Human Activity Recognition Using Smartphones

Koyel Pramanick (201333, koyel20@iitk.ac.in) ; Sahil Yadav (201399, sahilydv20@iitk.ac.in);
Shahana M A (191132, shana@iitk.ac.in); Vijay Kumar (201454, kumarvijay20@iitk.ac.in)

Data Mining(CS685A) , Guided by Dr.Arnab Bhattacharya, Department of CSE.

ABSTRACT:

We approach Human activity recognition as a supervised classification task. The challenging issue here is how to feed this classifier with a fixed number of features where the real input is a raw signal of varying length. The sensor signals were sampled in fixed width sliding windows of 3 sec and 20% overlap. Selected relevant features from the 335 new features created. Classified data using Logistic regression, SVM and Random Forest. Final model gives 98.33% and 94.37% accuracies over our two out of time data.

INTRODUCTION

Smartphones has become an inevitable factor in our daily life. Sensors has a big role in making smartphones more functional and aware of the environment thus most smartphones comes with different sensors and this makes it possible to collect vast amounts of information about the user's daily life and activities.

Accelerometer, Gyroscope and Linear Acceleration sensors are three among them. Accelerometers in mobile phones are used to detect the orientation of the phone. An accelerometer measures linear acceleration of movement, while a gyroscope on the other hand measures the angular rotational velocity. Both sensors measure rate of change; they just measure the rate of change for different things. Linear acceleration sensors, also called G-force sensors, are devices that measure acceleration caused by movement, vibration, collision. Since there is a meaningful difference of characteristics between datas retrieved from these sensors, many features could be generated from these sensors data to determine activity of the person that is carrying the device.

In this study we recognize the action done by the user using the data retrieved from these sensors which are affected from human movements.

Dataset consists of signals from accelerometer, gyroscope and linear acceleration sensors of a smartphone carried by different persons while doing 7 different activities are classified using various machine learning algorithms. In this study we create new features from two possible feature sets namely time-domain and frequency-domain statistics to represent motion signal obtained from accelerometer and gyroscope reads from the smartphone. Performance of different approaches are analysed and compared in terms of presicion and efficiency.

Dataset:

Dataset consists of signals from accelerometer and gyroscope sensors in smartphones recorded while users executed 7 different physical activities. Those are

1. walking
2. sitting
3. standing
4. jogging
5. biking
6. walking upstairs
7. walking downstairs

which are mainly used in the related studies and they are the basic motion activities in daily life. There were ten participants involved in data collection experiment who performed each of these activities for 3-4 minutes. All ten participants were male, between the ages of 25 and 30. Each of these participants was equipped with five smartphones on five body positions:

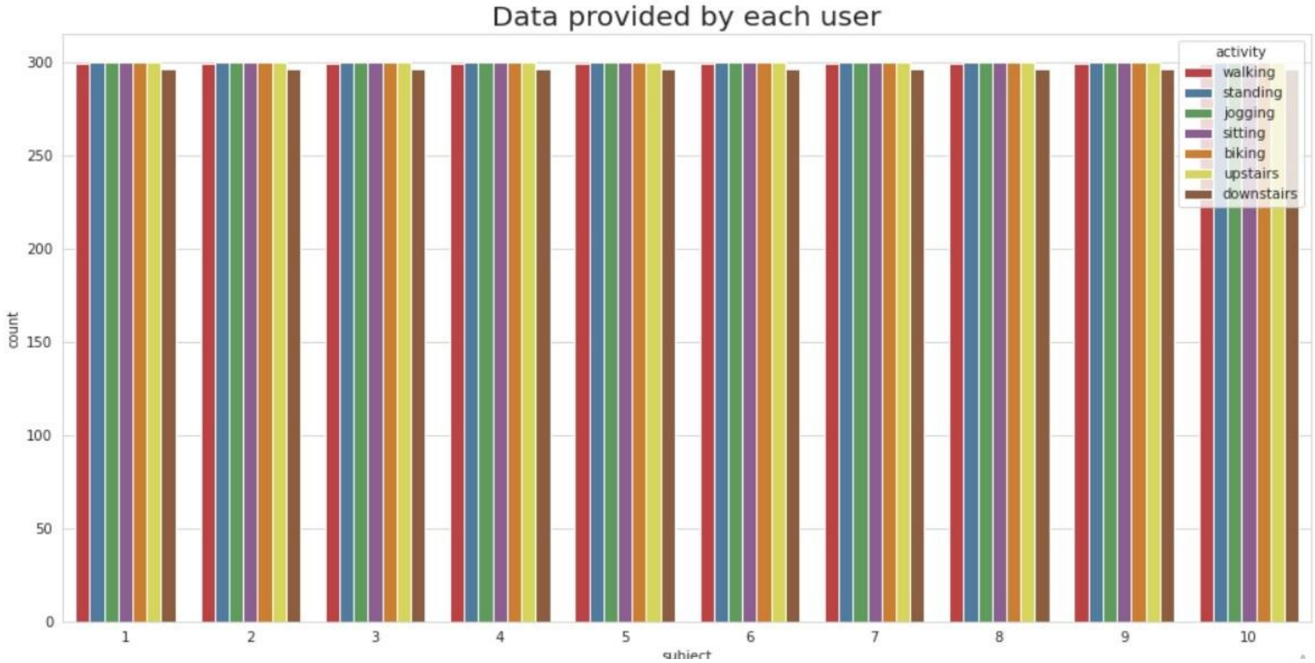
One in their right jean’s pocket, one in their left jean’s pocket, one on belt position towards the right leg using a belt clipper, one on the right upper arm and one on the right wrist.

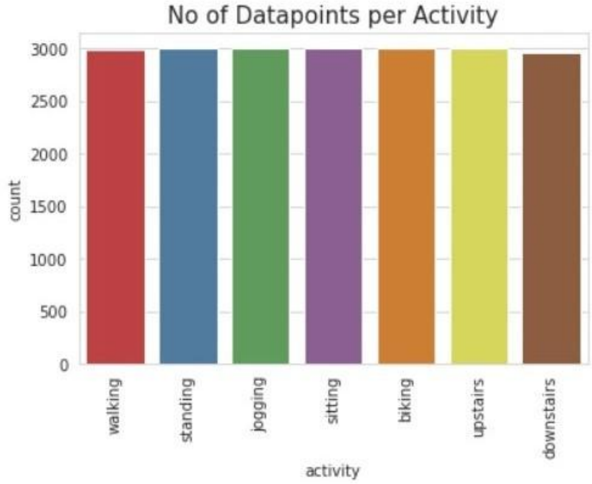
The orientation of the smartphones was portrait for the upper arm, wrist, and two pockets, and landscape for the belt position. The data was recorded for all five positions at the same time for each activity and it was collected at a rate of 50 samples per second. This sampling rate (50 samples per second) is enough to recognize human physical activities.

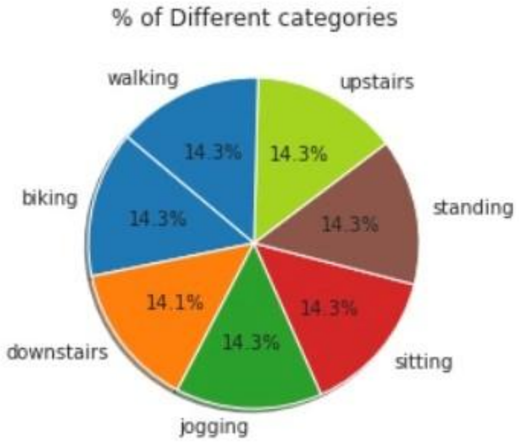
Each file contains data for each participant's seven physical activities for all five positions. Each dataset is perfectly balanced with approximately equal number of observations in each class.

Exploratory Data Analysis:

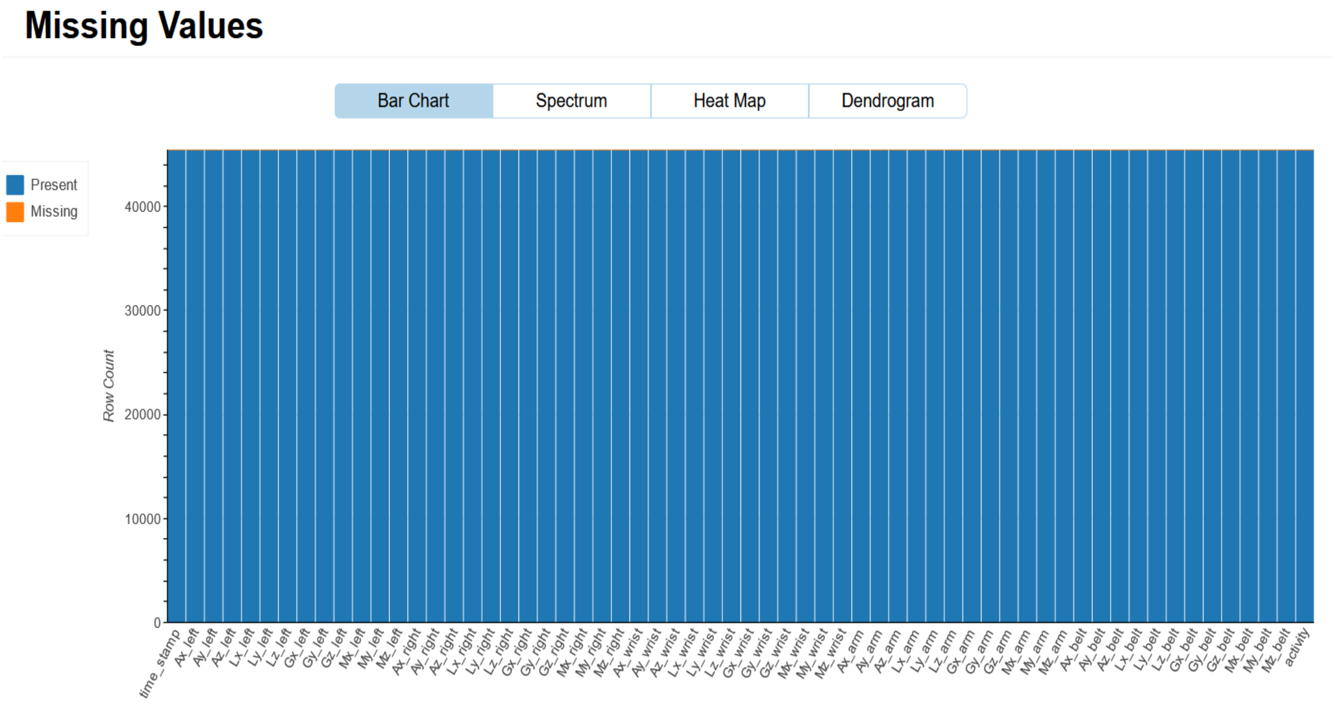
Our dataset contains 10 different files from 10 different persons.. Primary exploratory data analysis show that the data is perfectly balanced. Figures shown below shows that number of datapoints from each activity are approximately equal.



