

MSE 491 - Lab 1

Using LDA to classify EMG sensor values

Harkirat Gill

301219830, hsg12@sfu.ca

Table of Contents

[Feature extraction 1](#_Toc517373633)

[RMS 1](#_Toc517373634)

[Waveform Length 1](#_Toc517373635)

[AR coefficients 2](#_Toc517373636)

[Classifier 2](#_Toc517373637)

[Results 2](#_Toc517373638)

[Accuracy 2](#_Toc517373639)

[Plots 4](#_Toc517373640)

[Raw Data 4](#_Toc517373641)

[Window Size = 100 5](#_Toc517373642)

[Window Size = 1000 9](#_Toc517373643)

[Window Size = 50 13](#_Toc517373644)

[Window size = 200 18](#_Toc517373645)

Table of Figures

[Figure 1 Raw Values 4](#_Toc517373646)

[Figure 2 RMS, window size 100 5](#_Toc517373647)

[Figure 3 Waveform length, window size 100 6](#_Toc517373648)

[Figure 4 AR, window size 100 7](#_Toc517373649)

[Figure 5 Classified Train vs Labeled, window size 100 8](#_Toc517373650)

[Figure 6 Classified Test vs Labeled, window size 100 8](#_Toc517373651)

[Figure 7 Accuracy of training and testing dataset, window size 100 9](#_Toc517373652)

[Figure 8 RMS, window size 1000 9](#_Toc517373653)

[Figure 9 Waveform, window size 1000 10](#_Toc517373654)

[Figure 10 AR, window size 1000 11](#_Toc517373655)

[Figure 11 Classified test vs Labeled data, window size 1000 12](#_Toc517373656)

[Figure 12 Classified train vs Labeled data, window size 1000 12](#_Toc517373657)

[Figure 13 Accuracy of training and test dataset, window size 1000 13](#_Toc517373658)

[Figure 14 RMS, window size 50 13](#_Toc517373659)

[Figure 15 Waveform, window size 50 14](#_Toc517373660)

[Figure 16 AR, window size 50 15](#_Toc517373661)

[Figure 17 Classified test data vs Labeled, window size 50 16](#_Toc517373662)

[Figure 18 Classified train data vs Labeled, window 50 17](#_Toc517373663)

[Figure 19 Accuracy of training and testing dataset, window size 50 17](#_Toc517373664)

[Figure 20 RMS, window size 200 18](#_Toc517373665)

[Figure 21 Waveform, window size 200 19](#_Toc517373666)

[Figure 22 AR, window size 200 20](#_Toc517373667)

[Figure 23 Classified train data vs Labeled data, window size 200 21](#_Toc517373668)

[Figure 24 Classified test data vs Labeled data, window size 200 21](#_Toc517373669)

[Figure 25 Accuracy of training and testing dataset, window set 200 22](#_Toc517373670)

[Figure 26 Accuracy after discarding half of training dataset, window size 200 22](#_Toc517373671)

# 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 coeffiecients. AR\_coeff is 5x8 long (1 normalized vector + 4 order x 8 sensors)

