Abstract

The use of machine learning algorithms has become relevant in the medical field. A comparison between two different neural networks has been evaluated for classifying hand gestures EMG signals in order to build an application for the recognition of movements. The signals were filtered to remove noise. Then, they were rectified and normalized for the LSTM. Instead, for the implementation of the DNN, we manually extracted features from the filtered signals. Both networks show good results, reaching a final accuracy above 90%. An analysis on robustness has been done, but more investigations are necessary to improve the performances.

Introduction

Machine learning algorithms are being used in the field of medicine for the purpose of diagnosis and monitoring human activities that involve muscles. In addition, sEMG signals have a significant impact in engineering, ergonomics and sports. This work is focused on discussing classification accuracy of recorded time series sEMG signals using Long short term memory (LSTM) and Deep Neural Network (DNN).

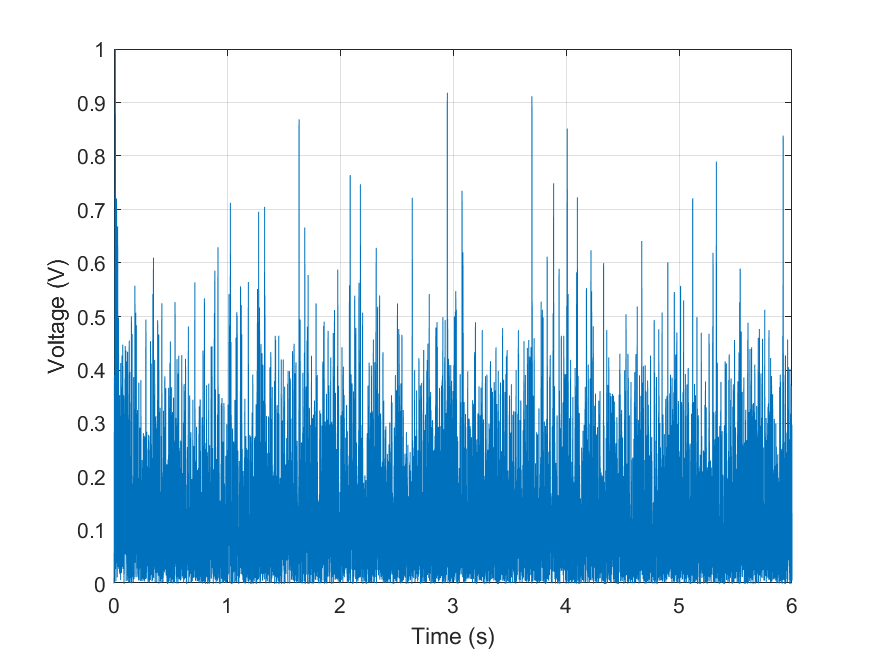
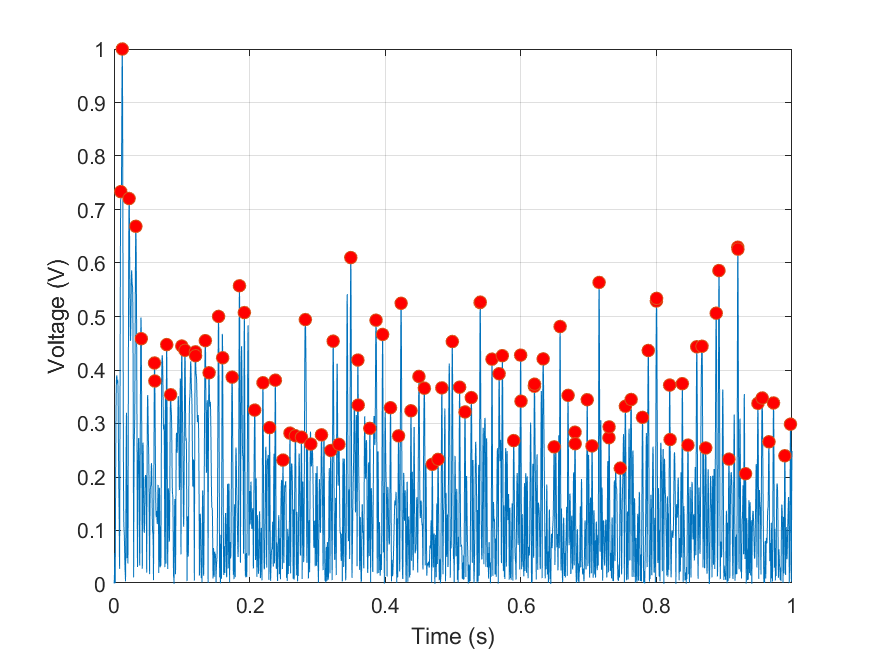
DNN features

