2023

Electromyogram (EMG) For Prosthetic Fingers Control

Name	SEC	BN
Ghufran Mohammed	2	8
Kareman Yasser	2	9
Mayar Fayez	2	42
Nada Ahmed	2	46
Naira Youssef	2	48

Dr.Eman Ayman , TA. Abderahman Hesham

Abstract

Surface Electromyogram (EMG) signals are usually utilized as a control source for multifunction powered prostheses in this report we will use machine learning classification algorithms to identify the movement of each finger. EMG dataset collected for the purpose of this report from eight subjects with eight electrodes attached on their forearm. Five classes of fingers movements where considered. In this report we illustrate our dataset and show the steps of features extraction and selection.

Problem definition

EMG signals recorded from amputees residual muscles have been extensively investigated as a source of control for prosthetic devices, denoted as myoelectric control. In this report we use different machine learning classification algorithms such as SVM, KNN to identify the movement of each finger and use that to control prosthetic devices movements, various features in time and frequency domain were extracted, using features selection methods such as Wrapper Method we managed to drop unrelated features.

Literature review

Hand amputees would highly benefit from a robotic prosthesis, which would allow the movement of a number of fingers. In this paper we propose using the electromyographic signals recorded by two pairs of electrodes placed over the arm for operating such prosthesis. Multiple features from these signals are extracted whence the most relevant features are selected by a genetic algorithm as inputs for a simple classifier. This method results in a probability of error of less than 2%. [1]

In the paper, two EMG features, namely, enhanced wavelength (EWL) and enhanced mean absolute value (EMAV) are proposed. The EWL and EMAV are the modified version of wavelength (WL) and mean absolute value (MAV), which aims to enhance the prediction accuracy for the classification of hand movements. [2]

A challenge that arises with the current demands of such prostheses is the ability to accurately control a large number of individual and combined fingers movements and to do so in a computationally efficient manner. As a response to such a challenge, we present a combined feature selection and projection algorithm, denoted as Mutual Components Analysis (MCA). The proposed MCA algorithm extends the well-known Principal Components Analysis (PCA) by pruning the noisy and redundant features before projecting the data. To implement the feature selection step, the mutual information concept is utilized to implement a new information gain evaluation function. [3]

The Data Description

Eight subjects, six males and two females, aged between 20-35 years were recruited to perform the required fingers movements. The subjects were all normally limbed with no neurological or muscular disorders. Subjects were seated on an armchair, with their arm supported and fixed at one position. The datasets were recorded using eight EMG channels as shown in Fig.1, the EMG signals were then bandpass filtered between 20-450 Hz with a notch filter implemented to remove the 50 Hz line interference. Five classes of movements were collected during this experiment, which are the flexion of each of the individual fingers, Thumb (T), Index (I), Middle (M), Ring (R), Little (L) as shown in Fig.2. The subjects were asked to perform each of the aforementioned five movements, and hold that movement for a period of 5 seconds in each trial. 3 trials, or repetitions, of each movement were collected. The data collected was saved in a CSV files, each file has 8 columns representing each electrode. The files were named according to the movement done while recording for example L_L1 is the little finger movement in the first trail.

Notice that the test data should have the same shape and size as our collected data then it should be gone into feature extraction process as it will be clarified in the next section.