‘a’ intakes the xi+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

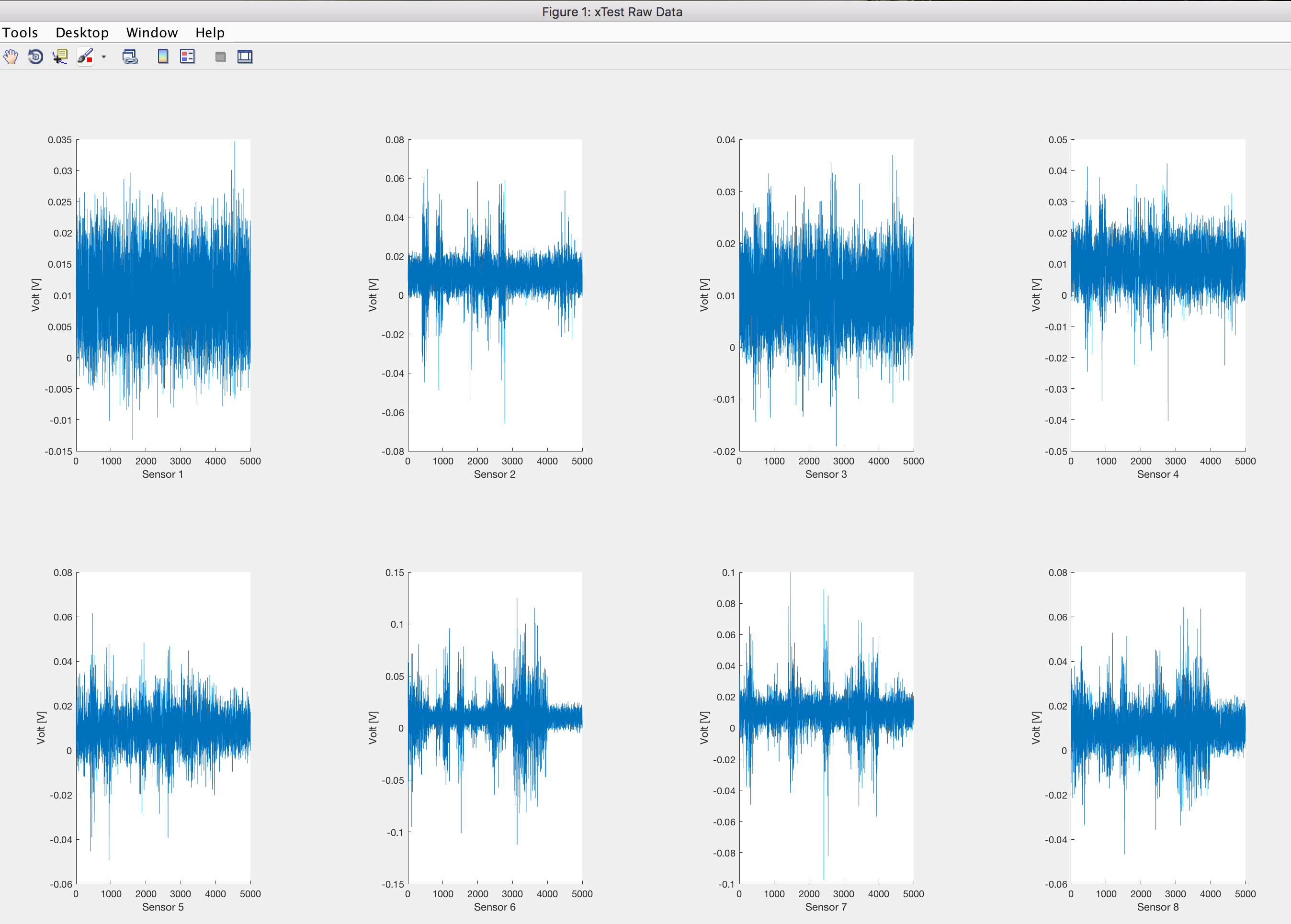


Figure 1 Raw Values

### Window Size = 100

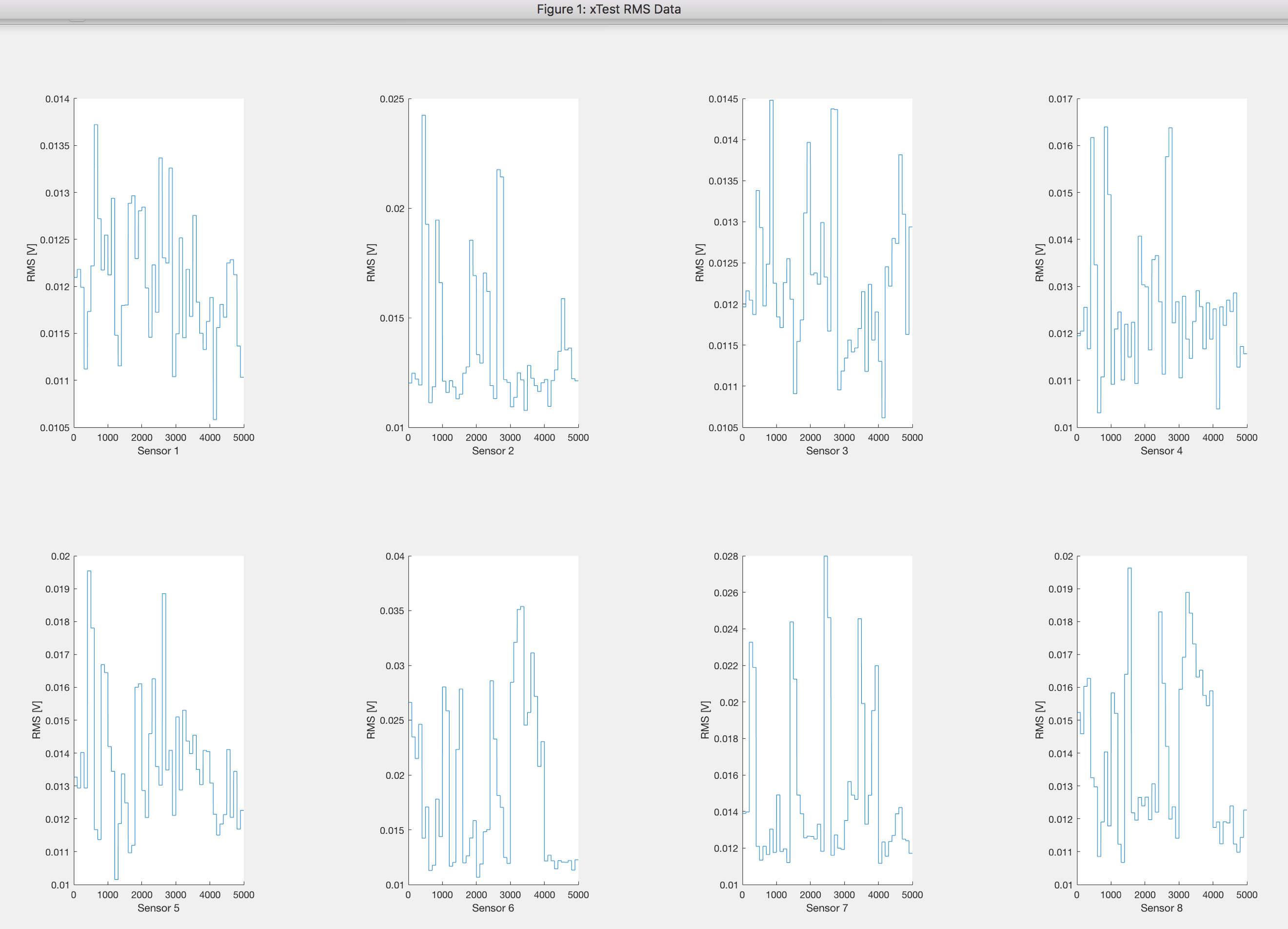


Figure 2 RMS, window size 100

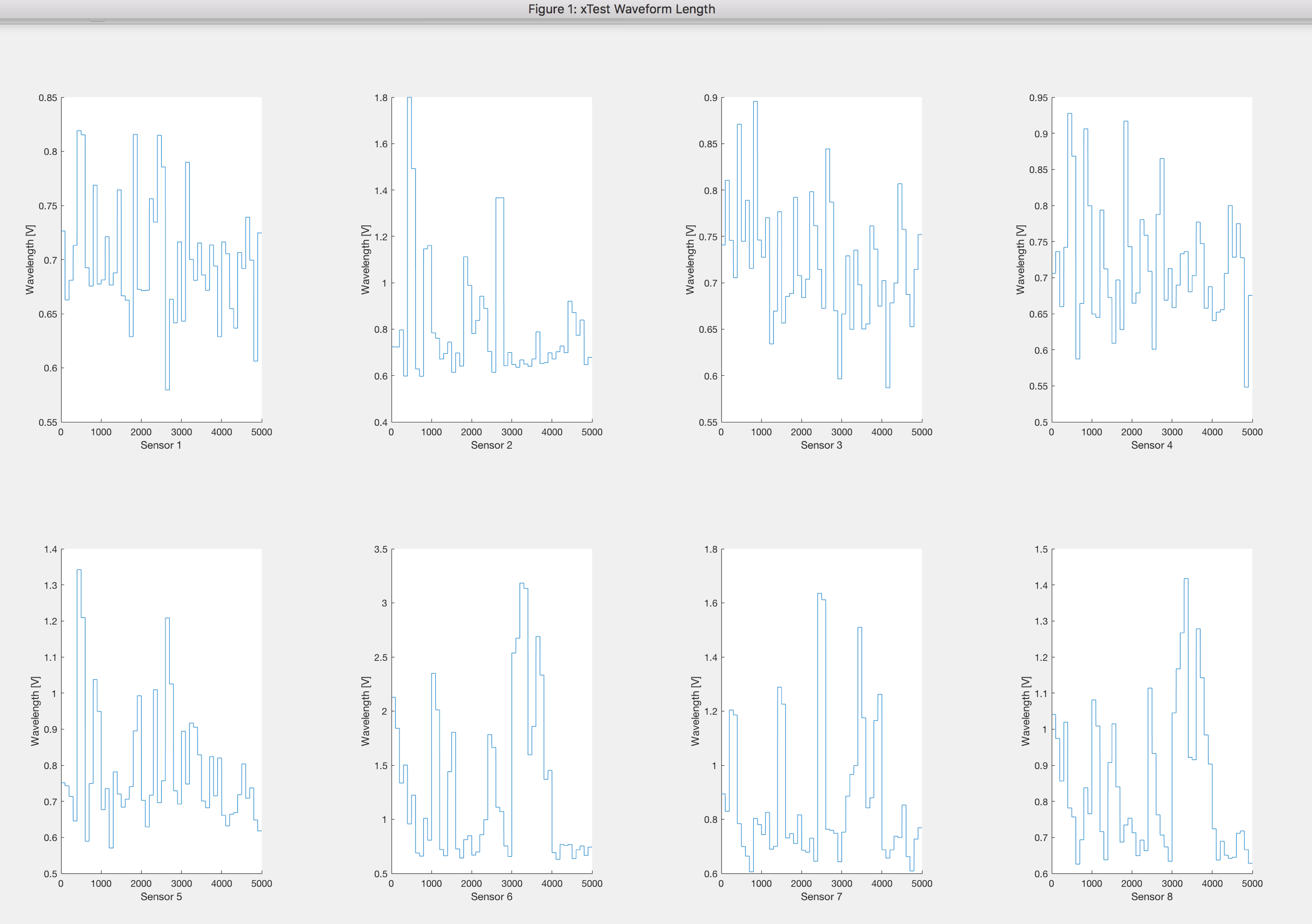


Figure 3 Waveform length, window size 100



