive actions and stacked them together to get a single dataframe representing each subject.

We chose the models for classification using the following criteria:

1. Interface of data
    a. All the features in our data are numeric, and we have binary output. Logistic Regression and Support Vector Machines are hence viable options for our setting.
2. Interpretability:
    a. We need an interpretable model to understand the features that discriminate aggressive action against normal actions. Support Vector Machine can hence not be used.
    b. Random Forest classifier is a good candidate since we can easily interpret the importance it gives to different features.
3. Amount-to-dimensionality ratio:
    a. The amount-to-dimensionality ratio is very low. To avoid overfitting, we need to use simpler models. Linear Logistic regression and Naive Bayes are hence good candidates.
    b. Random Forest classifier is also a good candidate since it is unaffected by the difference in scales between the features in a high dimension dataset.

Thus we chose Logistic Regression, Random Forest, and Naive Bayes as our classification models. We also used a default (random choice) model to compare the performance of our models.

4.1 Standardization

Since the chosen subjects vary in terms of age, sex, and possibly strength, the muscle activation varies between the subjects when performing the same action. According to [9], normalizing the dataset is required before we use machine learning models (e.g. logistic regression). Failure to do so may affect the performance of the models as the difference in scales of the features prevent the models from learning data better. Hence, we standardized all subjects using the mean and variance of subject 1. We used scikit-learn StandardScaler [9], which converts data into standard normal distribution. All the actions performed by each subject were now on the same scale.

4.3 Four-fold cross-validation

The amount-to-dimensionality ratio is poor in our dataset making it prone to overfitting. Thus, we ran the models using 4-fold cross validation which involved training models on three subjects and testing on one left out the subject in each loop.

For each cross-validation loop:

1. We calculated the confusion matrix, accuracy, recall, false-positive rate, true-positive rate and precision at 50% cutoff.
2. We also obtained the ROC curve and corresponding AUC value at each fold to assess the models at different cut-off values.

We also obtained an overall ROC curve generalizing the performance of models on all folds.

The two important metrics for choosing a particular model are Accuracy and ROC curves. Accuracy is the metric that will tell us how correctly the models can classify the unseen labels and informs us of the model's generalization capability. ROC curves tell us the overall performance of the model irrespective of the cut-off threshold selected and hence tell us about the utility of the model.

4.5 Choosing the best combination from the pre-processed data

We used the above-described method to run the classification models on the three processed dataset combinations we created earlier. We will use Accuracy (50% cutoff) and ROC curve to select one combination. All the models are running on the default parameters except random forest which is using 100 trees to learn the models.

