Overview

A human activity recognition (HAR) system was designed to recognize the user's hand gestures and send input orders to the virtual reality engine that will act accordingly. Additionally, a brain-computer interface (BCI) is going to be constructed to be used as a feedback system to determine what the student is feeling while using the proposed system. Hence, it can be inferred whether the proposed system is effective in the educational process or not. The process of building a HAR or a BCI is quite similar. The process consists of four main steps (i) data collection, (ii) segmentation, (iii) feature extraction, (iv) classification and feature selection. A block diagram of the process is shown in figure 1. with some differences in the process of preprocessing the signals and selecting the useful features.

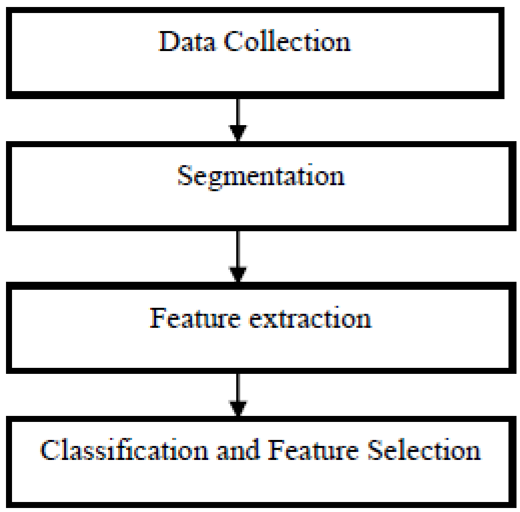


Figure 1

For the HAR system, the UCI-HAR dataset is used. It is a publicly available dataset collected in collaboration with 30 volunteers who were asked to perform a protocol of different activities including walking, walking upstairs, walking downstairs, sitting, standing, and lying. All participants were wearing a smartphone that includes an accelerometer and a gyroscope on the waist during the experiments' execution. During each experiment, 3-axial acceleration signals and 3-axial angular velocities signals were captured using smartphone sensors. The sampling rate for all sensors is 50 Hz. The data includes the participant's ID from 1:30, the starting and end time of each experiment, the accelerometer signals on the x,y, and z axes, and the gyroscope signals on the x,y, and z axes. Dataset description is summarized in Table 1.

Dataset	Sensors	Sampling Rate	No. of Users	No. of Activities	Activities
UCI HAR	• Accelerometer • Gyroscope	50Hz	30	6	• Walking • Walking upstairs • Walking downstairs • Sitting • Standing • Laying

Table 1

HAR System

Human activity recognition system full process in figure2.

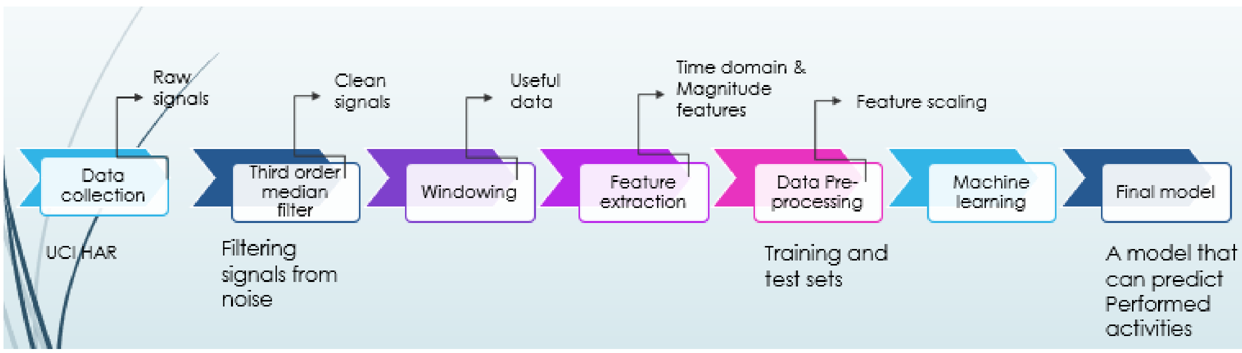
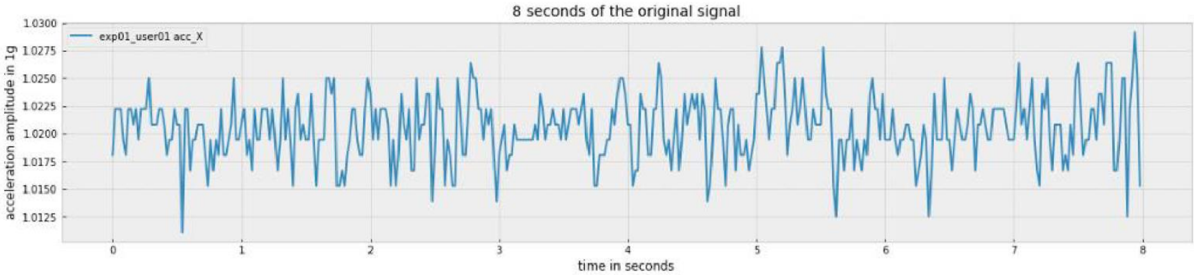