Figure 4 AR, window size 100

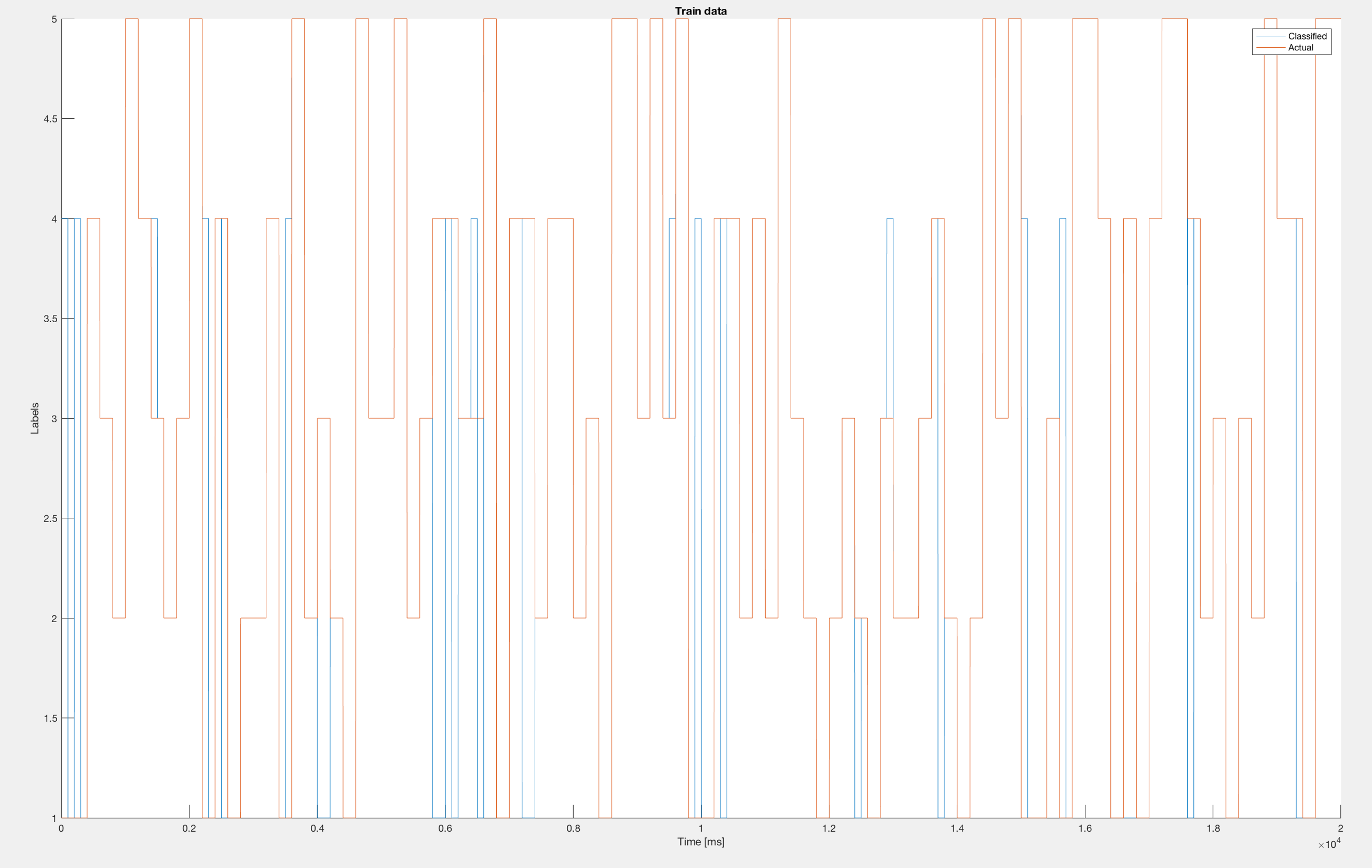


Figure 5 Classified Train vs Labeled, window size 100

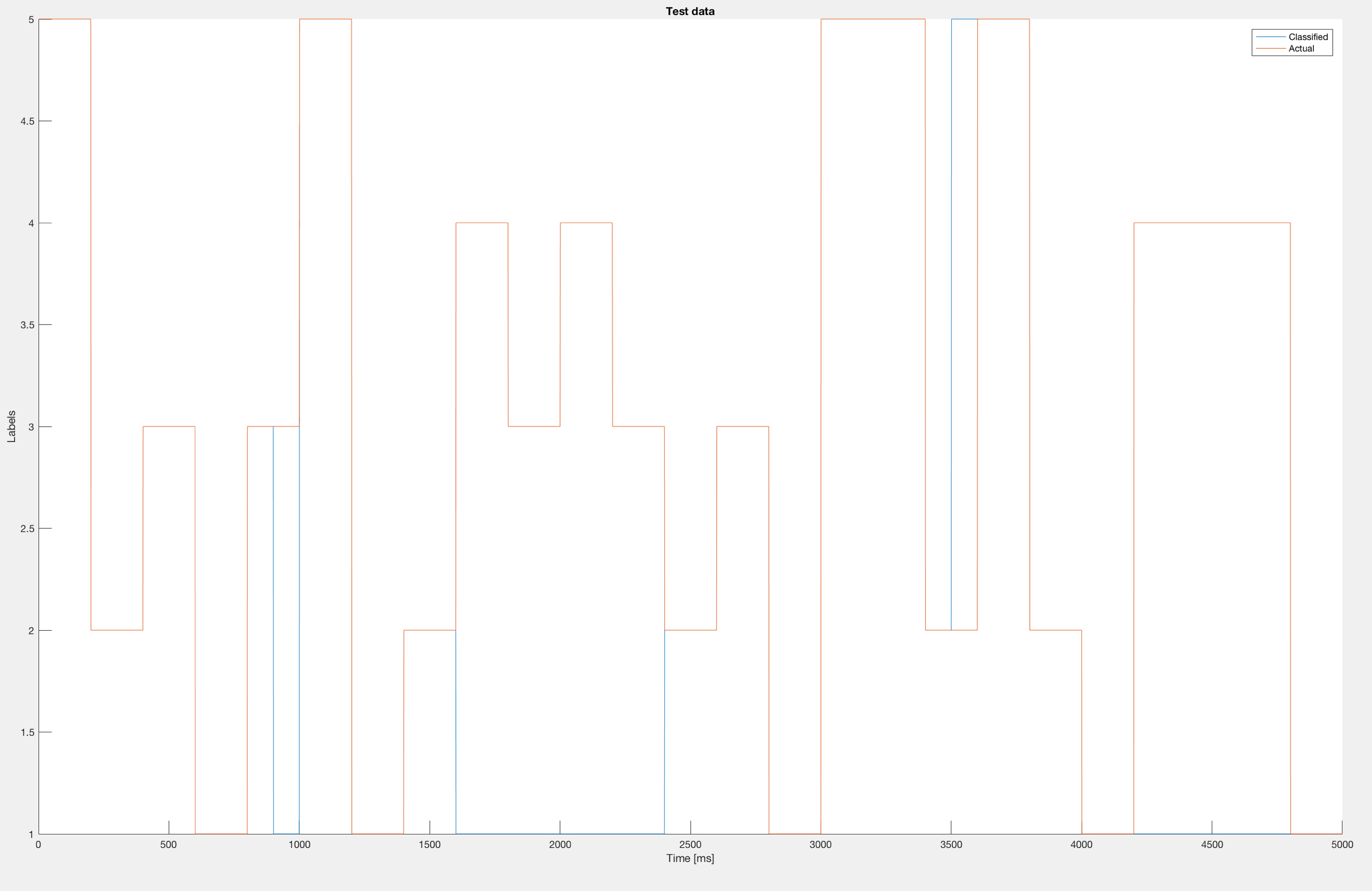


Figure 6 Classified Test vs Labeled, window size 100

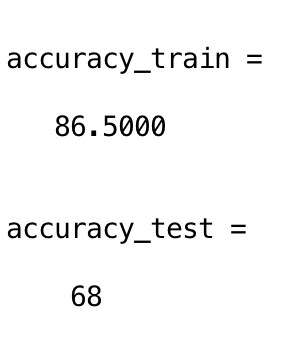


Figure 7 Accuracy of training and testing dataset, window size 100

