**Project Report**

The first step in the implementation of the project is data collection. Specifically, according to Publication 3 of the assignment, the gestures that were recorded for the letters A, D, M, L, and U. For each letter, five recordings of approximately 10 seconds were taken, resulting in a total of 25 recordings and 50 seconds of data per letter. Each recording was conducted while the user was seated, wearing the watch on their right hand, and starting the gesture before the recording began and ending it after the gesture was completed. The data are located in the *data* folder, and the directory is exactly as described in the assignment (e.g., data/class\_A/data\_A\_01.csv). Subsequently, the data are stored in MongoDB, in a collection titled ‘aiot\_db’. The five .csv files for each class are stored in the same collection, meaning there are a total of five collections, each labeled with the class they belong to.

For the data processing part, we initially load the data from the database and visualize them using the functions included in utils\_visual.py. Then, we segment the data into windows. After considerable experimentation, we found that the training produced the best results with a window size of 100 and an overlap of 50. Afterwards, we apply a low-pass filter to the windows to smooth the plots, making subsequent analysis easier. Once filtering is complete, we perform a flattening operation on the windows by converting each window to a vector, for the training process that follows. The flattened data is then split into training and testing data with 80-20 ratio. The data are then scaled using standard scaler, have their labels encoded using label encoder and are transformed back into 2D. Three kinds of PCA procedures are done on the data. PCA with 2 and 3 components and PCA with 36 components that are needed to explain 95% of the variance of the dataset. The data is then transformed by being projected onto the axes of the principal components. The projected data of the last PCA are then being used to train two statistical models. One SVM and one random forest. The models have 0.79 and 0.80 accuracy correspondingly. Following the creation of the two models, a grid search algorithm is used to find the best parameters. Two new models emerge out of this process with 0.85 and 0.90 accuracy correspondingly.

What follows is the neural network phase. In that phase, a CNN was chosen due to the nature of the data being a multidimensional time series. The parameters were chosen through trial and error. The main architecture is four convolutional layers with each having double the filters of the previous. The data used are the original data, split into train-test-validation subsets. The model had at most 0.88 accuracy. The accuracy over time goes up when it comes to train data and moves chaotically when it comes to validation data. The loss over time does the same thing but going down. Those are indicators that the model was learning each epoch but it was also overfitting.

The results of the testing indicate that the gesture recognition system performs adequately, yielding generally reliable outcomes. However, it falls short of the 90% accuracy threshold, primarily due to occasional inconsistencies in the input data (as illustrated in the Time Domain plot of class\_A.csv) and not the absolute best parameter selection in the training models.

Theofrastos Paximadis 1093460

Konstantinos Anastasopoulos 1093320