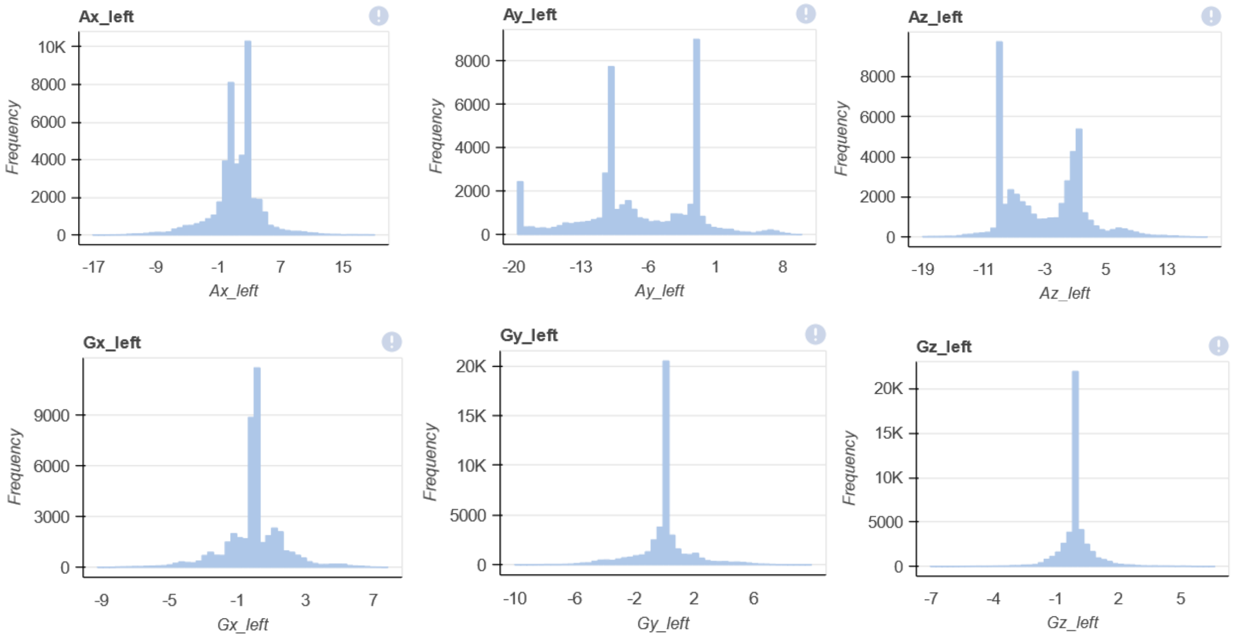


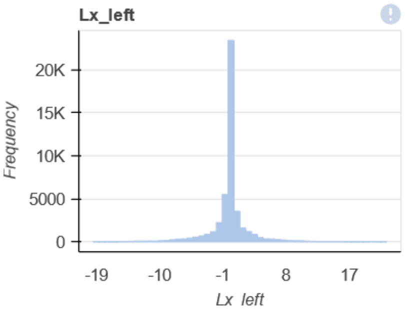


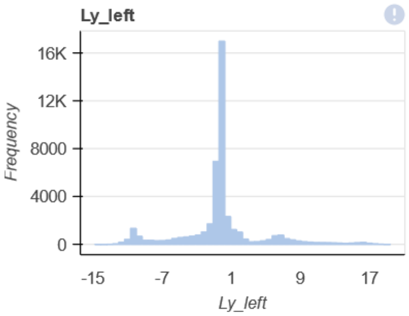
The data is free from missing values as well. Figure below validates this statement.

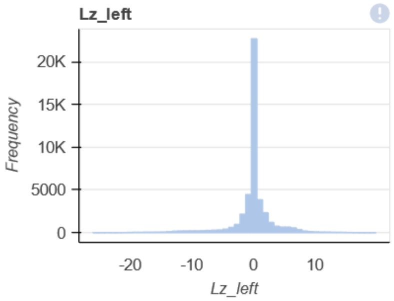


Several analysis like distribution and Normal Q-Q plot of all the x,y,z axis values in all sensors in all positions were analysed. A sample of which was done for position left pocket are shown below .

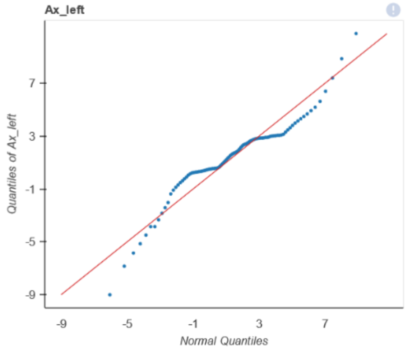


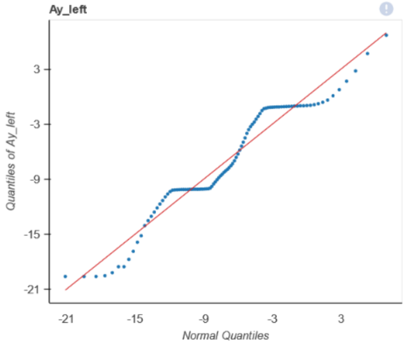


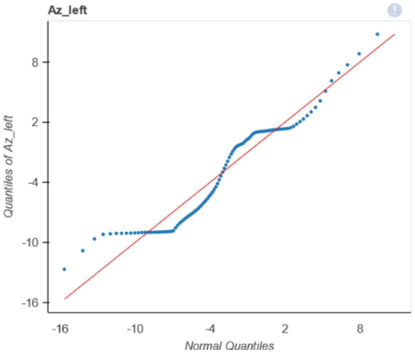


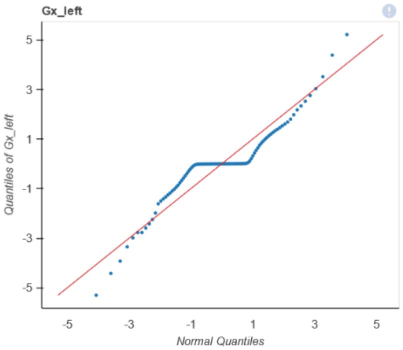


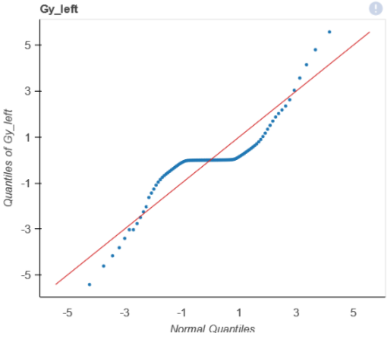
As expected from most real world data, the duration of walking on staircase is normally distributed.

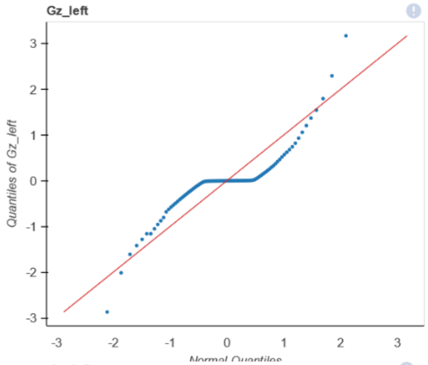


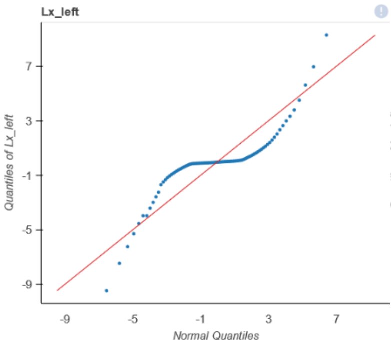


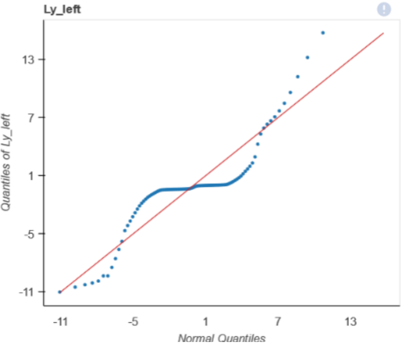


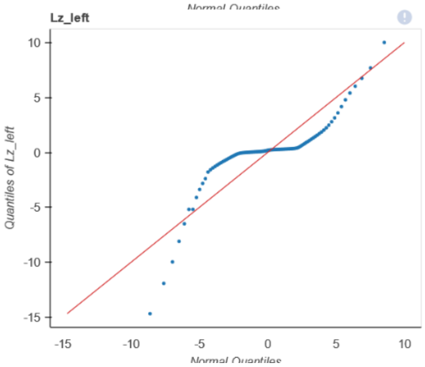








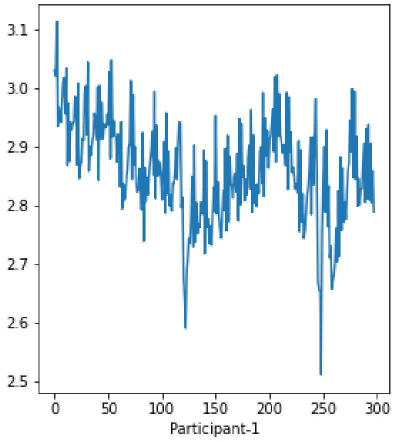


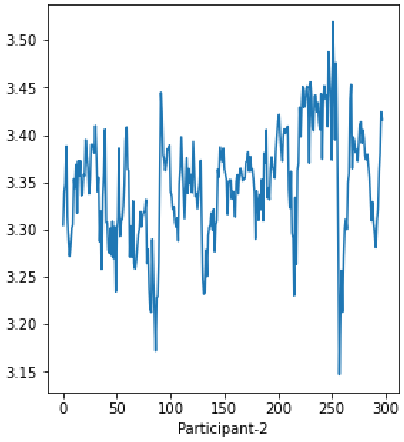


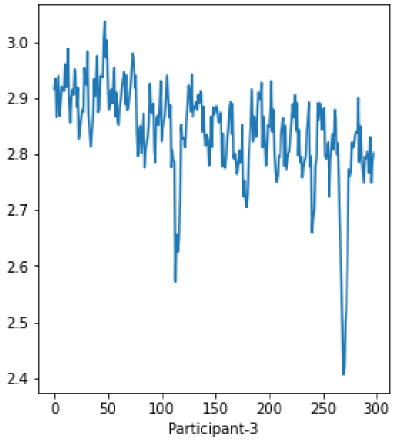
Is there a unique walking style for each person?