### Window Size = 1000

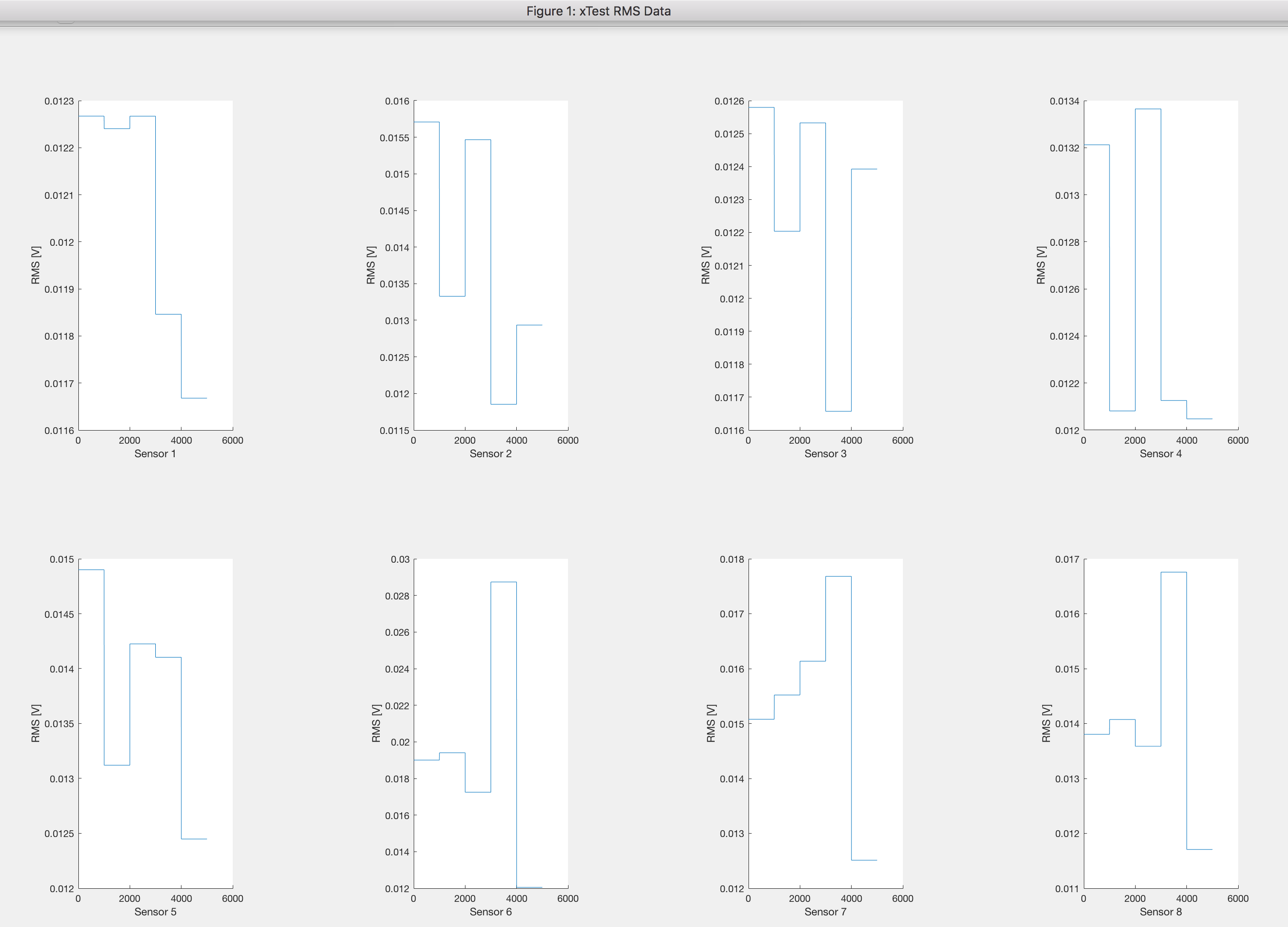


Figure 8 RMS, window size 1000

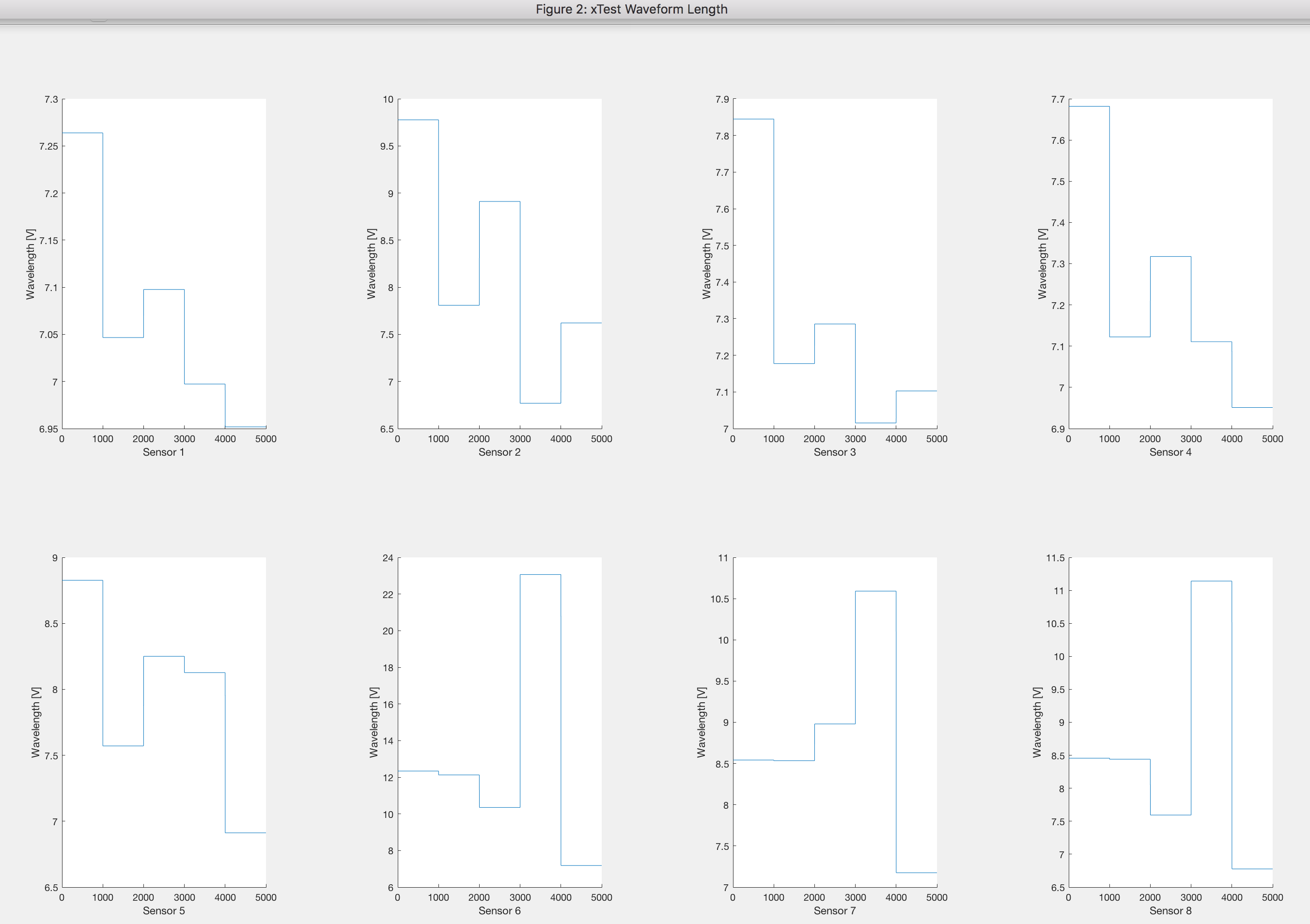


Figure 9 Waveform, window size 1000

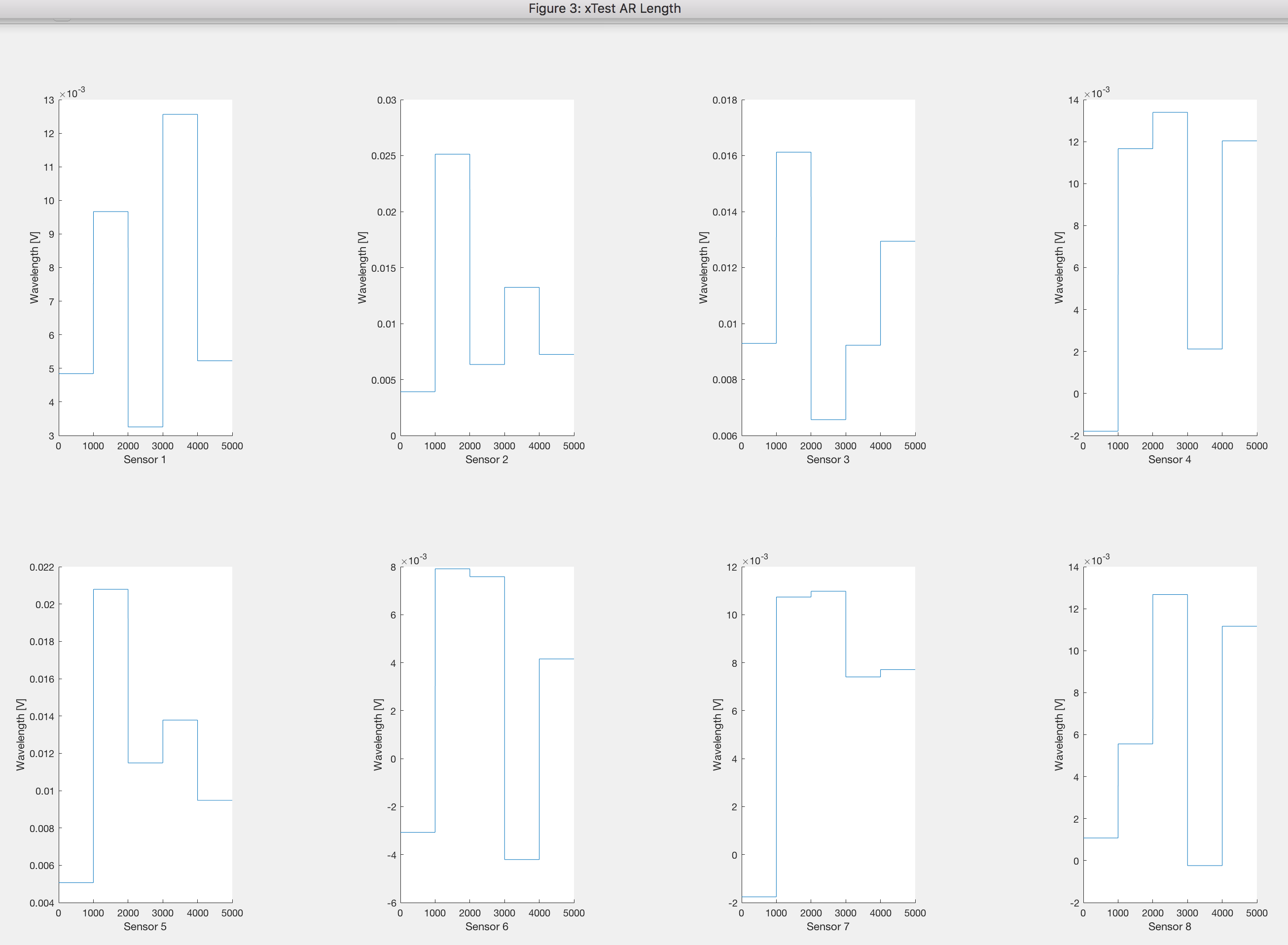


Figure 10 AR, window size 1000

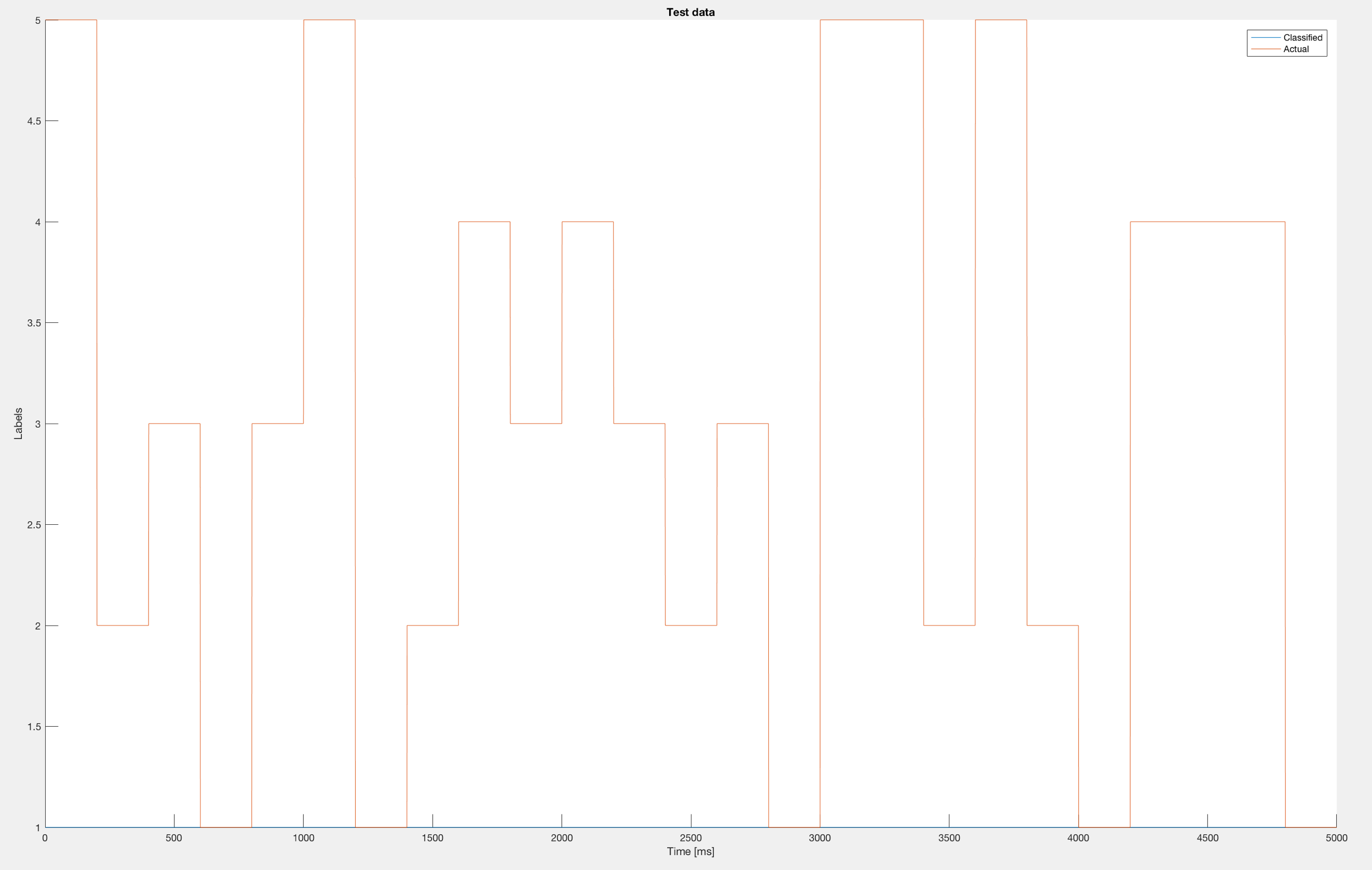


Figure 11 Classified test vs Labeled data, window size 1000