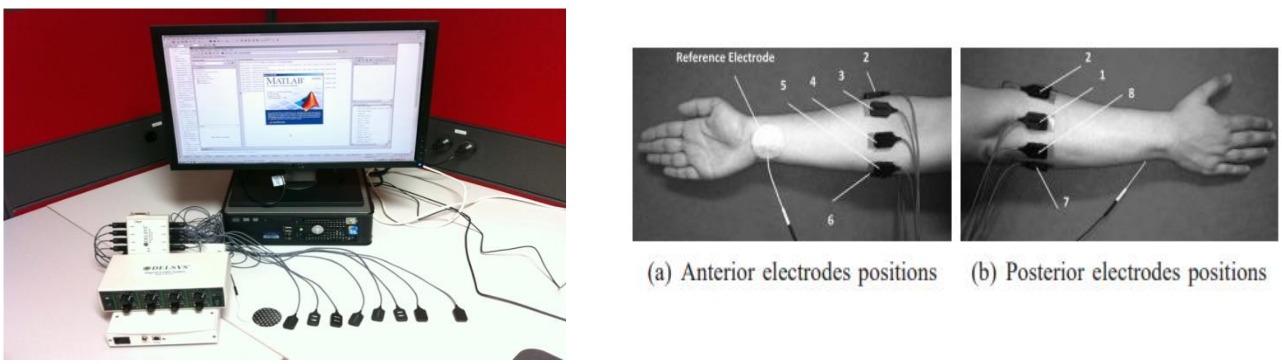


Fig.1 Electrodes placement on the right forearm

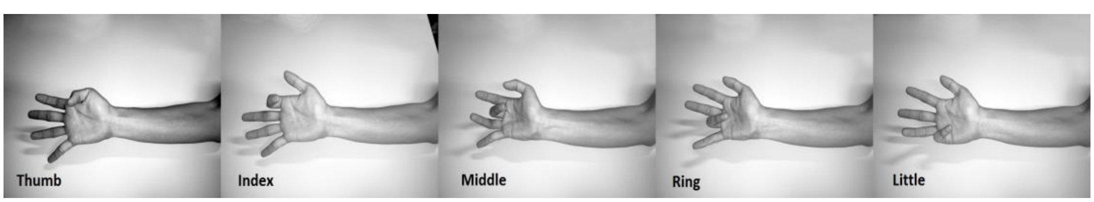
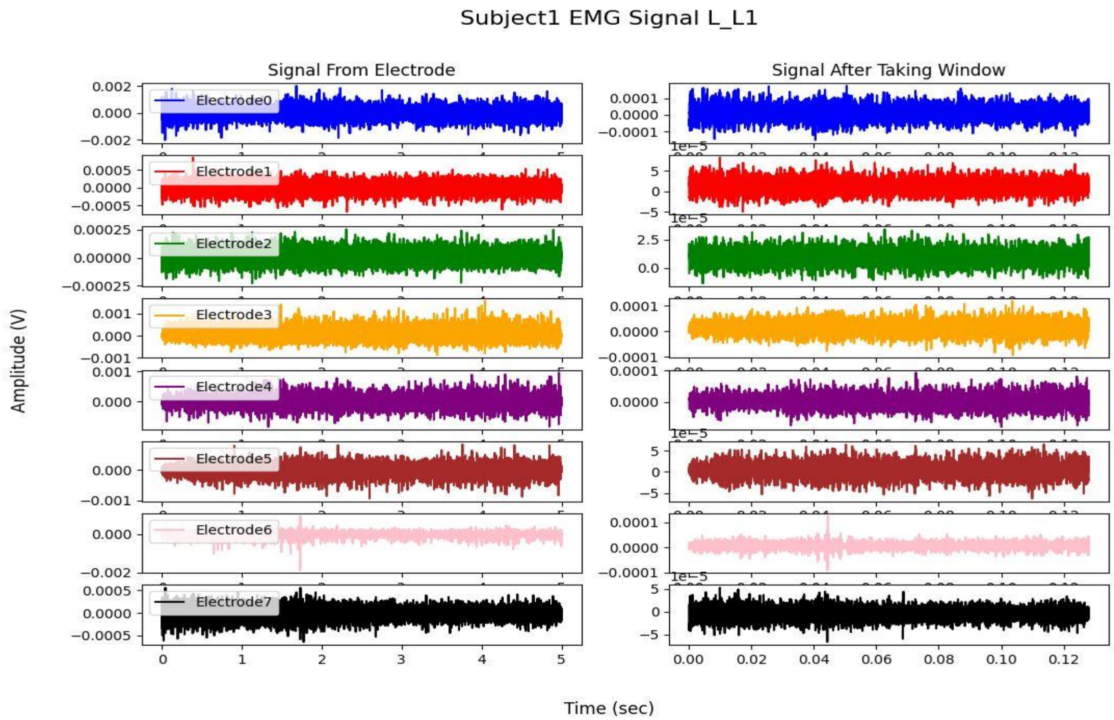
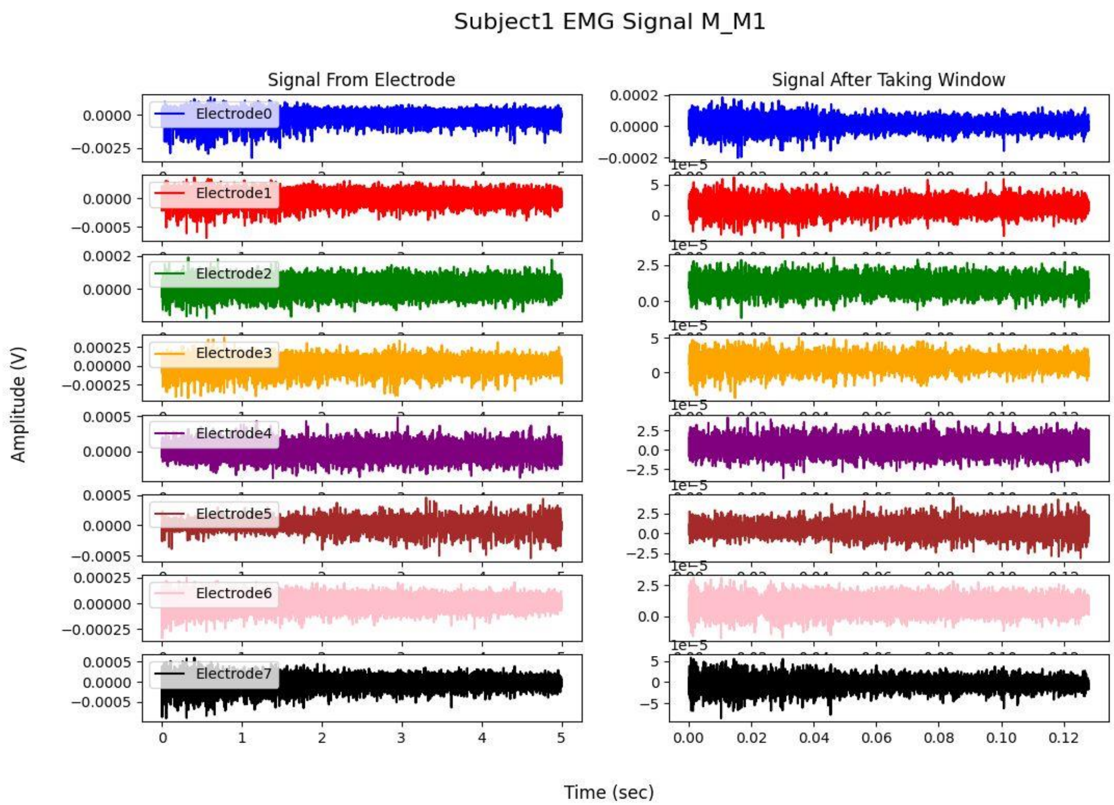
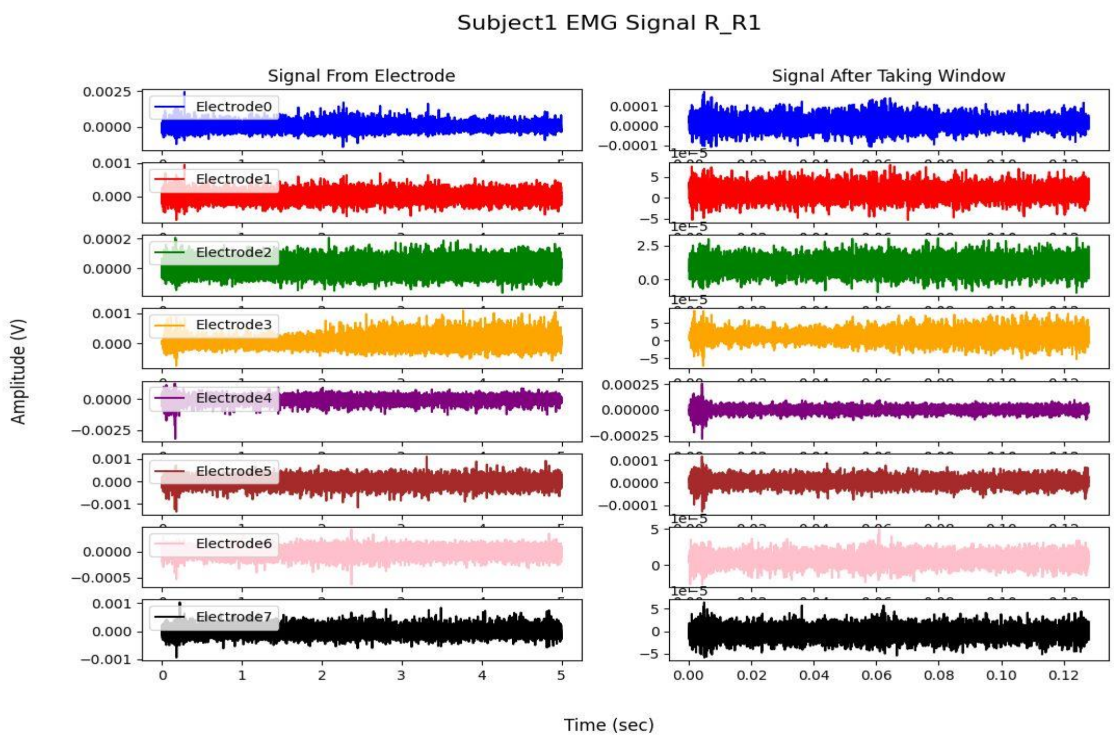


Fig.2 Different classes of individual fingers movement considered in this report

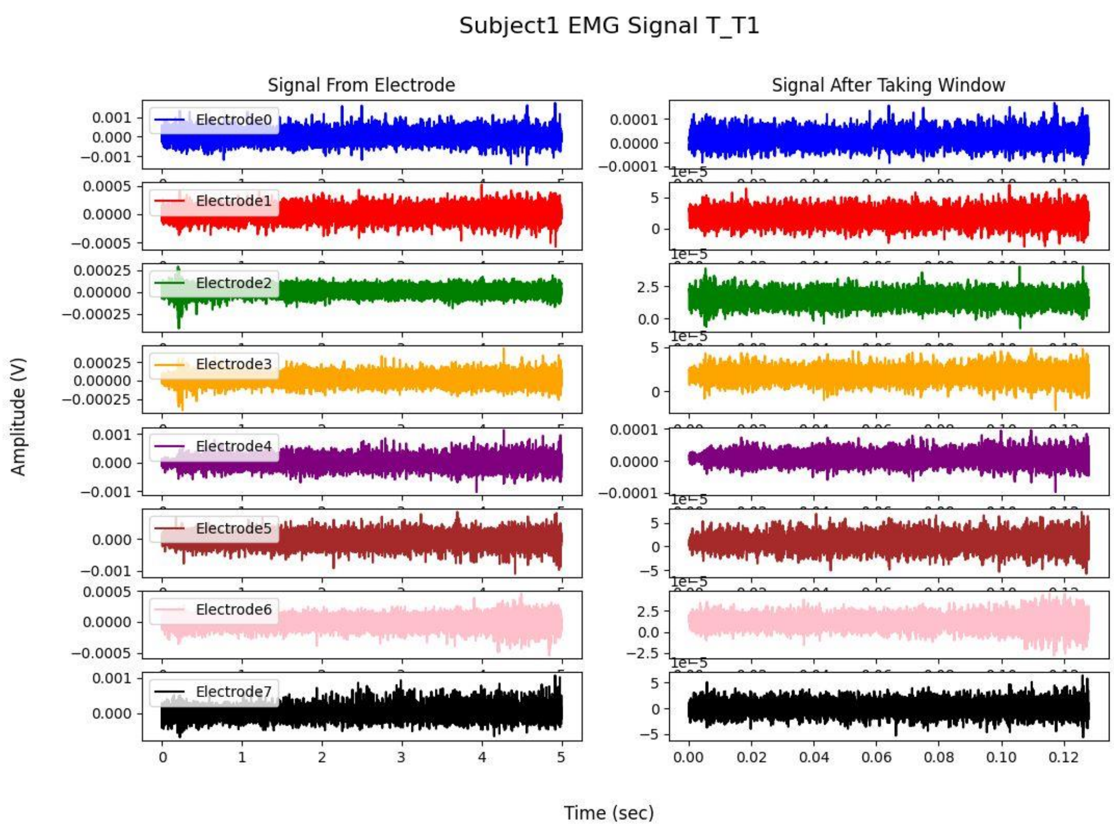
We visualized our records here is an example of subject1 1st trail records of movements and a 128msec window of each record.







