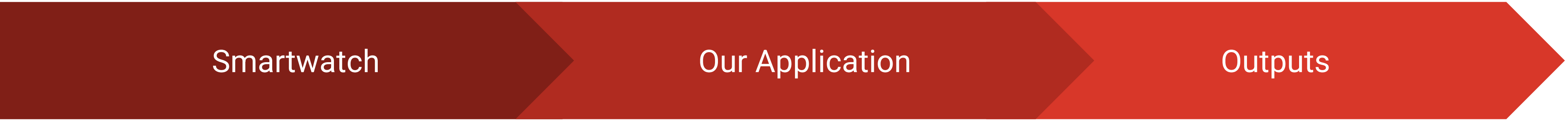


Smartwatch

Our Application

Outputs

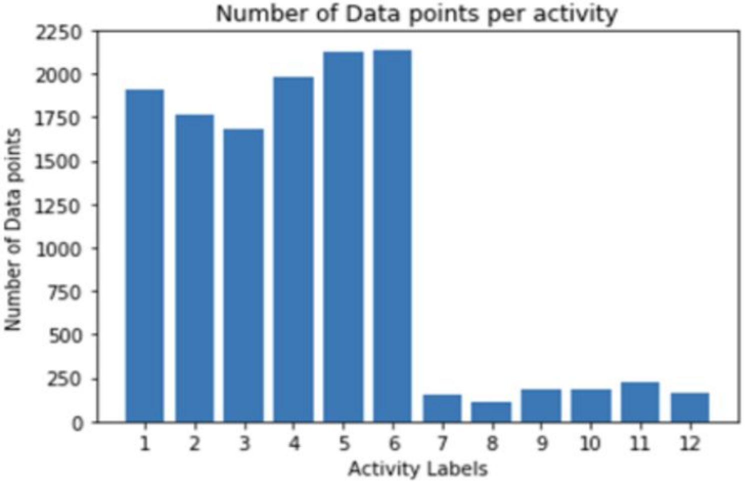
Accelerometer and Gyroscope

Two main: 1. Data Processing
2. Recognition

1. Data processing
   a. Signal processing
   b. Features creation
   c. Dataset generation

2. Recognition:
Inputs from the above are fed into ML model(SVM)

1. Activities prediction



- Applied feature scaling
- Tried decision trees
- Gaussian NB
- logistic regression classifiers
- Next need to work on SVM

1

Generating time domain signals

Median filtering, component selection, Jerks and magnitudes, Time domain signal generation

2

Windowing

Windowing type 1 and type 2, Frequency windows generation

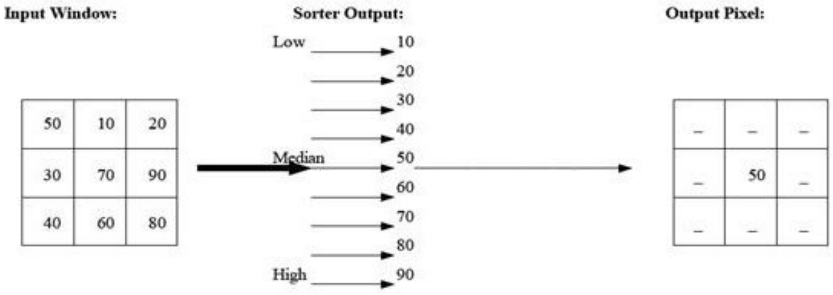
3

Features generation

Step 1: Define fast_fourier_transform functions Step 2: Apply it to one sample Step 3: Generation and Storage of Frequency Windows Discussion

Median Filtering : It helps in **reducing noise** and identifying features in dataset.

Scipy library has the median filter function *medfilt*



Input Window:

