Machine Learning Engineer Nanodegree

Additional Report

BOUCHKARA Abdelilah

July 26th ,2018

Additional Report

Before explaining the signal processing pipeline, I will first explore in details the Two original datasets called HAR Dataset and HAPT dataset to justify the production of datasets type I and II explored in the capstone report using RawData signals only located in V2 folder.

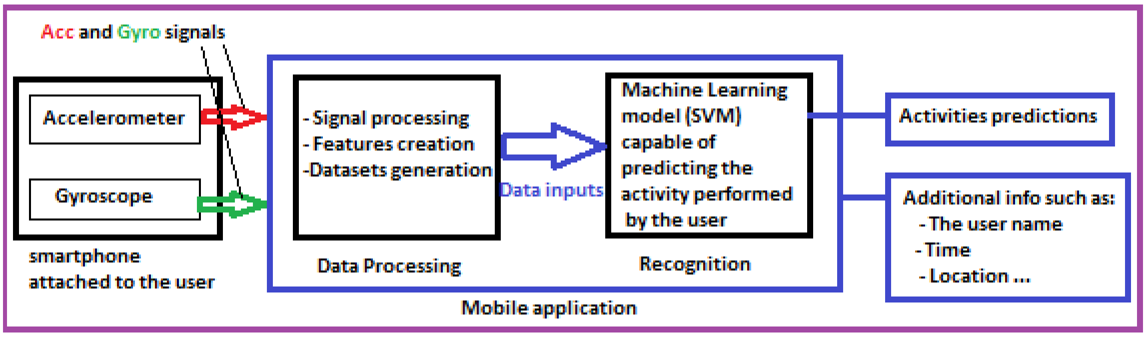
I. Project Origins:

HAR system name: **Smartphone Based Human Activity Recognition**

Objective: Detecting and Recognizing human daily activities using smartphone sensors (accelerometer and gyroscope) and informing remote monitoring systems about patient's activities in real time.

Project owners: **Jorge Luis Reyes Ortiz [1]**

Jorge Luis Reyes Ortiz during in his PHD research [2] has developed a mobile application containing this HAR system capable of real-time detection and recognition of basic activities and postural transitions (see the figure bellow) using a smartphone device attached to users.



The HAR system working in Online Mode

Human Daily Activities predicted by this HAR system are:

- Six Basic Activities (BAs):
  - Three Dynamic Activities:
    - 01) Walking (When the user is Walking)
    - 02) Walking up stairs (When the user is Walking up stairs)
    - 03) Walking down stairs (When the user is Walking down stairs)
  - Three Static Activities:
    - 04) Sitting (user in a sitting position)
    - 05) Standing (user in a standing position)
    - 06) Lying down (user in a lying down position)

- Six Postural Transitions (PTs):
  - 07) stand-to-sit (when the user is moving from a standing position to a sitting position)
  - 08) sit-to-stand (when the user is moving from sitting to standing)
  - 09) sit-to-lie (when the user is moving from sitting to lying down)
  - 10) lie-to-sit (when the user is moving from lying down to sitting)
  - 11) stand-to-lie (when the user is moving from standing to lying down)
  - 12) lie-to-stand (when the user is moving from lying down to standing)

When a smartphone is attached to Human body, activity's motion of this body will have an effect on the smartphone acceleration and rotation. As these quantities can be measured and captured by the device's accelerometer and gyroscope it is possible to describe and recognize the activity performed by analysing these quantities called **inertial raw signals**.

A multitude of transformations will be applied to these signals to build relevant features of each type of motion. These features will be fed to the recognition part (see the figure above) to predict daily activities performed by users.

In online mode the recognition part contains a machine learning model (already trained) capable of predicting performed activities with high accuracy. This HAR system can transfer activity predictions with some additional information such as the user's Identifiers to other applications within the same device or externally located through wireless communications.

II. Datasets and inputs:

This HAR system has an offline part(mode). Its goal is to build the optimal machine learning model mentioned in recognition part (figure above). To produce this optimal model a data collection of raw inertial signals with enough users should be done. A signal processing procedure and other Data processing steps will be applied to produce a clean and huge dataset. Dataset inputs (features and targets) will be fed to variety of machine learning algorithms this time not for predicting activities performed by users but for training and evaluating the algorithms to produce that final model.

To achieve this task the project owners have already organized an experiment to obtain HAR and HAPT datasets:

II.1. The experiment:

They carried out a set of experiments with a group of 30 volunteers in laboratory conditions volunteers age is within 19-48 years. The volunteers were asked to perform freely a sequence of the activities mentioned earlier in order to simulate a more naturalistic dataset. Each volunteer performed the protocol of activities twice which means two experiments per user. In the first type of experiments the smartphone (Samsung galaxy S2) was fixed on the left side of the waist of user's body the phone always faced the same direction with respect to the case in order to maintain the same orientation of the triaxial sensors coordinate axis. On the second type of experiments the smartphone was placed by the user himself as preferred. All the experiments were recorded on video with the consent of the participants in order to facilitate activities labelling tasks. The video recording was done with the video camera of another smartphone at a frame rate of 30 Hz.

A normal question that may come to the reader of this report is: Why this HAR system use complex quantities such acceleration and rotation which are not easy interpret by the human eye?

Why not simplifying things and use cameras to survey users and use computer visions techniques to recognize activities using videos and/ or photos captured as a source of data? It won't be less complex, clear and simple for machines as for humans?

Many reasons are behind not to choose cameras as activity detectors. The objective of this HAR system is to improve patient's quality of living by sending trustful data to other monitoring systems without embarrassing patients with camera surveillance.

As part of the Definition of Activity Recognition field **[3]**: "Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions" (Wikipedia).

From the definition above Agents in this case are humans (Human Activity Recognition). the sensors used to collect the series of observations should not influence agents' behaviour. Agents should act as natural as possible to collect naturalistic data. Using cameras as sensors in online mode will influence the patient's behaviour this may lead to wrong predictions. During the conducted experiments (in offline mode) project owners were forced to use a camera video to record all experiments to facilitate the activity labelling. Since the protocol of activities performed by volunteers was already defined (semi-controlled experiments) using camera to record performed activities won't have much influence on their behaviour compared to the protocol itself.

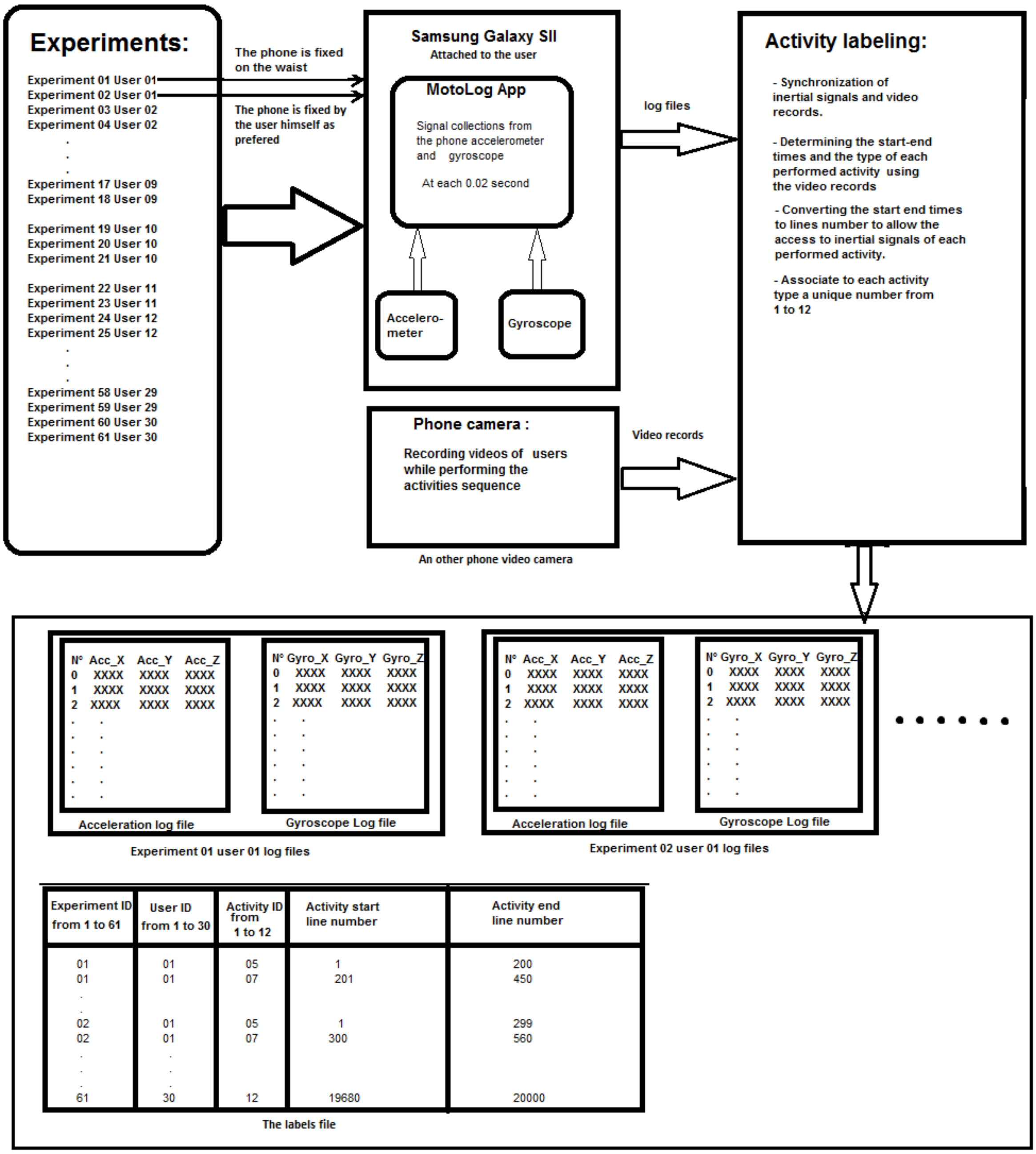
Every mechanic motion can be described by accelerations and rotations. These quantities captured by the smartphone sensors are more than sufficient to collect necessary information about patient's activities.