Subject1 EMG Signal R_R1
Signal From Electrode
Signal After Taking Window
Electrode0
Electrode1
Electrode2
Electrode3
Electrode4
Electrode5
Electrode6
Electrode7
Amplitude (V)
Time (sec)



Subject1 EMG Signal T_T1
Signal From Electrode
Signal After Taking Window
Electrode0
Electrode1
Electrode2
Electrode3
Electrode4
Electrode5
Electrode6
Electrode7
Amplitude (V)
Time (sec)

Methodology

Feature extraction

We employed the entire signal, a window of the signal, and an overlap signal. When extracting the features from a window of the signal, and an overlap signal, a 128msec analysis window size was used. To depict the EMG activity, different time-domain features were recovered.

Features [4]

- number of zero crossings (1 feature)
- waveform length (1 feature)
- number of slope sign changes (1 feature)
- skewness (1 feature)
- root-mean-square (1 feature)
- mean absolute value (1 feature)
- integral absolute value (1 feature)
- parameters of an autoregressive (AR) model with an order of 11 providing significant enhancements upon smaller model orders (11 features)
- the Hjorth time-domain parameters (3 features)

In a problem of 8 channels, the total number of extracted features is 168 features (21 features/channel × 8 channels = 168 features).

Features visualization

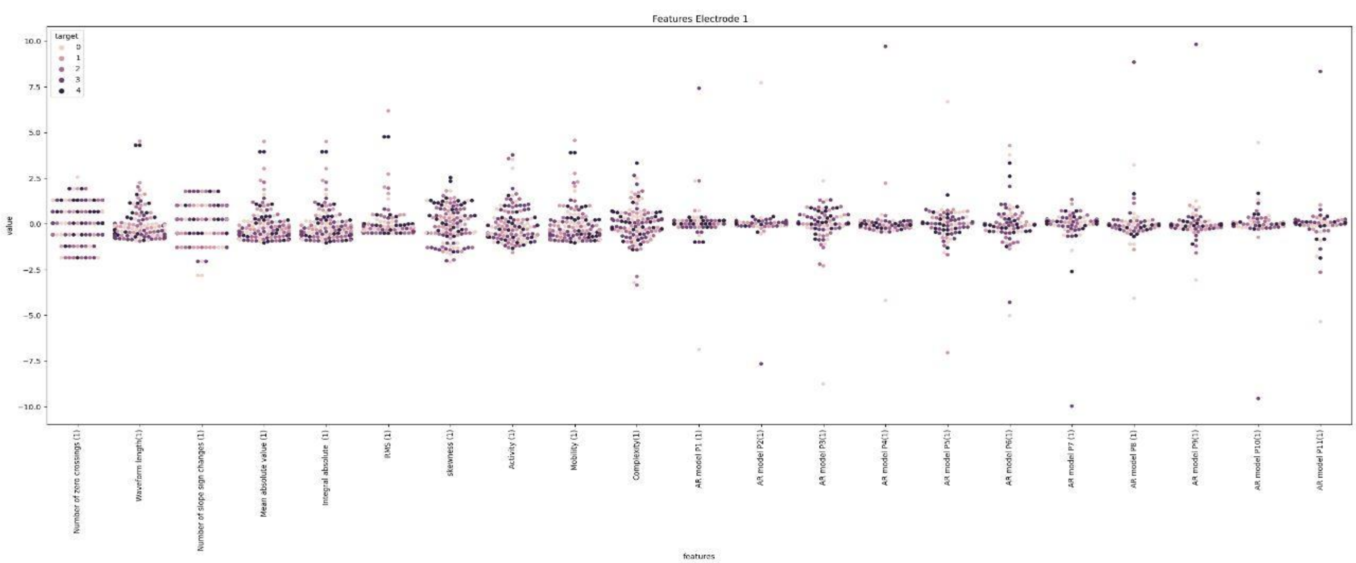


Fig4. Features for electrode 1

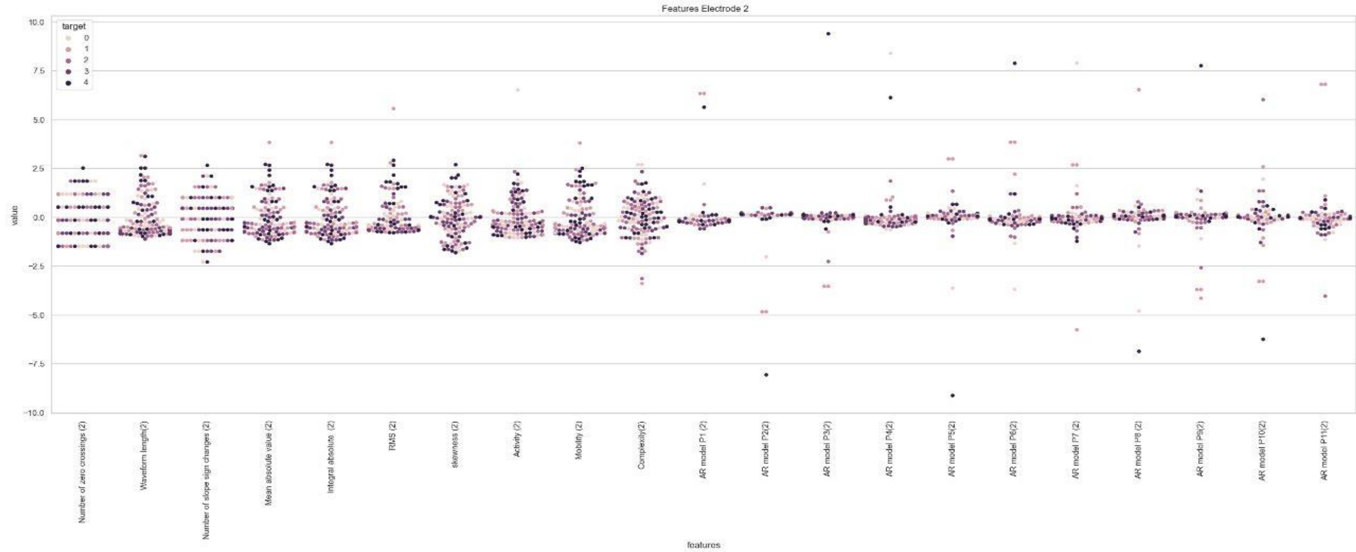


Fig5. Features for electrode 2

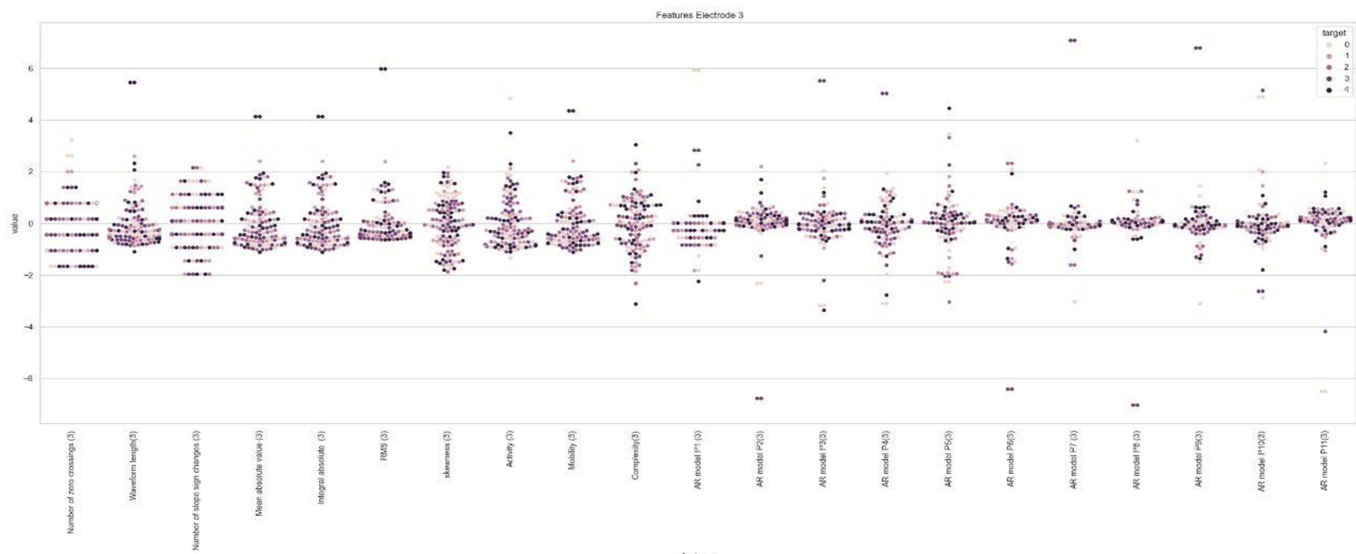


Fig6. Features for electrode 2

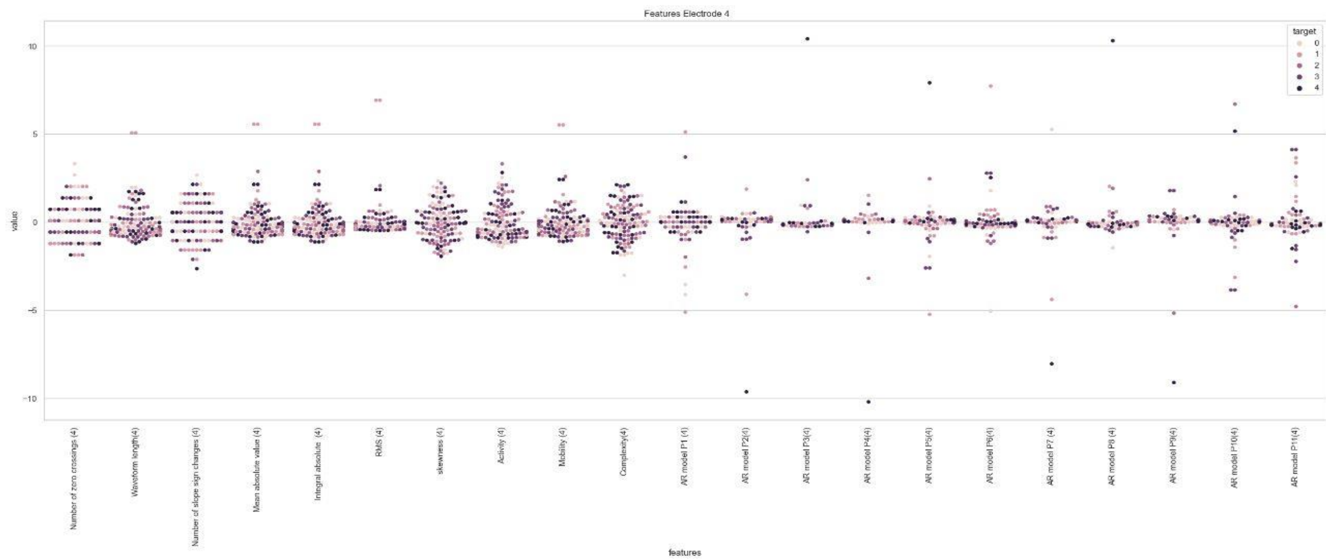


Fig7. Features for electrode 4

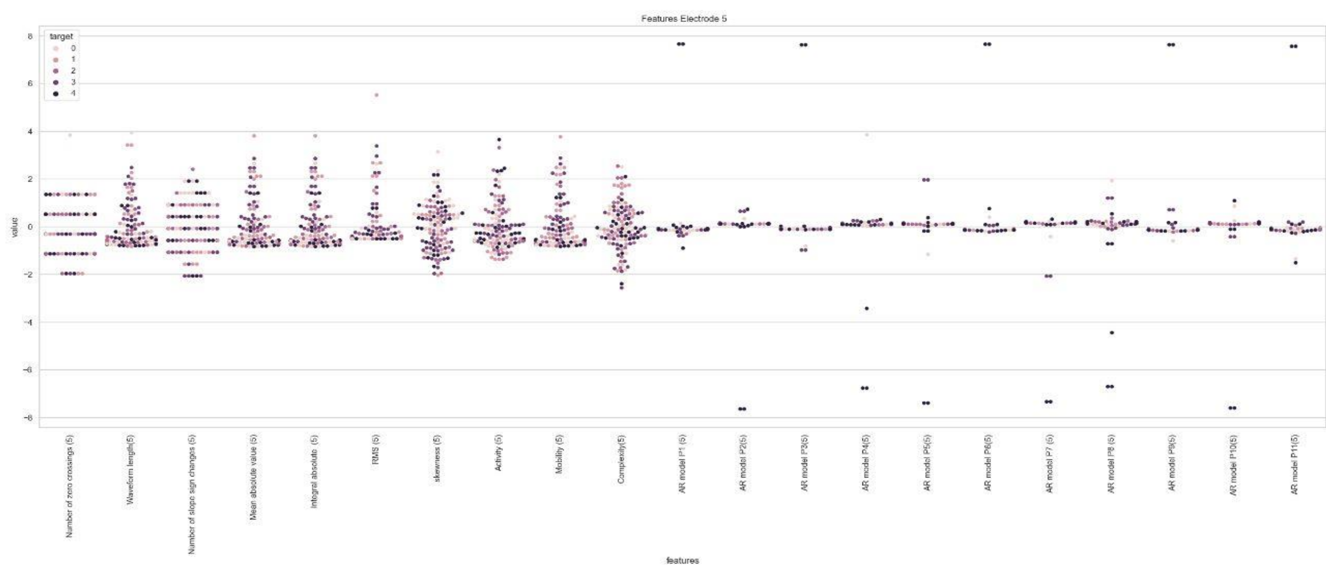


Fig8. Features for electrode 5

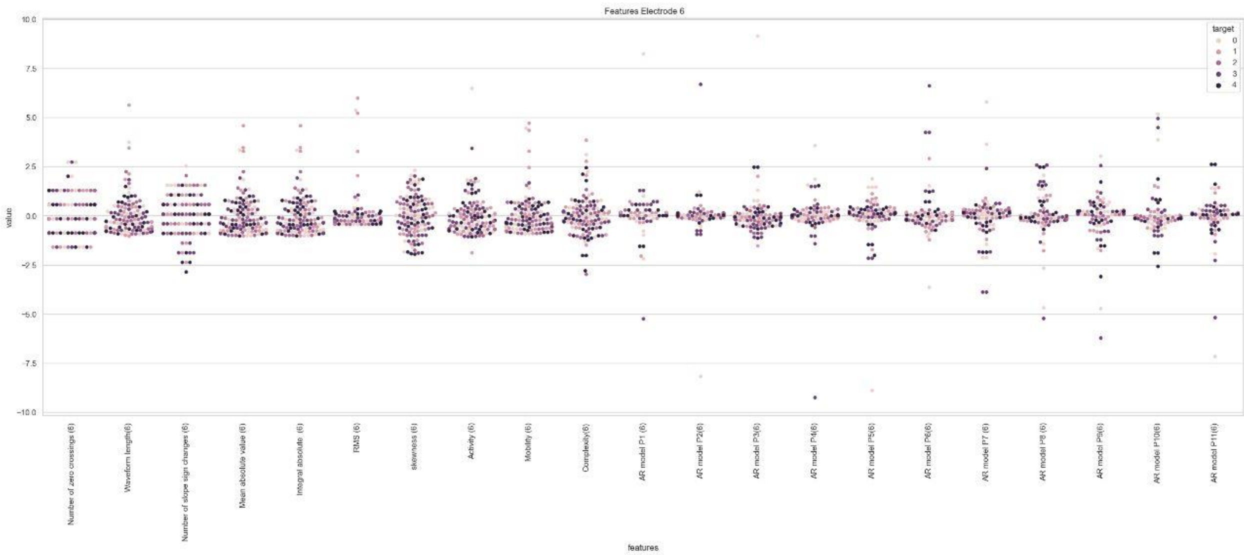


Fig9. Features for electrode 6

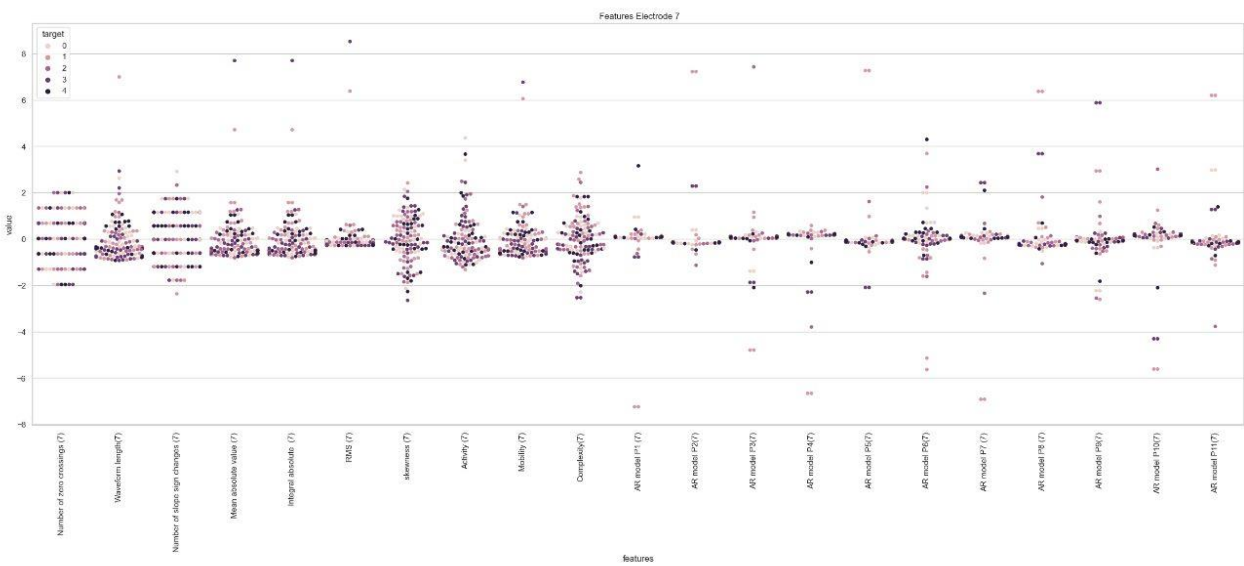


Fig10. Features for electrode 7

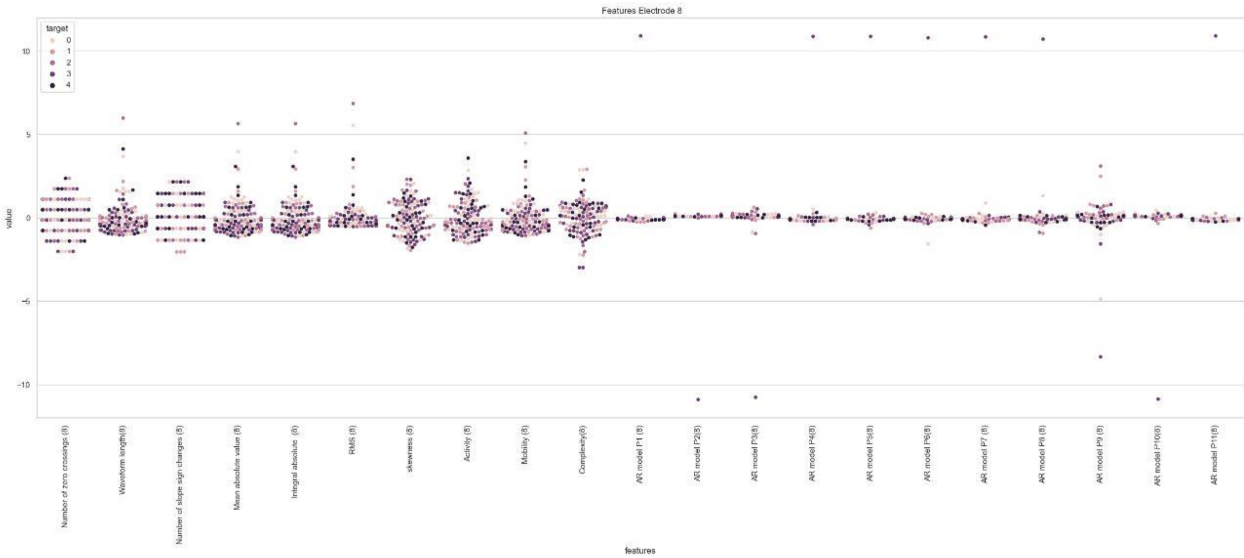


Fig11. Features for electrode 8

Features selection

As stated in previous works, in the classification step, different classifiers were used such as SVM and kNN where SVM classifier parameters were optimized as cost parameter was c = 8 and kernel type was set to a radial basis function with $\gamma = 12/\text{Number of features}$. On the other hand, the number of neighbors in kNN was set to 5 (selected experimentally according to pervious works). [1]

Wrapper methods for feature selection were used such as Sequential Forward Selection (SFS) and step-wise Selection(SFFS), each method was used on the whole signal, 128msec window and 128msec overlapped window to get the number of features to be selected in each case and the name of said features so we have three cases for each case we use two different ways to select features and compare the result to choose the model that preform the best.

Whole signal case

After running our code to determine the best number of feature for this case for kNN and SVM classifiers we got the following results

For kNN

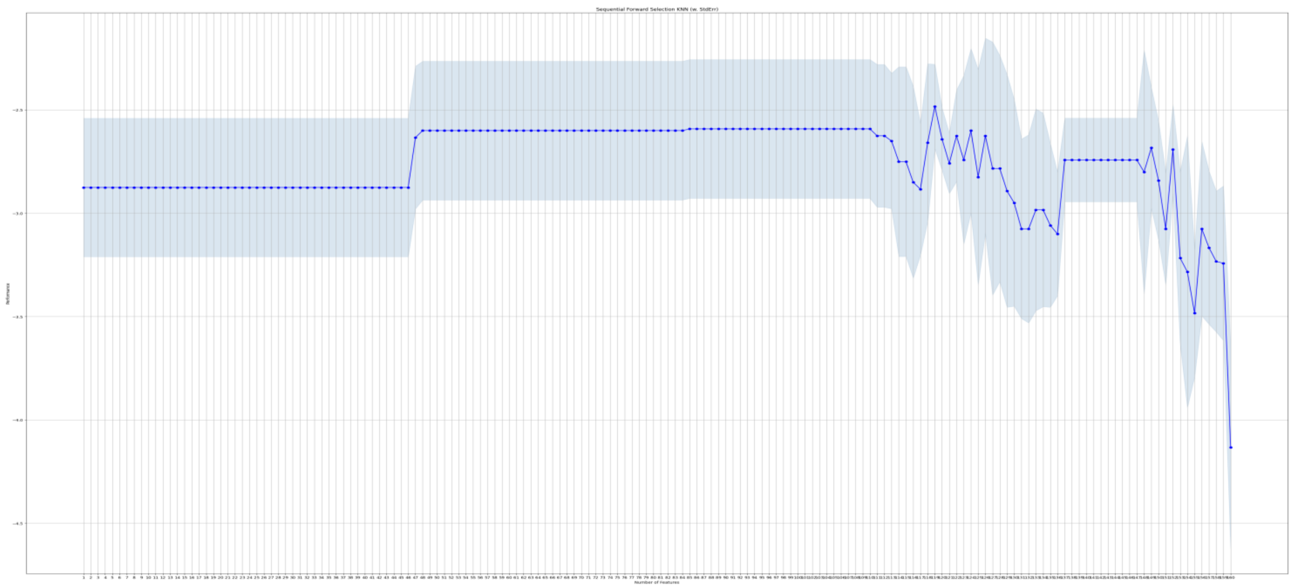


Fig.12. The best number for features is the peak in this graph

It's better illustrated in our source code, deciding from this graph k_features is 119 in SFS and SFFS.

