



# MSE 491 - Lab 1

Using LDA to classify EMG  
sensor values

---

Harkirat Gill

301219830, hsg12@sfu.ca

## Table of Contents

<b>Feature extraction .....</b>	<b>1</b>
RMS.....	1
Waveform Length .....	1
AR coefficients .....	2
<b>Classifier.....</b>	<b>2</b>
<b>Results .....</b>	<b>2</b>
Accuracy .....	2
Plots .....	4
Raw Data.....	4
Window Size = 100 .....	5
Window Size = 1000 .....	9
Window Size = 50 .....	13
Window size = 200.....	18

## Table of Figures

Figure 1 Raw Values .....	4
Figure 2 RMS, window size 100 .....	5
Figure 3 Waveform length, window size 100 .....	6
Figure 4 AR, window size 100 .....	7
Figure 5 Classified Train vs Labeled, window size 100 .....	8
Figure 6 Classified Test vs Labeled, window size 100 .....	8
Figure 7 Accuracy of training and testing dataset, window size 100 .....	9
Figure 8 RMS, window size 1000 .....	9
Figure 9 Waveform, window size 1000.....	10
Figure 10 AR, window size 1000 .....	11
Figure 11 Classified test vs Labeled data, window size 1000 .....	12
Figure 12 Classified train vs Labeled data, window size 1000 .....	12
Figure 13 Accuracy of training and test dataset, window size 1000 .....	13
Figure 14 RMS, window size 50 .....	13
Figure 15 Waveform, window size 50.....	14
Figure 16 AR, window size 50 .....	15
Figure 17 Classified test data vs Labeled, window size 50 .....	16
Figure 18 Classified train data vs Labeled, window 50 .....	17
Figure 19 Accuracy of training and testing dataset, window size 50 .....	17
Figure 20 RMS, window size 200 .....	18
Figure 21 Waveform, window size 200.....	19
Figure 22 AR, window size 200 .....	20
Figure 23 Classified train data vs Labeled data, window size 200.....	21
Figure 24 Classified test data vs Labeled data, window size 200 .....	21
Figure 25 Accuracy of training and testing dataset, window set 200 .....	22
Figure 26 Accuracy after discarding half of training dataset, window size 200 .....	22

## Feature extraction

### RMS

```
function result = rms(data, window_size, moving_window)

result = [];
for i = 1:window_size:size(data,1)
    a = sqrt(1/window_size*sum(data(i:i+window_size-1, :).^2));
    result = [result; repmat(a,window_size,1)];
end
```

We traverse through the training data and calculate RMS of all 8 channels over a window size. Variable 'a' takes in the RMS value of all 8 channels. Then we repeat the vector 'a' upto the window size and concat it to the 'result' matrix.

At the end of for-loop, the result contains the rms values of entire data over static windows with size of window\_size.

### Waveform Length

```
for i = 2:window_size:size(data,1)
    a = sum(abs( data(i:i+window_size-2,:) - data(i-1:i+window_size-3,:) ));
    result = [result; repmat(a,window_size,1)];
end
```

Similar format applies to waveform length. 'a' takes the waveform of a fixed static window and 'result' returns the entire waveform of all static windows with no overlap.

### AR coefficients

```

result = [];
for i = 1:window_size:size(data,1)
    for sensor = 1:1:8
        ar_data = data(i:i+window_size-1, sensor);
        AR_coeff = aryule(ar_data , p);
        a(sensor) = -AR_coeff(2:end) * ar_data(end:-1:end-p+1);
    end
    result = [result; repmat(a,window_size,1)];
end

```

'ar\_data' represents the data over which we want to calculate the AR coefficients. 'AR\_coeff' represents the AR coefficients. AR\_coeff is 5x8 long (1 normalized vector + 4 order x 8 sensors)

'a' intakes the  $x_{i+1}$  by utilizing the AR coefficients and last 4 data points from ar\_data.

### Classifier

```

trainer = fitcdiscr([xTrain_rms], yTrain);
classified_train = predict(trainer, [xTrain_rms]);
accuracy_train = sum(yTrain==classified_train)/size(yTrain,1)*100.0
classified_test = predict(trainer, [xTest_rms]);
accuracy_test = sum(yTest==classified_test)/size(yTest,1)*100.0

```

The trainer that yields the highest accuracy is trained using only rms values. Using waveforms or my perception of AR values decreases accuracy significantly.

### Results

#### Accuracy

Window Size	Accuracy on Training Set	Accuracy on Testing Set
100	86.5%	68%
1000	44%	20%
50	83.25%	59%
200	90%	80%

As we note, window size 200 gives the highest accuracy. This is because the labels, i.e., labels in yTrain and yTest, are 200 ms long.

Additionally, I was able to achieve higher accuracy by discarding half of the training dataset.

```
xTrain = xTrain(1:10000,:);  
yTrain = yTrain(1:10000,:);
```

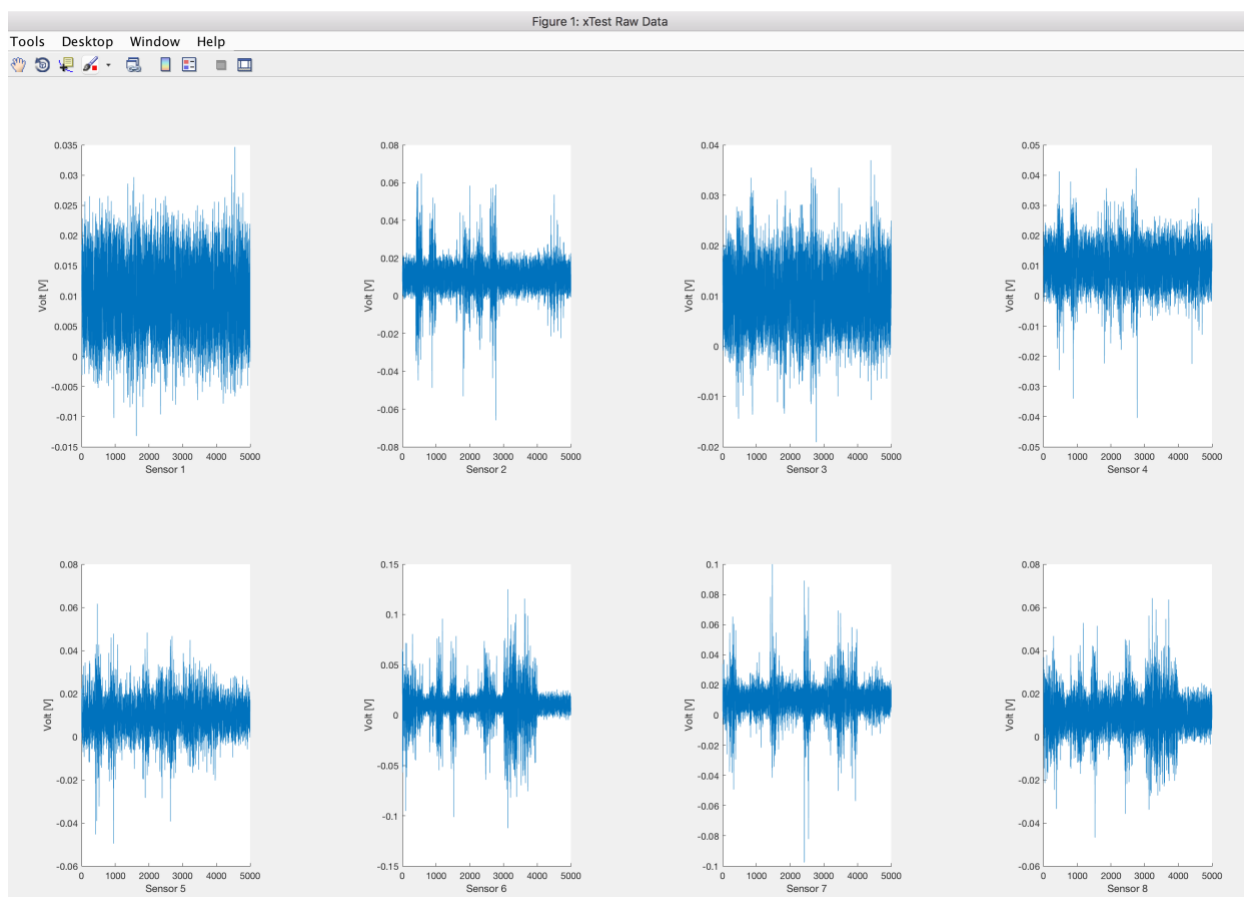
By utilizing the above code, I discarded half of the training dataset and the accuracy after doing so was the following:

Window Size	Accuracy on Training Set	Accuracy on Testing Set
200	86%	84%

This might be because dataset of 20,000 might be overfitting our classifier. The more data is introduced to our classifier, the more variance will exist within classes. This variance might yield a different plane on which the variance within classes might overlap between classes.