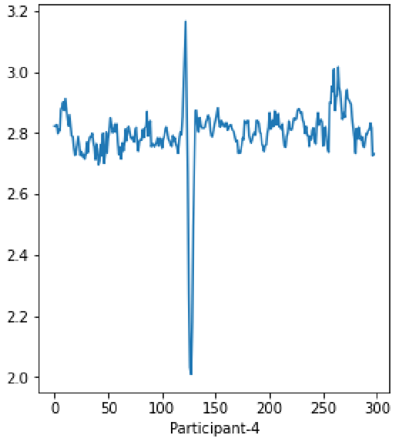
Walking for each participant:

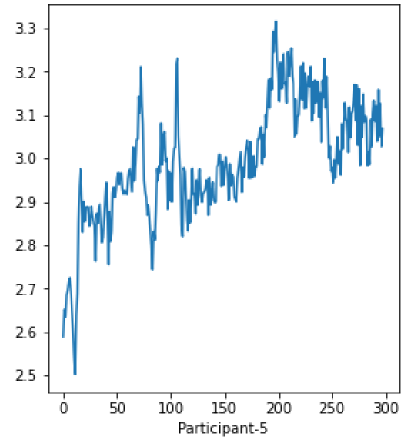
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **walking**.

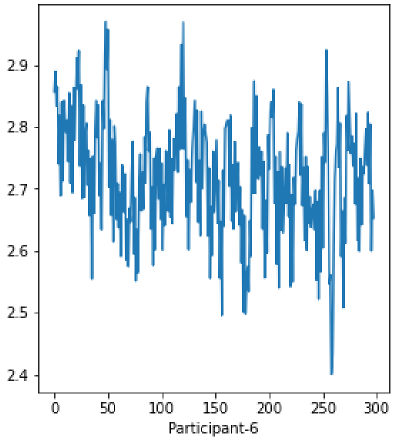


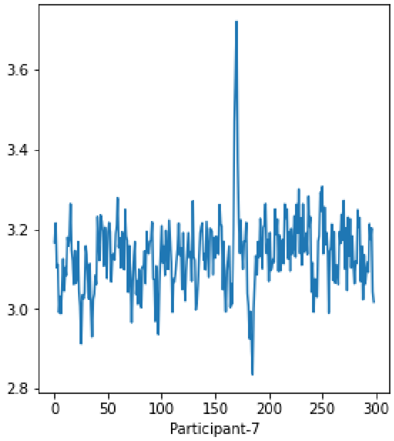


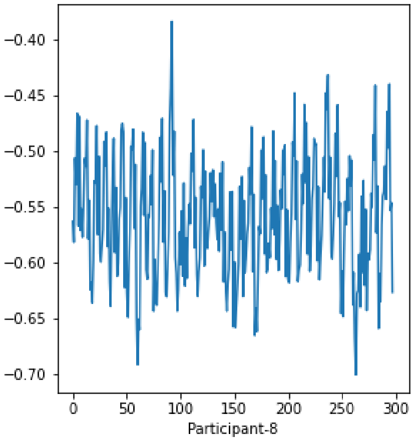


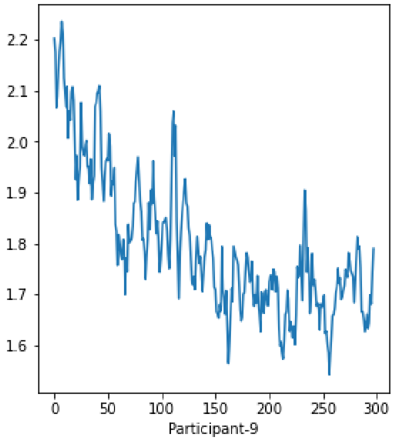


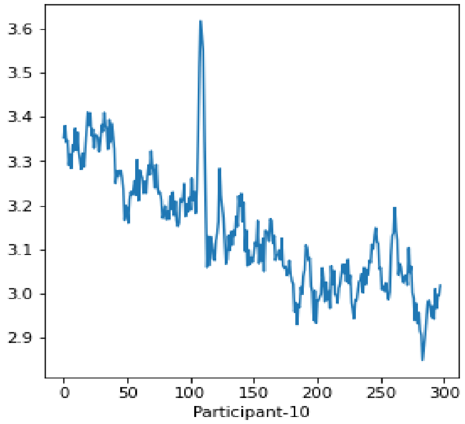






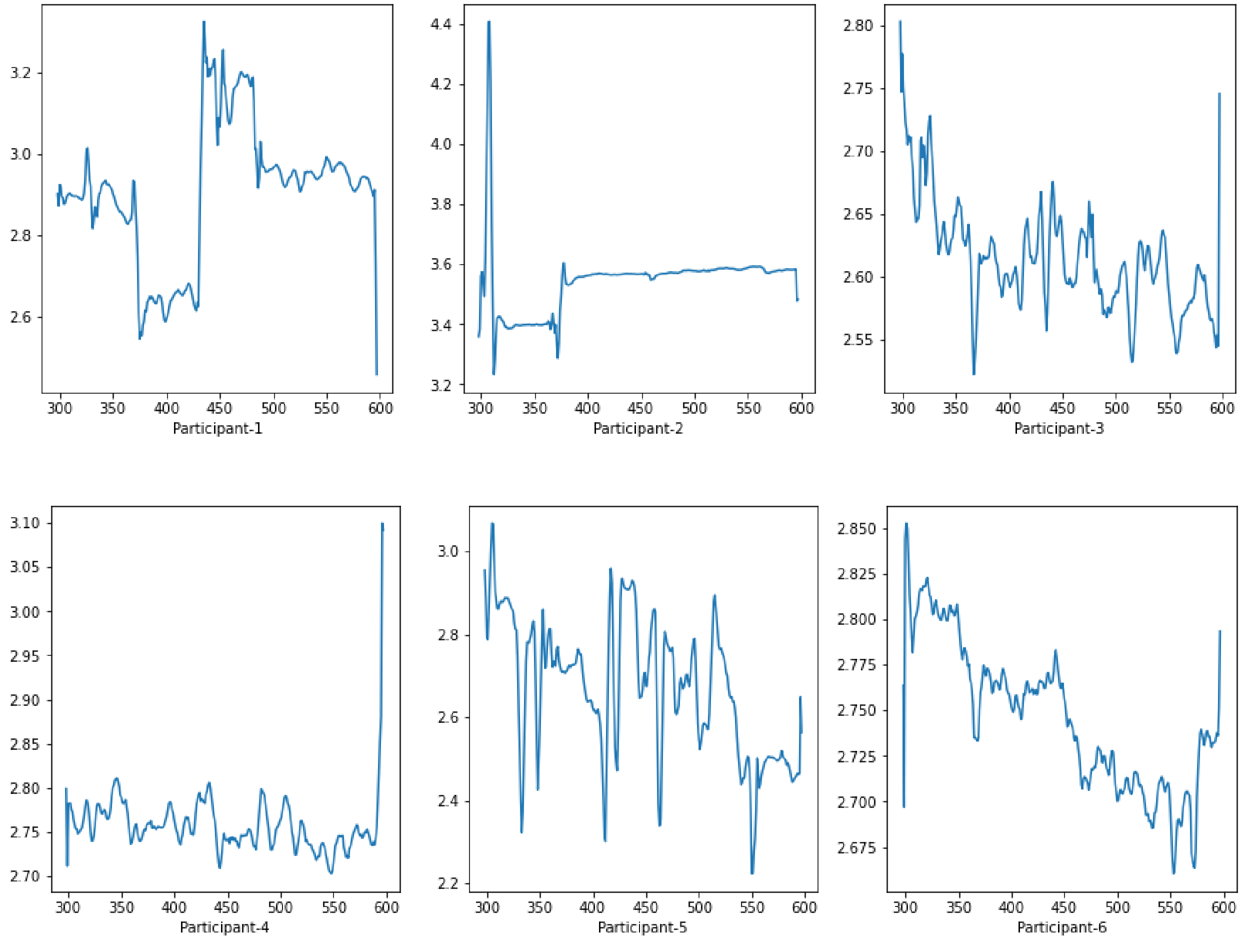


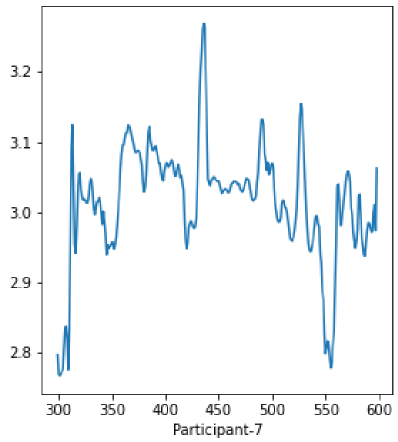


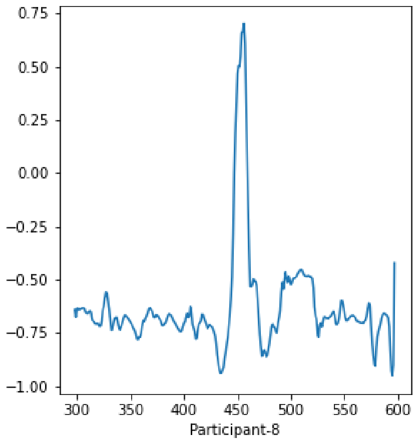


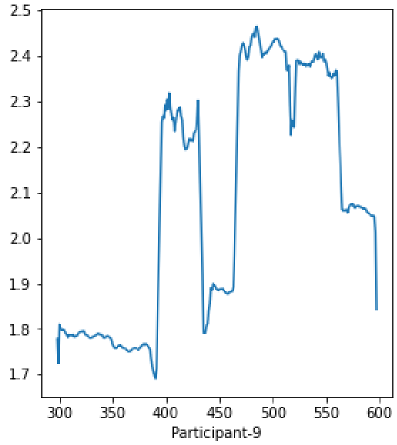
Standing for each participant:

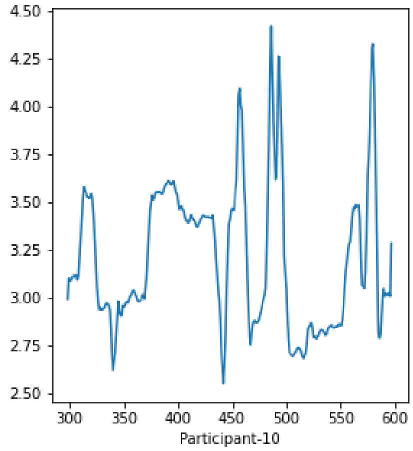
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **standing**.