For SVM

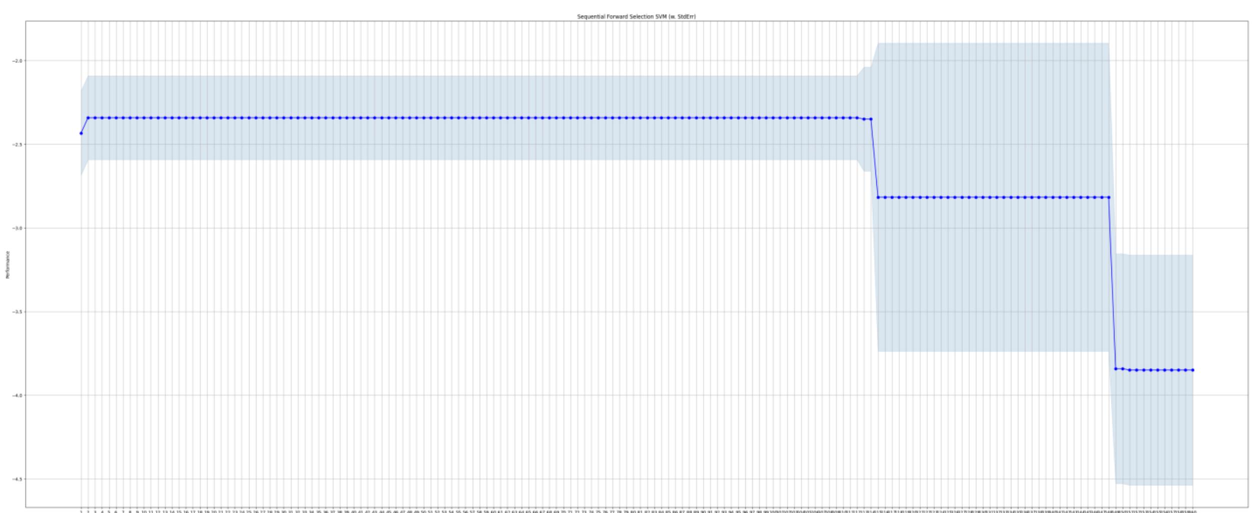


Fig.13. The best number for features is the peak in this graph

Deciding from this graph k_features is 110 in SFS and SFFS. The results indicate that SFFS method has better results for both KNN and SVM so we used the features selected by this method in the classification step as our independent values.

128msec Window case

There was a preprocessing step in this case as we noticed that 8 columns were just repeated ones so we dropped those columns as they are meaningless to us, after that we run our code to determine the best number of feature for this case for kNN and SVM classifiers.

For kNN

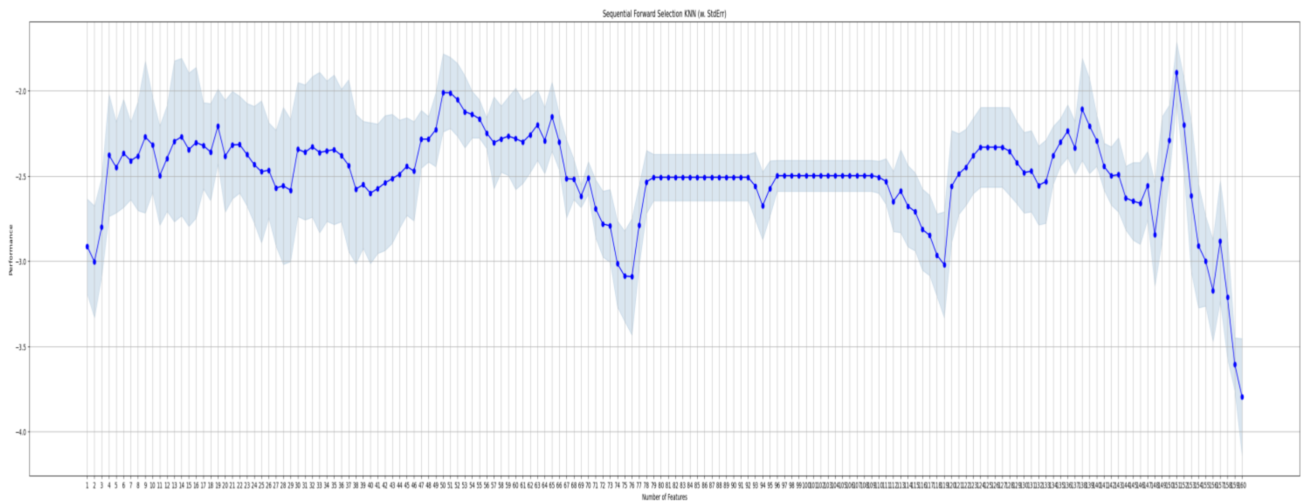


Fig.14. The best number for features is the peak in this graph

Deciding from this graph k_features is 151 in SFS and SFFS.

For SVM

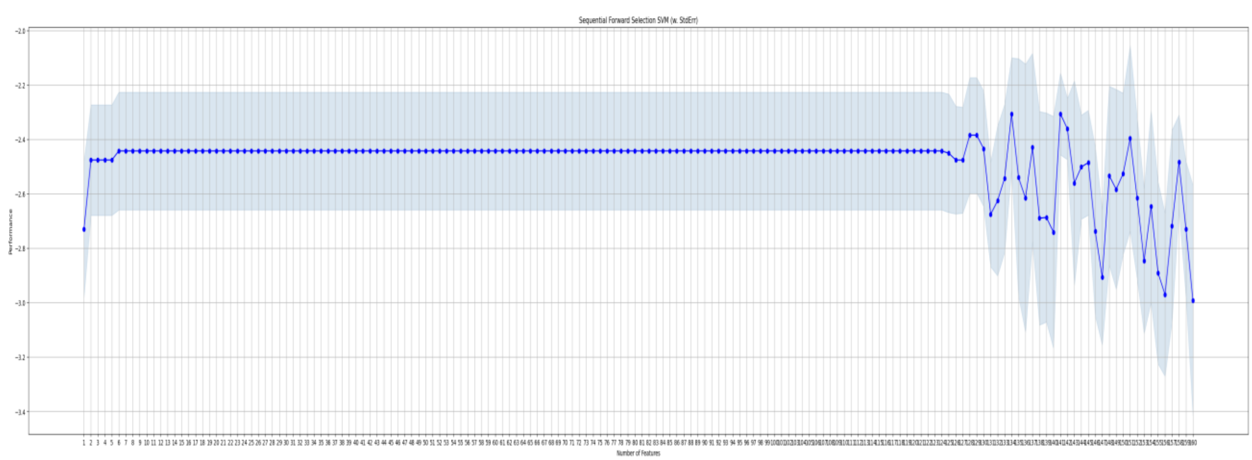


Fig.15. The best number for features is the peak in this graph

Deciding from this graph k_features is 134 in SFS and SFFS. The results indicate that SFS method has better results for both KNN and SVM so we used the features selected by this method in the classification step as our independent values.

128msec Overlap Window case

Similar steps were done as whole signal case the results are clearly illustrated in our source code file.

Example of featured selected by SFS and SFFS methods

	feature_idx	cv_scores	avg_score	feature_names
1	(22,)	[-0.22557588075880752]	-0.225576	(Number of slope sign changes (2),)
2	(22, 102)	[0.03675474254742561]	0.036755	(Number of slope sign changes (2), Number of s...
3	(22, 80, 102)	[0.20071138211382122]	0.200711	(Number of slope sign changes (2), Number of z...
4	(22, 80, 100, 102)	[0.46304200542005425]	0.463042	(Number of slope sign changes (2), Number of z...
5	(22, 42, 80, 100, 102)	[0.7704607046070461]	0.770461	(Number of slope sign changes (2), Number of s...
...	...	...	...	...
130	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
131	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
132	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
133	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
134	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...