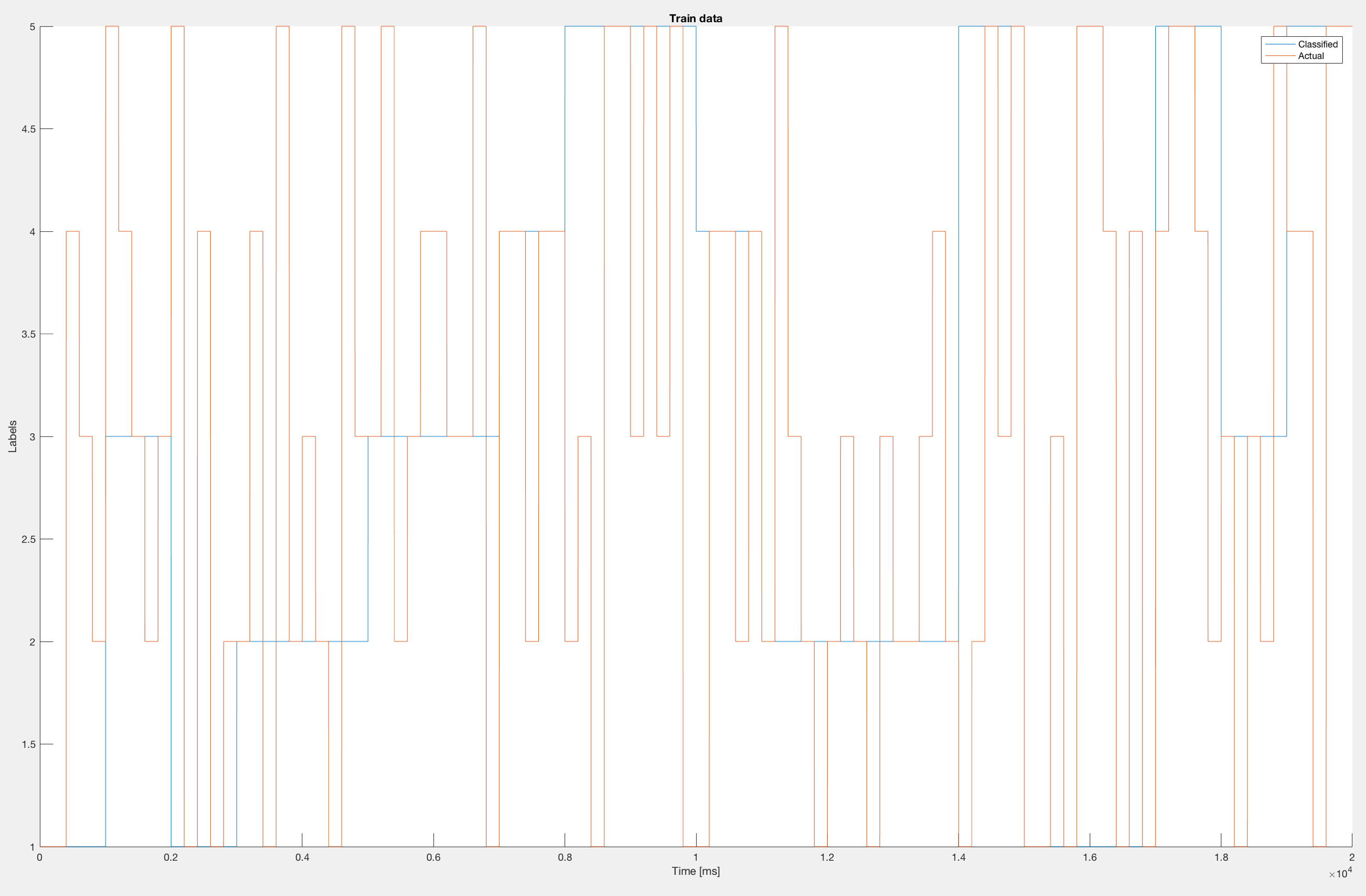


Figure 12 Classified train vs Labeled data, window size 1000

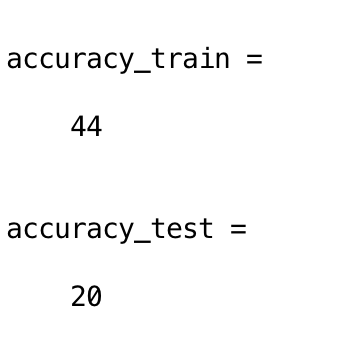


Figure 13 Accuracy of training and test dataset, window size 1000

### Window Size = 50

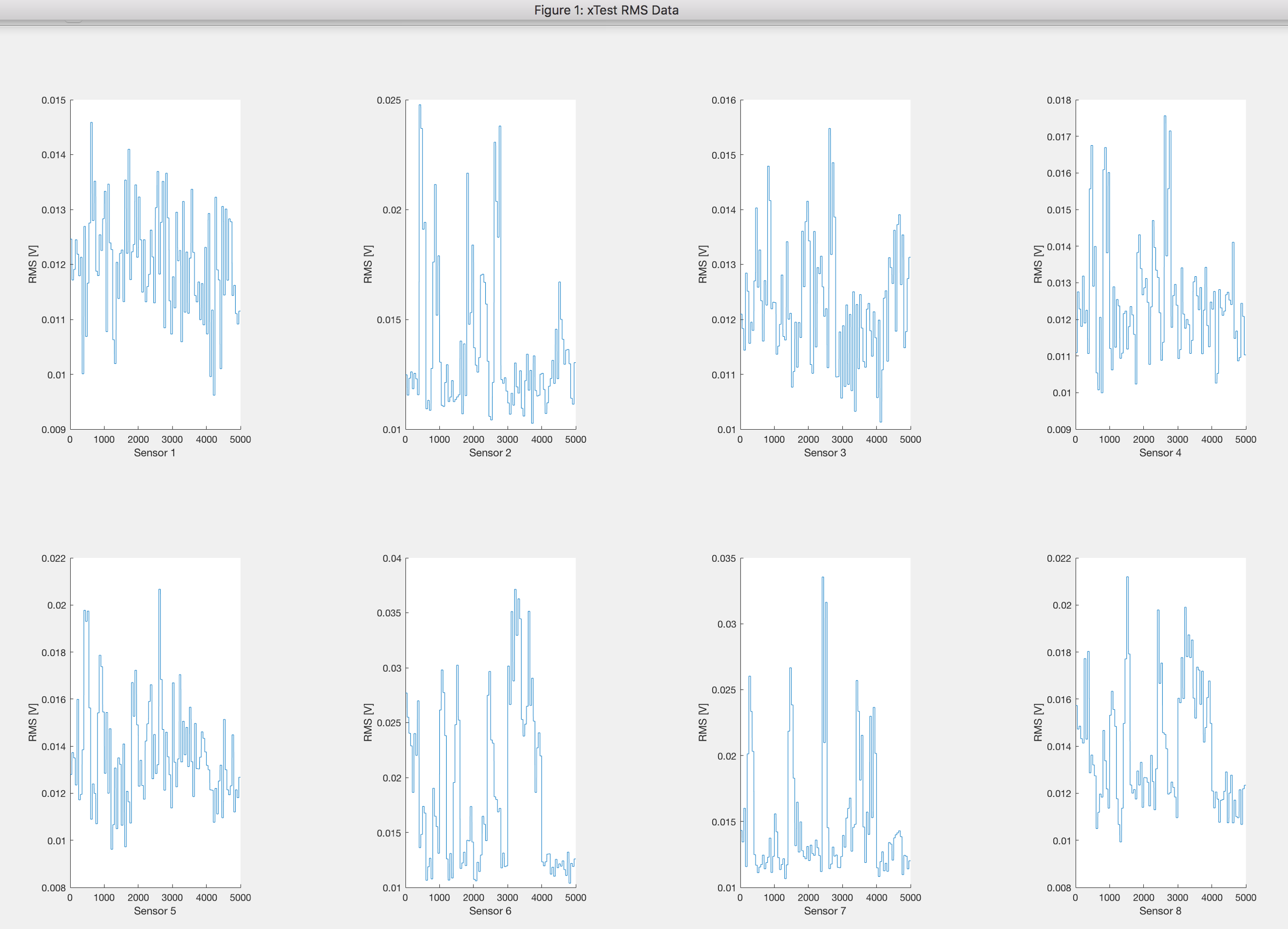


Figure 14 RMS, window size 50