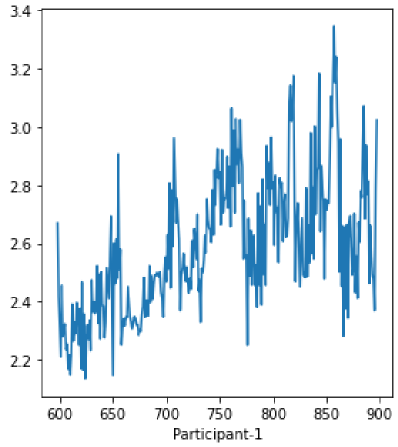


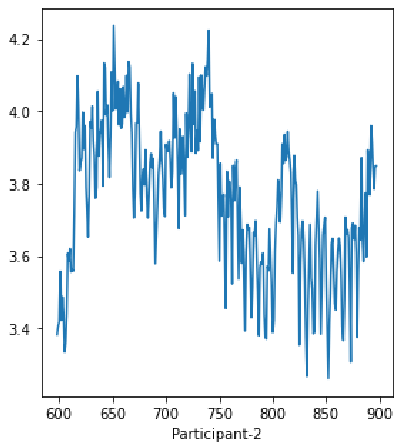


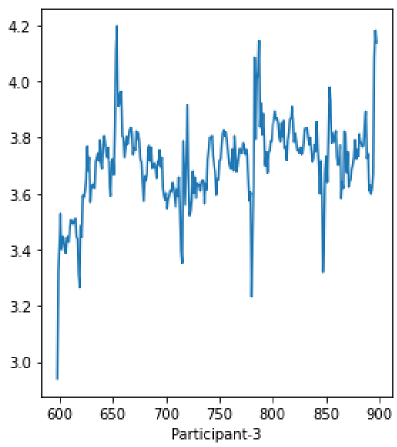


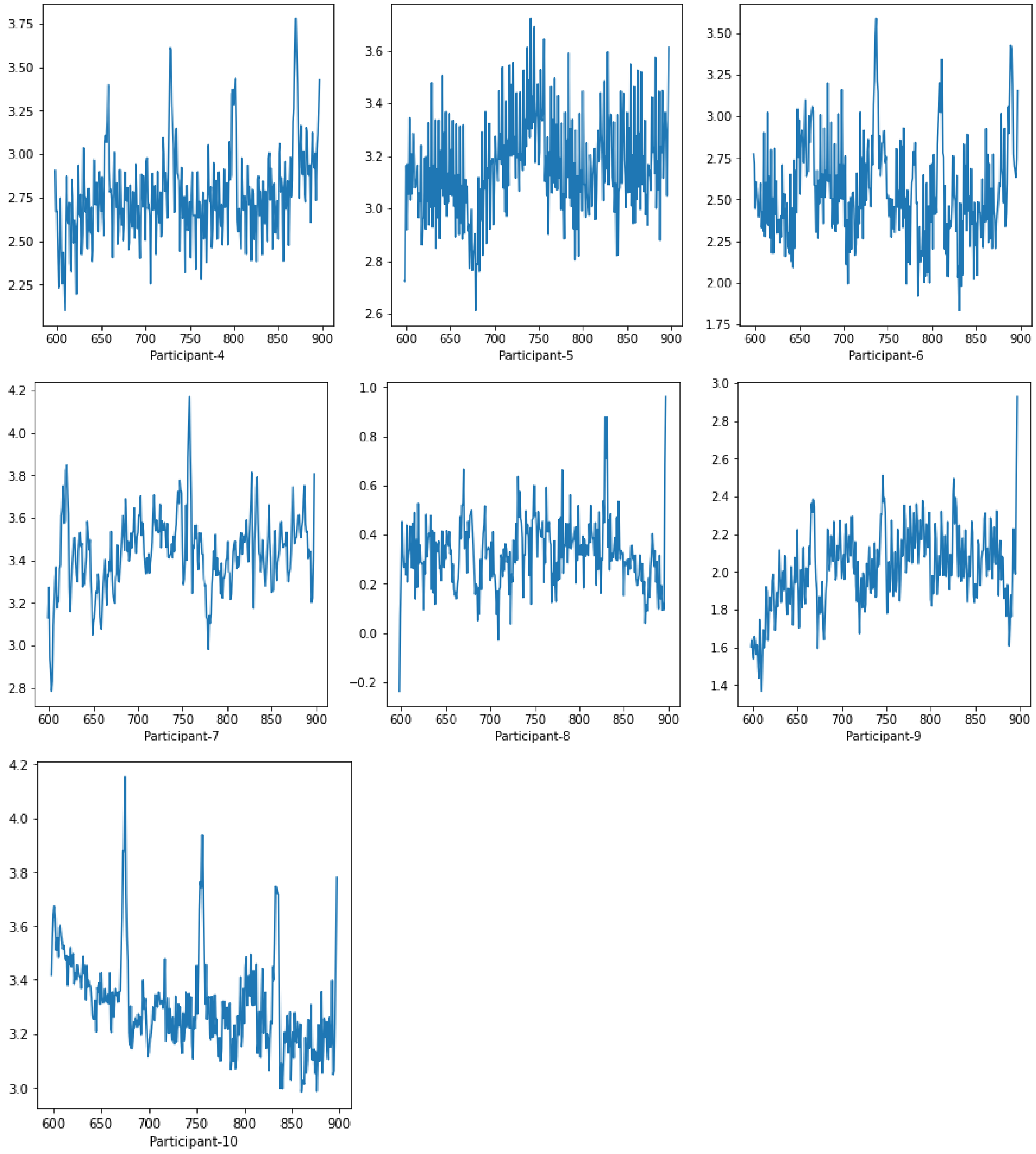
Jogging for each participant:

Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **jogging**.



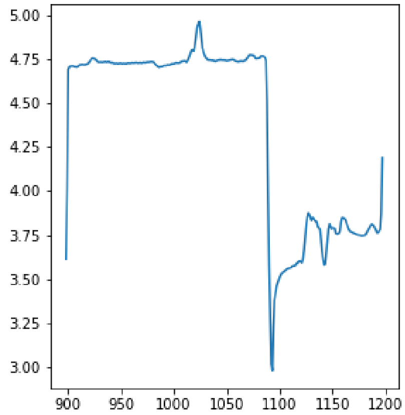




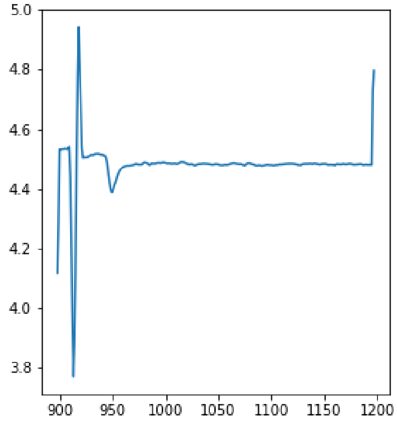


Sitting for each participant:

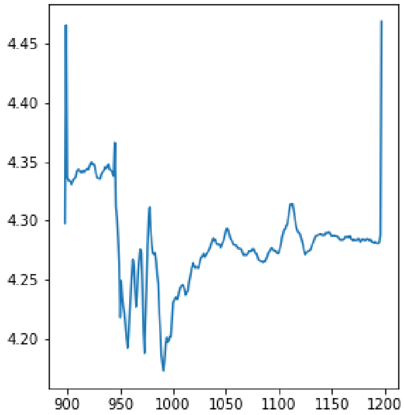
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **sitting**.