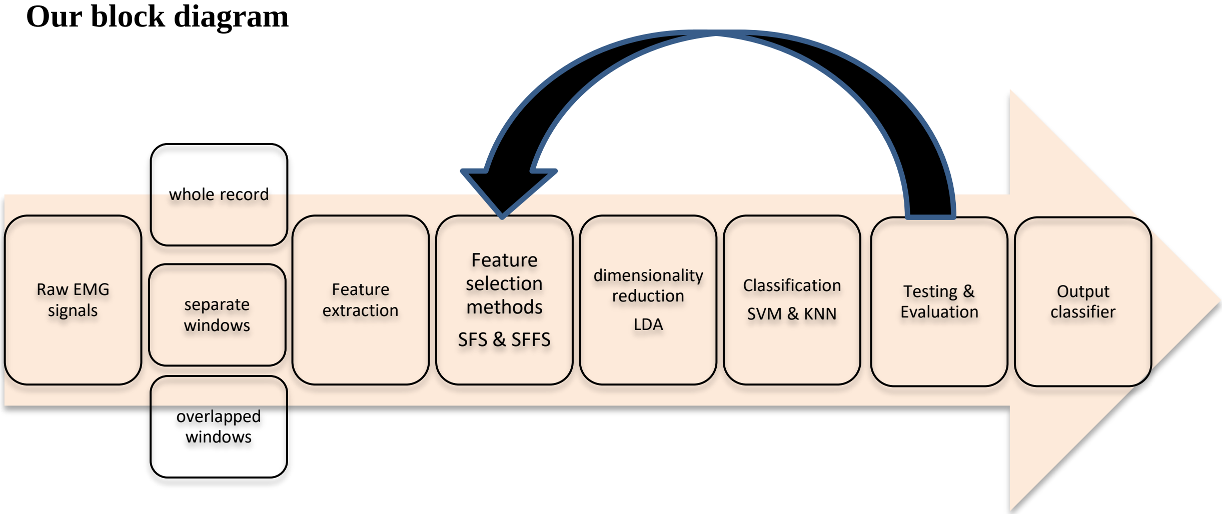
134 rows × 4 columns

	feature_idx	cv_scores	avg_score	feature_names
1	(22,)	[-0.22557588075880752]	-0.225576	(Number of slope sign changes (2),)
2	(22, 102)	[0.03675474254742561]	0.036755	(Number of slope sign changes (2), Number of s...
3	(22, 80, 102)	[0.20071138211382122]	0.200711	(Number of slope sign changes (2), Number of z...
4	(22, 80, 100, 102)	[0.46304200542005425]	0.463042	(Number of slope sign changes (2), Number of z...
5	(22, 42, 80, 100, 102)	[0.7704607046070461]	0.770461	(Number of slope sign changes (2), Number of s...
...	...	...	...	...
130	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
131	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
132	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
133	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...
134	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[1.0]	1.0	(Number of zero crossings (1), Waveform length...

134 rows × 4 columns

Fig16. Feature selected

Our block diagram



Classification step

We used five different classifiers which are SVM, KNN, DT, LR, RF with 5 folds cross validation we also used grid search to improve our parameters, the data was spilt into 20% test and 80% train.

Parameters

```
#Knn parameters
knn = KNeighborsClassifier(n_neighbors=5)
#DT parameters
param_dict ={
    "criterion":['gini','entropy'],
    "max_depth":(150, 155, 160),
    "min_samples_split":range(1,10),
    "min_samples_leaf":range(1,5)
 }
DT_model= DecisionTreeClassifier(random_state=42)
# RF parameters
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}
RF_model= RandomForestClassifier(random_state=42)
#  LR parameters
grid_values = {'penalty': ['l1','l2'], 'C': [0.001,0.01,0.1,1,10,100,1000]}
LR_model= LogisticRegression(random_state=42)
# SVM parameters
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}

svc = SVC(kernel = 'rbf', C = 8, gamma = 12/160, probability=True)
```

Results

Processed data	Classifiers	Accuracy	Precision
All data (No feature selected)	kNN	0.958	0.96
	DT	0.708	0.718
	RF	0.83	0.88
	LR	0.37	0.37
	SVM	0.12	0.50
Whole signal (Selected Features1) SFFS by SVM	kNN	0.958	0.96
	DT	0.58	0.558
	RF	0.875	0.968
	LR	0.375	0.338
	SVM	0.58	0.66
Whole signal (Selected Features 2) SFFS by kNN	kNN	0.83	0.84
	DT	0.66	0.768
	RF	0.916	0.9375
	LR	0.625	0.726
	SVM	0.666	0.694
128msec window (selected Features 1) SFS by SVM	kNN	0.45	0.63
	DT	0.16	0.34
	RF	0.33	0.56
	LR	0.20	0.25
	SVM	0.33	0.30
128msec window (selected Features 2) SFS by kNN	kNN	0.75	0.63
	DT	0.25	0.34
	RF	0.20	0.56
	LR	0.08	0.25
	SVM	0.33	0.30
128msec overlapped window (selected Features 1) SFS by SVM	kNN	0.92	0.94
	DT	0.62	0.76
	RF	0.79	0.85
	LR	0.375	0.52
	SVM	0.75	0.89
128msec overlapped window (selected Features 2) SFS by kNN	kNN	0.87	0.935
	DT	0.62	0.76
	RF	0.75	0.85
	LR	0.29	0.52
	SVM	0.75	0.89

Table1

Examples of confusion matrix

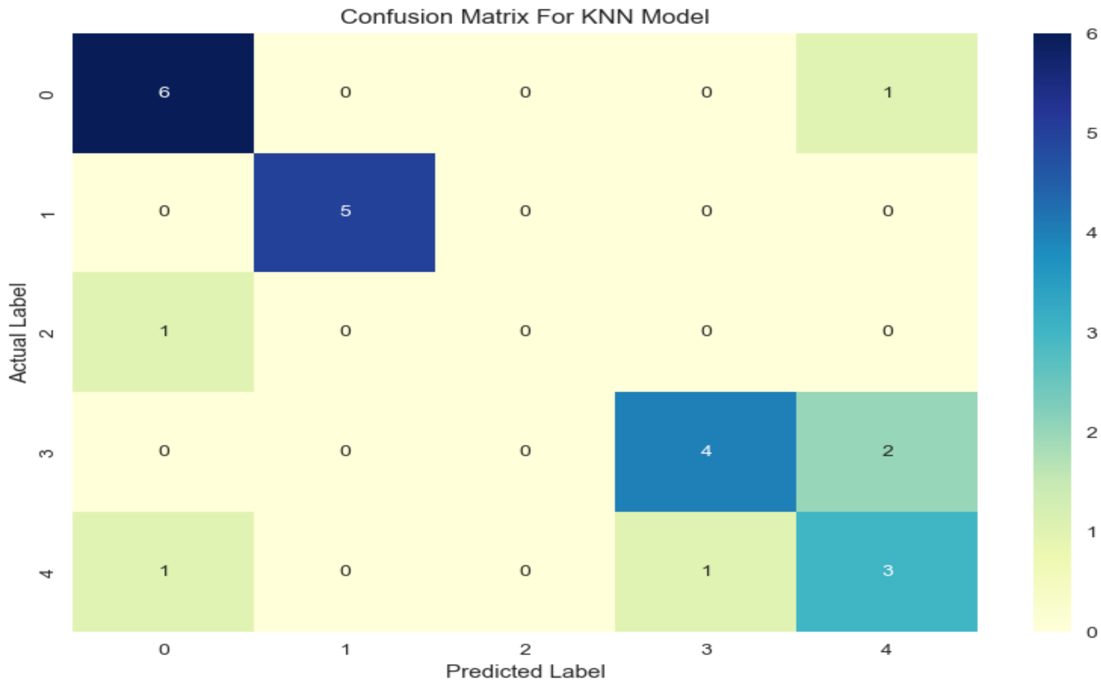


Fig17.