II.2 Data collection:

Jorge Luis Reyes Ortiz had developed an android application called **MotoLog** [2] **(figure bellow),** to capture and store signals coming from the inertial sensors (accelerometer and gyroscope) In log files, while volunteers were performing the protocol. **This application is capable to capture signal's values at each 0.02 second (frequency=50hz) 50 values of each axial inertial signal per second.**

II.3 Activities Labelling:

Once data collection is done. Log files and video records were synchronized to allow the activity labelling by setting manually the activity Ids and the start-end times of each performed activity. These start-end times can be transformed to line numbers knowing that the duration between two successive rows in the same log file; containing inertial signals; is 0.02 second.



Data collection and labeling in offline mode

The project owners have already published two versions of this dataset:

The first version **(V1 folder)** named as UCI HAR Dataset doesn't contain the original log files. It contains only some semi-processed basic inertial signals of basic activities from 1 to 6 **(Inertial signals folder inside train and test folders).** It doesn't contain postural transitions (activities from 7 to 12). It contains also the final datasets already processed and splatted into train and test files. The target files (**y_train and y_test**) also contain activity labels ids from 1 to 6.

The second version **(V2 folder)** called HAPT Dataset which is more useful actually it contains the original 122 log files and the labels file of all experiences and activities. These files are stored in a folder called **RawData inside HAPT Dataset folder**. This version contains also the final datasets (train and test files) which are inside the train and test folders.

Both versions contain some additional text files: **features names, activity labels, features info, and README file.**

Train and test files from both versions were obtained from the same original log files located in **RawData** folder in **V2**. But these log files were processed in two different ways to produce V1 and V2 train and test files.

Table of files and folders included in UCI HAR Dataset

<table>
<tr><th></th><th colspan="2">File or folder name</th><th>Details</th></tr>
<tr><td rowspan="12">-
UCI HAR Dataset (V1)</td><td rowspan="4">Train (Folder)</td><td>Subject_train.txt</td><td>This file contains user identifier of each row of features used in training (21 unique user identifier) 7352 user id labels in total</td></tr>
<tr><td>X_train.txt</td><td>This file contains data points used for related to train users:
- 7352 rows in total
- 561 features or columns in total:
  - 265-time domain features
  - 289-frequency domain features
  - 7 additional features</td></tr>
<tr><td>Y_train.txt</td><td>- This file contains: activity labels of each row of features used in training.
Activity labels used in this dataset are from 1 to 6
Obviously, it contains the same number of rows as X_train.txt</td></tr>
<tr><td>Inertial Signals (folder)</td><td>Each file in this folder contains one semi processed basic signal for user’s ids used in training.
Each row contains 128 values were captured in a chronological order</td></tr>
<tr><td rowspan="4">Test (Folder)</td><td>Subject_test.txt</td><td>This file contains the user identifiers used for testing. (9 unique user identifiers) 2947 user id labels in total.</td></tr>
<tr><td>X_test.txt</td><td>This file includes rows of features related user identifiers used for testing.
561 features or columns in total:
2947 rows</td></tr>
<tr><td>Y_test.txt</td><td>Contains activity labels related to each row in X_test.txt
Activity labels are from 1 to 6.
Obviously, it contains the same number of rows as X_test.txt</td></tr>
<tr><td>Inertial Signals (folder)</td><td>The same as inertial signals folder mentioned above the only difference is that these files are related to user’s ids used in testing</td></tr>
<tr><td colspan="2">activity_labels.txt</td><td>It maps the name of each activity to its identifier</td></tr>
<tr><td colspan="2">features.txt</td><td>It Includes the names of all 561 features used in X_train and X_test</td></tr>
<tr><td colspan="2">features_info</td><td>Explains how signals and features were obtained</td></tr>
<tr><td colspan="2">README.txt</td><td>Contains general information about this dataset</td></tr>
</table>

HAR Dataset (V1) link:
https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

Use of this dataset in publications must be acknowledged by referencing the following publication [4]

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Table of files and folders included in HAPT Dataset

	File or folder name		Details
HAPT Dataset (V2)	RawData (Folder)		This folder includes 122 logfiles mentioned in the figure above: 61 experiences in total 2 files per experience (1 acc file and 1 gyro file) - Each Acceleration file contains 3 columns acc[X, Y ,Z] - Each gyroscope file contains 3 columns gyro[X, Y, Z] It includes also labels.txt which contains activity labels of each capture recorded in all experiments.
	Train (Folder)	Subject_Id_train.txt	This file contains user identifier of each row of features used in training (21 unique user identifier) 7767 user id labels in total.
		X_train.txt	This file contains data points used for related to train users: - 7767 rows in total - 561 features or columns in total: - 265-time domain features - 289-frequency domain features - 7 additional features
		Y_train.txt	- This file contains: activity labels of each row of features used in training. Activity labels used in this dataset are from 1 to 12 Obviously, it contains the same number of rows as X_train.txt
	Test (Folder)	Subject_Id_test.txt	This file contains the user identifiers used for testing. (9 unique user identifiers) 3161 user id labels in total.
		X_test.txt	This file includes rows of features related user identifiers used for testing. 561 features or columns in total: 3161 rows
		Y_test.txt	Contains activity labels related to each row in X_test.txt Activity labels are from 1 to 12. Obviously, it contains the same number of rows as X_test.txt
	activity_labels.txt		It maps the name of each activity to its identifier
	features.txt		It Includes the names of all 561 features used in X_train and X_test
	features_info		Explains how signals and features were obtained
	README.txt		Contains general information about this dataset

HAPT Dataset (V2) link: https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

Use of this dataset in publications must be acknowledged by referencing the following publications

- Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, Davide Anguita. Transition-Aware Human Activity Recognition Using Smartphones. Neurocomputing. Springer 2015.

The figure bellow is summarizing the general process of this HAR system in offline mode and explaining the meaning of each file and folder included in Both Datasets V1 and V2:

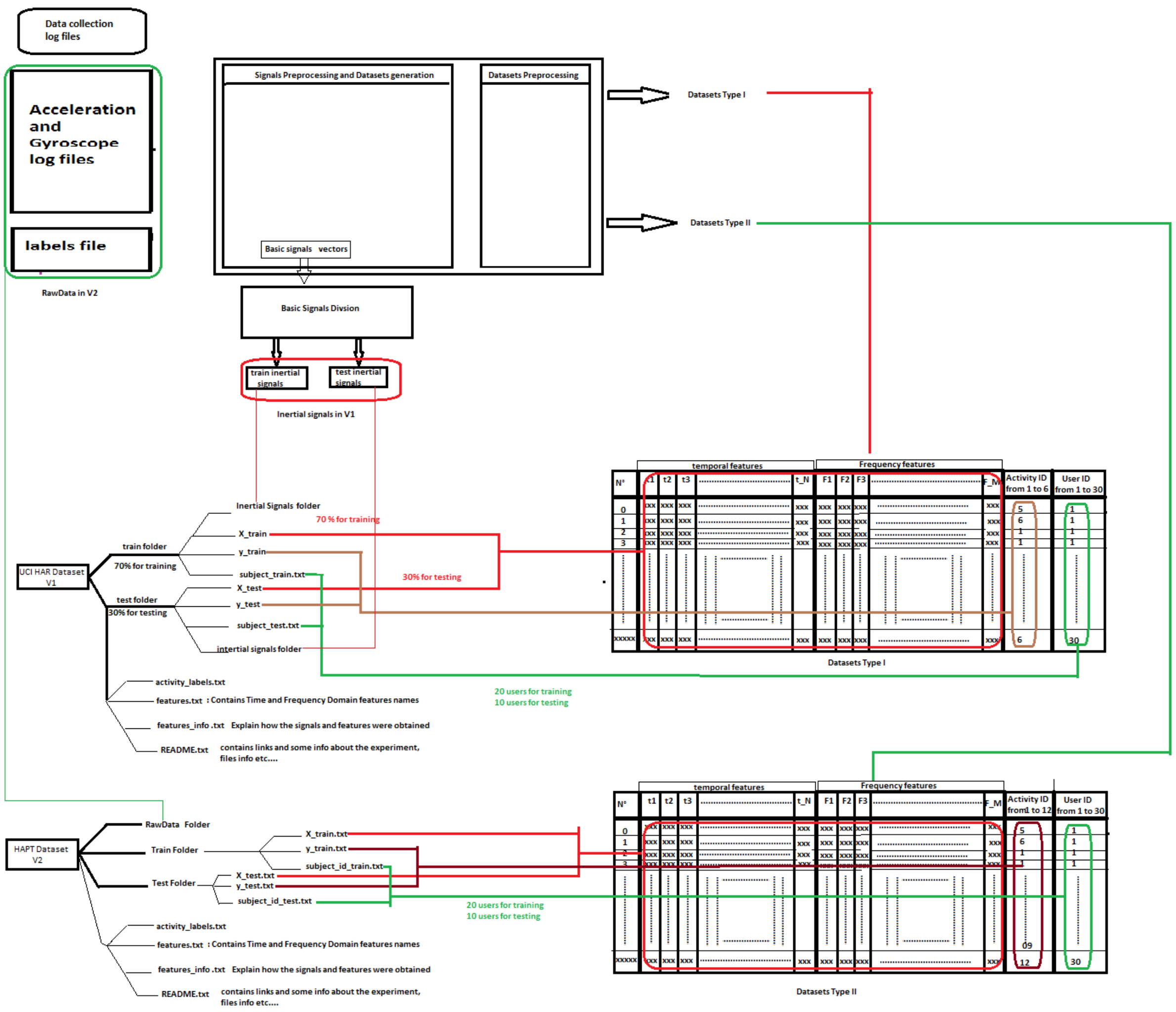


Figure: Original Data Architecture

As It appears they achieved great results in their machine learning research **[4]** using supervised learning approach with SVMs algorithms as an optimal model not only for its high accuracy level but also for its low computation cost as this system will be used on smartphones.