## Plots

## Raw Data

*Figure 1 Raw Values*

Window Size = 100

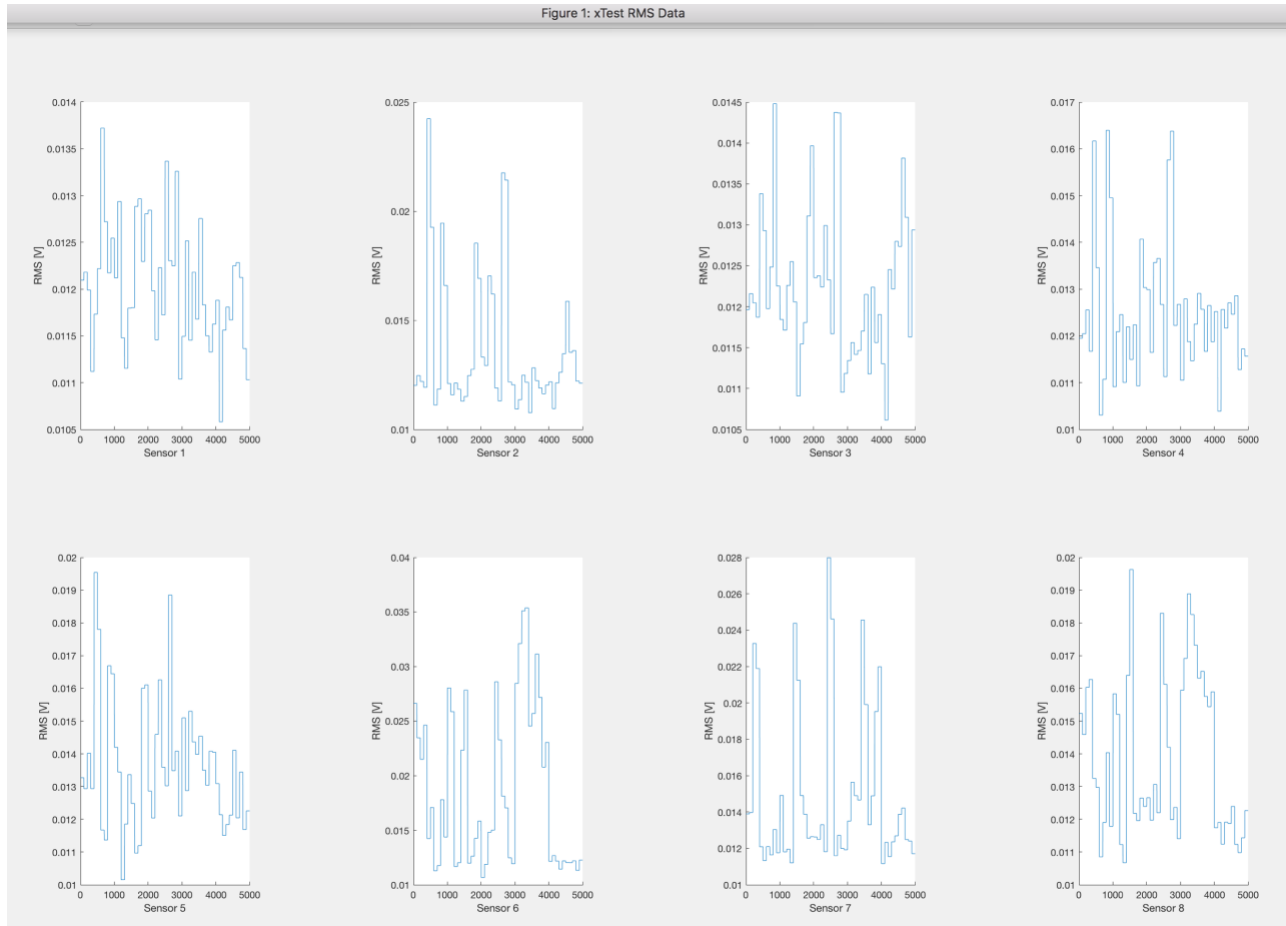


Figure 2 RMS, window size 100



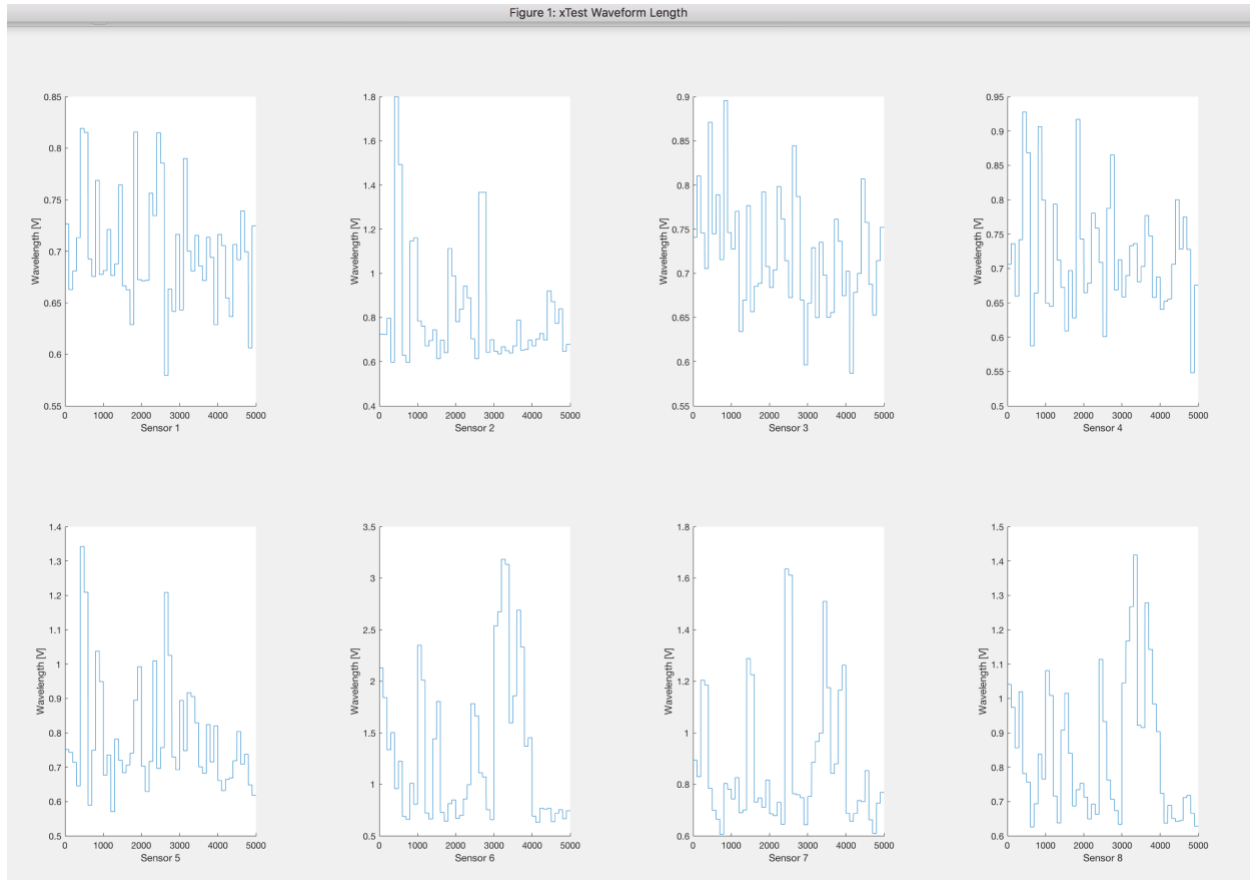


Figure 3 Waveform length, window size 100

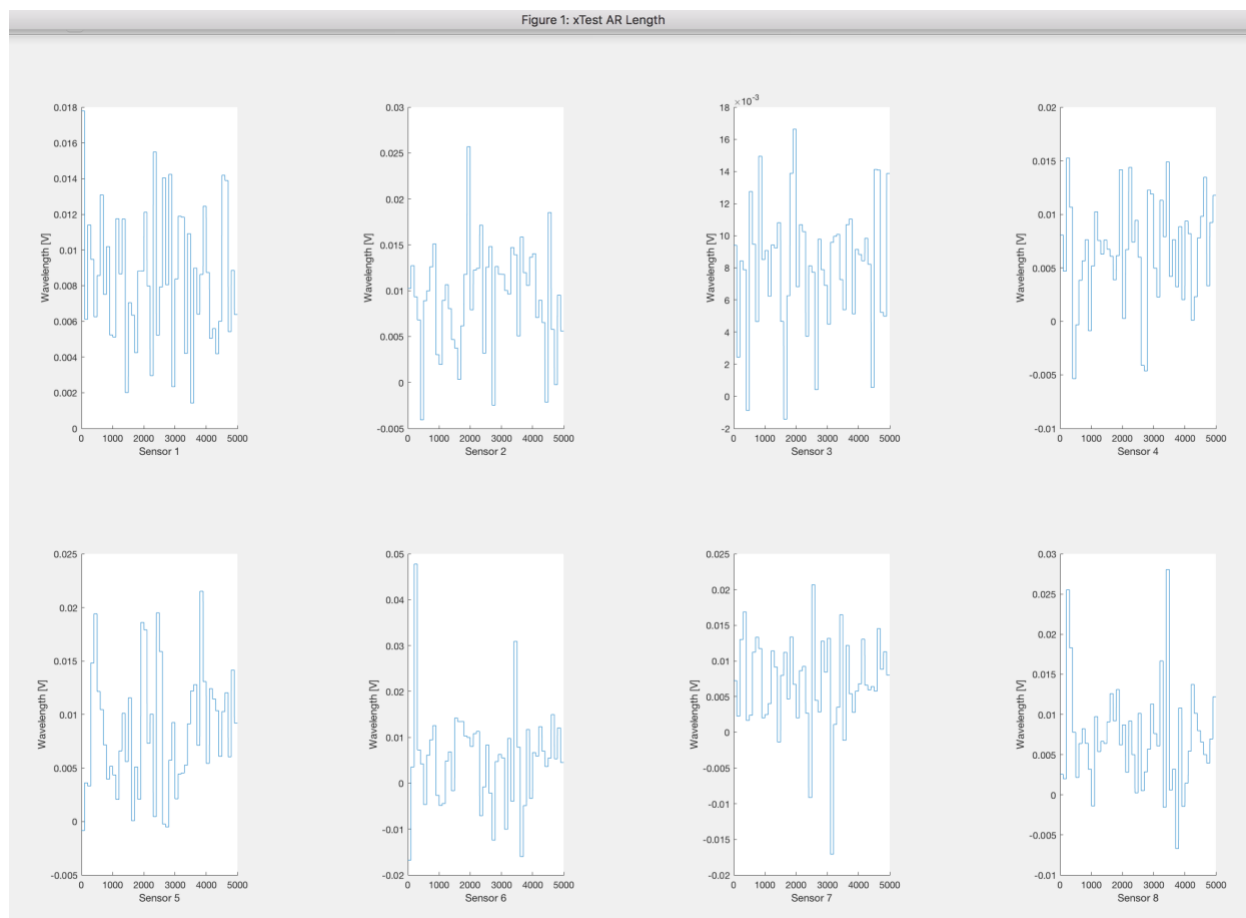


Figure 4 AR, window size 100

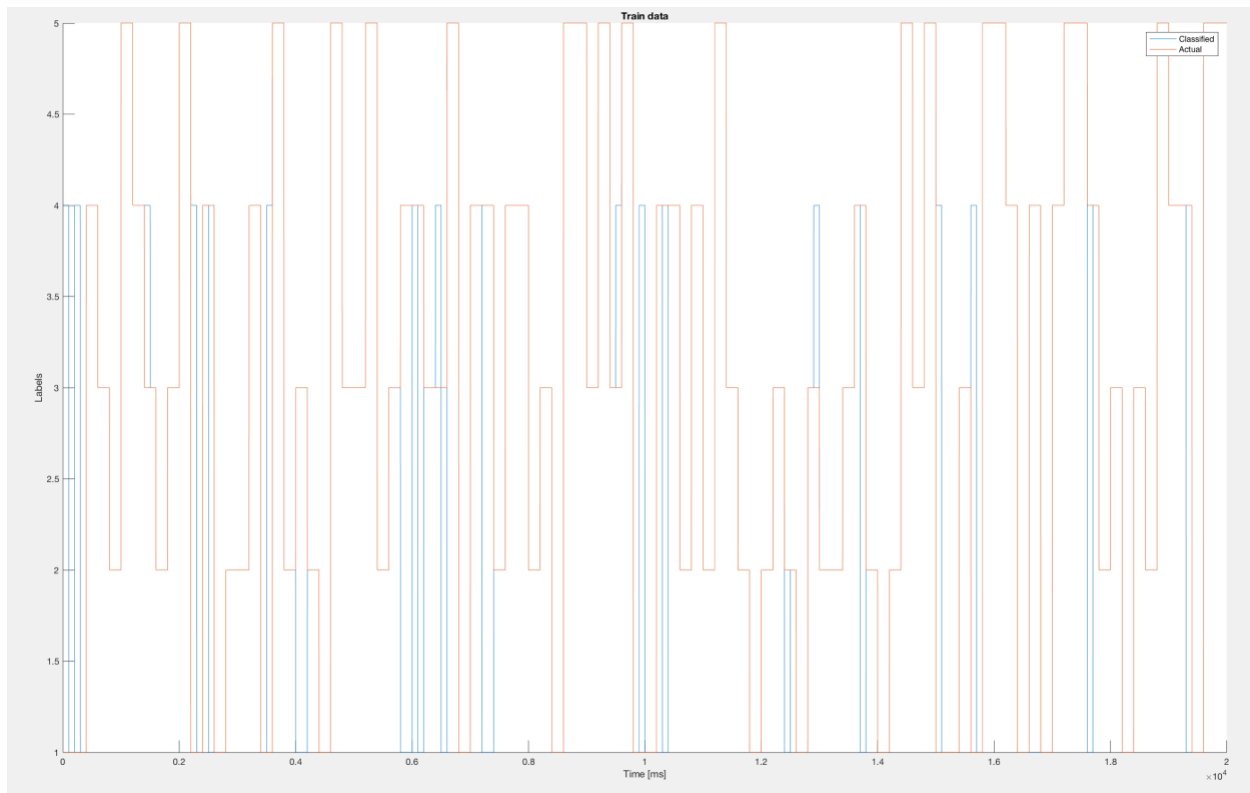