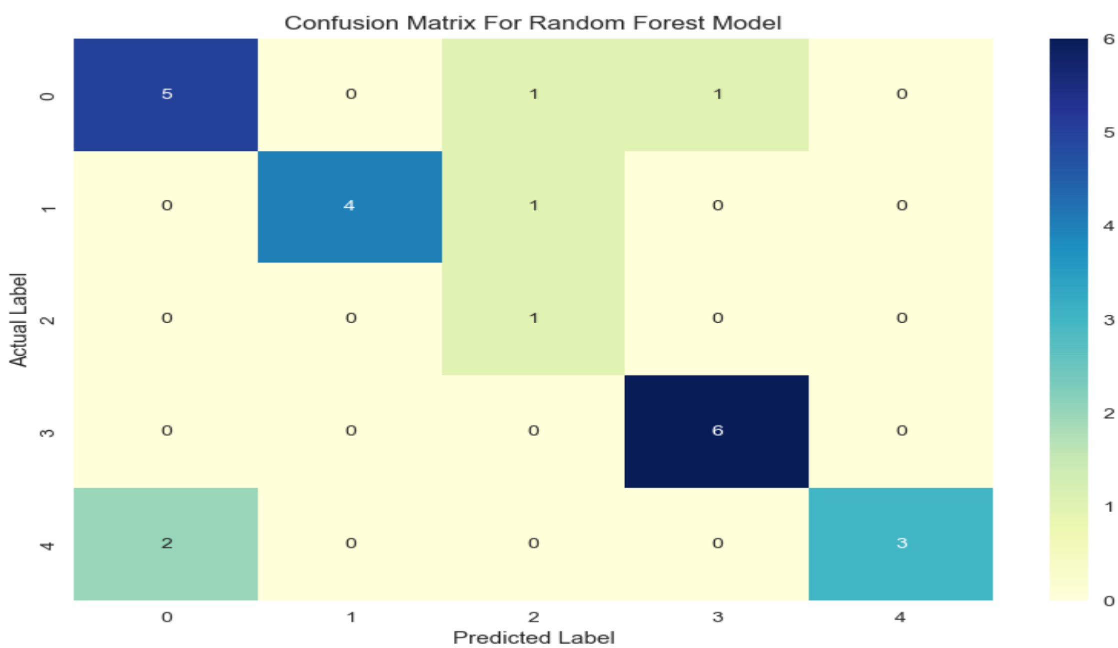


Fig18.

The rest of confusion matrices are well illustrated in the source code file

Examples of ROC curves

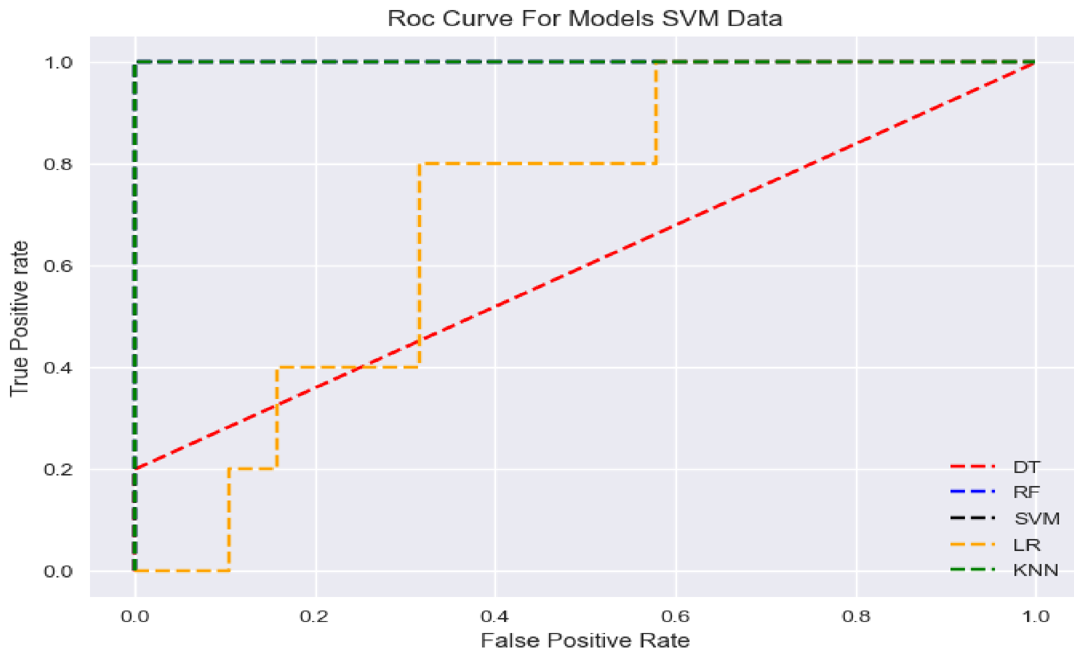


Fig19.

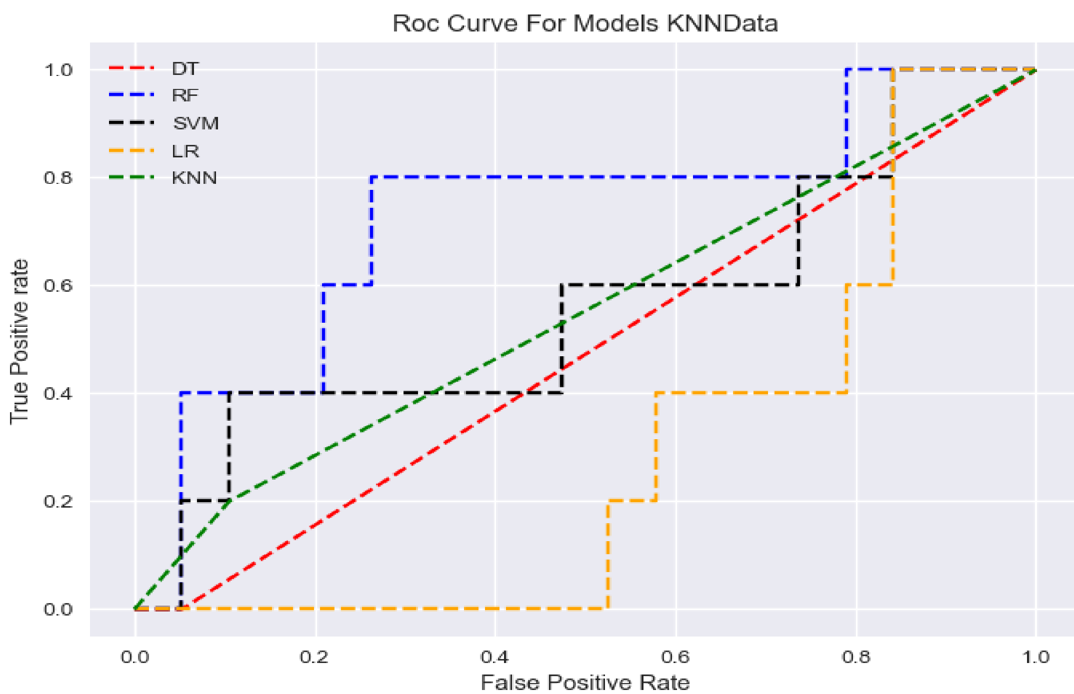


Fig20.

The rest of ROC curves are well illustrated in the source code file

Discussion

As shown in table1 our results indicate that LR classifier has the lowest accuracy and Precision while kNN has shown better results in all our cases with the best two accuracies 0.95 in the all data case and 0.92 in the 128msec overlapped window case. For SVM, RF, and DT there were a variety in the resulted accuracy in all the cases. We also noticed that the 8^{th} electrode is useless and we could have used 7 electrodes for more simple data acquisition process. In addition to what the paper suggested we tried three different cases, as stated in the paper they only took a 128msec window (128msec window case) while we did the process on the whole signal a case without selecting any features (all data case)and another with selected feature (whole signal case) and a case on an overlapped windows with 128msec size (128msec overlapped window case).

Overall the 128msec overlapped window case and all data case has better results than other cases

Challenges we faced during processing

As stated in the paper different dimensionality reduction techniques were applied on the selected features we tried only one technique which is LDA and we got overfitted models and we were tight on time because some feature selection methods took a lot of run time from 30 min to 2 hours so we didn't get to try other techniques**.**

Source code and info documents

https://github.com/Naira06/CDSS_final-project.git

References

[1] Dori Peleg, Eyal Braiman, Elad Yom-Tov and Gideon F. Inbar, "Classification of Finger Activation for Use in a Robotic Prosthesis Arm", IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2002.

[2] Jingwei Too, Abdul Rahim Abdullah and Norhashimah Mohd Saad, "Classification of Hand Movements based on Discrete Wavelet Transform and Enhanced Feature Extraction", International Journal of Advanced Computer Science and Applications(IJACSA), Volume 10 Issue 6, 2019.

[3] Rami N. Khushaba, Sarath Kodagoda, Dikai Li and Gamini Dissanayake, " Electromyogram (EMG) Feature Reduction Using Mutual Components Analysis for Multifunction Prosthetic Fingers Control", IEEE, 2012.

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