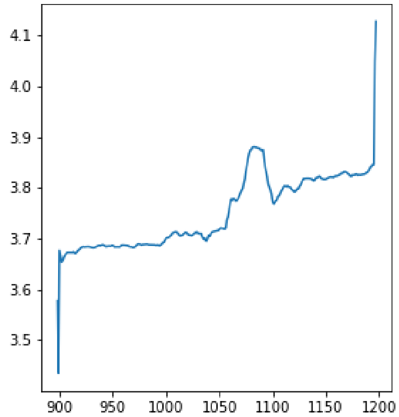
Participant-1



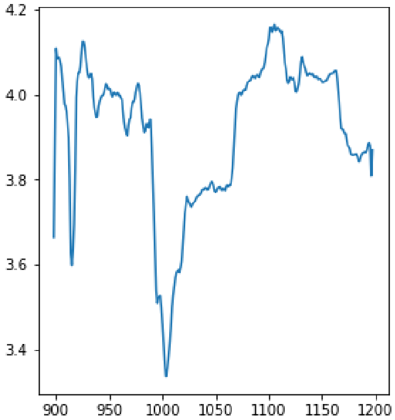
Participant-2



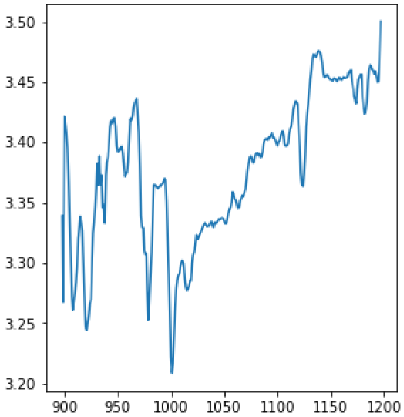
Participant-3



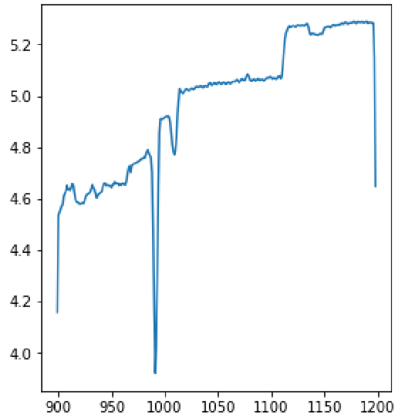
Participant-4



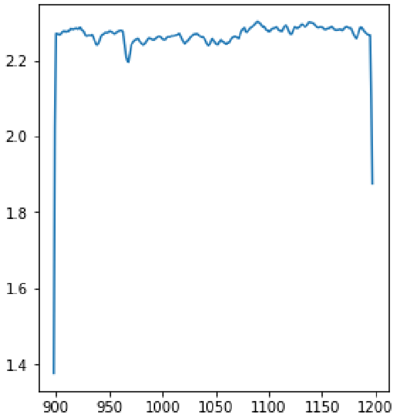
Participant-5



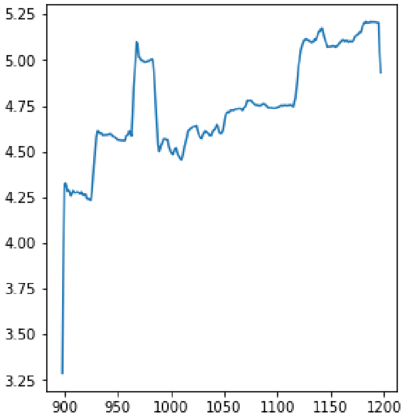
Participant-6



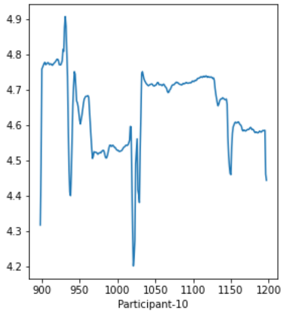
Participant-7



Participant-8



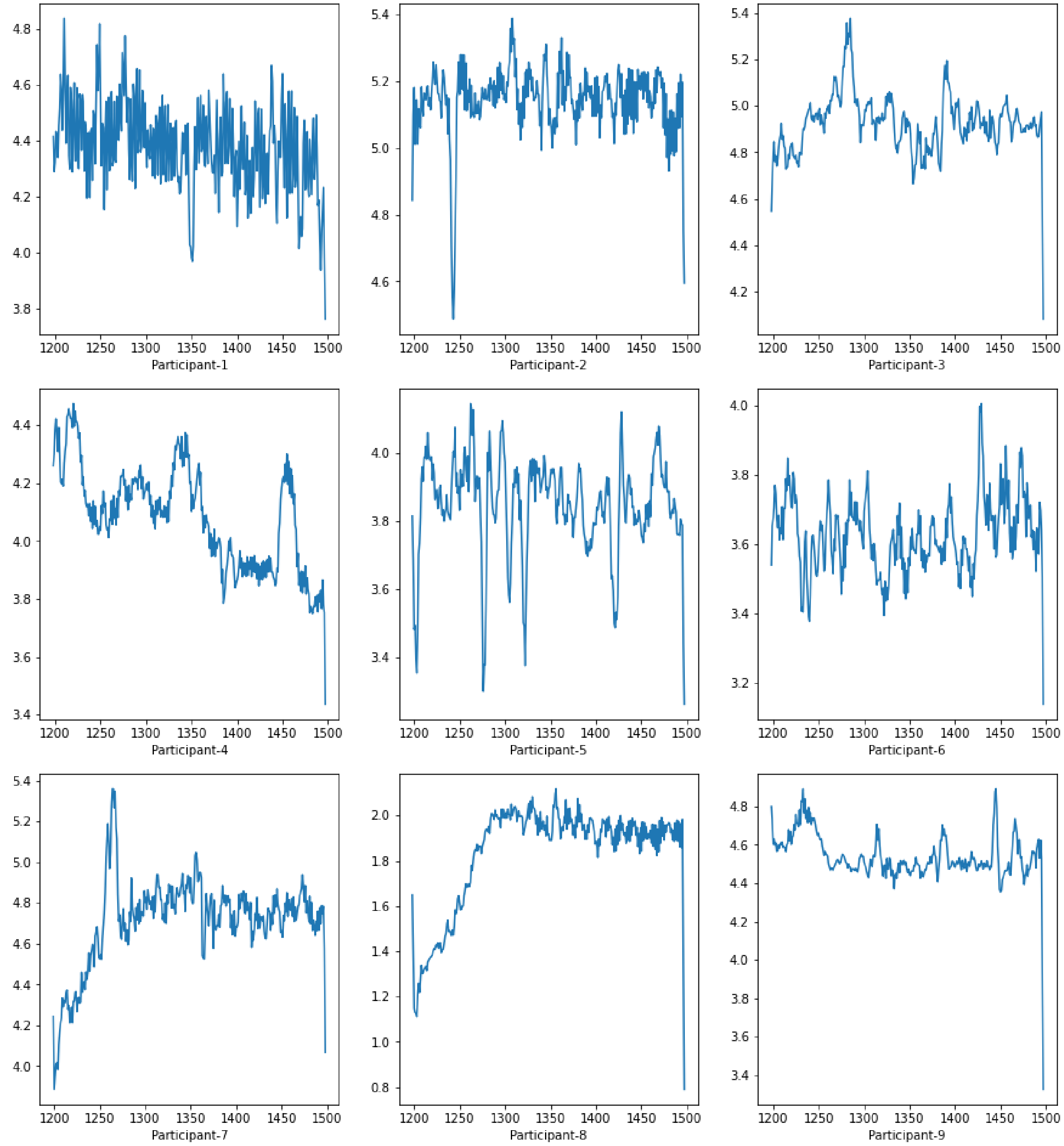
Participant-9

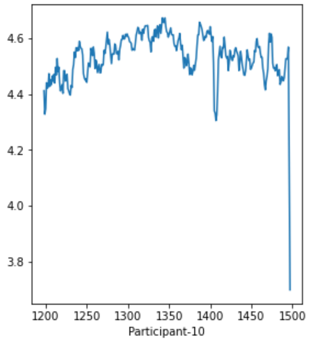


Participant-10

Biking for each participant:

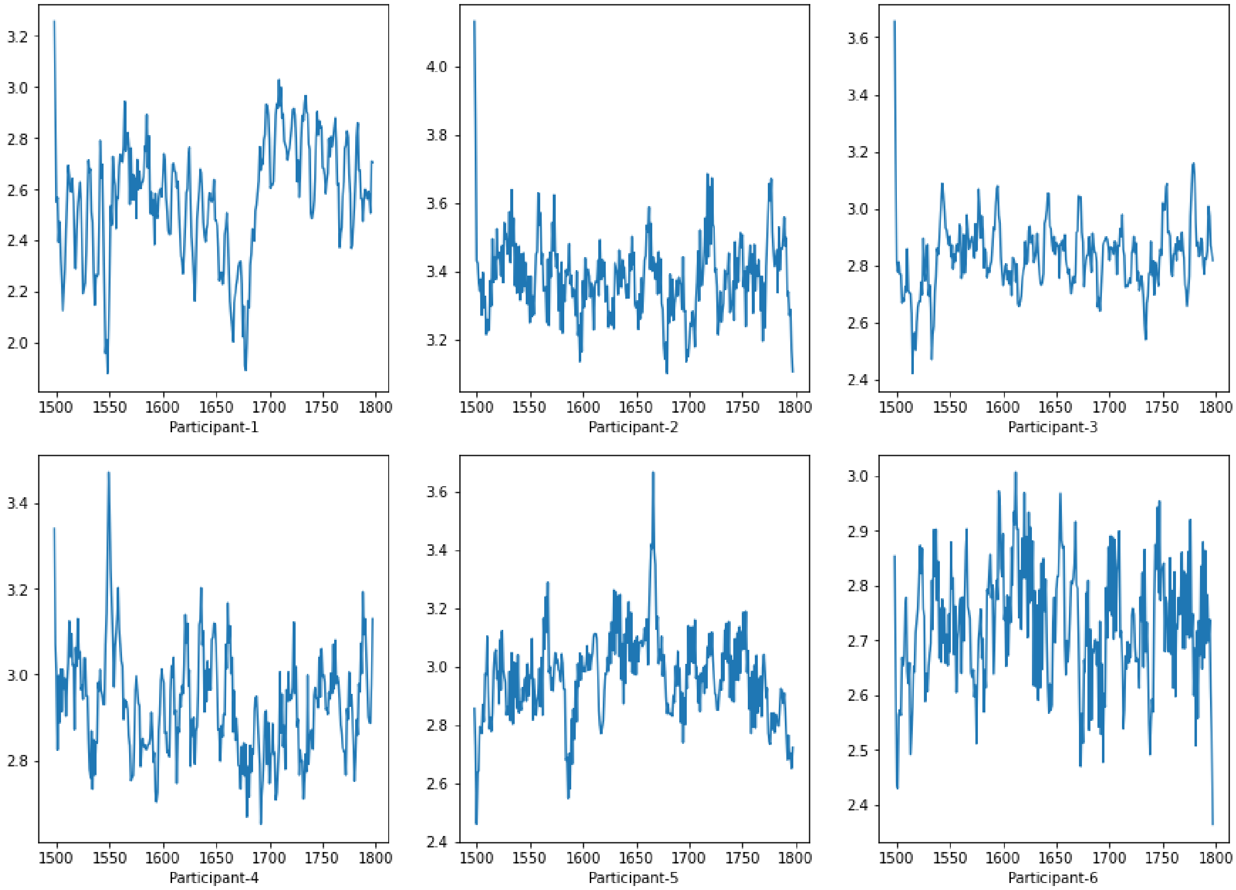
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **biking**.

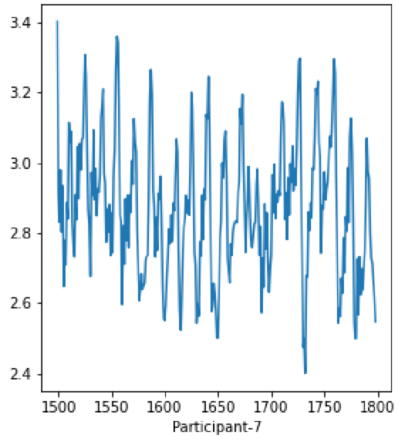


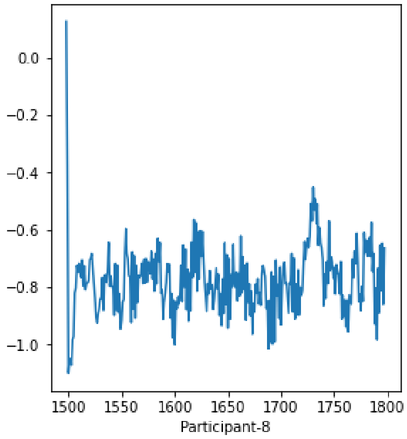


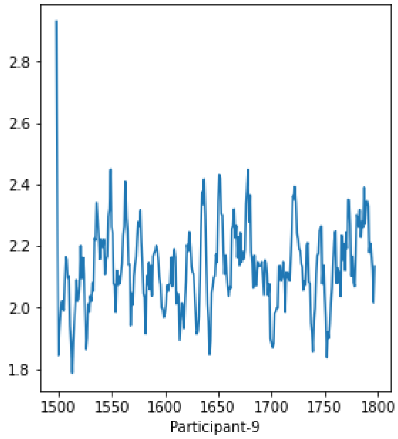
Upstairs for each participant:

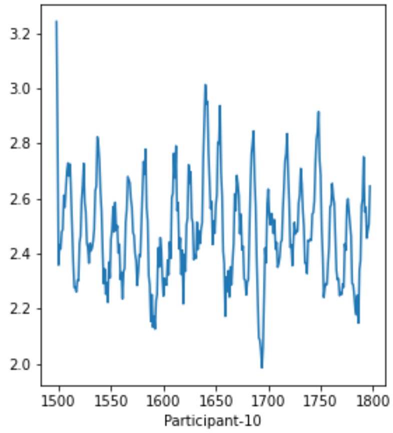
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **upstairs**.