4.5.1 Assessing Accuracy

1. Combination 1 (Window size = 500 Step size = 50)

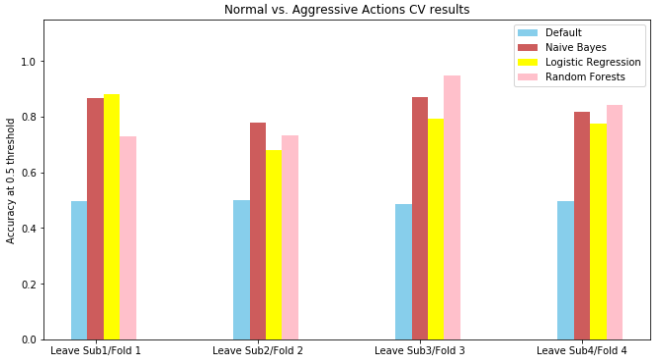

Figure 4: Accuracy of the models on 4 folds in Combination 1

2. Combination 2 (Window size = 1000, Step size = 100)

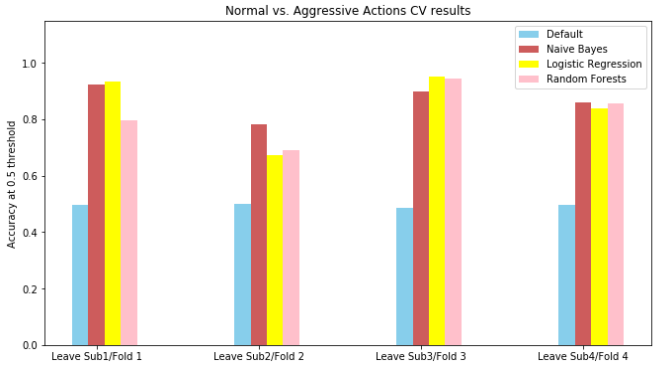

Figure 5: Accuracy of the models on 4 folds in Combination 2

3. Combination 3 (Window size = 2000 Step size = 200)

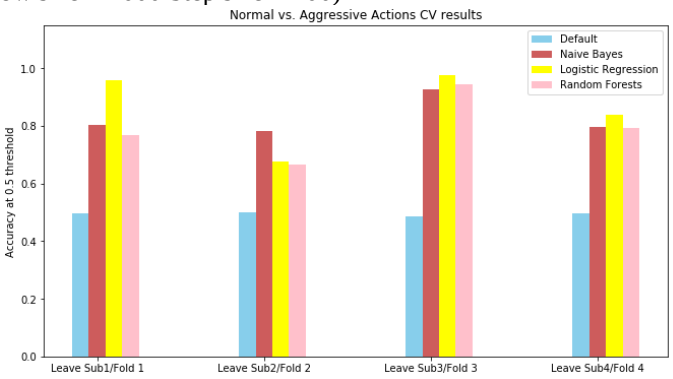

Figure 6: Accuracy of the models on 4 folds in Combination 3

Average Accuracy	**Combination 1**	**Combination 2**	**Combination 3**
Default	49.54%	49.50%	49.43%
Naives Bayes	83.28%	86.62%	82.70%
Logistic Regression	78.21%	84.90%	86.15%
Random Forest	81.27%	84.67%	79.25%

Table 1: Average accuracy over all folds.

Although all combinations are performing equally well, combination 2 (step size 1000 and window size 100) seems to be performing consistently using all models. We shall now compare the ROC curves of the three combinations.

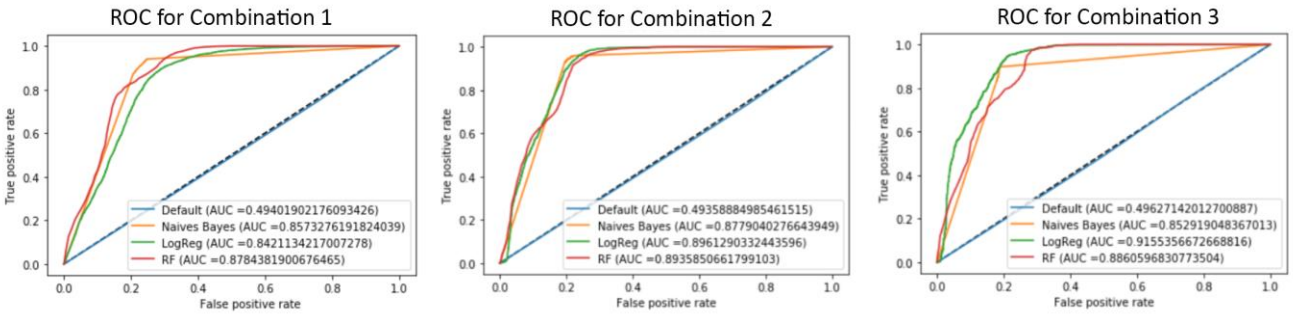

Figure 7: ROC curves of the models on all 4 folds for all combinations

ROC curve also favors Combination 2 in which all the models are performing consistently. Hence, we will go ahead with Combination 2 featurized data and do further analysis on it.

5. Feature engineering

The preliminary models were run using all the features to select a particular window and step-size combination of feature extraction. There were about 6000 features generated for each time window, and since not all of the features are relevant to our classification, we decided to perform feature selection by running our models only on the specific set of features and assess the changes in the performance. Once we have selected an optimal feature space, we will then choose one classification model and interpret its results.

5.1 Defining Feature Spaces

After reviewing all of the features calculated by tsfresh, we identified four categories of features as the basis for defining our feature spaces. We made a list of “top features” that ranked high in terms of feature importances from the random forest classifier and was potentially meaningful concerning our data. There were a few features that either was poorly defined or performed unclear operations on our data. We classified them as “uncertain features.” We made a list of “unfit features” containing features that we believe provided unmeaningful information for our classification task. Examining the features more deeply, we identified some “low-priority” features that we suspect would only have marginal effects on our results.