III. RawData Exploration and Exploratory Visualization:

In this part, I will explore and present all files used as inputs of the **signal processing pipeline**:

RawData is a folder located in HAPT Dataset V2. It includes 122 log files of 61 experiences and 1 labels file.

In total we have 61 experiences [exp ids from 1 to 61]. Two log files per each experience:

- One Acceleration file named as **acc_expXX_userYY.txt :**

'acc': is for acceleration

'XX': is the exp id (between 1 and 61)

'YY': is the user identifier (between 1 and 30).

- One Gyroscope file named as **gyro_expXX_userYY.txt :**
  - **'gyro':** is for gyroscope
  - **XX:** is the experience identifier (between 1 and 61)
  - **YY:** is the user identifier (between 1 and 30).

Each file is composed of rows and columns. The number of rows(captures) depend on the number of captures and duration of each experience.

acc and **gyro** files from the same experience (having the same exp_Id) will have the same number of rows. The actual difference in time between each two successive rows in the same file: is 0.02 seconds since the recording frequency of the phone sensors is 50 captures per second (50Hz).

Each log file contains 3 columns:

- **Acc** file columns are: **[acc_X, acc_Y, acc_Z] or acc[X,Y,Z]** each column contains float acceleration values of its axis in 'g's (gravity of earth) which is equal to: 9.80665 meter/second^2 (m/s^2 is the universal unit used for acceleration measures).

- **Gyro** file columns are: **[gyro_X, gyro_Y, gyro_Z] or gyro [X, Y, Z]** each column contains float gyro values which are actually the rotation speed called also angular velocity of its axis during the experiment. Units used for measuring the gyroscope values is radian/second (universal units).

	acc_X	acc_Y	acc_Z
0	0.918056	-0.112500	0.509722
1	0.911111	-0.093056	0.537500
2	0.881944	-0.086111	0.513889

	gyro_X	gyro_Y	gyro_Z
0	-0.054978	-0.069639	-0.030849
1	-0.012523	0.019242	-0.038485
2	-0.023518	0.276417	0.006414

The tables above represent the first 3 rows of acc and gyro files of the 1st experiment.

These files were already verified no missing values are present.

The labels file: This file includes all activity labels of each row in all logfiles.

It contains: XXX row and 5 columns

	experiment_number_ID	user_number_ID	activity_number_ID	Label_start_point	Label_end_point
0	1	1	5	250	1232
1	1	1	7	1233	1392
2	1	1	4	1393	2194

The first 3 rows in labels file

For example, the meaning of the first row (row 0):

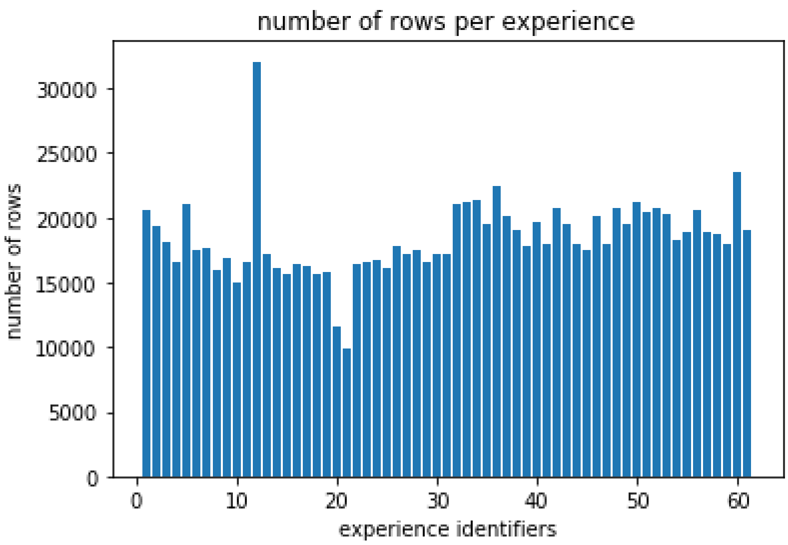
In acc_exp01_user01.txt and gyro_exp01_user01.txt:

Rows having indexes between [250 and 1232] (1232 included) are related to activity ID number 05 which is standing.

250 is the labels starting point of the activity number 5 Performed by the user 01 in the experience 01.

1232 is the label end point of the activity number 5 Performed by the user 01 in the experience 01.

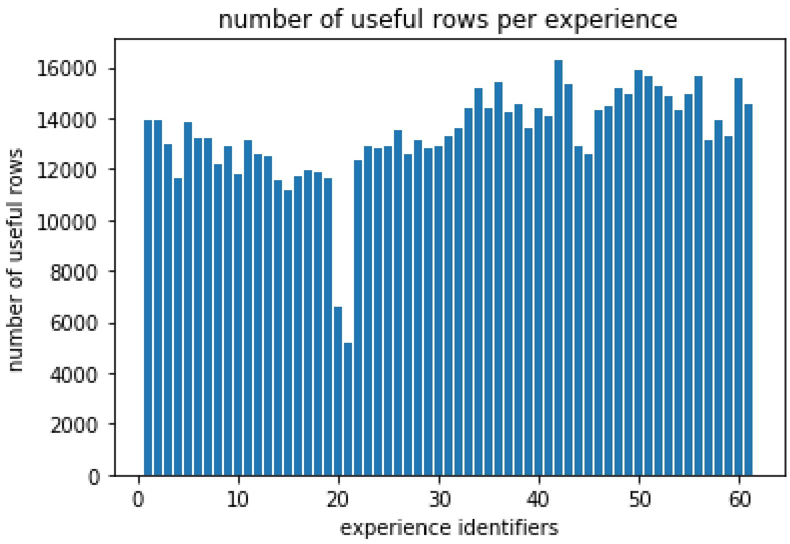
III.1. General Visualizations:



The figure above represents the number of captures (rows) for each experience. It appears that:

- The majority of logfiles have between 15000 and 20000 captures in totale.
- The experience N° 12 has more than 30000 captures . This may be justified by the presences of captures without activity ids (non useful captures) where the volunteer N° 6 was not performing any activity.
- The volunteer N° 10 has conducted 3 experiences exp 19 exp 20 and exp 21 . The number of rows in exp 20 and 21 is less 13000 captures per file. We can assume that exp 20 and exp 21 logfiles concatenated actually represents 1 experience .

To have a clear idea about the number of useful rows (having an activity Id) see the figure bellow:

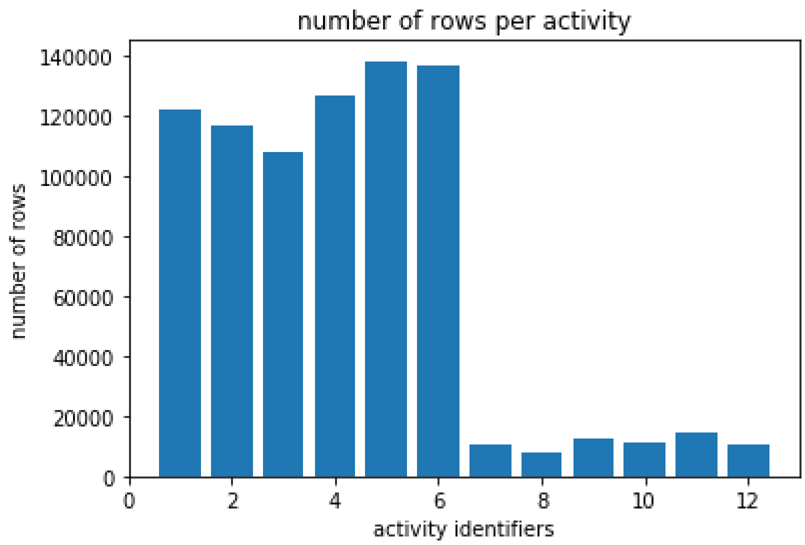


In this figure It appears clearly that number of captures is between 11000 and 16000 rows for the majority of experiences .Except The two experiences 20 and 21 ,they have less 8000 captures each. This reinforces hypothesis mentioned earlier.