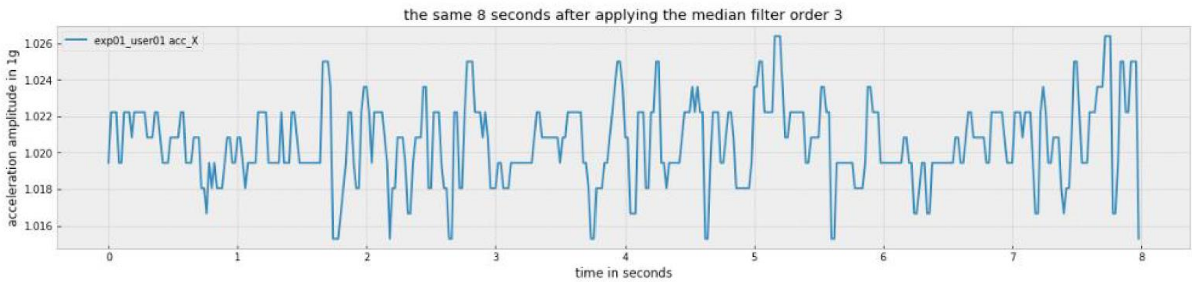
Figure2

Data collection

As mentioned earlier the UCI-HAR dataset is used to build the HAR system. The data includes raw signals that cannot be fed directly to the machine learning model. Therefore, signal preprocessing is needed to filter the raw signals from noise and divide them into small segments or windows. From each window, a number of features will be extracted to obtain clean data that will be trained using machine learning classifiers.

The process of filtering noise was done by defining a third-order median filter. The third order median filter works through for each value in a column the filter selects a 2 other values around the central value: 1 before and 1 after. It returns the median value of this list which will be stored in the output column with the same index of the original value. The first value in the output column is the median of the first 3 values in the original one and the second has the same value as the first one. The third value in the output column is the median of the 2nd 3rd and 4th values in the original. From the two figures below, it appears clearly the effect of the median filter on the original signal.





After extracting clean signals from the raw signals, the magnitude of each three axes will be calculated using the Euclidian magnitude equation in figure3.

$$\sqrt[2]{X^2 i + Y^2 i + Z i^2}$$

Figure3

Segmentation

For the used dataset, a default 50 Hz sampling frequency is used for all of the two sensor data. A sliding window of 2.56s is applied on the dataset with a 50% overlap between two consecutive windows.50% overlap means that the starting point of the next window is obtained by shifting the starting point of the actual window by 64 rows.

Feature Extraction

To train a machine learning model, discriminative features are needed to be extracted. Features are different variables that are extracted from the data and can differentiate between different activities of a dataset. Eight statistical features are extracted in time-domain from each axis of the three axes of the two sensors. Moreover, magnitude features are extracted from magnitude values. The extracted features are as follows:

i. mean(): Mean value of one array.
ii. std(): Standard deviation of one array.
iii. mad(): Median absolute deviation of one array.
iv. max(): Largest value in one array.
v. min(): Smallest value one array.
vi. energy(): Energy measure. Sum of the squares of one array divided by the number of values.
vii. iqr(): Interquartile range, the value of the third quartile minus the value of the first quartile.
viii. entropy(): Signal entropy.

Data pre-processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Data usually contains missing values, outliers, categories, or names. Only feature scaling is used in order to standardize the data to have a mean of 0 and a standard deviation of 1. Afterward, the data is divided into two sets. The first set is the training set that includes 70% of the data. The training set will be used to train the machine learning classifiers to predict similar data in the future. The second set is the test set that includes 30% of the data. The test set will be fed to the classifiers to predict the performed activity. Then, the results of the predicted values will be used to evaluate the classifier.

Machine learning classifiers

To train the data, three machine learning classifiers are constructed. The random forest (RF), the support vector machine (SVM), and the logistic regression(LR). The random forest (RF) is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A 100 tree is used in the forest. Support vector machine (SVM) is a machine algorithm that finds decision boundaries called hyperplanes that help classify the data points. Logistic regression (LR) is used to model the probability of a certain class or event existing such as pass/fail, win/lose. It uses the sigmoid function with a threshold to classify data.

Model evaluation.

To Evaluate models' performances, accuracy, sensitivity, and the F1 score are used.

$\boldsymbol{Accuracy}(\boldsymbol{model})= \boldsymbol{number\ of\ samples\ correctly\ predictedtotal} / \boldsymbol{number\ of\ samples}.$

$\boldsymbol{True\ Positives}\ (\boldsymbol{TP})$: $\boldsymbol{number\ of}\ samples\ of\ a\ \boldsymbol{class}\ correctly\ predicted\ as\ samples\ of\ the\ same\ \boldsymbol{class}.$

$\boldsymbol{False\ Negatives}\ (\boldsymbol{FN})$: $\boldsymbol{number\ of}\ samples\ of\ a\ \boldsymbol{class}\ incorrectly\ predicted\ as\ a\ samples\ of\ another\ \boldsymbol{class}.$

$\boldsymbol{False\ Positives}\ (\boldsymbol{FP})$: $\boldsymbol{number\ of}\ samples\ of\ other\ \boldsymbol{classes}\ incorrectly\ predicted\ as\ samples\ of\ \boldsymbol{class}.$

Sensitivity measures the model's ability to correctly predict samples of a class.

$$\boldsymbol{Sensitivity}\ = \boldsymbol{TP} / (\boldsymbol{TP} + \boldsymbol{FN})$$

$$\boldsymbol{Precision} = \boldsymbol{TP} / (\boldsymbol{TP} + \boldsymbol{FP})$$

The F1-score is calculated to evaluate the system performance. Classification accuracy is commonly used as it is a single measure used to summarize model accomplishment. F-Measure provides a way to combine both precisions and recall into a single measure and captures both properties.

F1 Score = (2 *Precision * Recall) / (Precision + Recall)