Figure 5 Classified Train vs Labeled, window size 100

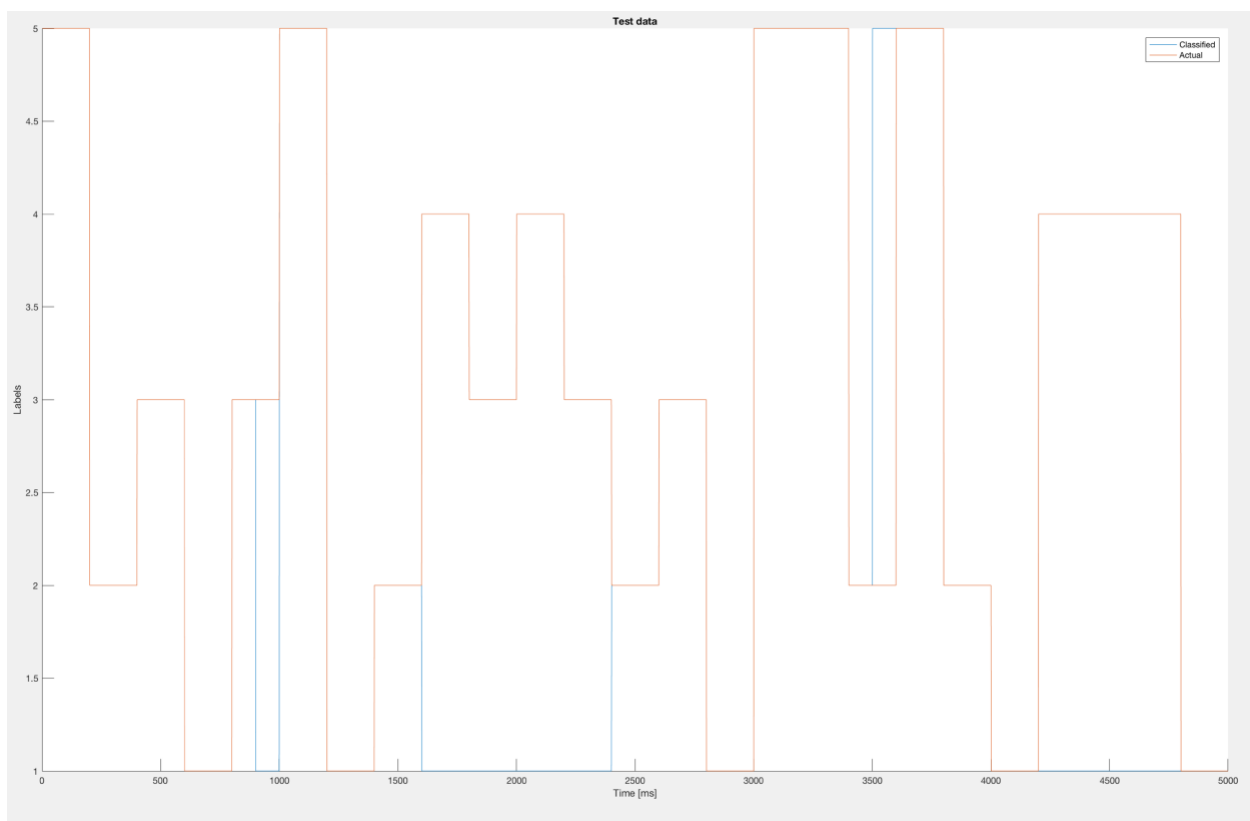


Figure 6 Classified Test vs Labeled, window size 100

accuracy\_train =

86.5000

accuracy\_test =

68

Figure 7 Accuracy of training and testing dataset, window size 100

Window Size = 1000

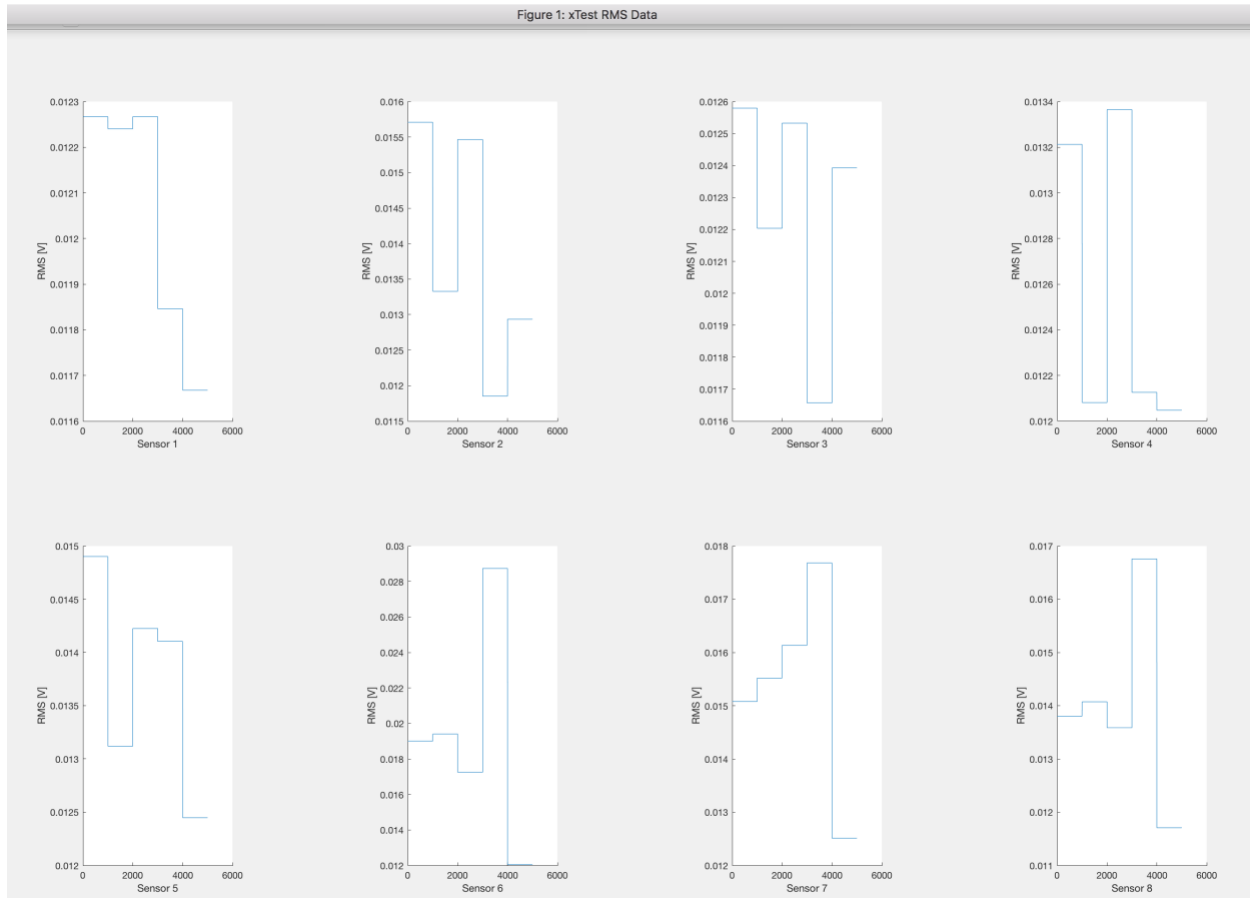