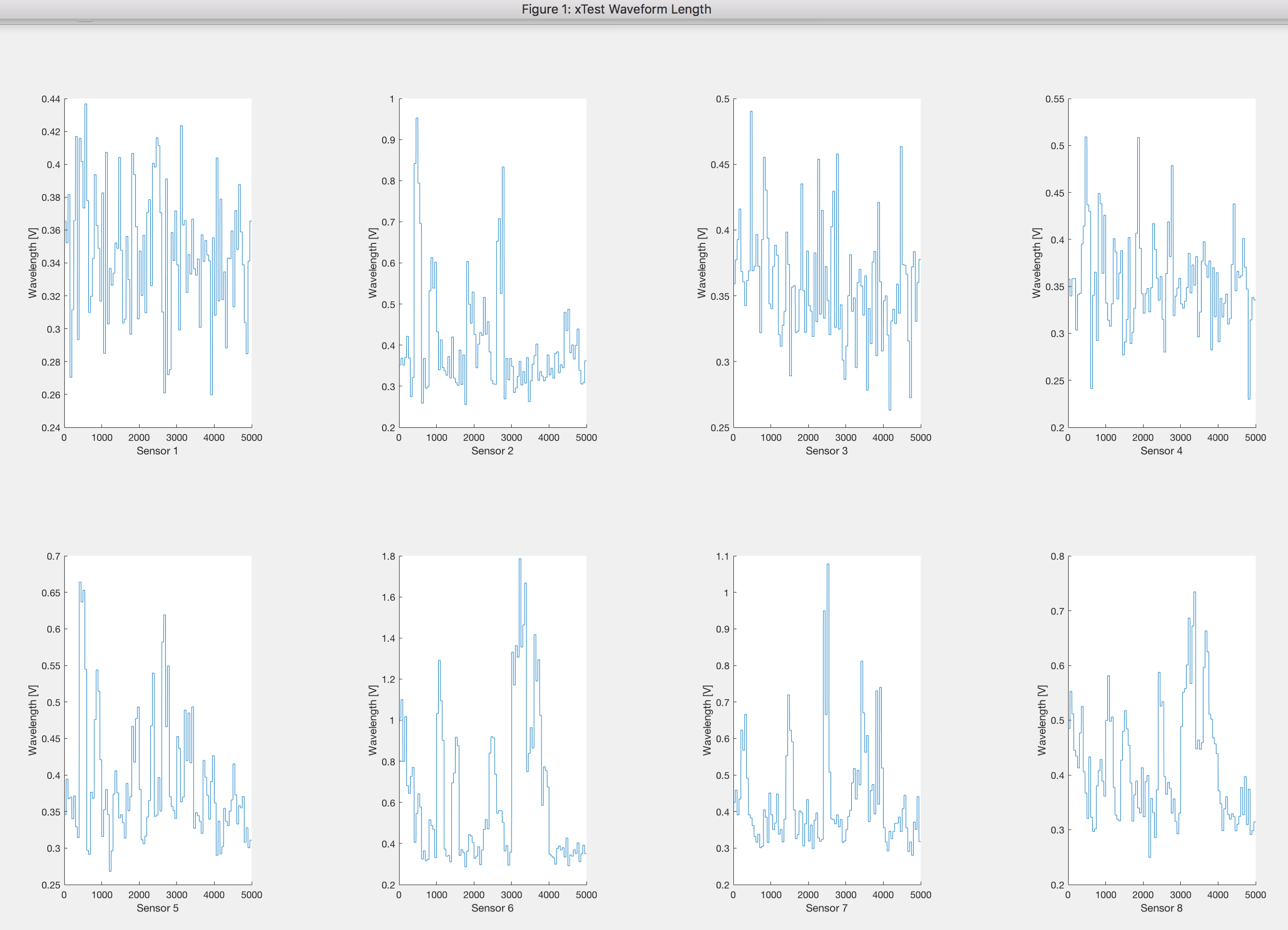


Figure 15 Waveform, window size 50

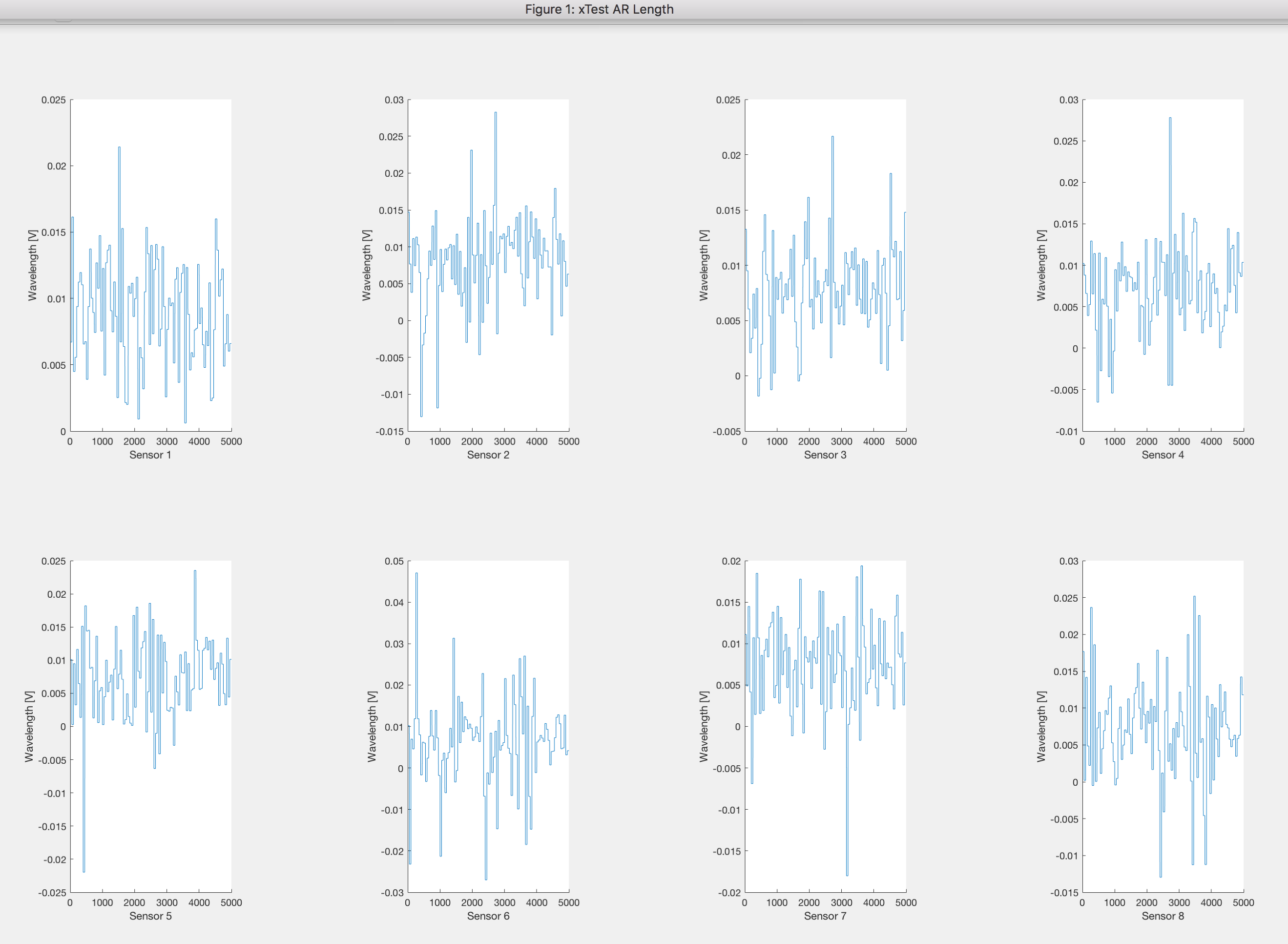


Figure 16 AR, window size 50



Figure 17 Classified test data vs Labeled, window size 50

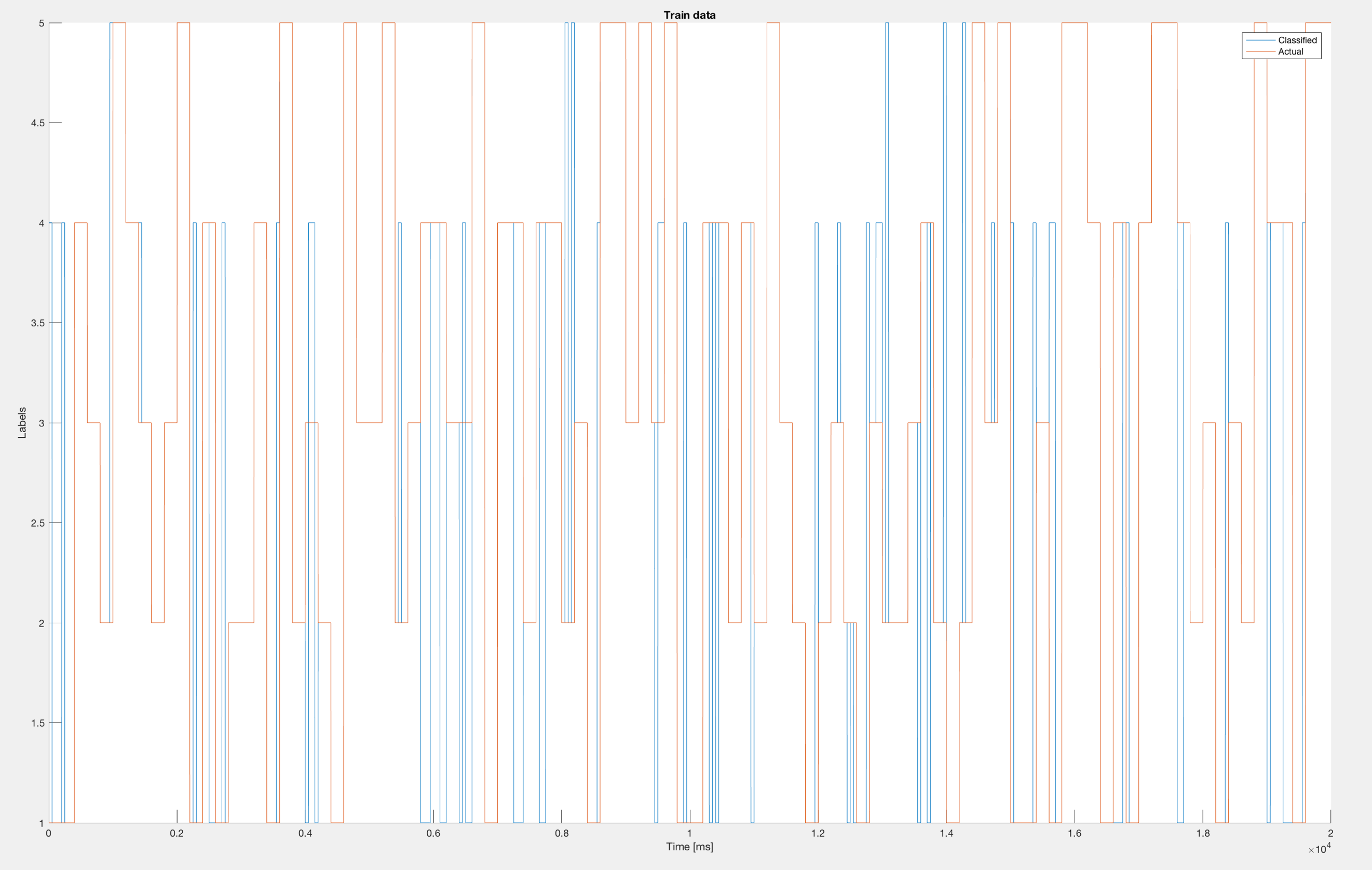


Figure 18 Classified train data vs Labeled, window 50

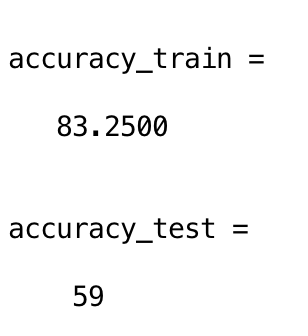