The movements were divided into segments using a sliding time window.  
For the DNN we extracted the following time domain features from the filtered signals:

* MAV
* WAMP
* WL
* AR model coefficients
* RMS

LSTM compression

In order to compress the data, 100 values were extracted from each sensor in each segment. This was done dividing the segment into 100 intervals and taking the maximum value in each interval. We chose to use the maximum value because we thought it would be representative of the activation of the muscles during the movements.



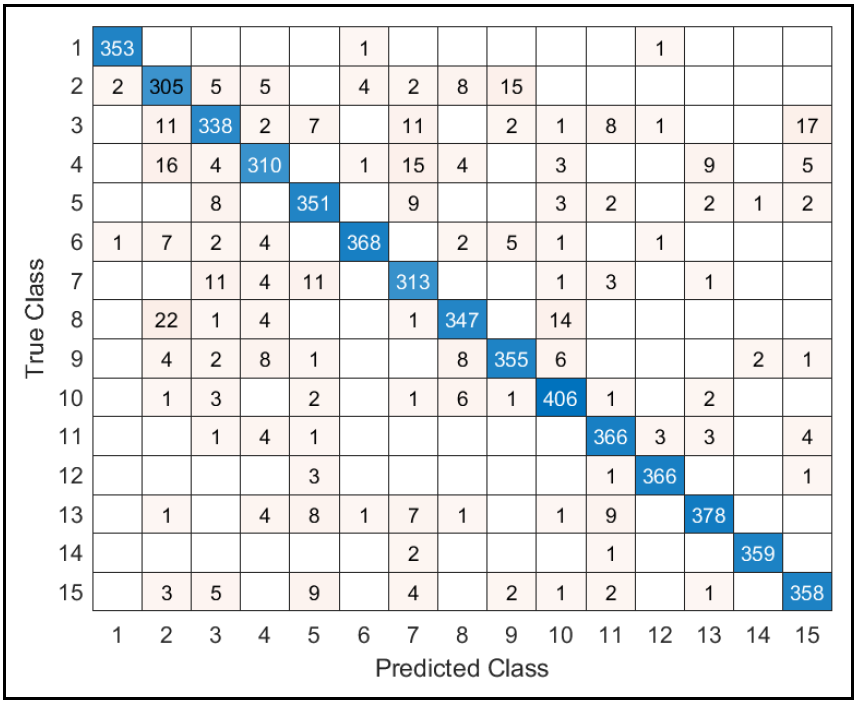
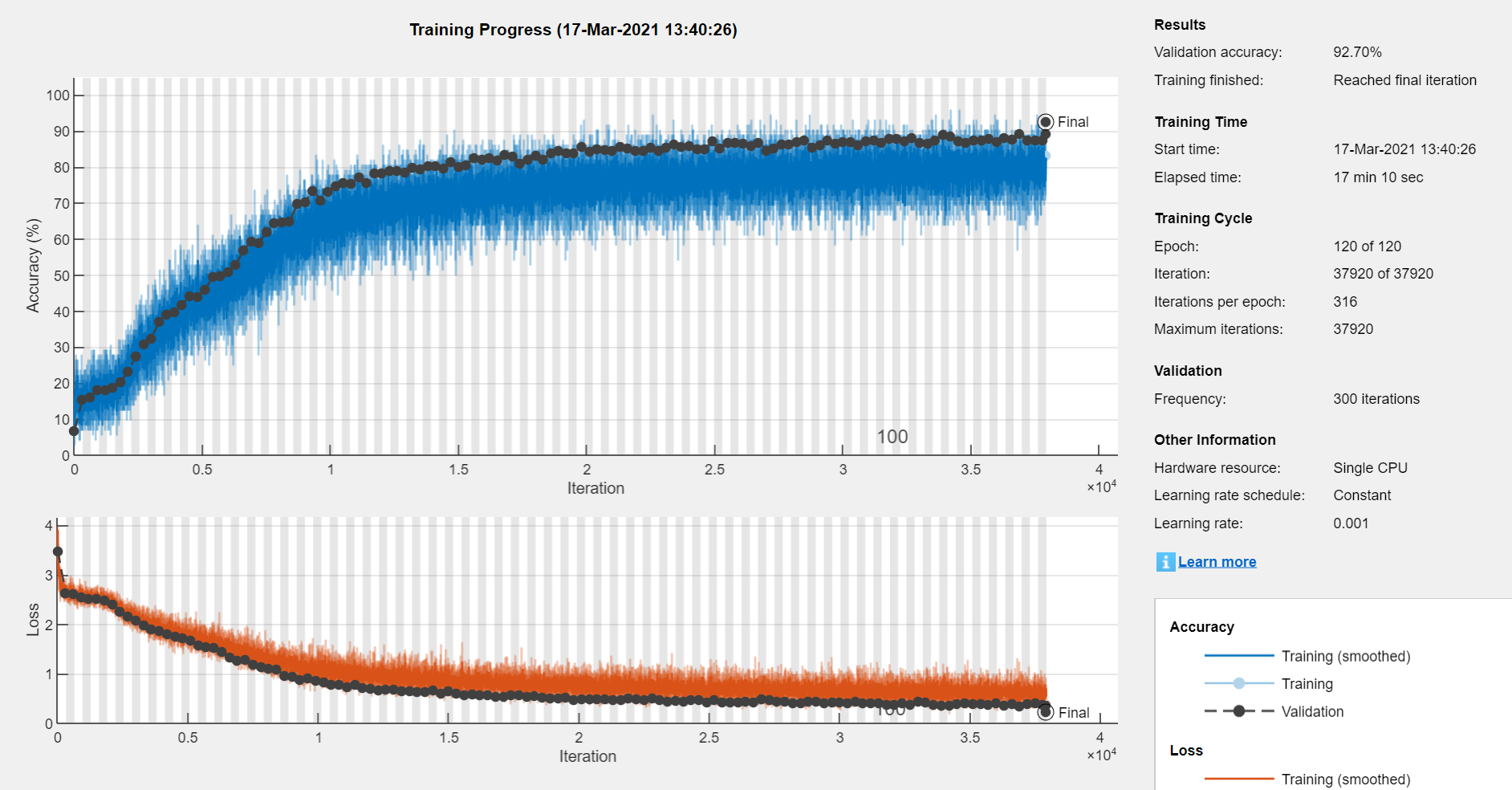
Network structure