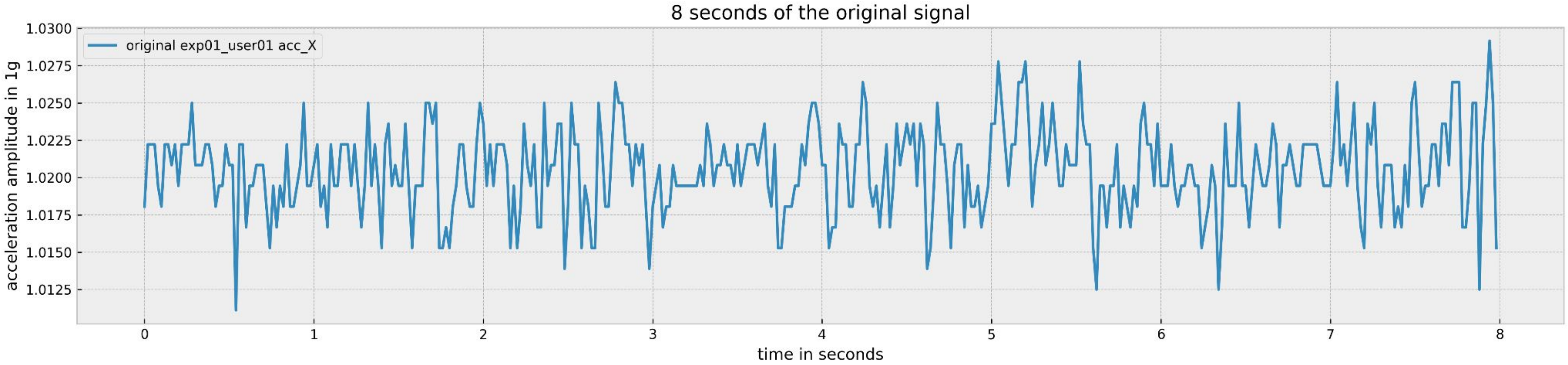
50	10	20
30	70	90
40	60	80

Sorter Output:

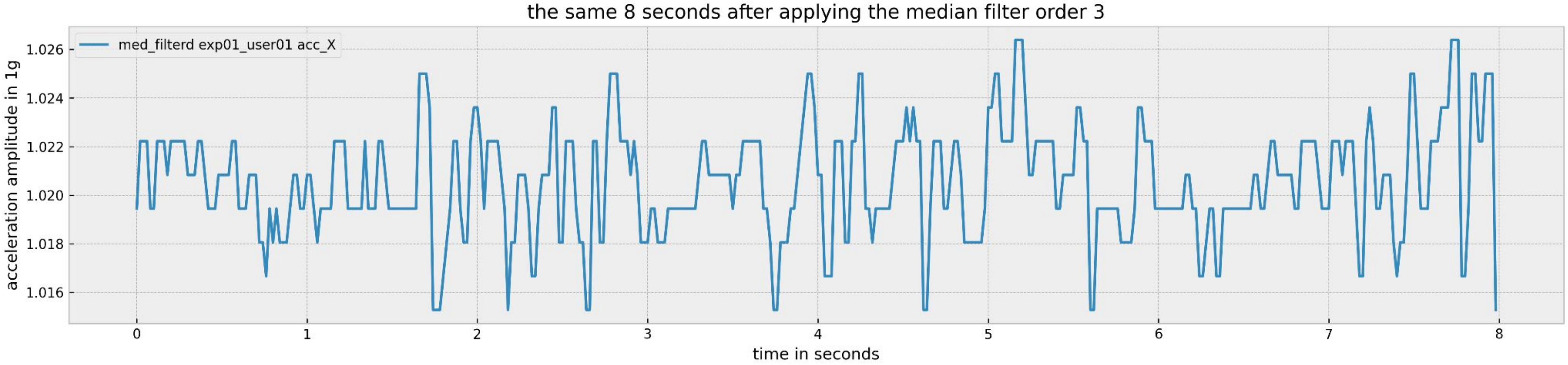
Output Pixel:

_	_	_
_	50	_
_	_	_

The filter collects a window of samples from the input signal, and then performs the median operation on those samples.



8 seconds of the original signal



the same 8 seconds after applying the median filter order 3

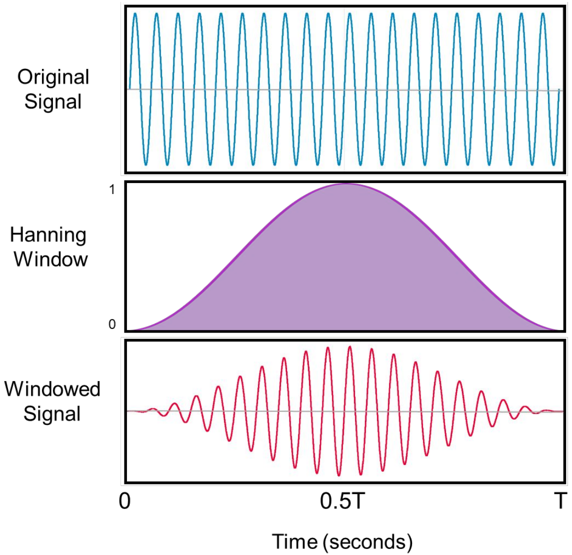
Useful Components Selection:

Import FFT fast fourier transform function to convert SIGNAL from *time domain* to *frequency domain.*

Jerking function & Magnitude function:

jerk(signal(x0)) is equal to (signal(x0+dx)-signal(x0))/dt

Windowing is a signal processing technique that involves multiplying a signal by a window function to reduce discontinuity effects and signal leakage



EXAMPLE

Windowing helps to break down continuous signals into smaller manageable chunks(windows)

In the code, the sliding window has 50% overlap.
Overlapping windows ensure that important features that may span multiple windows aren’t missed.