Figure 19 Accuracy of training and testing dataset, window size 50

### Window size = 200

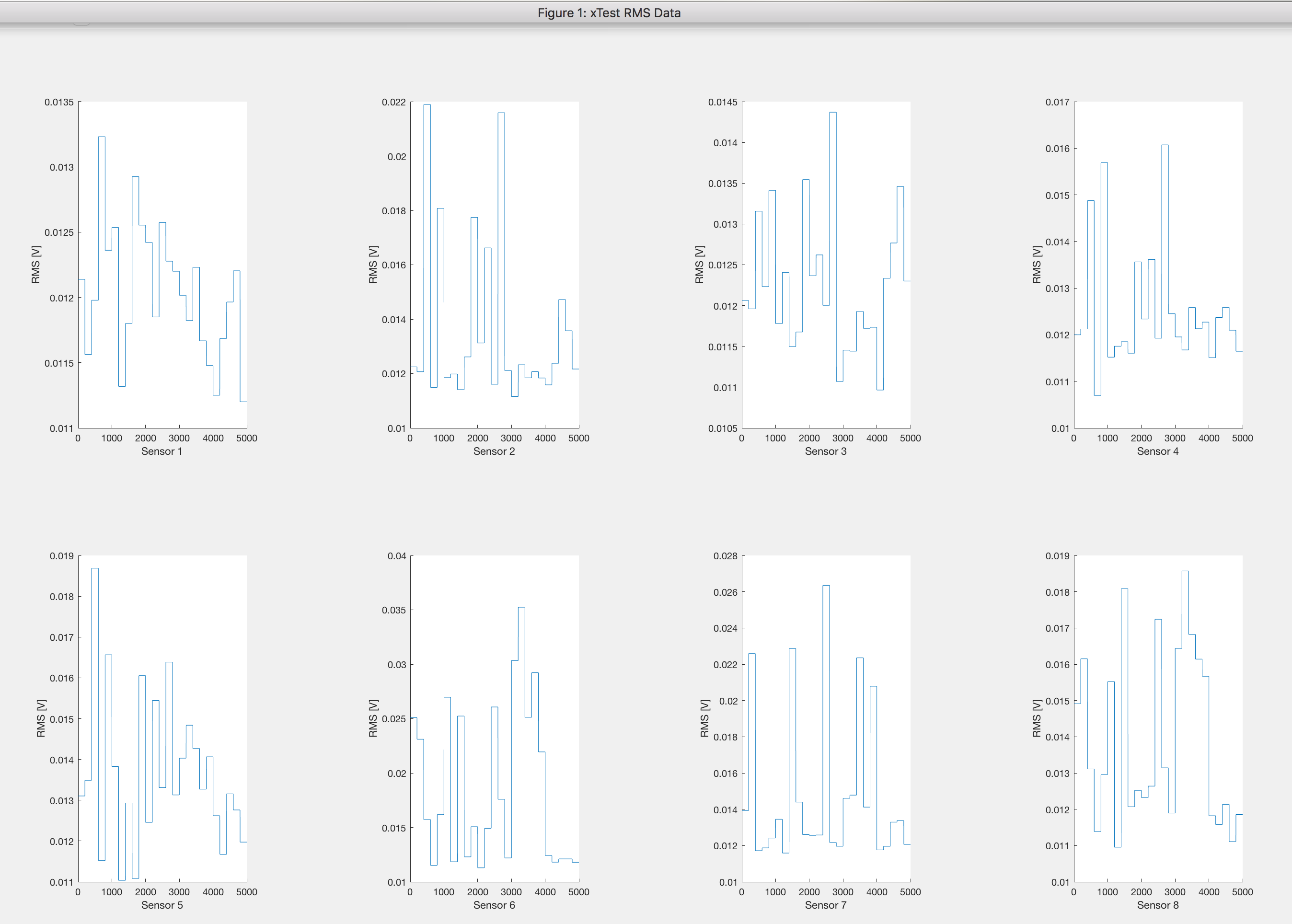


Figure 20 RMS, window size 200



Figure 21 Waveform, window size 200

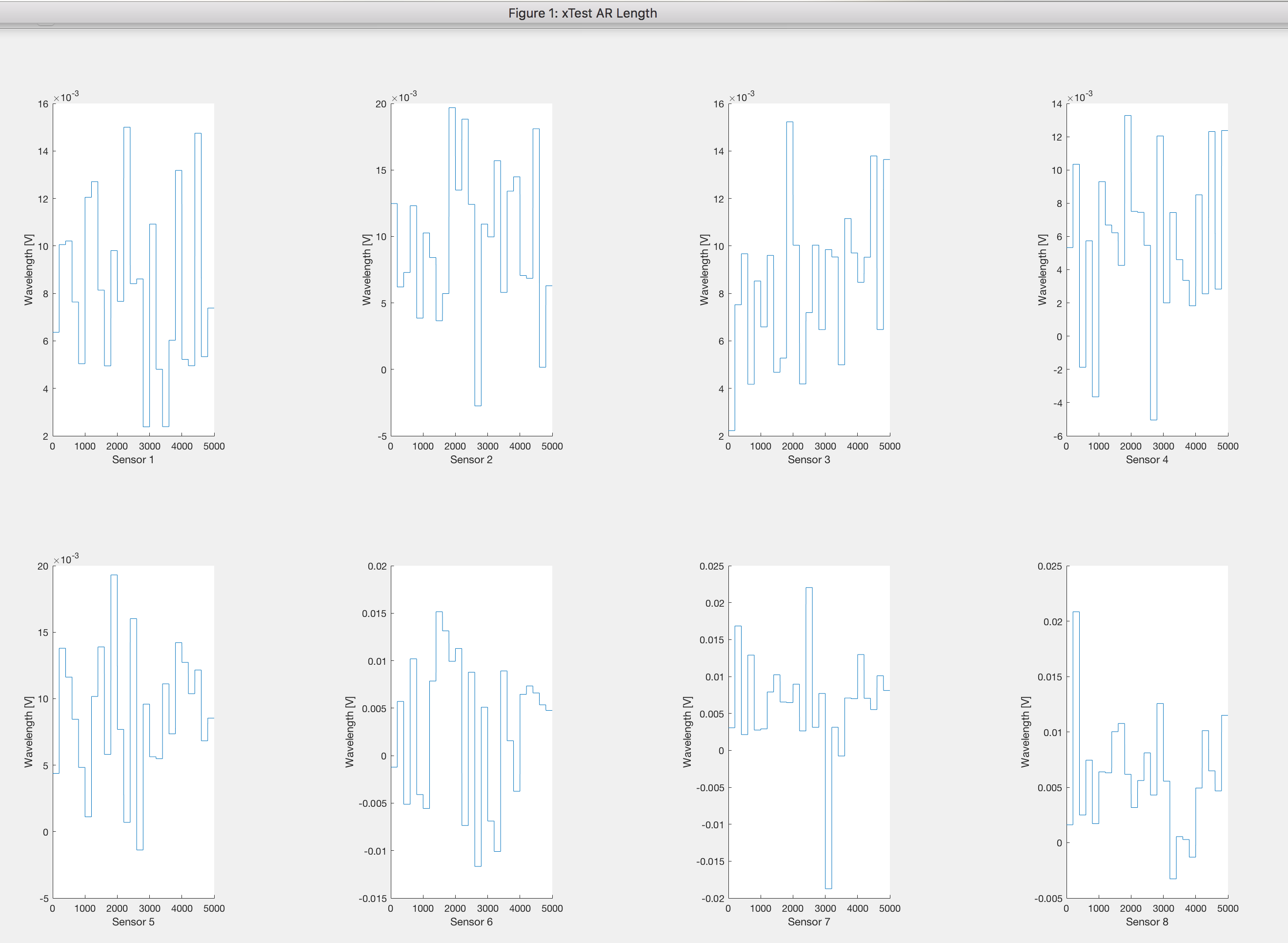


Figure 22 AR, window size 200

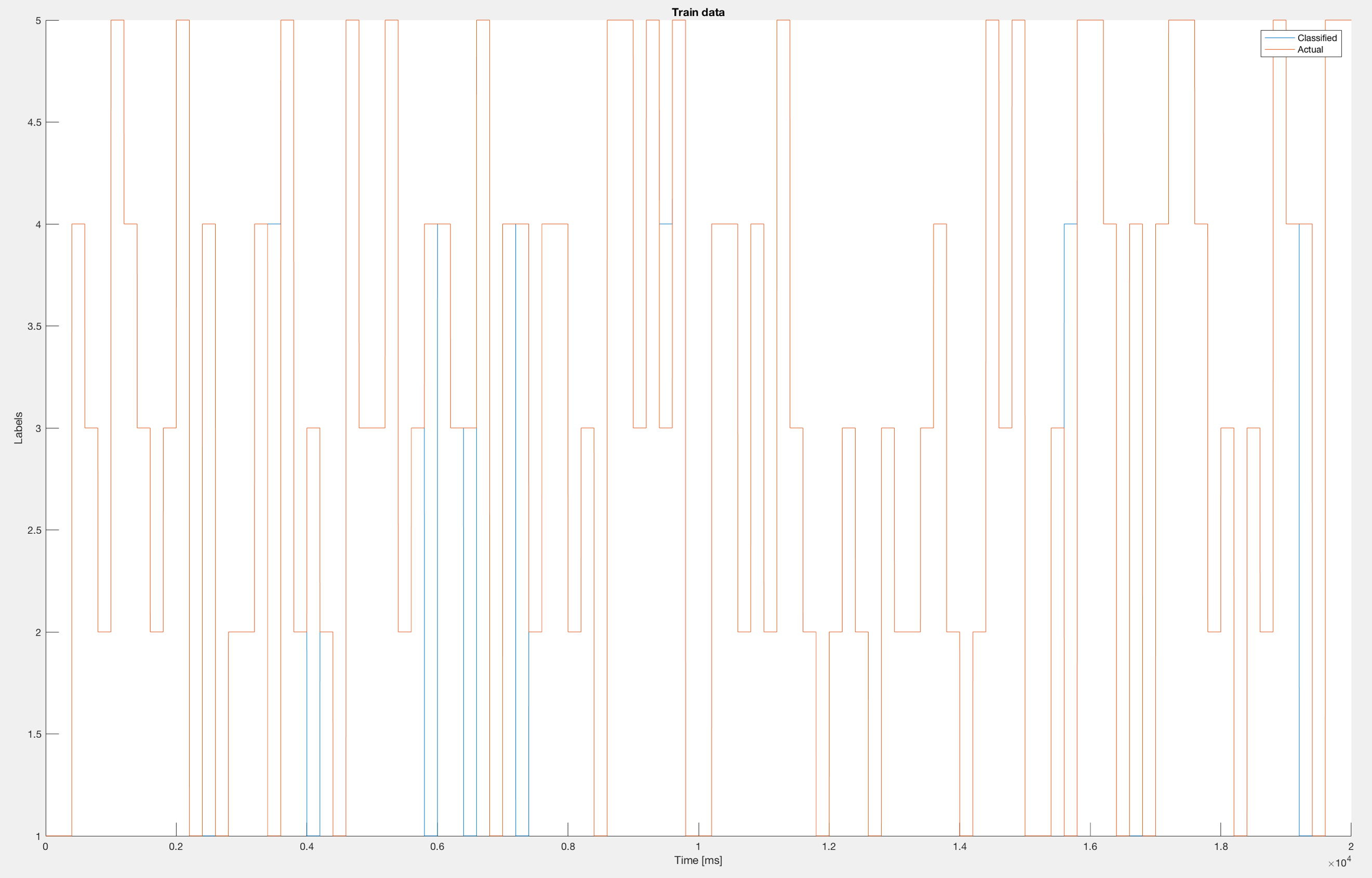