The layers used for the structure of the networks are the following:

* sequenceInputLayer: takes as input the segments of the dataset (LSTM)
* featureInputLayer: takes as input the features and applies a normalization (DNN)
* bilstmLayer: is the LSTM cell, which extracts the features from the given time series (LSTM)
* fullyConnectedLayer: a layer made of a given number of neurons, with their weights and biases
* sigmoidLayer: the activation function
* dropoutLayer: used to reduce overfitting
* batchNormalizationLayer: used to make the training faster and more stable
* softmaxLayer: converts a vector of numbers into a vector of probabilities
* classificationLayer: gives the output class to which the input belongs

Training

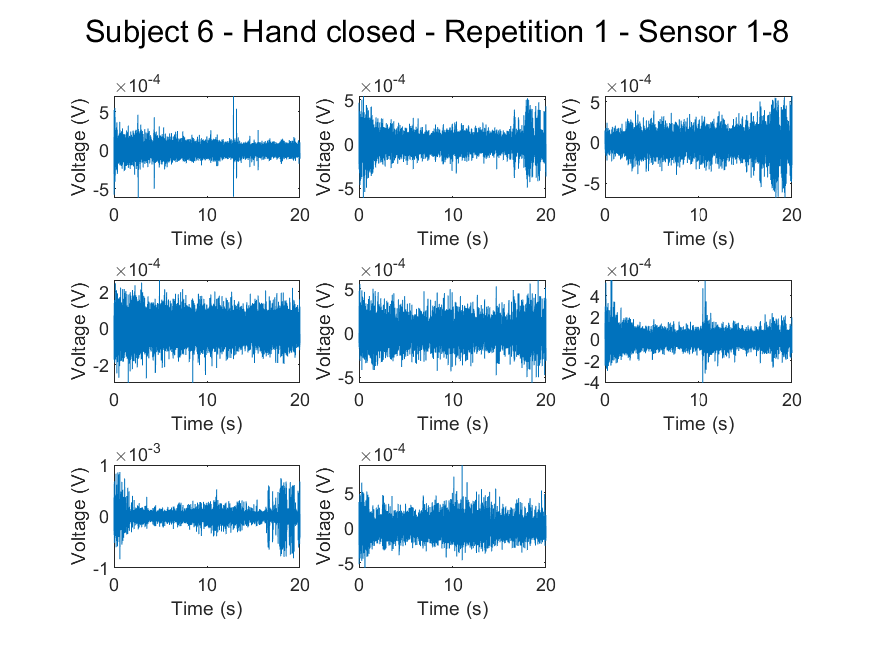
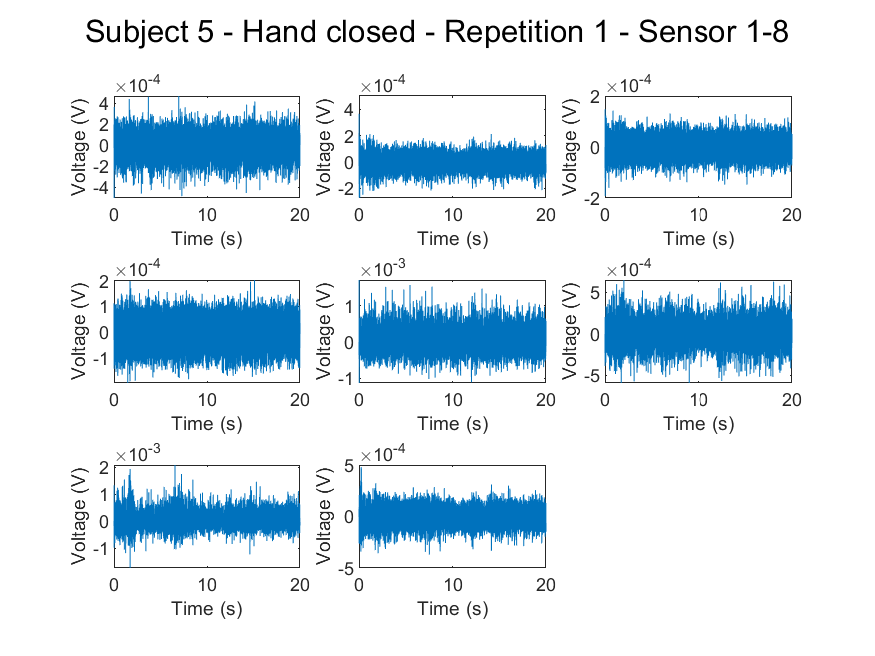
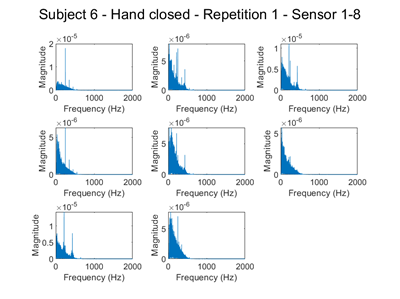
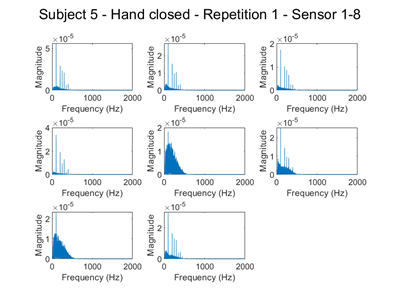
The dataset was randomly shuffled and divided into Training Set (80%) and Testing Set (20%).



Robustness

Lastly, we wanted to evaluate the robustness of the networks.  
First, we trained the networks using 7 subjects, and used the remaining one for the testing.  
We saw that both networks were unable to classify the movements of the remaining subject.  
Then we repeated the training using 2 of the 3 repetitions and left the remaining one for testing. This time, the DNN was able to classify the testing repetitions quite well, while the LSTM could not.

To understand why this happens, we compared the same movement performed by 2 different subjects. We can see that both the time domain signals and their FFT’s are noticeably different, which could explain why the networks struggle when presented a new subject.



Conclusions

We implemented an LSTM and a DNN structure and achieved accuracy of 92.66% and 92.7% respectively. We have found issues regarding the robustness of the networks when a new subject or repetition is analysed. Starting from the results we achieved, further investigations could be made on the robustness of the networks. Both networks could benefit from trying different structures, a larger dataset and data augmentation. For the DNN, implementing more features could also improve the performance.