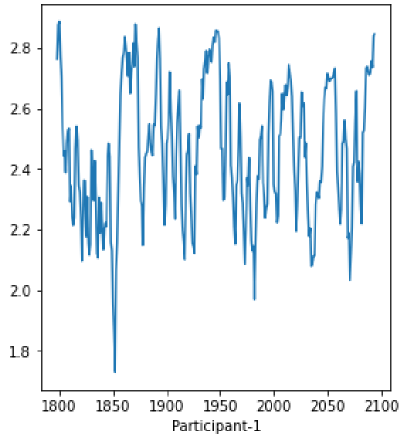


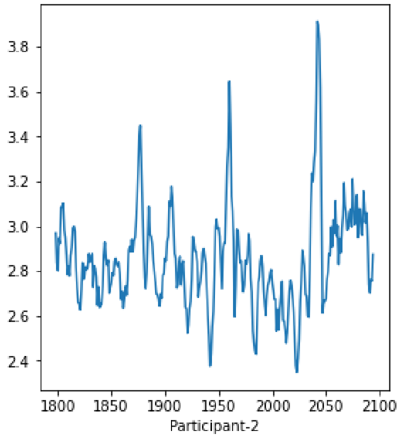


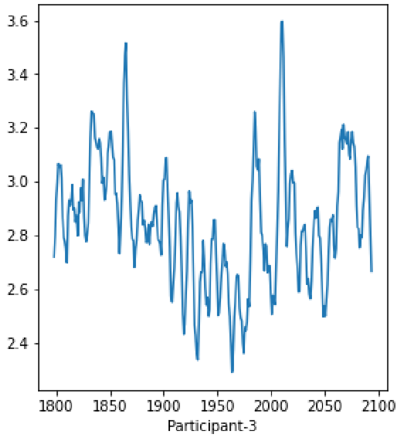


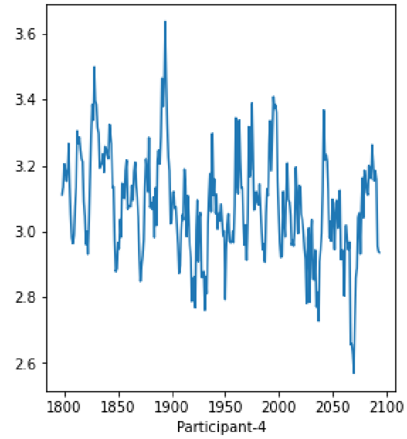
Downstairs for each participant:

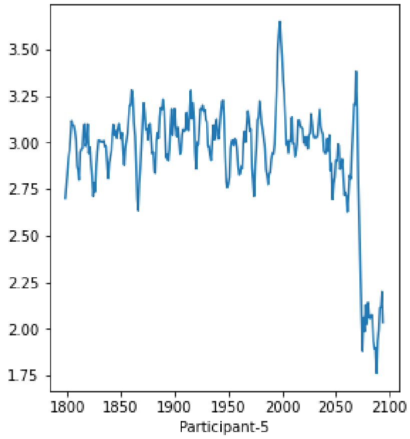
Here we plot mean values of each window (containing 150 samples) of x-values from accelerometer of 5 different sensors for **downstairs**.

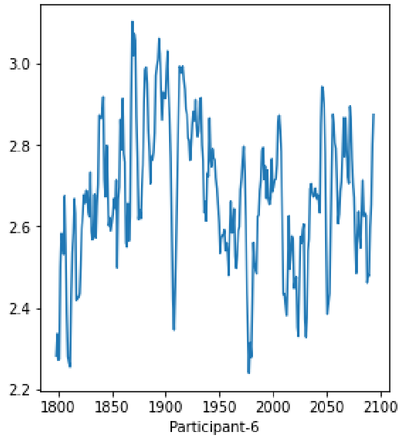


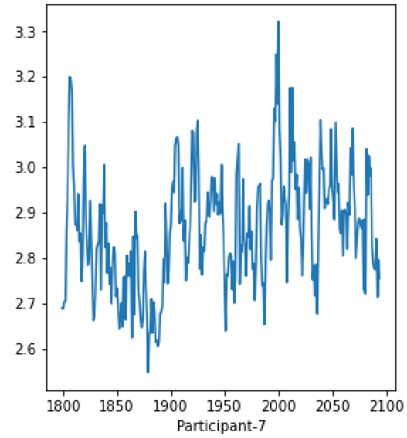


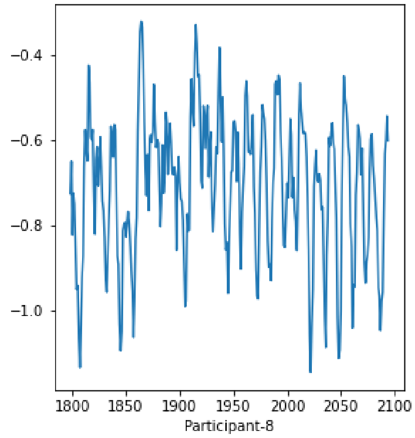


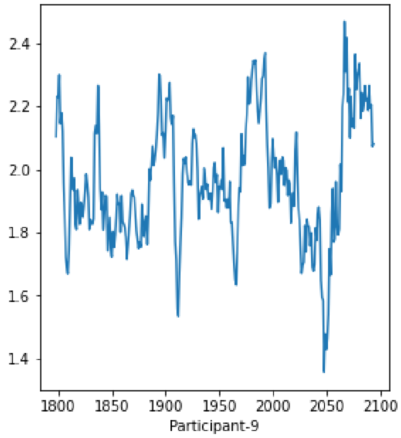

3.6
3.4
3.2
3.0
2.8
2.6
1800
1850
1900
1950
2000
2050
2100
Participant-4

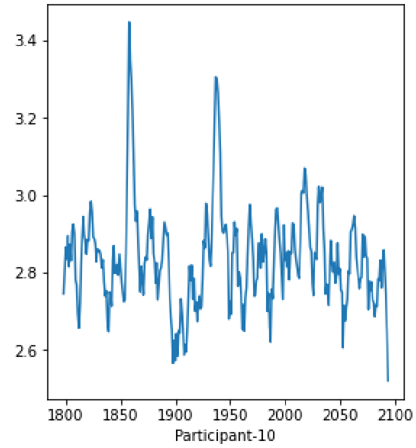

3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1800
1850
1900
1950
2000
2050
2100
Participant-5


3.0
2.8
2.6
2.4
2.2
1800
1850
1900
1950
2000
2050
2100
Participant-6


3.3
3.2
3.1
3.0
2.9
2.8
2.7
2.6
1800
1850
1900
1950
2000
2050
2100
Participant-7


−0.4
−0.6
−0.8
−1.0
1800
1850
1900
1950
2000
2050
2100
Participant-8


2.4
2.2
2.0
1.8
1.6
1.4
1800
1850
1900
1950
2000
2050
2100
Participant-9


3.4
3.2
3.0
2.8
2.6
1800
1850
1900
1950
2000
2050
2100
Participant-10

METHODOLOGIES:

Feature Creation and Selection:

Since we have inertial signals (raw signal data) in our raw data, over different sensors from 5 different positions which were collecting data at the same time, we have taken mean of equivalent axis values for each sensor from each 5 positions. For example, there are in total 5 x-values of Accelerometer sensors (one from each sensor). We have taken their mean and got first column (Acc_x) of our mediatory data. In this way we get 9 columns (acc_x, Acc_y, Acc_z, Lin_x, Lin_y, Lin_z, Gyro_x, Gyro_y, Gyro_z).

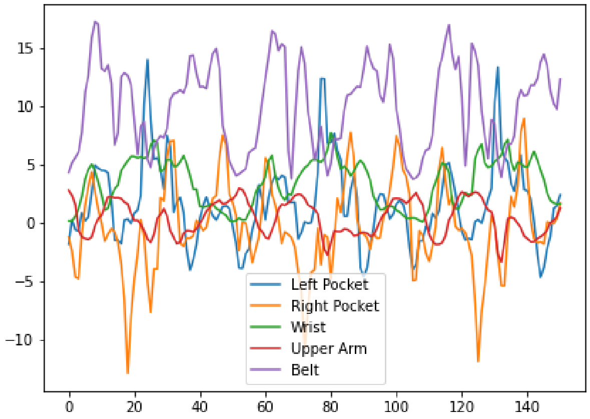


Fig: (a) x-values of accelerometer in 5 different sensors (for participant-1)

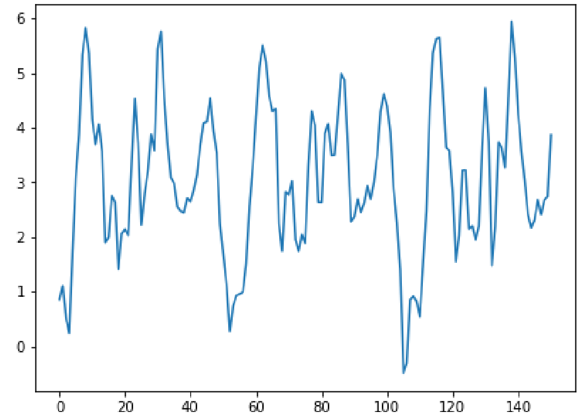


Fig: (b) Mean value of x-values of 5 accelerometers (for participant-1)