Figure 23 Classified train data vs Labeled data, window size 200

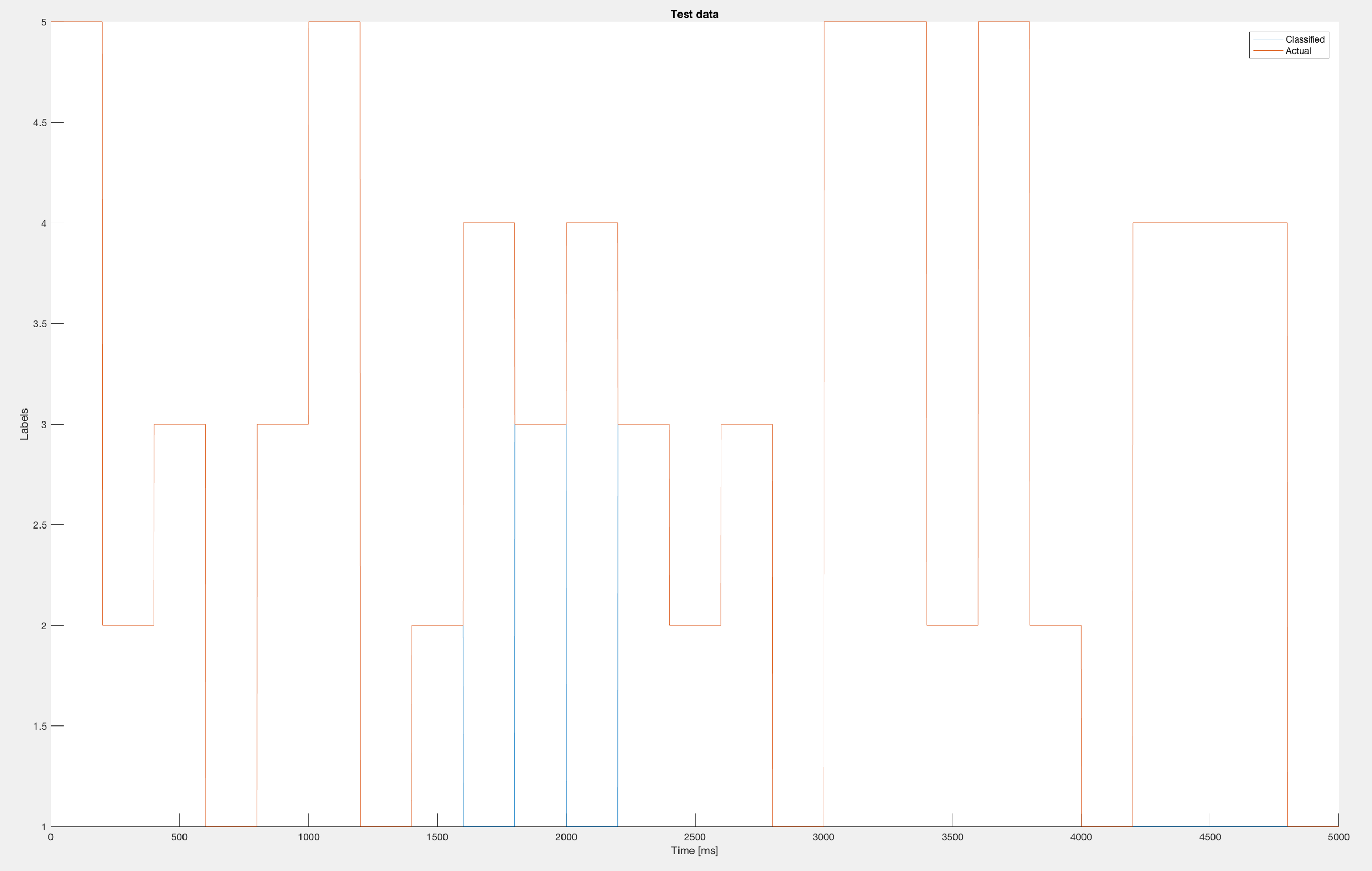


Figure 24 Classified test data vs Labeled data, window size 200

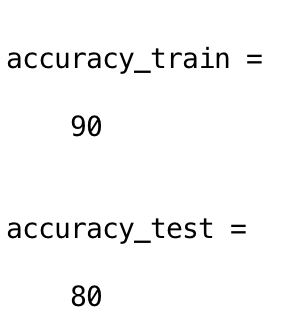


Figure 25 Accuracy of training and testing dataset, window set 200



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