Figure 8 RMS, window size 1000

Figure 2: xTest Waveform Length

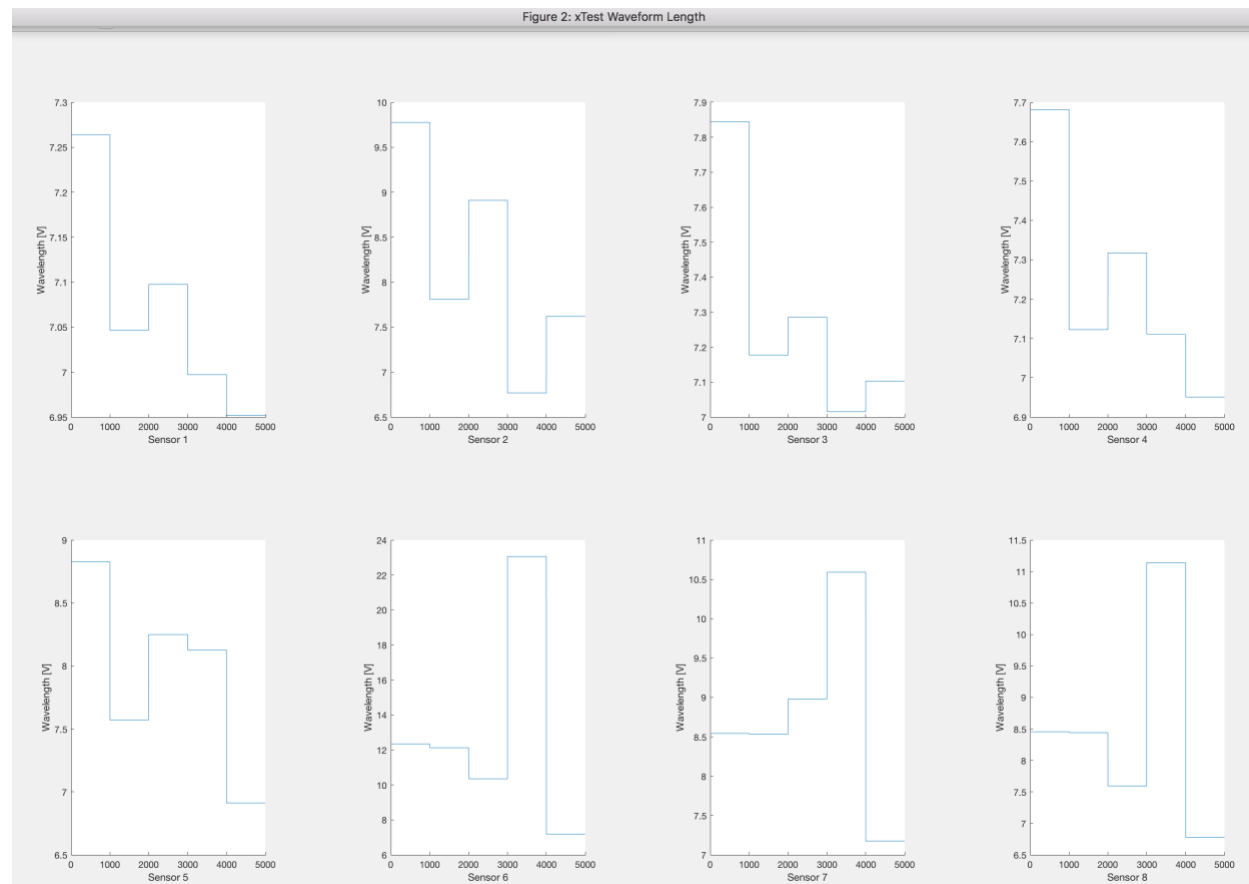


Figure 9 Waveform, window size 1000

Figure 3: xTest AR Length

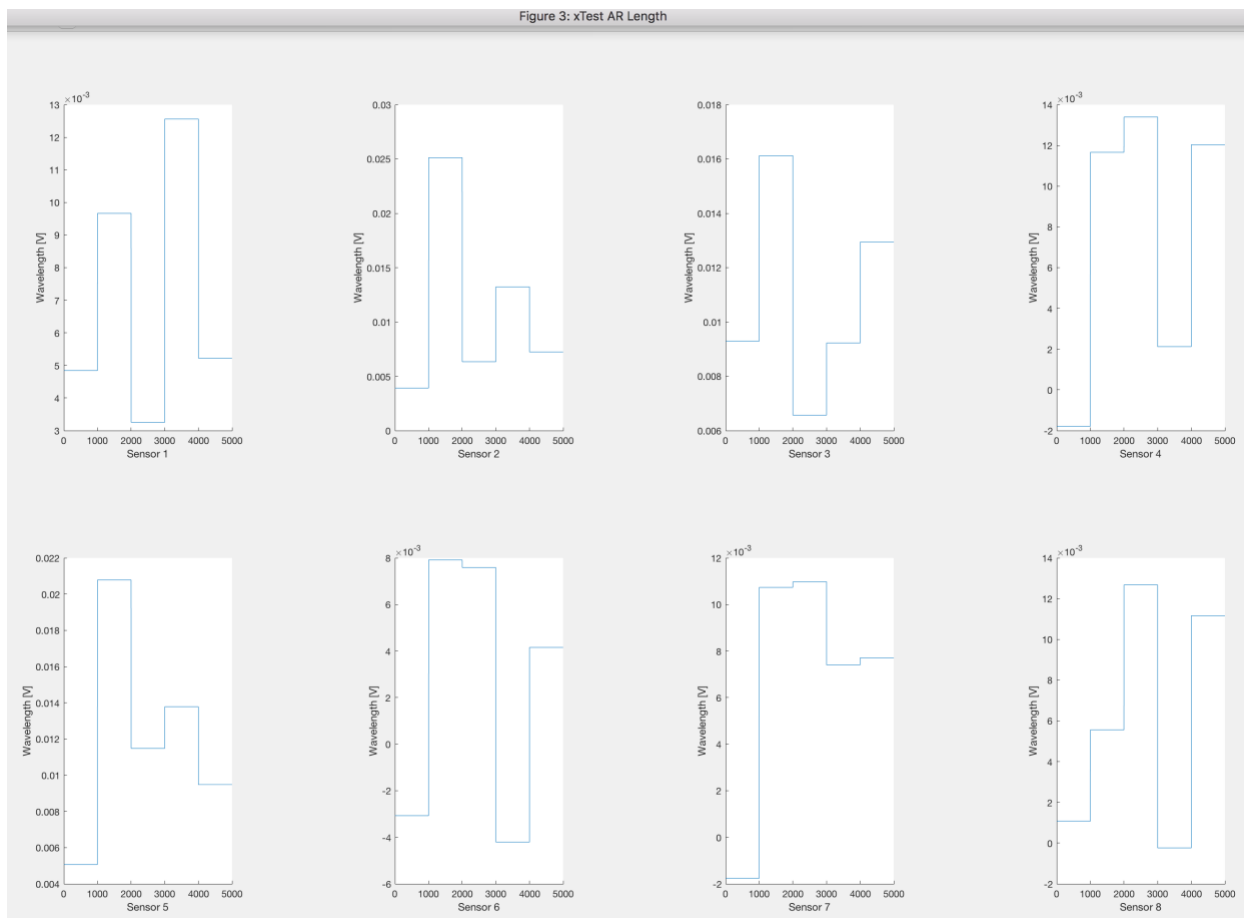


Figure 10 AR, window size 1000

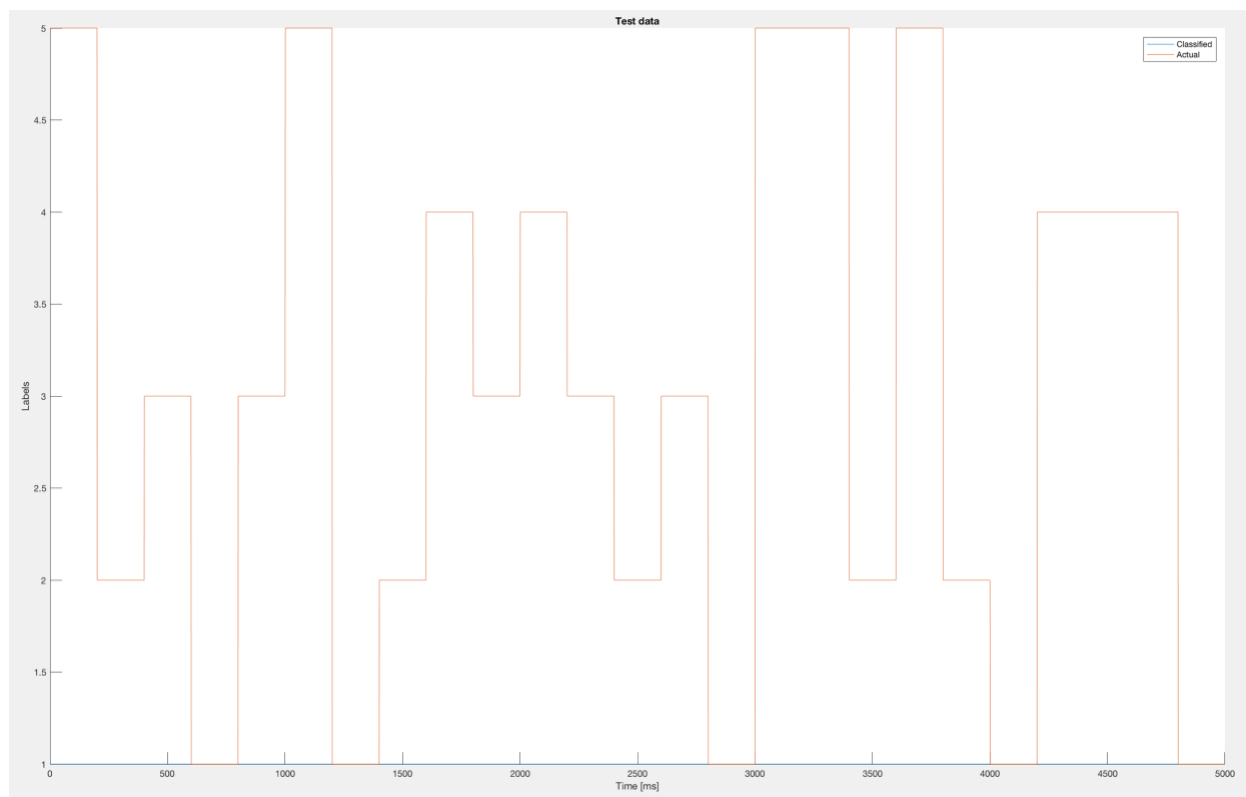


Figure 11 Classified test vs Labeled data, window size 1000