The number of captures in general is enough to record all activities. In duration terms the majority of experiences have a **useful durations** between : 220 and 320 seconds which I think is enough to contain all activities in the protocol. The figure bellow shows the number of rows per activity in all records. Basic activities (activity Ids from 1 to 6) have the majority of rows since they have the first priority in this study. Postural Transitions have low number of rows due to the short duration of the transition from a static state to anther static state.

Basic Activities contains at least more 10000 capture per second for all users which I think is enough in terms of datapoints number will be fed to machine learning algorithms.

Basic activities columns are approximately balanced if we ignore postural transitions(from 7 to 12). In a nutshell ,This Raw Dataset is unblanced Basic activties and postural transitions should be treated differently.



The figure bellow shows more detailed info about each activity performed in all experiences. It shows the mean duration of each activity type. For example if all users performs activity N° 2 three times during each experience. The mean duration here is sum of all durations related to this activity devided by the number of times this activity was performed in the protocol in all experiences.

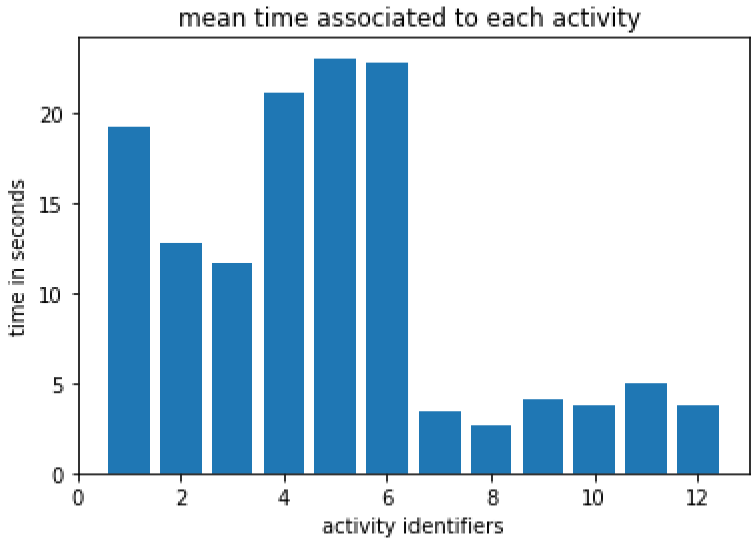
Mean_D =(N_rows X 0.02)/ (183)

Mean_D: Mean duration of activity number 2 in seconds .

N_rows: number of rows having activity id N° 2 in all experiences.

0.02 : The duration between two successive rows in seconds

183 =3*61: number of times this activity was performed in the protocol of all experiences



From the figure above, it appears that:

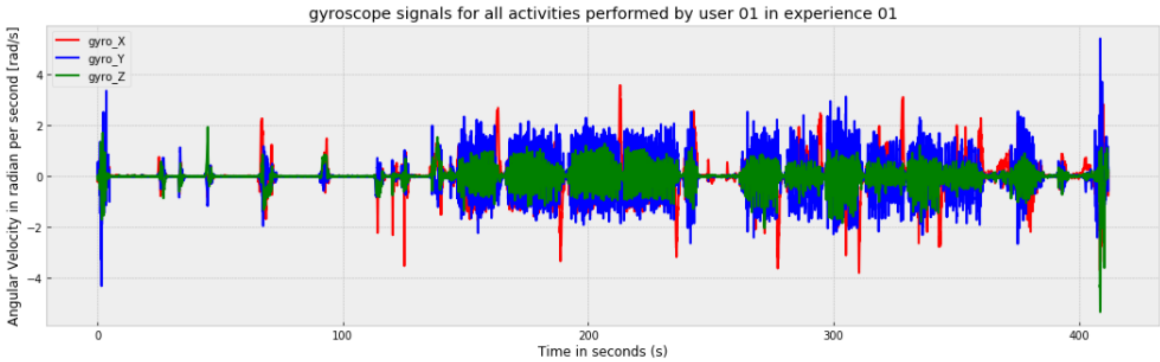
- Activities: 1(walking) ,4 , 5 and 6 (are static postures) have a mean duration near to 20s.
- Activities 2 and 3 (walking upstairs and Walking down stairs) have approximately 12s as a mean duration due to the short path of upstairs.
- For postural transitions: all postural transitions durations are less than 5 seconds they have approximately the same duration mean.

III.2. Detailed visualizations:

The figure bellow represents acceleration signals of the 1st experience.

Acc values values varie approximately from -1g to 2g. During static postures acc signals should be constant. During dynamic activities and postural transitions acc signals will varie with a high speed.

The figure bellow shows the gyro signals variation during the same experiment. Gyro values are approximately between -4 rad/second and 4 rad/second



During static postures acc and gyro signals should be constant. During dynamic activities and postural transitions gyro signals will varie with a high speed.

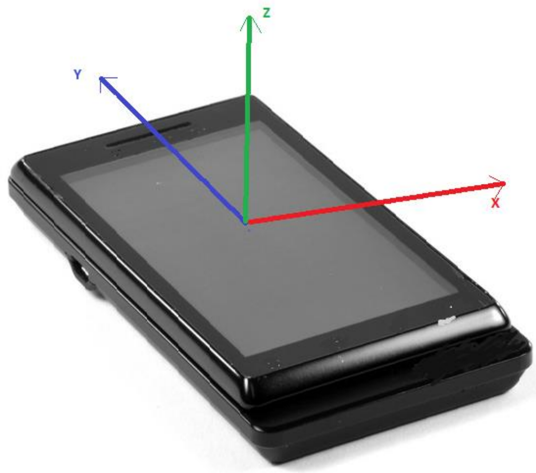
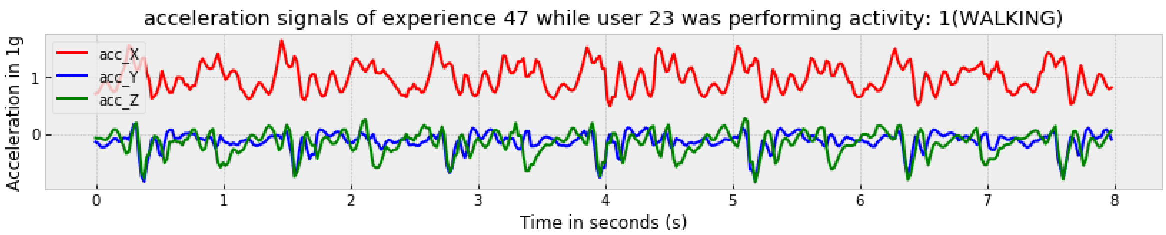
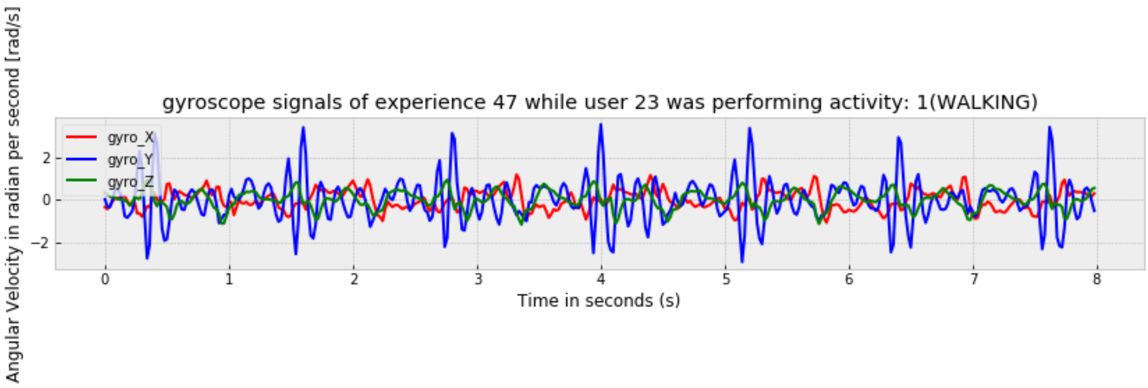


Figure: Axis Directions



The figure above represents acc signals captures while the user 01 in experiment 01 was walking. Signals are periodic which reflects the periodicity of walking. Acc y and acc Z means are near or bellow 0g. acc_X mean is near to 1g this can be justified by the phone orientation.

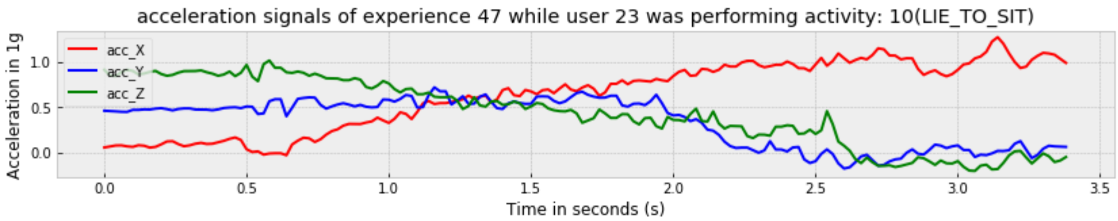
X axis of the phone have the same direction and the same orientation as the earth gravity force which is approximately equal to 1. acc_Y and acc_Z are not perpendicular exactly on the gravity force during walking as a result the gravity force will have negative projections on these axes.