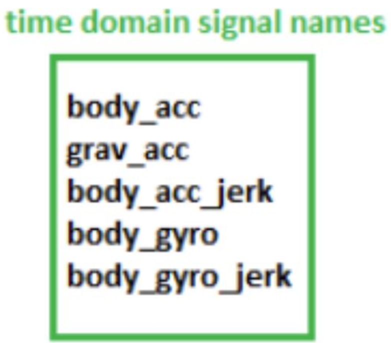
Overlapping windows of time series data help to capture patterns and features.

Features Generation:

By using mean, std, median_deviation, max and min on the dataset we try to extract axial features for *3 axial signals X,Y,Z*

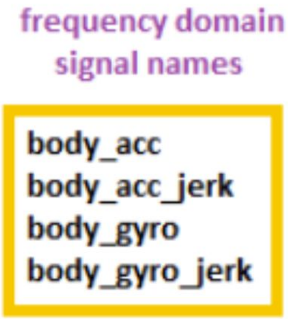
Also we can define magnitude feature functions

Finally we get: 1. Time features
2. Frequency features
3. Common features



time domain signal names

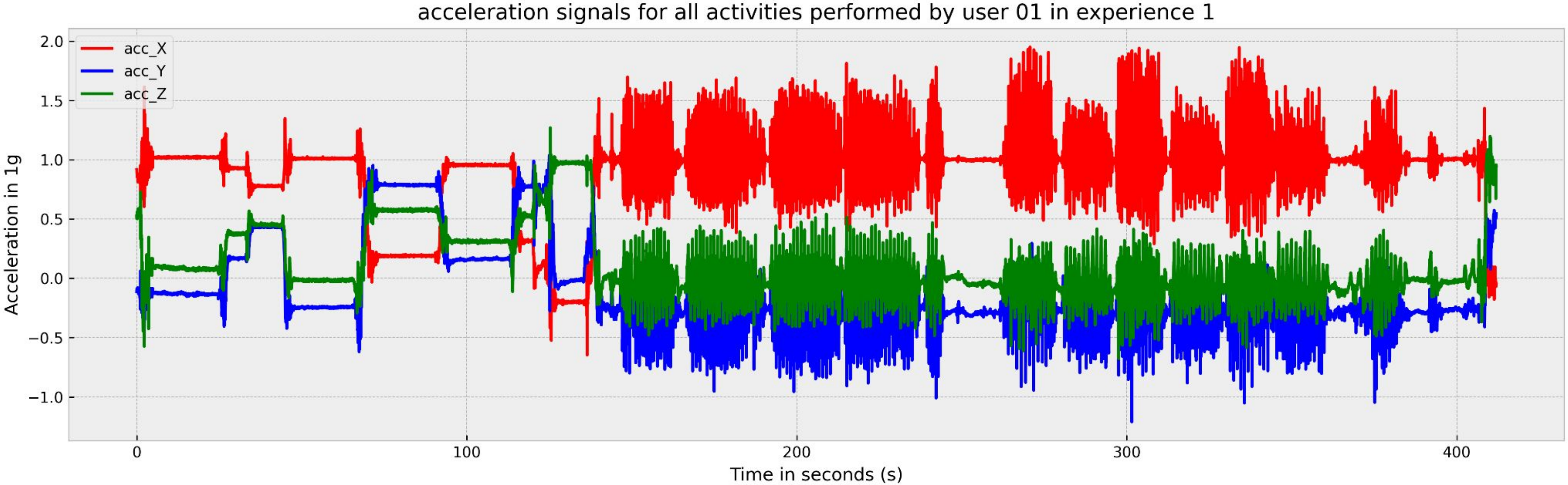
body_acc
grav_acc
body_acc_jerk
body_gyro
body_gyro_jerk



frequency domain signal names

body_acc
body_acc_jerk
body_gyro
body_gyro_jerk

561 features was obtained by calculating variables from the time and frequency domain.


acceleration signals for all activities performed by user 01 in experience 1
acc_X
acc_Y
acc_Z
Acceleration in 1g
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
100
200
300
400
Time in seconds (s)