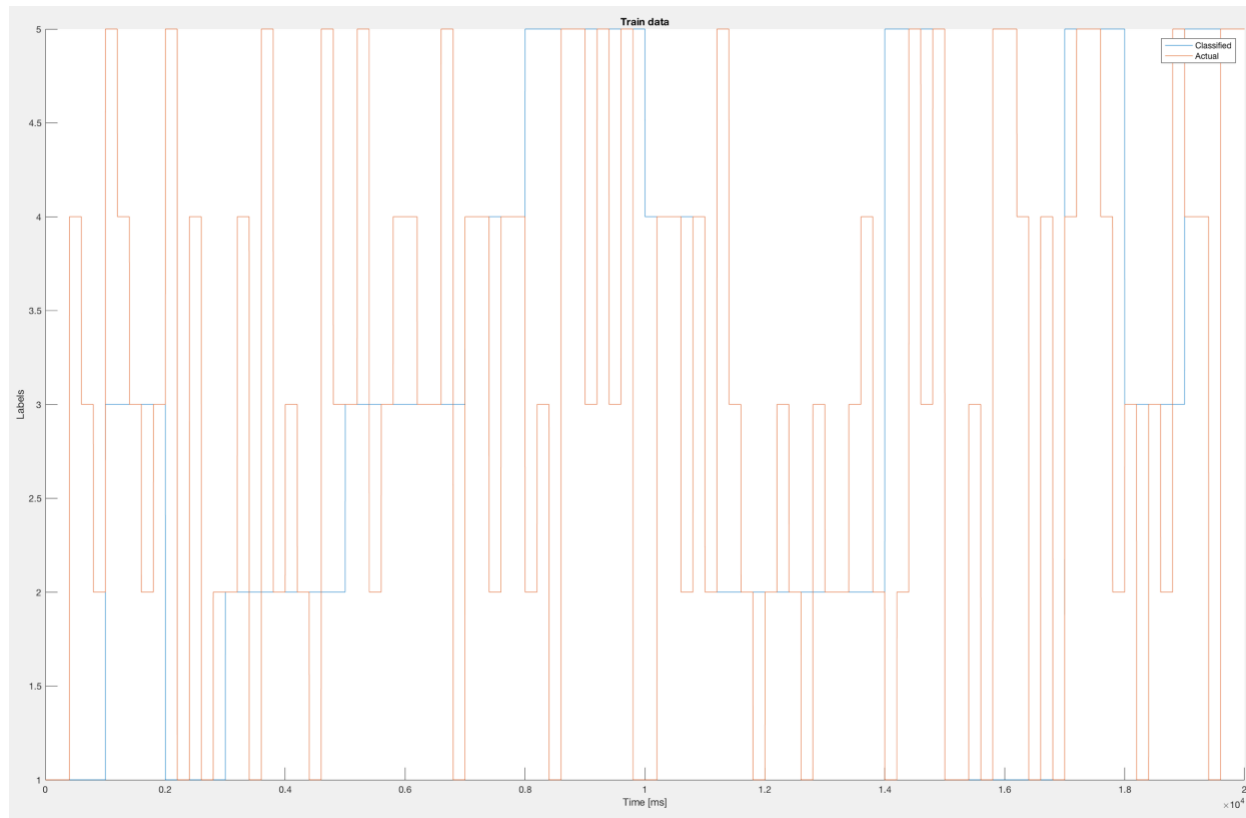


Figure 12 Classified train vs Labeled data, window size 1000

accuracy\_train =

44

accuracy\_test =

20

Figure 13 Accuracy of training and test dataset, window size 1000

Window Size = 50

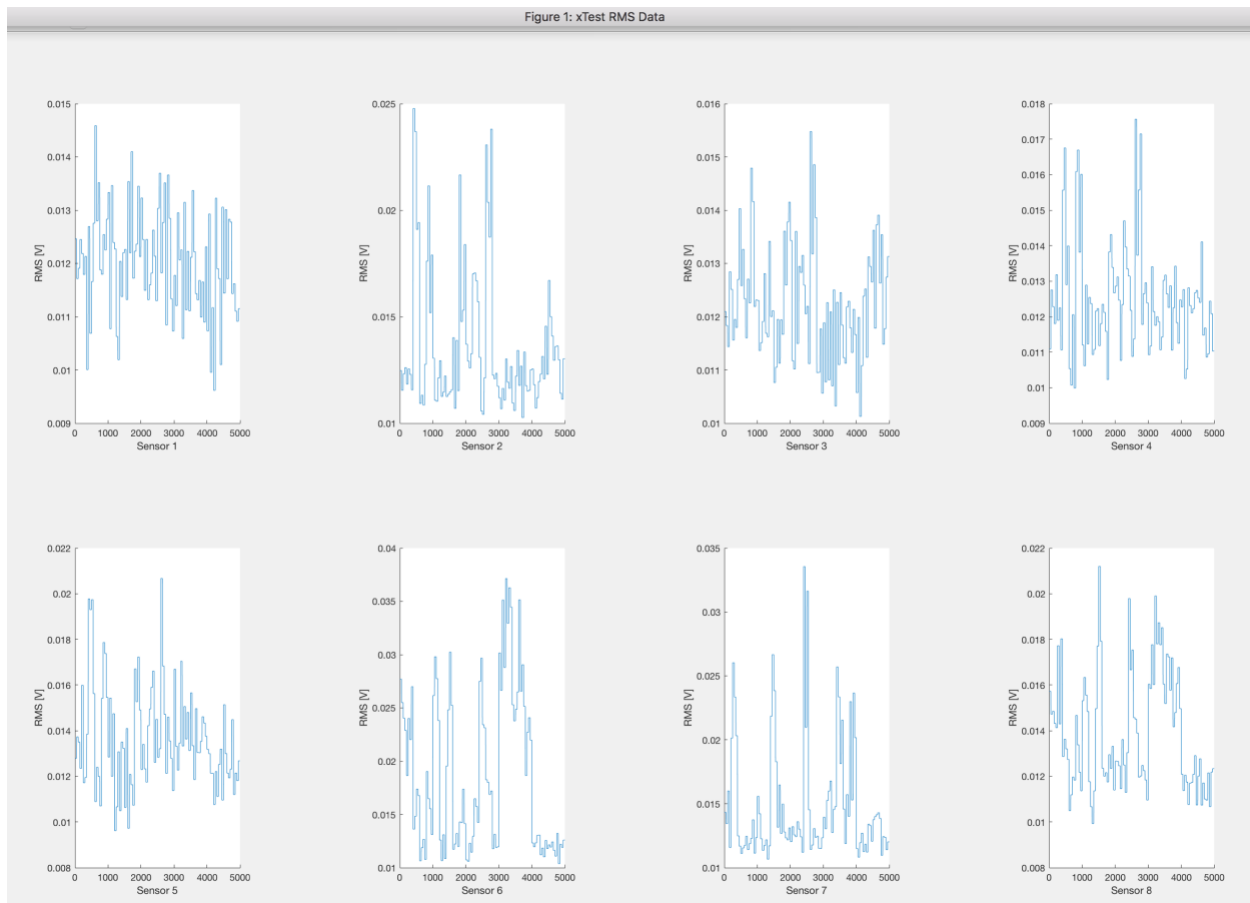
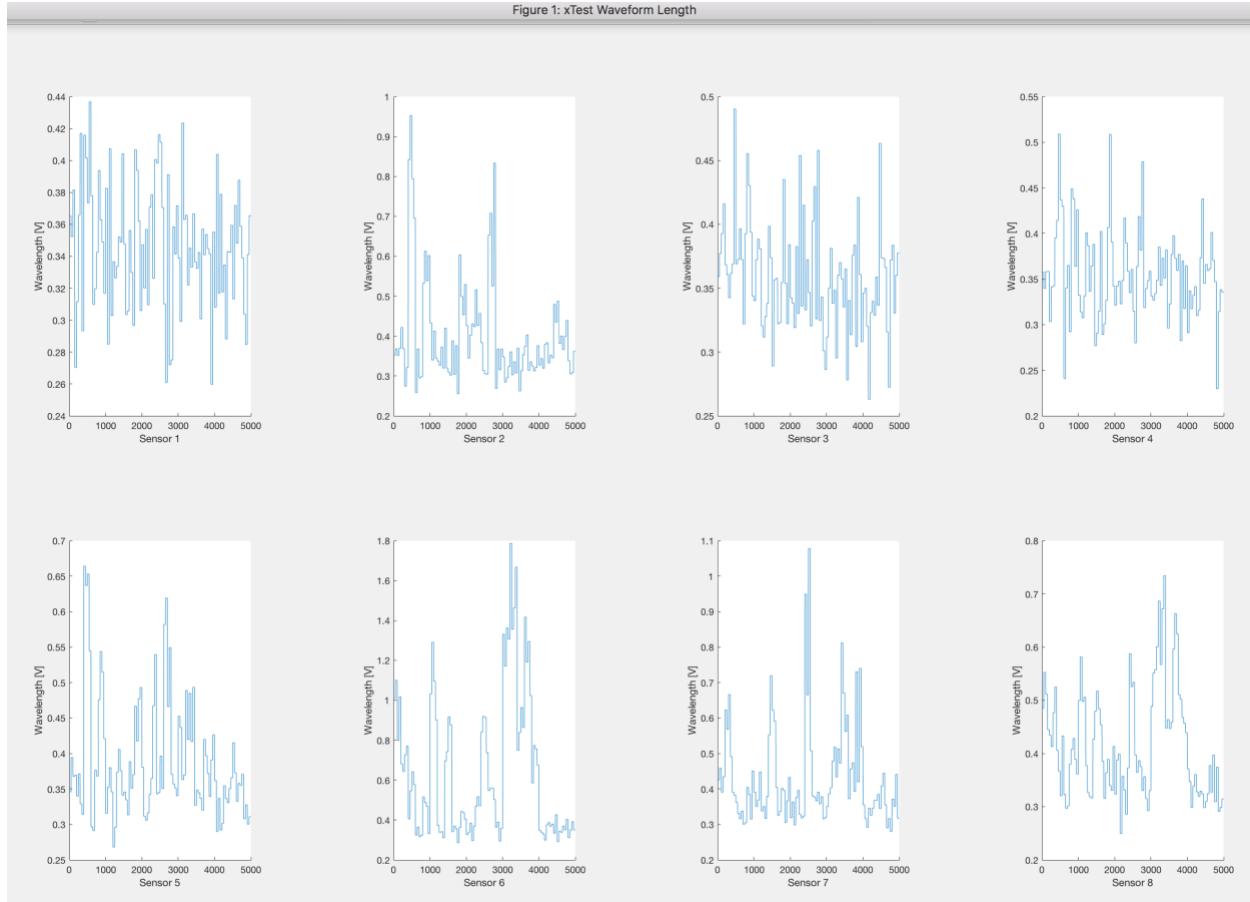