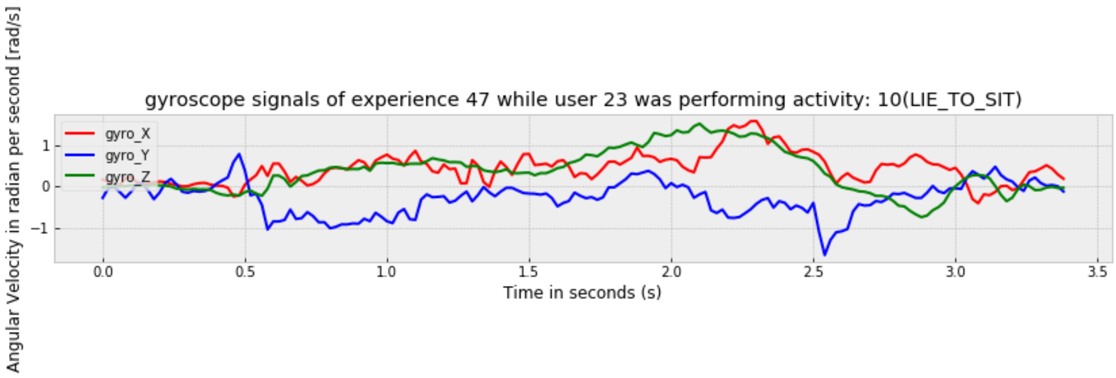


For gyro signals the figure above shows gyro signals related to this activity. X and Y axis mean values are near to 0. The angular velocity of Z axis varies periodically with a high speed from -3 rad/s to 3 rad/s. The Z axis is the axis perpendicular on the phone screen it's normal that its angular velocity varies the most due the fact of moving legs during walking.

For postural transitions the most important thing is the start and end points values of each **acc** signal. Since the gravity force projections on each axis will vary from static posture to another static posture.

As said before the **Z_axis** is perpendicular on the phone screen and oriented to the user's body. When the user is lying down this axis will have a positive orientation as the gravity force. This will justify first values at the beginning of the transition below. At the end of this transition which is Sitting (a static posture) the **X axis** will have the same orientation as the gravity force which justifies **acc** values at the end of this transition.





Gyro signals are hard to analyse during the transition but the start points values of all axis and end point values of all axis should be near to 0 rad/s since it is a transition from a static posture to another static posture.

Signal processing steps:

Inputs: **raw_dic**, a python dictionary contains 61 data frames: keys are: **expXX_user_YY.**

signal_X, signal_Y ,signal_Z columns are notated as : **signal[X,Y,Z]**

Each data frame contains six columns: **acc [X, Y, Z] and gyro [X, Y, Z]**.

I- Generating all time domain signals data frames:

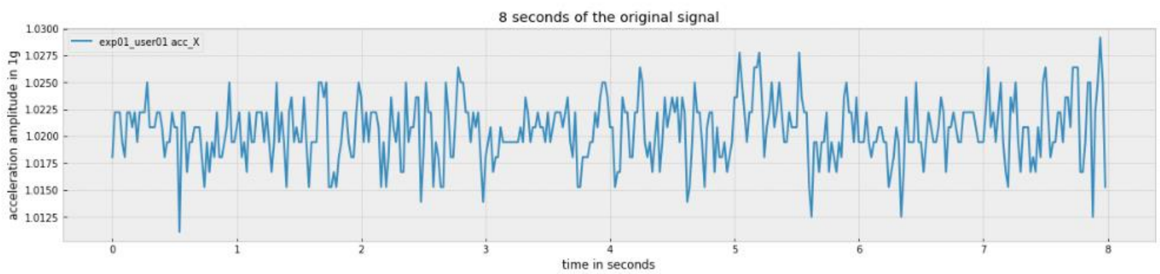
1- Median filter (3^{rd} order): function name: median_filter.

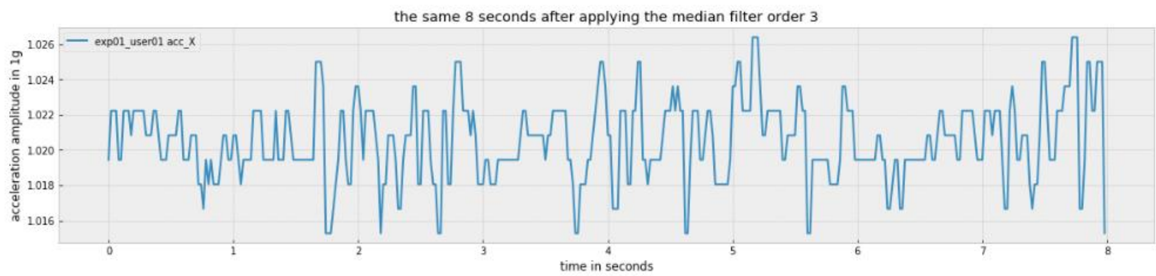
Inputs: **1column (axial component)**.

Median filtering process: for each value in this 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**
- **The second has the same value as the first one**
- **The third value in the output column is the median of 2^{nd} 3^{rd} and 4^{th} value in the original column**
- **The before last values is the median of [L-2,L-1,L]**
- **The last value in the output column is the same as last value in the original column (won't be changed)**

Output: med_filtred is median-filtered signal (1D column having the same length as the input column).





- **From the two figures above, it appears clearly the effect of the median filter on the original signal. Some peaks were cut using the median value (fast ups and downs can be considered as a background noise [5]. The role of the median filter is to reduce this noise as much as possible.**

2- Useful component selection:

function name: components_selection_one_signal

Inputs: **med_filtered signals.**

components_filtering process: This function converts each time domain signal (1D column each time) to frequency domain using **Fast Fourier Transform** function.

The selection of useful components is done based on frequency ranges (see figure). after that a reverse Fast Fourier Transform is applied to useful components to obtain these signals in time domain.

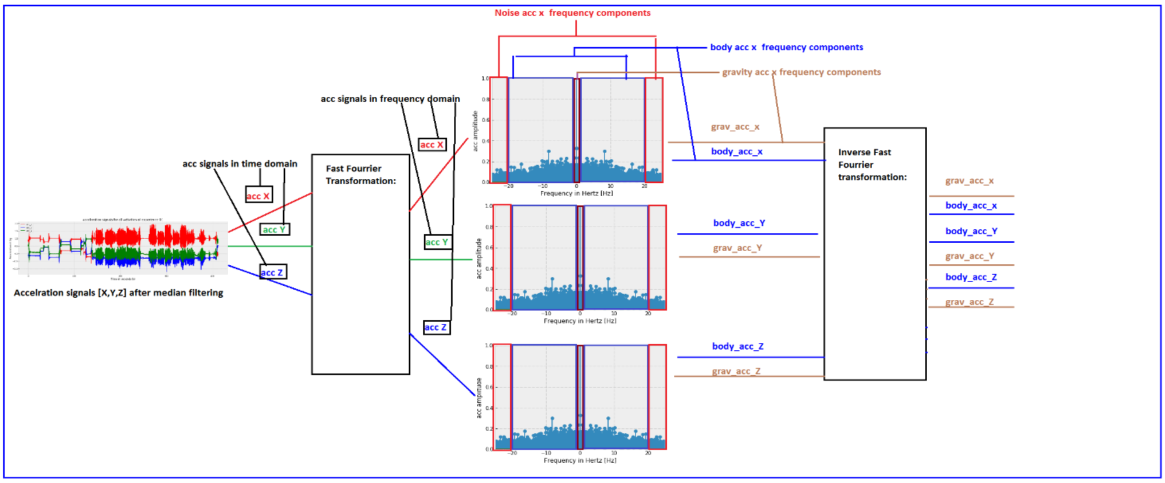
Outputs: **total_component , DC_component , body_component, noise**.

If the input signal is an acceleration signal **(acc_X or acc_Y or acc_Z)** already filtered using **med_filter**: **useful components are** :

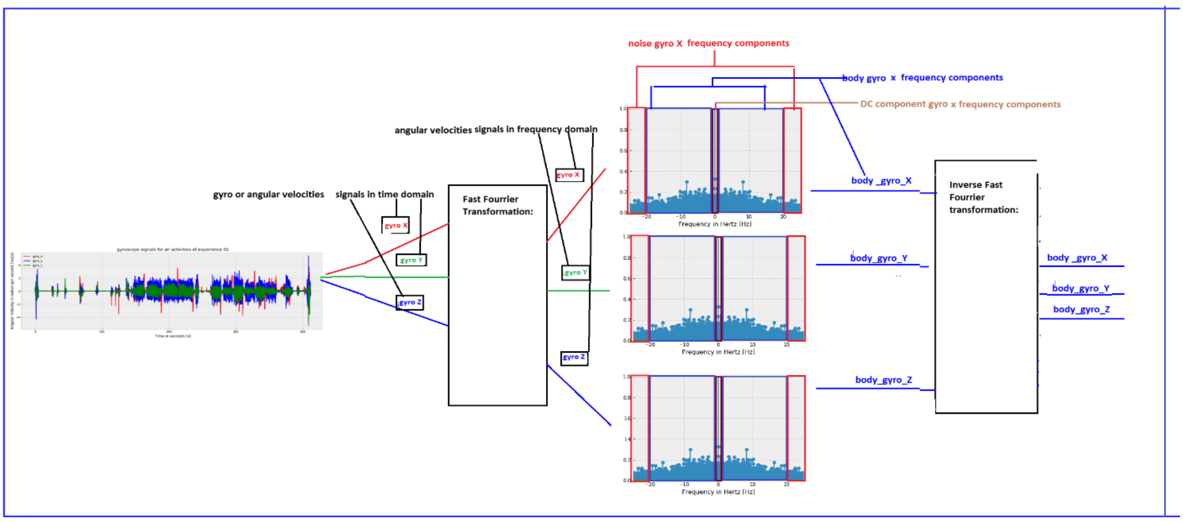
- **DC_component:** will be considered as gravity component **Grav_acc_[X or Y or Z]**
- **Body_component:** will be considered as the acceleration of the body: **Body_acc [X or Y or Z]**