	List of Features
Top Features	Quantiles, fft coefficient, absolute_sum_of_changes, change_quantiles, maximum, minimum, mean, median, variance, agg_linear_trend, abs_energy, mean_abs_change, autocorrelation, entropy, linear_trend, kurtosis
Uncertain Features	Lag , ar_coefficient, mean_second_derivate_central, time_reversal_asymmetry_statistic
Unfit Features	Last_location_of_maximum, last_location_of_minimum, length, number_crossing_m, partial_autocorrelation, range_count,set_property, value_count, variance_larger_than_standard_deviation
Low-priority Features	Augmented_dickey_fuller, agg_autocorrelation, energy_ratio_by_chunks, has_duplicate, has_duplicate_max, has_duplicate_min, friedrich_coefficients, index_mass_quantile, max_langevin_fixed_point, sum_of_reoccurring_values, ratio_value_number_to_time_series_length

Table 2: List of features in each feature category.

In addition to our original feature space (labeled feature space 0), 6 feature spaces were set up as follows.

Feature space #	Include top features	Include all features	Remove uncertain features	Remove unfit features	Remove low-priority features	Approx. number of features
0	✓	✓				6300
1	✓		✓			6050
2	✓			✓		6100
3	✓			✓	✓	5850
4	✓		✓	✓		5950
5	✓		✓	✓	✓	5800
6	✓					4750

Table 3: Features included and/or removed in each feature space

The feature names generated by tsfresh, however, often included multiple features names related to one feature. Each feature space is thus created by including/excluding features with names containing the keywords listed in the appropriate feature categories.

5.2 Selecting the best feature space

We conducted a 4-fold leave-one-subject-out cross-validation given each feature space. The results for Naive Bayes, Logistic Regression, and the Random forest is shown in Table 4.

We can see that Feature space 1 (without uncertain features) Feature space 4 (without uncertain and unfit features), and Feature space 6 (containing only the important features) perform consistently well on all the models. Feature space 6 is the top priority for us since the number of features (~4750) are greatly reduced compared to that of the original feature space (~6300) hence making it a good candidate for interpretation.

	Feature Space 0			Feature Space 1			Feature Space 2			Feature Space 3		
Fold	NB	LR	RF	NB	LR	RF	NB	LR	RF	NB	LR	RF
1	0.923	0.933	0.896	0.713	0.568	0.853	0.599	0.557	0.830	0.600	0.559	0.825
2	0.781	0.674	0.685	0.841	0.703	0.782	0.823	0.854	0.789	0.823	0.529	0.750
3	0.900	0.951	0.917	0.890	0.954	1.000	0.853	0.884	1.000	0.853	0.901	1.000
4	0.860	0.839	0.845	0.850	0.954	1.000	0.853	0.904	1.000	0.8532	0.9012	1.000
Avg	**0.866**	**0.849**	**0.845**	**0.835**	**0.809**	**0.909**	**0.782**	**0.805**	**0.900**	**0.782**	**0.803**	**0.893**

	Feature Space 4			Feature Space 5			Feature Space 6		
Fold	NB	LR	RF	NB	LR	RF	NB	LR	RF
1	0.934	0.841	0.841	0.720	0.569	0.833	0.913	0.947	0.8224
2	0.700	0.754	0.732	0.841	0.706	0.763	0.758	0.786	0.792
3	0.931	1.000	1.000	0.889	0.969	1.000	0.905	1.000	1.000
4	0.931	1.000	1.000	0.889	0.969	1.000	0.905	1.000	1.000
Avg	**0.834**	**0.803**	**0.904**	**0.835**	**0.803**	**0.899**	**0.870**	**0.934**	**0.905**

Table 4: 4-fold leave-one-subject-out cross-validation accuracies for each feature space.

From the 3 chosen models ,i.e., naives bayes, logistic regression and random forest, we will finally use the random forest for interpretation since it is consistent on all folds and provides ‘feature importance’ parameter that shall help us identify the important features and interpret the result better. We will now try tuning the random forest classifier to see if we can improve its accuracy further.

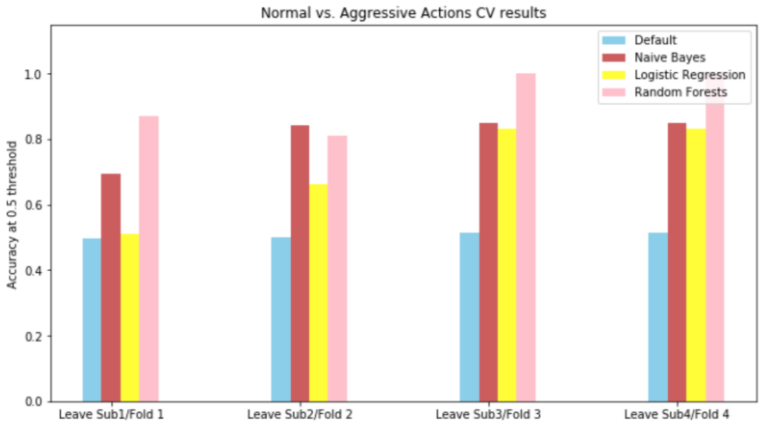

Figure 8: 4-fold leave-one-subject-out cross-validation results based on feature space 6.