Figure 14 RMS, window size 50





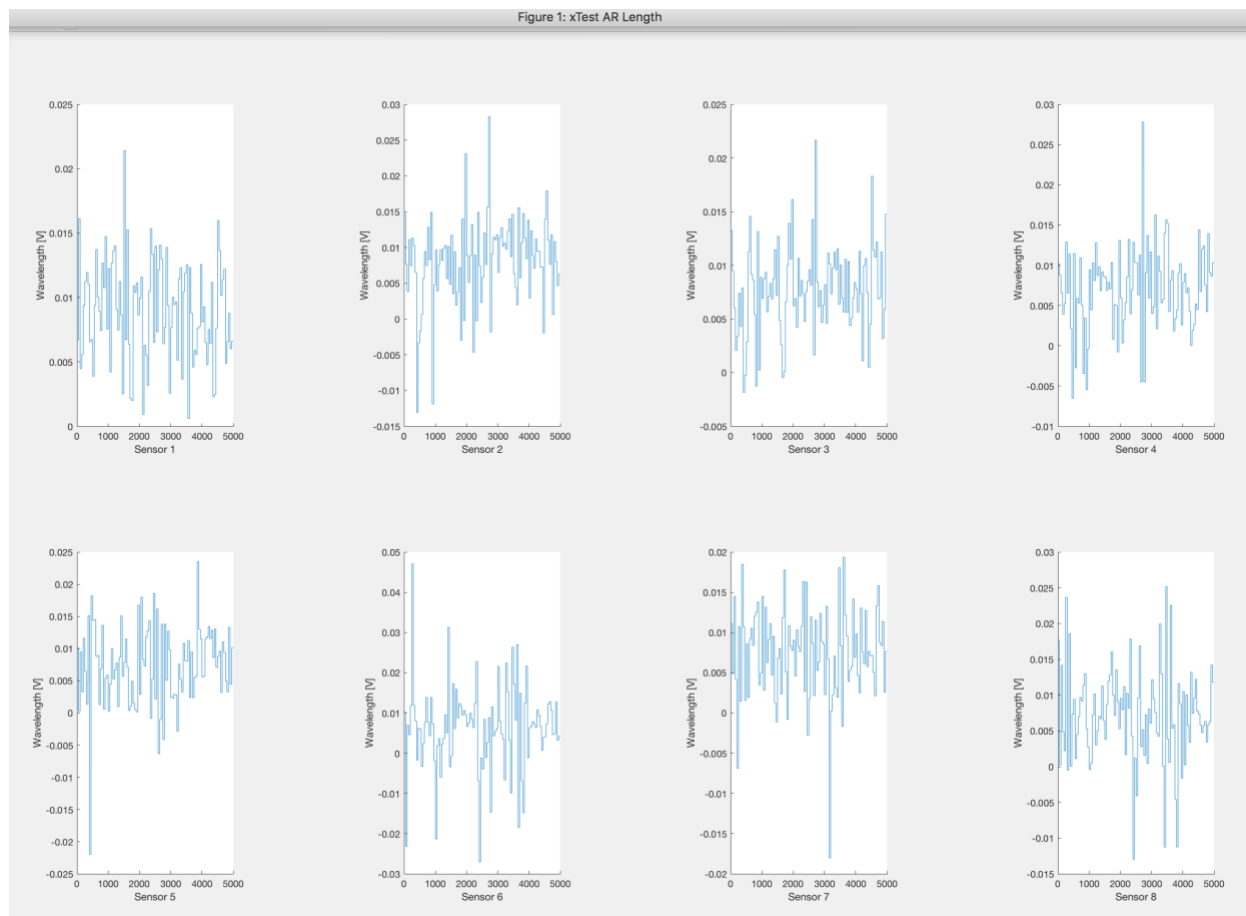


Figure 16 AR, window size 50

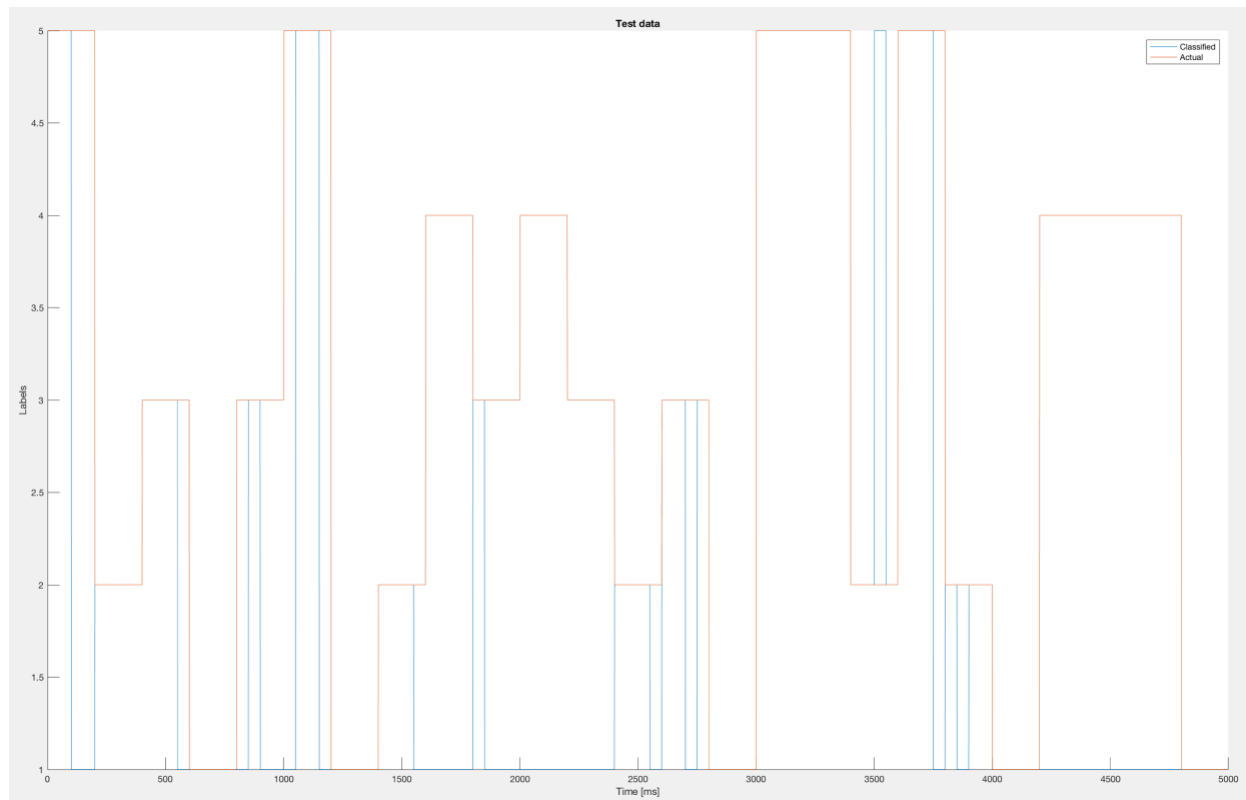


Figure 17 Classified test data vs Labeled, window size 50

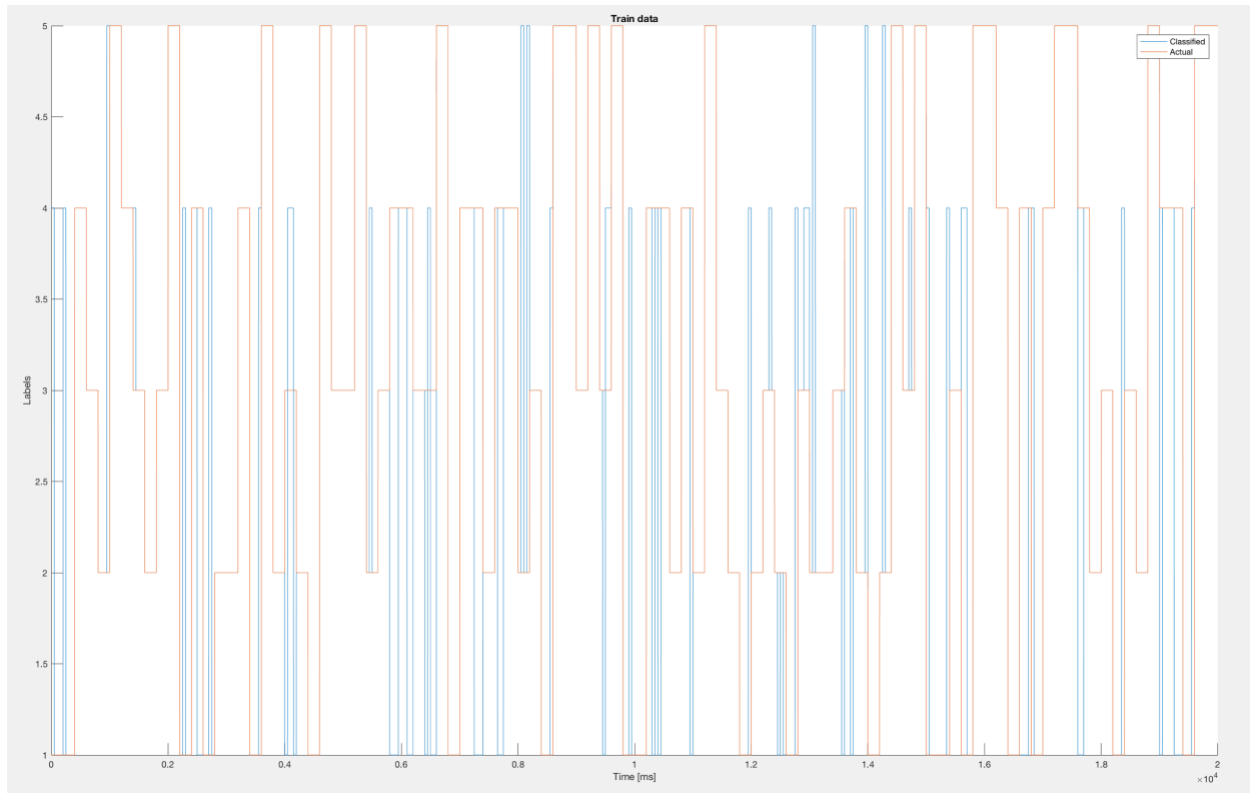


Figure 18 Classified train data vs Labeled, window 50

accuracy\_train =

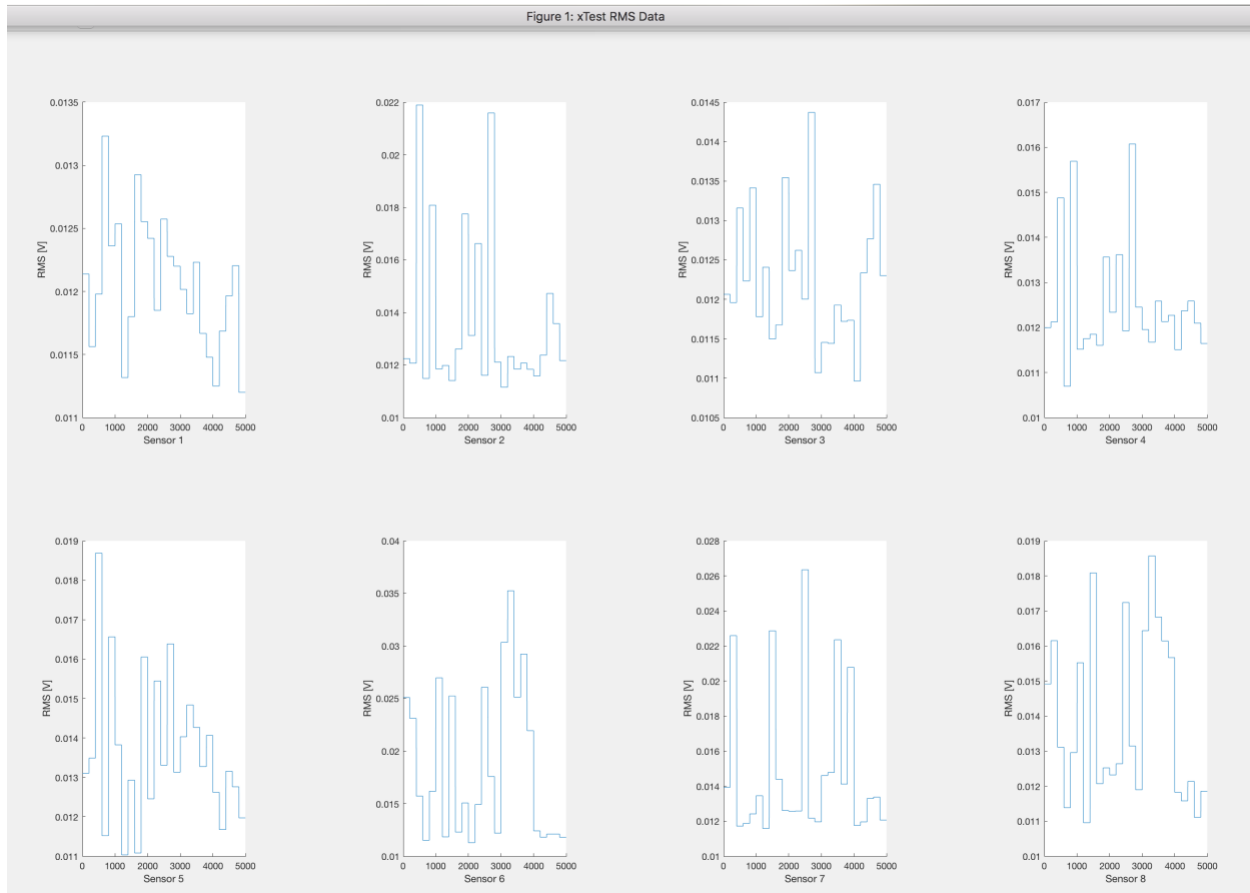
83.2500

accuracy\_test =

59

Figure 19 Accuracy of training and testing dataset, window size 50