If the input signal is a gyro signal (gyro[X or Y or Z]) already filtered using med_filter the useful components are:

- **Body_component: will be considered as the angular velocity caused by the user's body: Body_gyro[X or Y or Z]**



Acceleration Filtering

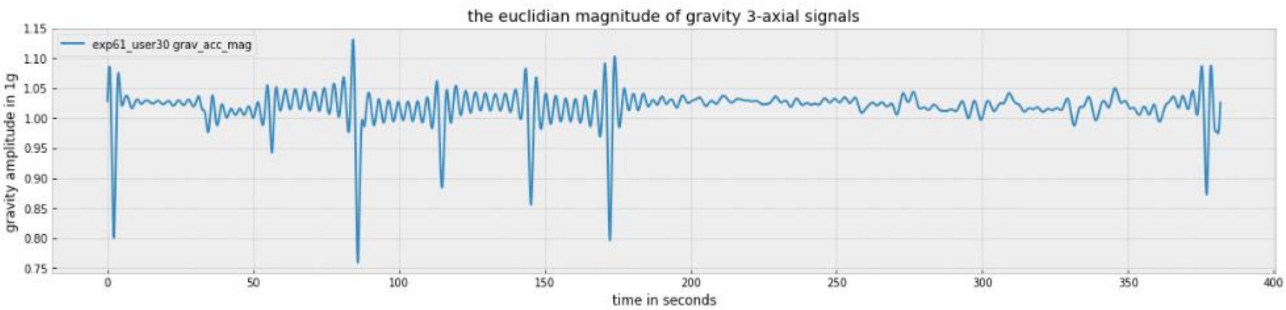


Angular velocities Filtering

Verification:

During the experiments the gravity magnitude (gravity of earth) is approximately constant since all experiments were carried out in the same place (laboratory). As a result, the gravity magnitude signal should be approximately equal to 1 for the majority of data points (since the unit used to measure acceleration is 1g).

I choose to generate **gav_acc[X,Y,Z]** using acceleration signals related to the experiment 61 performed by the user number 30(the last data frame in **raw_dic**). After calculating the **Euclidian magnitude** of these tri-axial signals I obtained the **grav_acc_mag** related to this experiment (see figure below).



From the figure above, it appears clearly that **grav_acc_mag** is approximately near to **1g** for the majority of data points. This proves that component selection was done correctly. The reader is invited to use the **Jupyter Notebook related to signal processing** to verify if the gravity magnitude is near to **1g** using other data frames.

Frequency Ranges:

DC_Component is generated from signal components having frequencies between: [-0.3Hz,0.3Hz]
Body_Component is generated from signal components having frequencies between: [-20Hz,0.3Hz[⋃]0.3Hz,20Hz]
Noise component is generated from signal components having frequencies between: [-25 Hz,-20Hz[⋃]20Hz,25Hz]
Total component = med_filtred signal(inputs) - Noise(high frequency)

Frequency ranges justification:

- As they mentioned in **README File (HAPT Dataset)**. The gravitational force is assumed to have only low frequency components as result they chose to select components having frequencies less than **0.3 Hz(|freq|<= 0.3 Hz; |X| is the absolute value of X)**.
- Since 99% of body motion's energy is contained below 15Hz [4], selecting components having absolute frequencies between **]0.3Hz, 20Hz]** as body motion's components is more than enough to conserve body motion's components properties.

Sampling Frequency Justification:

- Why **50hz** is enough to conserve the majority of signals properties?

When a real signal is captured using a sampling frequency **F in Hz** (F captures per second). The reconstitution of the same original signal (the real signal) is impossible (see Information theory Field **[6]**). We could only reconstitute components having Frequencies less or equal to the Nyquist frequency (Sampling Frequency F/2).

In our case since 99% of body motion's energy is contained below 15Hz **[4].** Choosing a sampling frequency **F=50Hz**, will allow the reconstitution of all signal components having frequencies less or equal to **Nyquist frequency equal to 25 Hz** in our case.

3- Jerking function: function name: jerking_function

Useful inputs are**: body_acc [X or Y or Z] or body_gyro [X or Y or Z].**

jerking_function process: It returns **time derivatives** of each body component.

The derivation formula:

signal_jerk [row i] = (signal [row i+1] – signal [row i]) /0.02

0.02s : is the time duration between two sequential values(1/50Hz) in a column.

The length of the output column = length of input column -1 (the last value of the input column cannot be jerked)

Output: **signal_jerk (1D array)**

Output names could be : body_acc_jerk[X or Y or Z] or body_gyro_jerk[X or Y or Z].

4- Magnitude function: function name: mag_function

After generating all 15 axial signals **(see figure bellow)** we will apply the **mag function** to each tri-axial signals **signal[X,Y,Z]** to generate magnitude signal: **Signal_mag**

t_body_acc_X	t_body_acc_jerk_X
t_body_acc_Y	t_body_acc_jerk_Y
t_body_acc_Z	t_body_acc_jerk_Z
t_grav_acc_X	
t_grav_acc_Y	t_: is for time domain
t_grav_acc_Z	
t_body_gyro_X	t_body_gyro_jerk_X
t_body_gyro_Y	t_body_gyro_jerk_Y
t_body_gyro_Z	t_body_gyro_jerk_Z

Axial signals generated for each raw_dic dataframe

Inputs: all triaxial signals mentioned before**: Body_acc[X,Y,Z],Body_gyro[X,Y,Z], Grav_acc_[X,Y,Z], columns body_acc_jerk[X,Y,Z] and body_gyro_jerk[X,Y,Z]**

Mag_function process: It calculates the Euclidian magnitude of each **3-axial columns** of the same **signal type** and it returns **one magnitude column**.

Euclidian magnitude formula:

$$\text{Signal\_mag[row i]} = \sqrt[2]{X^2 i + Y^2 i + Z i^2} \qquad \begin{array}{l} Xi : \text{signal\_X[row i]} \\ Yi : \text{signal\_Y[row i]} \\ Zi : \text{signal\_Z[row i]} \end{array}$$

Output: signal_mag (1D array) it could be:

body_acc_mag ,body_gyro_mag,Grav_acc_mag,body_acc_jerk_mag or,body_gyro_jerk_mag.

5- Generate all time domain signals data frames

Inputs: all useful signals (20 signals) per data frame

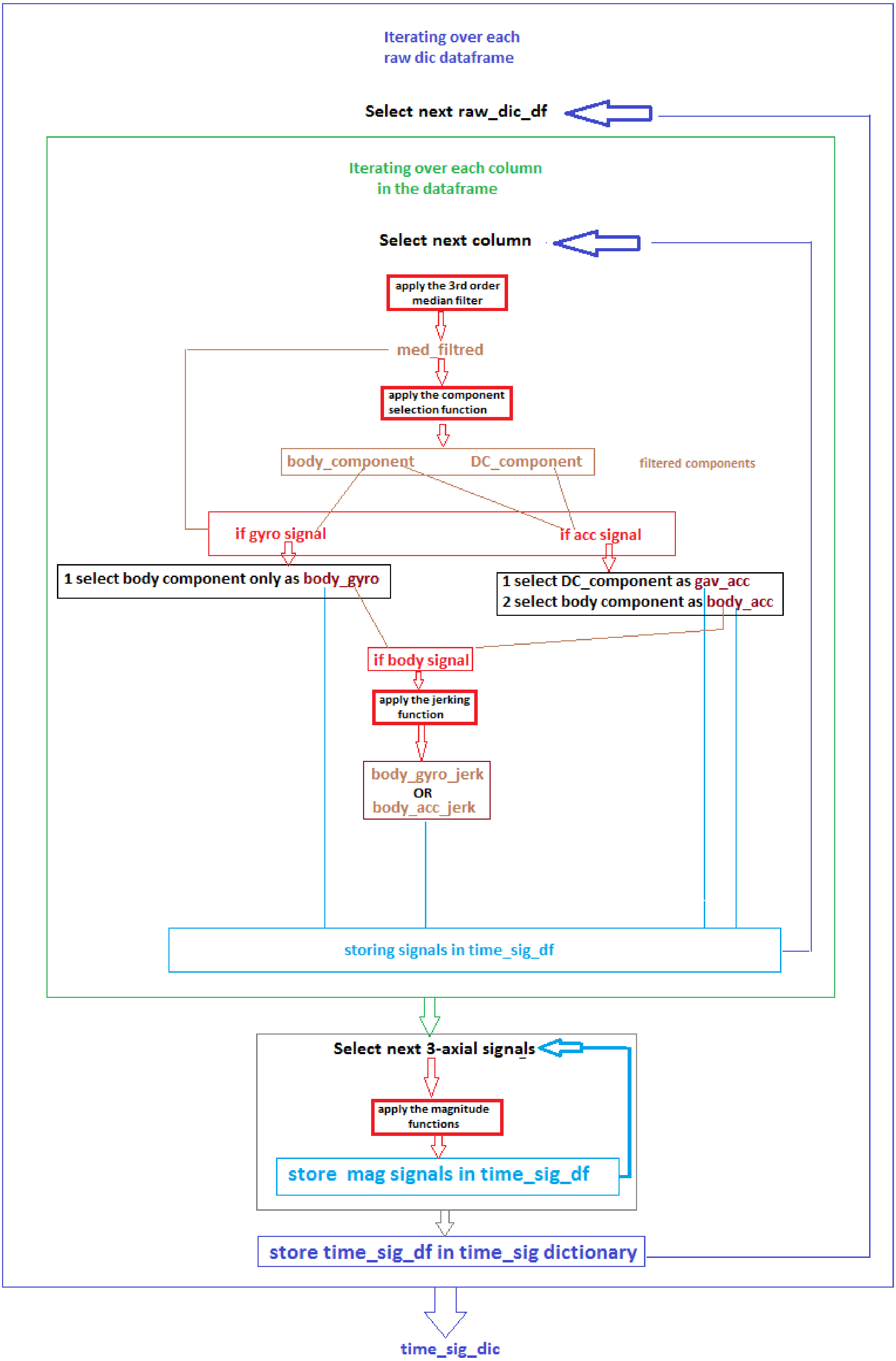
Output: **time_sig_dic** is a dictionary contains 61 data frames **(time_sig_Df),** keys are: **expXX_userYY** each **time_sig_Df** contains 20 signals (15 triaxial signals and 5 magnitude columns):