6. Final Results

Our best model selected was the random forest model based on feature space 6, which achieved an accuracy of 90.5% with about 4750 features.

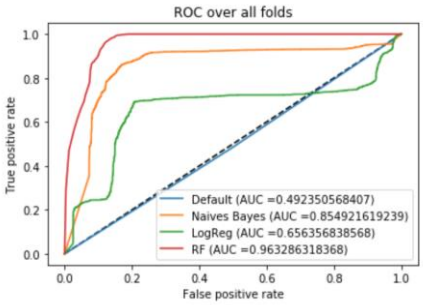

Figure 9: ROC curves for feature space 6 showing random forest to be the most optimal model.

Given the new feature space and model, we examined the top features based on random forest importance, as shown below.

	importance
R-Ham__change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.4	0.021349
R-Thi__change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.4	0.018265
R-Ham__absolute_sum_of_changes	0.016028
R-Thi__change_quantiles__f_agg_"mean"__isabs_True__qh_0.6__ql_0.4	0.014304
R-Ham__change_quantiles__f_agg_"mean"__isabs_True__qh_1.0__ql_0.2	0.011927
R-Thi__change_quantiles__f_agg_"var"__isabs_False__qh_1.0__ql_0.4	0.011496
R-Thi__change_quantiles__f_agg_"mean"__isabs_True__qh_0.8__ql_0.4	0.011410
R-Thi__change_quantiles__f_agg_"var"__isabs_False__qh_1.0__ql_0.2	0.011403
R-Ham__change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.2	0.011177
R-Thi__agg_linear_trend__f_agg_"mean"__chunk_len_10__attr_"stderr"	0.011054
R-Thi__agg_linear_trend__f_agg_"max"__chunk_len_10__attr_"stderr"	0.010956
R-Ham__agg_linear_trend__f_agg_"min"__chunk_len_5__attr_"stderr"	0.010901
R-Ham__change_quantiles__f_agg_"mean"__isabs_True__qh_0.8__ql_0.0	0.010731
R-Ham__change_quantiles__f_agg_"var"__isabs_True__qh_0.8__ql_0.2	0.010642
R-Ham__change_quantiles__f_agg_"var"__isabs_True__qh_0.8__ql_0.4	0.010623
L-Ham__change_quantiles__f_agg_"var"__isabs_False__qh_1.0__ql_0.4	0.010226
L-Ham__absolute_sum_of_changes	0.010196
L-Ham__change_quantiles__f_agg_"mean"__isabs_True__qh_0.8__ql_0.4	0.010012
R-Ham__change_quantiles__f_agg_"var"__isabs_False__qh_1.0__ql_0.8	0.009946
R-Ham__change_quantiles__f_agg_"var"__isabs_False__qh_0.2__ql_0.0	0.009196
R-Thi__change_quantiles__f_agg_"mean"__isabs_True__qh_1.0__ql_0.4	0.008445
R-Thi__change_quantiles__f_agg_"var"__isabs_True__qh_0.8__ql_0.4	0.008210
R-Tri__abs_energy	0.007704
R-Thi__change_quantiles__f_agg_"var"__isabs_False__qh_0.6__ql_0.4	0.007469
R-Thi__change_quantiles__f_agg_"mean"__isabs_True__qh_0.8__ql_0.2	0.006895

Figure 10: A sample from the feature importance list generated by our best random forest model.