Window size = 200



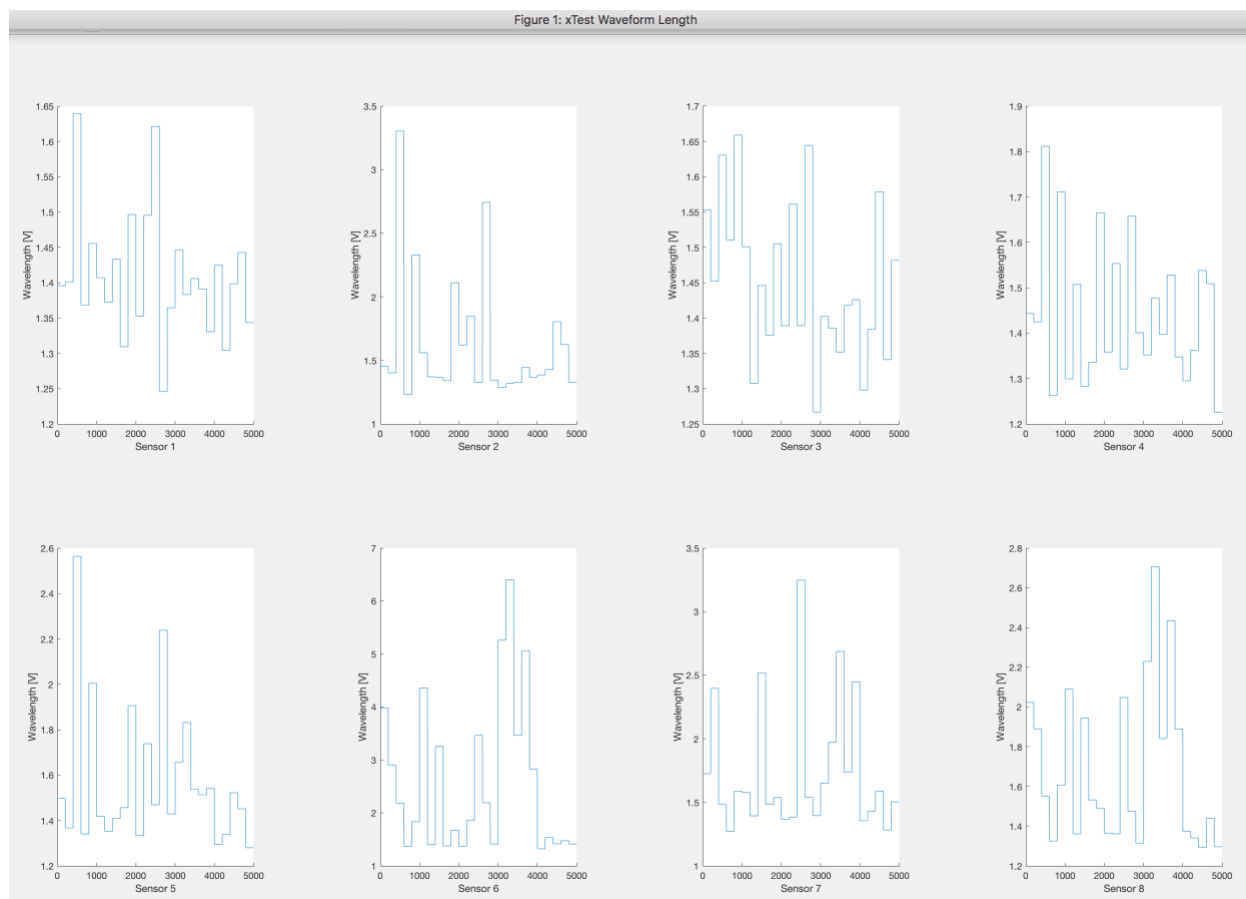


Figure 21 Waveform, window size 200

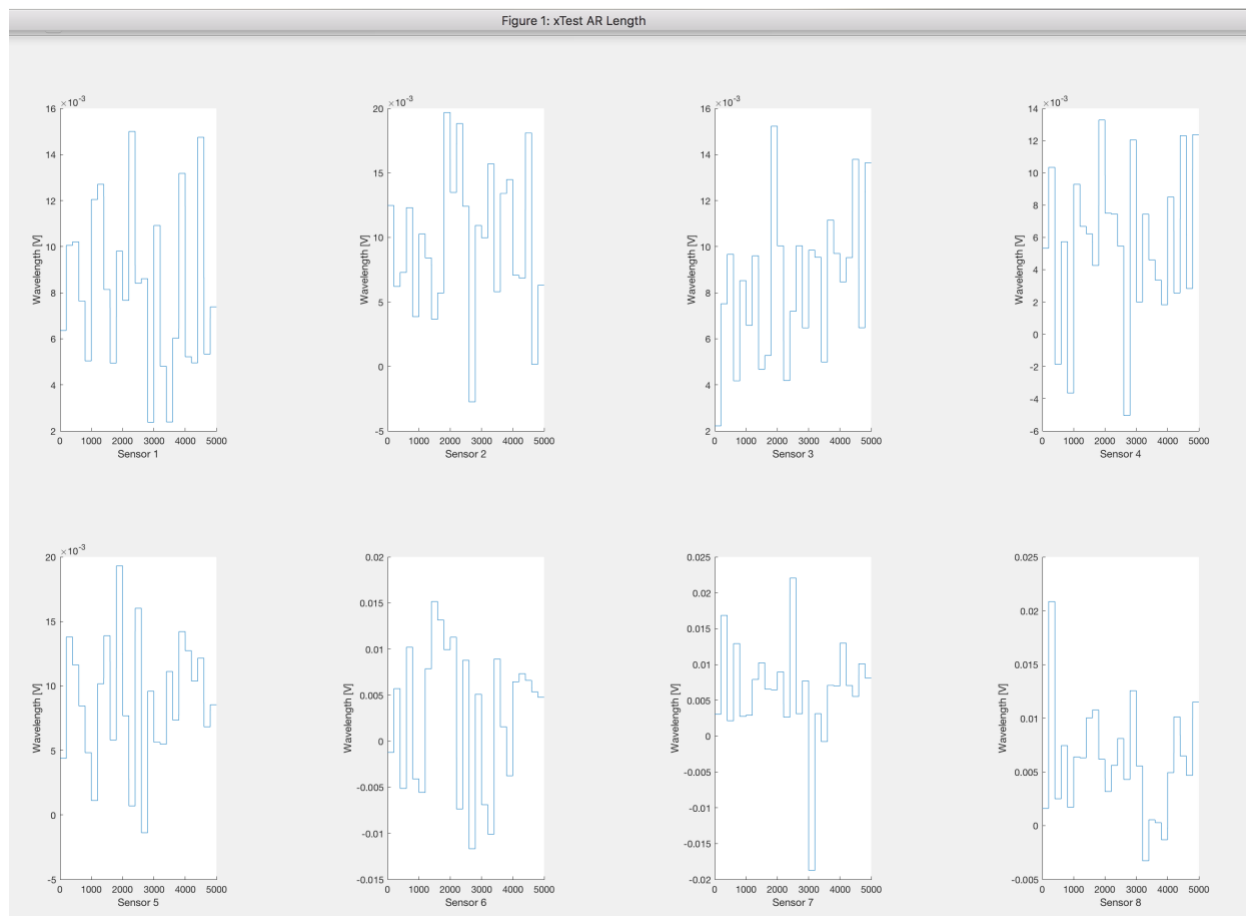


Figure 22 AR, window size 200

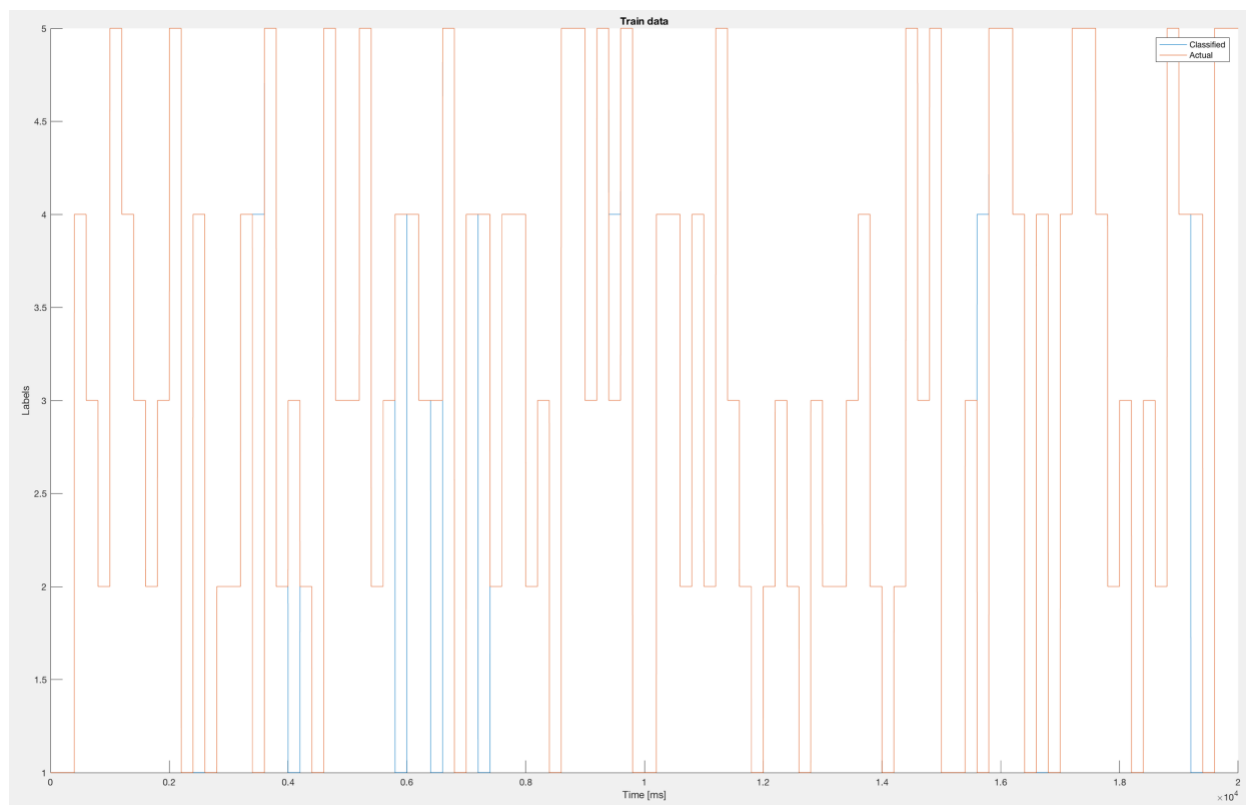


Figure 23 Classified train data vs Labeled data, window size 200

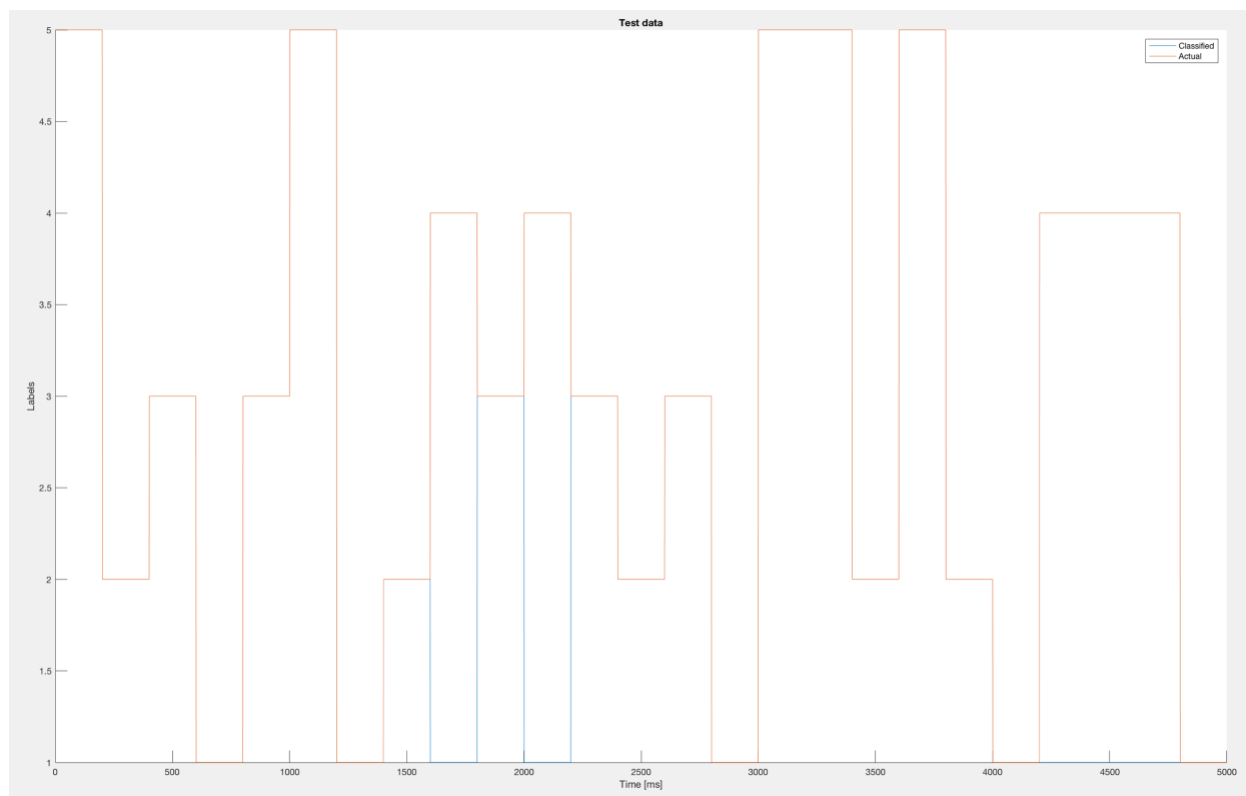


Figure 24 Classified test data vs Labeled data, window size 200



accuracy\_train =

90

accuracy\_test =

80

Figure 25 Accuracy of training and testing dataset, window set 200

```
5 %% inputs
6 - window = 200;
7 - moving_window = false;
8 - want_plots = 0;
9
10 %% loading data
11 - load('semgExcerciseRand.mat')
12
13 - xTrain = xTrain(1:10000,:);
14 - yTrain = yTrain(1:10000,:);
15
```

Command Window

accuracy\_train =

86

accuracy\_test =

84

Figure 26 Accuracy after discarding half of training dataset, window size 200