time_sig_dic = { 'expXX_userYY' → time_sig_Df }

Tri-axial signals are: Body_acc[X,Y,Z], Grav_acc_[X,Y,Z], Body_acc_jerk[X,Y,Z] , Body_gyro[X,Y,Z], Body_gyro_jerk[X,Y,Z]

Magnitude columns are: Body_acc_mag , Grav_acc_mag, Body_acc_jerk_mag, Body_gyro_mag, and Body_gyro_jerk_mag.

Generation Process:



II- Windowing:

As mentioned in README files time data frames in **time_sig_dic** will be sampled in fixed-width sliding windows of **0.02 x 128=2.56 sec** and **50% overlap** (128 readings or rows per window).

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.

Windowing type I:

After selecting the data frame to be windowed the windowing procedure mentioned above will be applied only to rows related to basic activities (activity ids from 1 to 6). These rows can be selected using the labels files which contains detailed info about each capture in all experiences.

Let say that a group of rows in the selected data frame belongs to the activity N° 1. The windowing procedure will be applied multiple times to this group of rows producing small data frames each one contains 128 rows and 20 columns. At the beginning of each procedure the algorithm will verify if the number of rows to be windowed is at least 128. If not, the procedure won't be applied.

As a result, all windows will contain 128 rows and all rows in a window will have the same activity Id which is considered as **the window's activity Id**.

Windowing type II:

After selecting the data frame to be windowed. The windowing procedure will be applied to all rows in this data frame (no selection by activity id). By the end of each procedure a voting function will be applied to each window to determine the **window_activity_Id .**

The voting function rules:

- If the window contains rows related to multiple activities and there is an **activity_id** with more than 65 rows this **activity_id** is chosen to be the window's **activity_id**.
- If each activity id in this window have at most 63 rows this window won't be stored.
- If the number of rows having no **activity_id** (non-useful rows see data exploration section) in this window is at least 65 rows. This window won't have an activity ID and it won't be stored.

Storing:

Windows obtained with windowing type I will be stored in dictionary: **t_dic_win_type_I**

Windows obtained with windowing type II will be stored in dictionary: **t_dic_win_type_II**

Each window should be stored with a unique key (no problem if we have windows from different dictionaries having the same key).

Each key is composed from:

- **the window unique Id** (integer auto increment)
- **the experiment** and **the user ids** obtained from the labels file
- **the activity Id** associated to the window (obtained by voting in windowing type II or from the labels file in windowing type I).

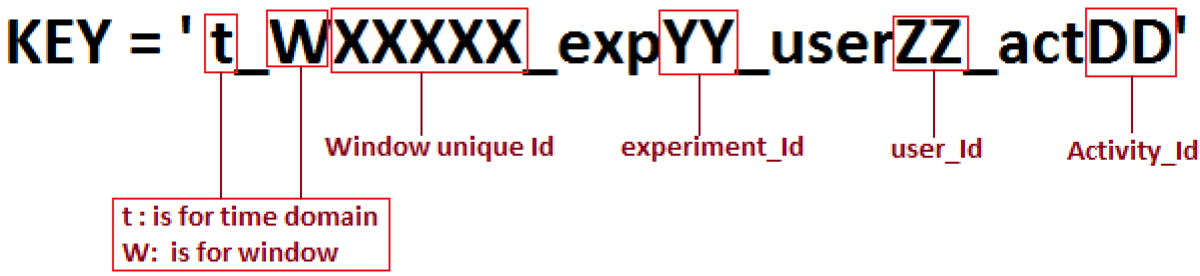


Figure: Time domain windows Keys structure

Frequency windows generation:

By applying Fast Fourier Transformation to each non-gravitation column in all windows for both dictionaries. We obtain frequency domain windows which will be stored in two other dictionaries (f_**dic_win_type_I** and **f_dic_win_type_II**) with the same keys associated to the time domain windows we replace 't' by 'f' **(f is for frequency domain).**

Each window in those dictionaries contains 128 rows and 16 columns.

12 Tri-axial signals: f_body_acc[X,Y,Z], f_body_acc_jerk[X,Y,Z] , f_body_gyro[X,Y,Z], f_body_gyro_jerk[X,Y,Z]

4 Magnitude columns: f_body_acc_mag , f_body_acc_jerk_mag, f_body_gyro_mag, and f_body_gyro_jerk_mag.

III- Features Generation:

From each **(t_window,f_window)** having the same **window_ID** in (t**_dic_win_type_I** , **f_dic_win_type_I**) a row of features will be generated by calculating variables from time and frequency domain To produce features matrix of Dataset type I

The same process will be applied to windows in (t**_dic_win_type_II** , **f_dic_win_type_II**) to obtain features matrix of Dataset type II.

A feature is a function will be applied to one column or 3-axial columns at once to produce (irrespectively) 1 value or multiple values depending on the type of the function.

Features can be categorized as Axial features or Magnitude features. They Can be categorized also as Time domain features and Frequency domain Features.

Axial features: are functions that will applied to each **3-axial signals[X,Y,Z]** at once to produce 1 value or multiple values (see the two figures bellow):

- Common axial features they can be applied only to axial columns from time and frequency domain.
- Time domain axial features: they can be applied only to time axial columns
- Frequency domain features: they can be applied to only to frequency axial columns

Magnitudes features: are functions that will be applied to each magnitude signal **(signalname_mag):**

- Common magnitude features: can be applied only magnitude columns from time and frequency domains.
- Time domain magnitude features: can be applied only to time domain magnitude signals

- Frequency domain magnitude features can be applied only to magnitude signals from time domain.

Features meaning:

mean(): Mean value of one array

std(): Standard deviation of one array

mad(): Median absolute deviation of one array

max(): Largest value in one array

min(): Smallest value one array

sma() and Sma_mag : is the Signal magnitude area (area under the signal magnitude).

energy(): Energy measure. Sum of the squares of one array divided by the number of values.

iqr(): Interquartile range, the value of the third quartile minus the value of the first quartile

entropy(): Signal entropy

arCoeff(): Autoregression coefficients with Burg order equal to 4 (using the burg method)**[7]**

correlation(): correlation coefficient between two signals (person's R coefficient)

maxInds(): the frequency values having largest signal value

meanFreq(): Weighted average of the frequency components to obtain a mean frequency

skewness(): skewness of the frequency domain signal

kurtosis(): kurtosis of the frequency domain signal

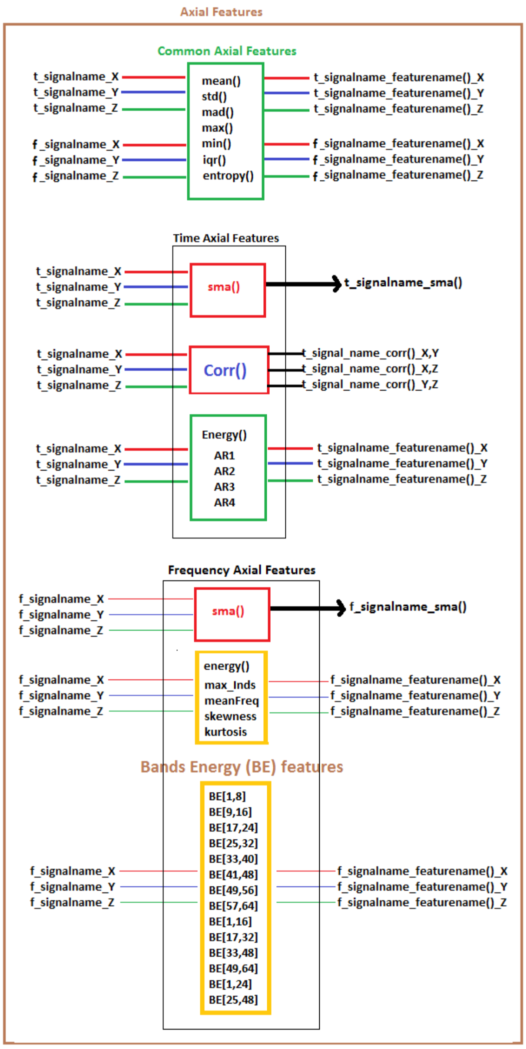
bandsEnergy(): Energy of a frequency interval within the 64 bins(strictly positive frequencies only) of the FFT of each window.