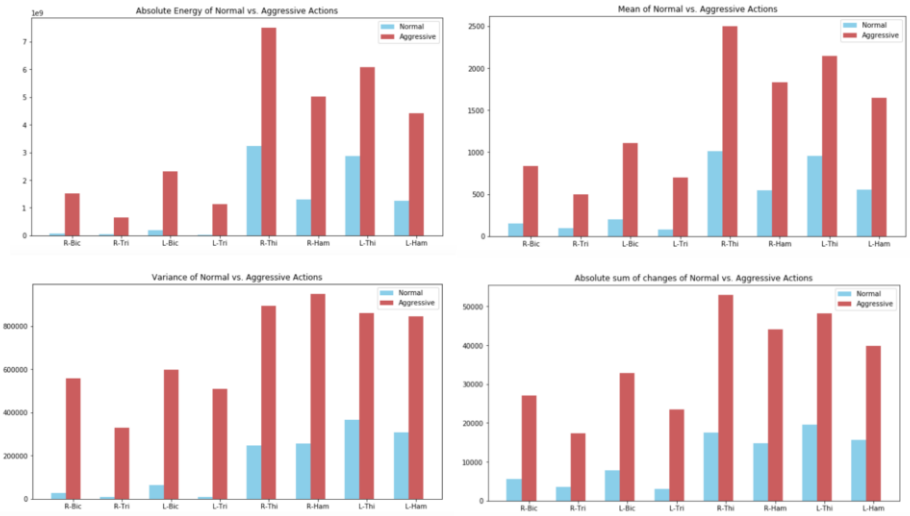

Figure 11: Graphs comparing aggressive vs. normal actions across the 8 channels for absolute energy, mean, variance, and the absolute sum of changes.

We examined top four features derived from Random Forest for our analysis:

1. Absolute energy
2. Mean
3. Variance
4. The absolute sum of changes.

Absolute energy is defined as the sum over squared values across time, and the absolute sum of changes is defined as a sum over the absolute value of consecutive changes in the series. We noticed that these feature values are generally always much higher in aggressive actions compared to normal actions (figure 11), and feature values are generally higher for right thigh, right hamstring, left thigh, and left hamstring compared to the rest of the muscle groups. We observed similar results from the feature importance of random forest on the remaining 5 feature spaces.

7. Conclusion and future work

The primary task of this project was to explore a way to process and classify EMG signals for the purpose of assisting the professionals in Biomedical field. We approached the problem by first processing the signals and trying different ways of extracting the time-series features. We then chose a particular combination and ran the models on different feature spaces. We came up with the feature space with best performance and chose Random Forest for results.

Random Forest associated the use of thigh and hamstring muscles as the major component of classifying between aggressive and normal activities. We also derived the top features of these muscle groups that affect the model the most. While our model successfully classifies aggressive actions against normal ones, it might be interesting to try running muli-class classification to further analyse the difference between each action.

Furthermore, in our current study, we only had 4 subjects for analysis. Since the features derived from these subjects were not independent of each other, the independence assumption for machine learning models does not hold true. For future work, we recommend that the study be conducted on a larger set of diverse people so that we get unbiased result.

References

[1]Mayoclinic.org. (2018). *Electromyography (EMG) - Mayo Clinic*. [online] Available at: https://www.mayoclinic.org/tests-procedures/emg/about/pac-20393913 [Accessed 9 May 2018].

[2]Reaz, M., Hussain, M. and Mohd-Yasin, F. (2006). Techniques of EMG signal analysis: detection, processing, classification and applications (Correction). *Biological Procedures Online*, 8(1), pp.163-163.

[3]Merriam-webster.com. (2018). *Definition of KINESIOLOGY*. [online] Available at: https://www.merriam-webster.com/dictionary/kinesiology [Accessed 9 May 2018].

[4] Sezgin, N. (2012). Analysis of EMG Signals in Aggressive and Normal Activities by Using Higher-Order Spectra. *The Scientific World Journal*, 2012, pp.1-5.
[5]Merlo, A., Farina, D. and Merletti, R. (2003). A fast and reliable technique for muscle activity detection from surface EMG signals. *IEEE Transactions on Biomedical Engineering*, 50(3), pp.316-323.

[6]Tsfresh.readthedocs.io. (2018). *Introduction — tsfresh 0.11.0.post0.dev1+nge7f2e56 documentation*. [online] Available at: http://tsfresh.readthedocs.io/en/latest/text/introduction.html [Accessed 9 May 2018].

[7]Archive.ics.uci.edu. (2018). *UCI Machine Learning Repository: EMG Physical Action Data Set Data Set*. [online] Available at: https://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set [Accessed 9 May 2018].

[8]Biomech.uottawa.ca. (2018). [online] Available at: http://www.biomech.uottawa.ca/english/teaching/apa6905/lab/MatLab%20Help%20-%20EMG%20Analysis.pdf [Accessed 9 May 2018].

[9]Scikit-learn.org. (2018). *sklearn.preprocessing.StandardScaler — scikit-learn 0.19.1 documentation*. [online] Available at: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [Accessed 9 May 2018].