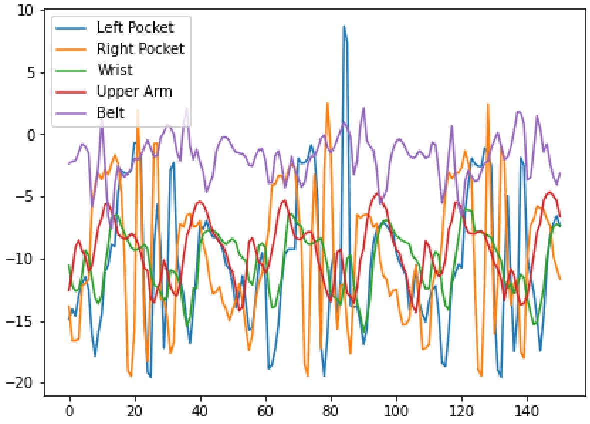


Fig: (a) y-values of accelerometer in 5 different sensors (for participant-1)

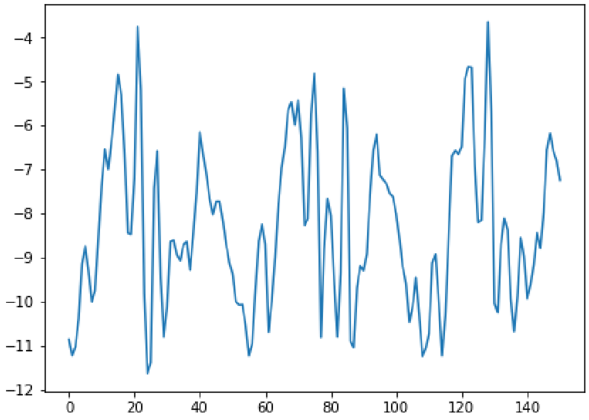


Fig: (b) Mean value of y-values of 5 accelerometers (for participant-1)

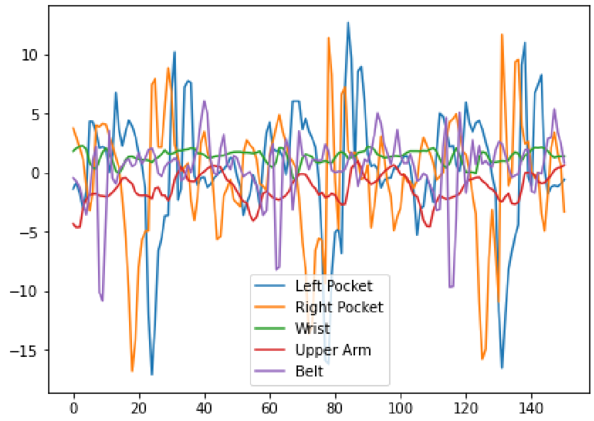


Fig: (a) z-values of accelerometer in 5 different sensors (for participant-1)

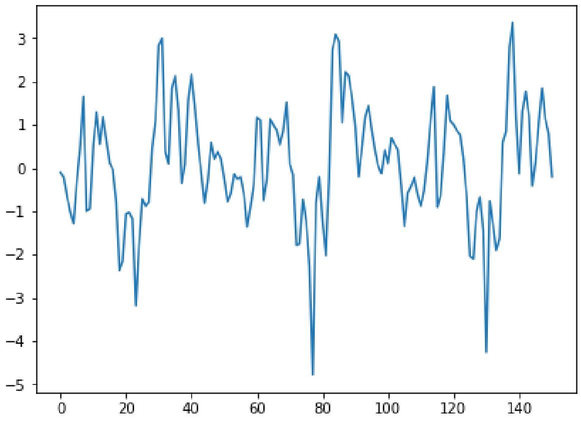


Fig: (b) Mean value of z-values of 5 accelerometers (for participant-1)

Now, we have sampled sensor signals in fixed width sliding windows of 3 seconds and with a 20% overlap in windows. i.e, 150 records per window with 60 records overlapping with adjacent windows (since the signals were initially sampled in the rate of 50 record per second). First, several statistical measures (mean, standard deviation, kurtosis, skewness etc.) were calculated on time domain signal data. Then these signals are transformed to frequency domain from time domain using Fast Fourier Transformation (FFT). Statistical measures (same as before) are again calculated from this transformed data. 6 other features are calculated from indices of minimum and maximum values of each window from time domain and frequency domain data. Altogether we create 336 new features from the raw data. We are using only these features for our project. (All features are provided in feature.txt file)

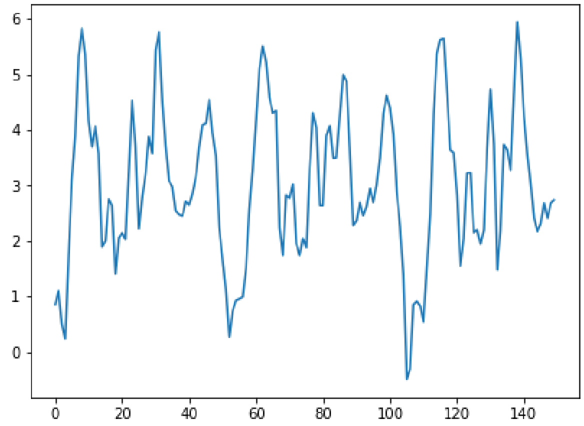


Fig: (a)

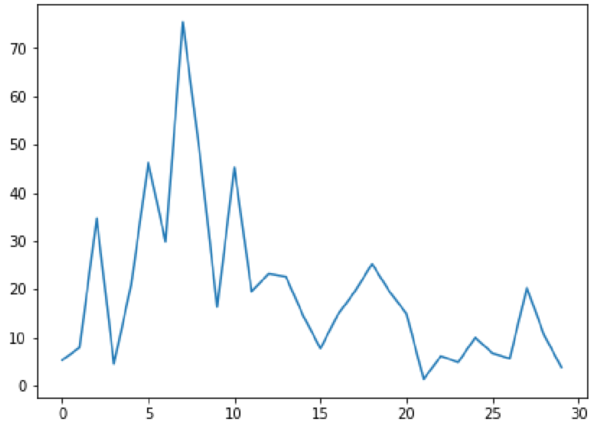


Fig: (b)

In above diagram Fig: (a) and Fig: (b) shows visual representation of time domain signal and transformed frequency domain signals for accelerometer data for participant-1 for first 150 samples.

Out of the 10 datasets, we use the dataset of last 2 participants for out-of-time prediction. From the remaining 8 datasets, we are using first 4 participants data for our training purpose and rest datasets are set aside for validation purpose (in order to tune the hyperparameters of our models and to select the optimal model). For each of our 4 training datasets, first we have splitted them into train-test data in 70:30 ratio randomly (since each datasets are perfectly balanced).

We have scaled the data using Min-Max scaling technique over the training data first.

Feature selection is performed using 4 feature selection criteria;

1.Constant feature

No constant features (zero variance) were found.

2. High Correlated features

Correlation among features are found and we have removed features having high correlation(>0.95) with other features.

Features removed due to high correlation (>0.95)

Dataset Name	Number of features removed
p1_feature	167
p2_features	156
p3_features	158

p4_features	164

3. Mutual information criteria

It is a quantity that measures how much one feature can tell us about another feature.

$$I(X ; Y) = H(X) - H(X|Y)$$

Where, I(X ; Y) is the mutual information between two variables.

H(X) is the entropy of X and H(X|Y) is the conditional entropy of X given Y.

Feature with very low mutual information are removed. we have fixed the threshold at 0.15 .

Features removed due to lower mutual information

Dataset Name	Number of features removed
p1_feature	32
p2_features	40
p3_features	32
p4_features	22

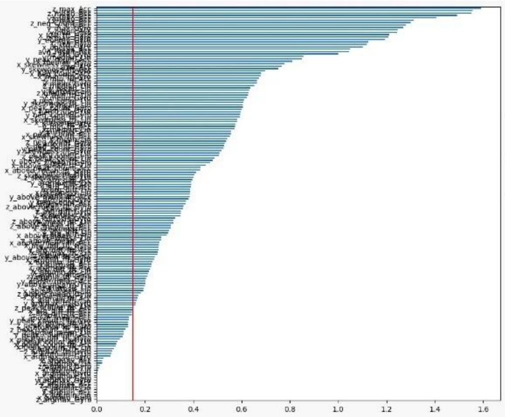


Fig 1.1

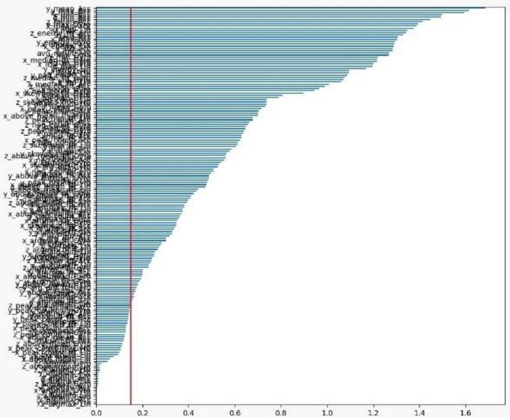


Fig 1.2

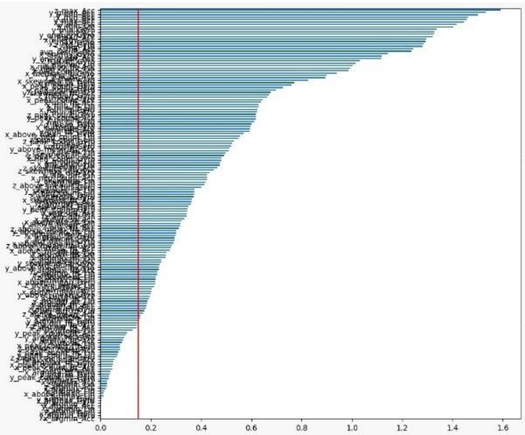


Fig 1.3

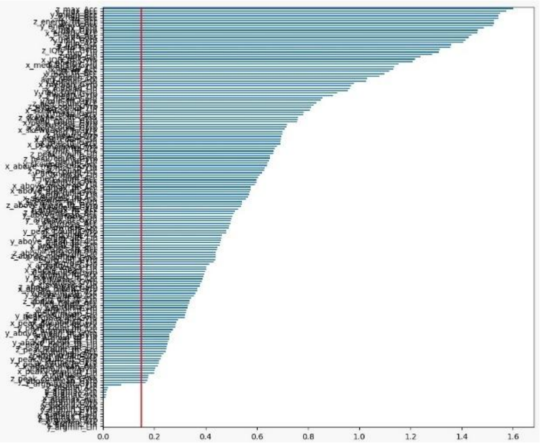


Fig 1.4

Figure 1.1,.1.2 , 1.3 and 1.4 shows the mutual information of each feature in decreasing order.
Red line indicates the threshold value (0.15) for mutual information

4. Extra Trees classifier Score

The higher the extra tree classifier score, the more important the feature. Hence features having score below a threshold values are removed. We are using 0.005 as threshold here.