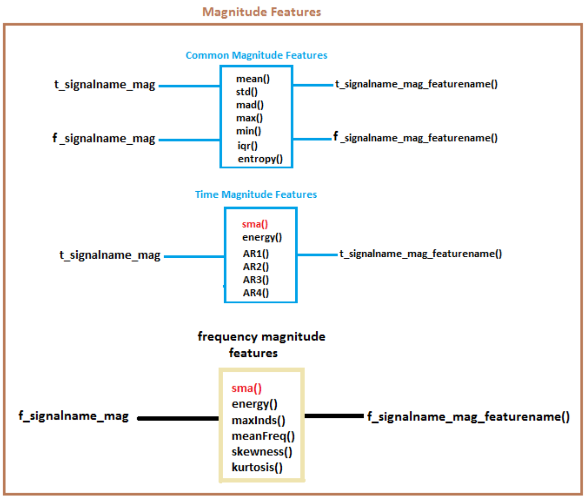
Angle(): Angle between to vectors.

- **Angle(0)**= angle between **t_body_acc[X.mean,Y.mean,Z.mean]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(1**)= angle between **t_body_acc_jerk[X.mean,Y.mean,Z.mean]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(2)** = angle between **t_body_gyro[X.mean,Y.mean,Z.mean]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(3)** = angle between **t_body_gyro_jerk[X.mean,Y.mean,Z.mean]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(4)** = angle between **X_axis =[1,0,0]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(5)** = angle between **Y_axis =[0,1,0]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**
- **Angle(6) =** angle between **Z_axis =[0,0,1]** and **t_grav_acc[X.mean,Y.mean,Z.mean]**

Feature name	Time domain Formula	Frequency Domain Formula	Axial	Mag
Mean	$Mean(S) = \frac{1}{N}\sum_{i=1}^{N} S_i$	Same formula	✓	✓
Standard Deviation	$\sigma(S) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(S_i - Mean(s))^2}$	Same formula	✓	✓
Median Deviation	$MAD(S) = median_i(\lvert S_i - median(S)\rvert) \quad where\ i \in [\lvert 1, N\rvert]$	Same formula	✓	✓
Max	$Max(S) = max_i(S_i), \quad where\ i \in [\lvert 1, N\rvert]$	Same formula	✓	✓
Min	$Min(S) = Min_i(S_i), \quad where\ i \in [\lvert 1, N\rvert]$	Same formula		
Signal Magnitude Area	$SMA(S_1, S_2, S_3) = \frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert S_{i,j}\rvert, \quad where\ 1,2,3\ are\ X,Y,Z\ compoents$	$SMA(S_1, S_2, S_3) = \frac{1}{3\sqrt{N}}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert S_{i,j}\rvert, \quad where\ 1,2,3\ are\ X,Y,Z\ compoents$	✓	X
Signal Magnitude Area	$SMA_{mag}(S_{mag}) = \sum_{i=1}^{N}\lvert S_{mag_i}\rvert$, where S_{mag} is a magnitude column	$SMA_{mag}(S_{mag}) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\lvert S_{mag_i}\rvert$, where S_{mag} is a magnitude column	X	✓
Energy	$Energy(S) = \sum_{i=1}^{N} S_i^2$	$Energy(S) = \frac{1}{N}\sum_{i=1}^{N} S_i^2$	✓	✓
Interquartile Range	$IQR(S) = Q3(S) - Q1(S)$	Same formula	✓	✓
Entropy	$Entropy(S) = \sum_{i=1}^{N}(C_i \log(C_i)), \quad Where\ C_i = \frac{S_i}{\sum_{j=1}^{N} S_j}$	Same formula	✓	✓
AutoRegression Coeficients	$AR_1, AR_2, AR_3, AR_4 = arburg(S, 4), \quad Where\ AR_i \in \mathbb{R}$		✓	✓
Correlation	$Correlation(S_i, S_j) = \frac{C_{i,j}}{\sqrt{C_{i,i}C_{j,j}}}, \quad Where\ C_{i,j} = Cov(S_i, S_j),\ i,j \in \{X,Y,Z\}$		✓	X
Skewness		$Skewness(S) = E\left[\left(\frac{S - Mean(s)}{\sigma(s)}\right)^3\right]$	✓	✓
Kurtosis		$Kurtosis(S) = \frac{E[(S - Mean(S))^4]}{E[(S - Mean(S))^2]^2}$	✓	✓
Max frequency index		$MaxfreqInd(S) = Freq(\arg max_i(S_i))$	✓	✓
Mean Frequency		$MeanFreq = \frac{\sum_{i=1}^{N} freq_i S_i}{\sum_{j=1}^{N} S_j}$	✓	✓
Bands of Energy		$EnergyBand(s, a, b) = \frac{1}{b - a + 1}\sum_{i=a}^{b} S_i^2$	✓	X
Angle	$Angle(S_X, S_Y, S_Z, v) = Cos^{-1}\left(\frac{([mean(S_x), mean(S_y), mean(S_Z)].v)}{\lVert[mean(S_x), mean(S_y), mean(S_Z)]\rVert_2 \lVert v\rVert_2}\right)$		✓	X

Figure: Features' mathematical formulas





For each tuple of dictionaries obtained from the same windowing method:

For each tuple of windows having the same **window_Id** in those dictionaries we obtain:

- Time domain features row is: (**Common axial features** + **Time axial features)** applied to all time domain axial signals in time domain window (**t_window**) and **(Common magnitude features + Time magnitude features)** applied to time domain magnitude columns in the same window.
- Frequency domain features row is: Common Axial Features + Frequency Axial Features applied to all frequency domain axial signals in a frequency domain window(**f_window**) and Common magnitude features + frequency magnitude features applied to all frequency domain magnitude columns in the same window.
- Additional features row: is the angle function applied to 7-time domain signals from **t_window** and will produce a row with 7 inputs
- Two additional inputs: activity Id and the **user_Id** associated to this tuple of windows.

All These Inputs will be combined to obtain one full row.

15 : number of Time axial signals in t_window
12: number of Frequency axial signals in f_window
7 : number of Common axial features
6 : number of Time axial features without sma
5 : number of outputs of sma applied to each 3-axial time signals
19: number of Frequency axial features without sma
4 : number of outputs of sma applied to each 3-axial frequency signals

Time features = 15 x 7 + (15 x 6 + 5) + 5 x 7 + 5 x 6 = 105 + 95 + 35 + 30 = 265

Frequency_features= 12 x 7 + (12 x 19 + 4) + 4 x 7 + 4 x 6 = 84 + 232 + 28 + 24 = 368

5 : number of time magnitude signals in t_window
4 : number of frequency magnitude signals in f_window
7 : number of Common magnitude features
6 : number of time magnitude features
6: number of frequency magnitude features

Additional features= 7

Total = 640 Features

Figure : Row's structure

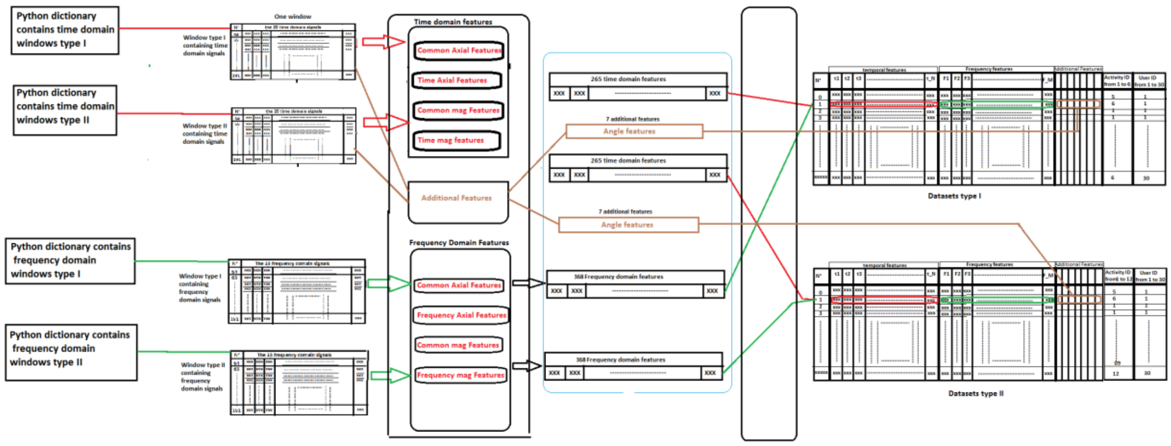


Figure: Features generation Process

By applying these functions to both tuples of dictionaries We obtain the final datasets Dataset type I and Dataset type II.

Dataset type I contains data related to basic activities only (activity Id from 1 to 6)

Dataset type II contains data related to all activities (basic activities and Postural Transitions).

References:

[1]: https://www.icephd.org/wiki/index.php/Jorge_Luis_Reyes_Ortiz

[2]: https://www.springer.com/us/book/9783319142739

[3]: https://en.wikipedia.org/wiki/Activity_recognition

[4] : https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2013-84.pdf

[5] : https://en.wikipedia.org/wiki/Background_noise

[6] : https://en.wikipedia.org/wiki/Information_theory

[7]: http://thomas-cokelaer.info/software/spectrum/html/user/ref_param.html#id15