Features removed due to low extra tree classifier score

Dataset Name	Number of features removed
p1_features	12
p2_features	10
p3_features	7
p4_features	13

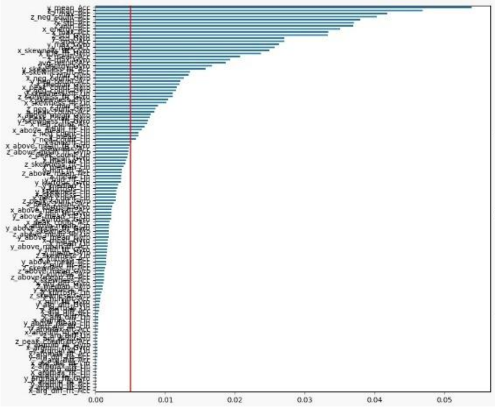


Fig 2.1

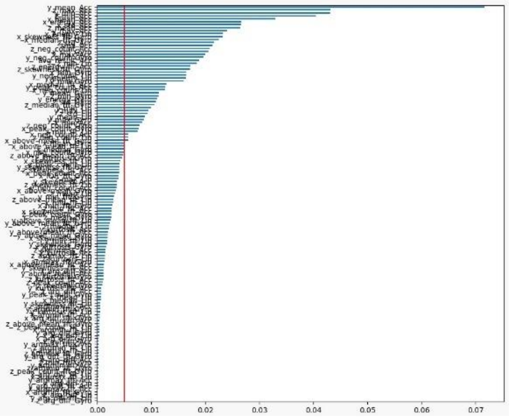


Fig 2.2

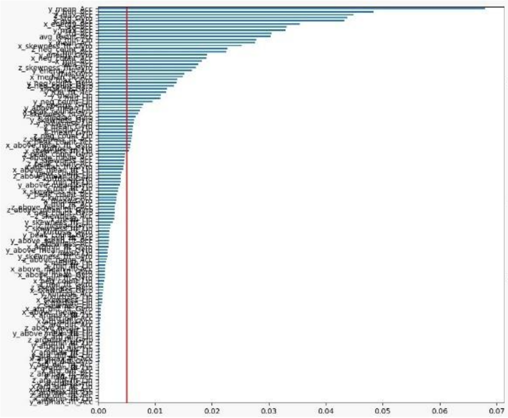


Fig 2.3

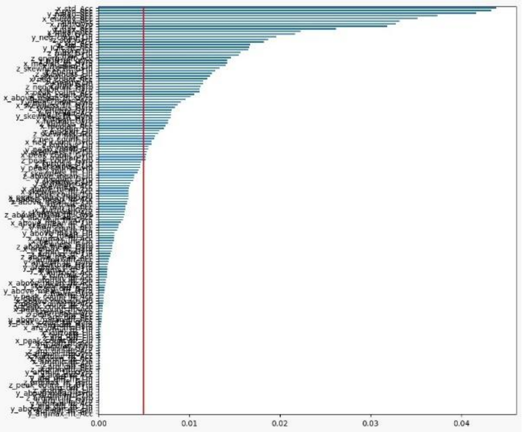


Fig 2.4

Figure 2.1,2.2 ,2.3 and 2.4 shows the extra trees classifier score of each feature in decreasing order

Red line indicates the threshold value (0.005) for extra trees classifier score .

Finally, 124, 129, 138 and 136 are number of features used for training of first four datasets respectively.

Before going into model fitting part of project, Let's discuss some algorithms in brief which we have used:

Logistic Regression:

Usually logistic regression is usually used for classification of two classes. Although there are some clever extensions of this . when we have more than two classes, some extensions like one-vs-rest first transform multi-class classification problems into multiple binary classification problems._then train it on each binary classification problem and predictions are made using the model that is the most confident. It assumes that each classification problem is independent. For fitting the model using python , the LogisticRegression class can be configured for one-vs-rest by setting the "multi_class" argument to "ovr*"* and the "*solver*" argument to a solver that supports it, such as "lbfgs"

An alternate approach involves changing the logistic regression model to support the prediction of multiple class labels directly. Specifically, to predict the probability that an input example belongs to each known class label. A logistic regression model that is adapted to learn and predict a multinomial probability distribution is referred to as Multinomial Logistic Regression. It is an extension of logistic regression that adds native support for multi-class classification problems. It involves changing the loss function to cross-entropy loss. For fitting the model here, we have to change the "multi_class" argument to "multinomial" in the above defined class.

Support Vector Machine:

SVM is a supervised machine learning algorithm that helps in both classification and regression problem statements. It tries to find an optimal boundary (a.k.a hyperplane) between different classes. In simple words, SVM does complex data transformations depending on the selected kernel function, and based on those transformations, it aims to maximize the separation boundaries between your data points If the data is perfectly linearly seperable, then we use linear kernel. If not, we use other kernels such as ‘rbf’, ‘polynomial’ etc. It maximizes the separation between two classes. The data points which are at the minimum distance to the hyperplane i.e, closest points are called Support Vectors. SVM works well for any kind of data .

SVM is by default is fit for binary classification. It is extended to Multiclass classification For multiclass classification, by breaking down the multi-classification problem into smaller subproblems, all of which are binary classification problems. Optimal type of kernel is chosen using hyper parametre tuning.

Random Forest:

Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.Decision tree with their maximum depth overfits the training data. At the same time when we reduce depth, it gives high bias as well. In order to tackle this issue we use ensemble method using decision trees and then decide the class of input by majority voting, i.e, the class selected by most trees. In this way we get low variance and low bias as well. Random forest works well for many kinds of data.

Performance metrics for multiclassification:

All of the metrics you we use are associated with confusion matrices in one way or the other. While a 2 by 2 confusion matrix is intuitive and easy to understand, larger confusion matrices can be truly confusing.

The **precision** is calculated by dividing the true positives by the sum of true positives and false positives. **Recall** is calculated by dividing the number of true positives by the sum of true positives and false negatives. Due to their nature, precision and recall are in a trade-off relationship. You may have to optimize one at the cost of the other. However, what if you want a classifier that is equally good at minimizing both the false positives and false negatives. This is where the **F1 score** comes in. It is calculated by taking the harmonic mean of precision and recall and ranges from 0 to 1.

$$F1\,Score = \frac{2 * (precision * recall)}{precision + recall}$$

Model Fitting:

On the train set of each participant, we fit three different classifiers such as Logistic Regression, SVM and Random Forest with different parameters. Optimal models are found for each dataset using hyperparameter tuning on training data using 5 fold cross validation technique.
Scaled test data using the min-max values that were found previously in the training data and dropped the features which were found irrelevant during training the data. Then, each models with optimal parameters obtained from training are tested on the test data (remaining 30%) of each participant separately.
Next, we have taken datasets over the activities of participants 5,6,7 and 8. Performed 70:30 split, Min-Max scaling on 70% data. Features that are found irrelevant in atleast one of the participant in previous training is removed. Now we have 88 features for all the datasets in this stage. Then we have made list of parameter values for each parameter for each algorithm as we obtained from each training cases. Then we have checked for each algorithm seperately which parameter set is working better than other combinations . Here all the average training and average testing accuracies from 4 datasets were approximately equal(approximately 99% for SVM and Random Forest and 57% for Logistic Regression). So there is no overfitting problem present in the model.
(We are considering the mean accuracy from 4 datasets).

Finally applied voting classifier with Logistic Regression with parameters C=15, max_iter = 50, multi_class='multinomial',solver ='lbfgs' ; SVM with parameters C=1000,gamma=0.1,kernel='rbf' ; Random Forest Classifier with parameters n_estimators=400 ,min_samples_split=2, min_samples_leaf=2, max_features='sqrt', max_depth=10, criterion='entropy' (parameter names are written according to sklearn package of python programming language).

We applied this final model on out-of-time dataset. (from participant 9 and 10) to check the performance.

Results:

Out of time data accuracy for final model for data 9 = 98.33%
Out of time data accuracy for final model for data 10 = 94.37%

Classification report (Precision, Recall, F1 Score) for participant 9

Class Labels	Precision	Recall	F1 Score	support
0	1.00	1.00	1.00	298
1	0.97	1.00	0.86	300
2	1.00	1.00	1.00	300
3	0.99	0.98	0.98	300
4	0.98	0.99	0.99	300
5	0.95	0.97	0.96	300
6	1.00	0.95	0.97	297

Confusion matrix for participant 9:

	Pred 0	Pred 1	Pred 2	Pred 3	Pred 4	Pred 5	Pred 6
Actual 0	298	0	0	0	0	0	0
Actual 1	0	300	0	0	0	0	0
Actual 2	0	0	300	0	0	0	0
Actual 3	0	4	1	294	1	0	0
Actual 4	0	0	0	4	296	0	0
Actual 5	0	0	0	0	1	290	1
Actual 6	0	0	0	0	0	14	282

Classification report (Precision,Recall, F1 Score) for participant 10

Class Labels	Precision	Recall	F1 Score	support
0	1.00	0.99	0.99	298
1	0.99	1.00	0.99	300
2	1.00	1.00	1.00	300
3	1.00	0.95	0.98	300
4	0.94	1.00	0.97	300
5	0.76	0.99	0.86	300
6	1.00	0.67	0.80	297

Confusion matrix for participant 10:

	Pred 0	Pred 1	Pred 2	Pred 3	Pred 4	Pred 5	Pred 6
Actual 0	295	3	0	0	0	0	0
Actual 1	0	299	1	0	0	0	0
Actual 2	0	0	300	0	0	0	0
Actual 3	0	1	0	286	13	0	0
Actual 4	0	0	0	0	300	0	0
Actual 5	0	0	0	0	2	298	0
Actual 6	0	0	0	0	3	95	288

Conclusion: In our whole project we have performed some data analysis over human activity recognition and done feature extraction from the raw data, feature selection from all created features, finally ended with fitted model having quite high accuracy on out-of-time-data.

References:

https://www.utwente.nl/en/eemcs/ps/research/dataset/