HUMAN ACTIVITY RECOGNITION AND ANALYSIS USING ACCELEROMETER DATA

By

MS. ASHWINI DHAREWA (11700114020)

MS. DISHA ROY CHOWDHURY (11700114031)

MS. SWAGATA KUNDU (11700114085)

MS. TANUSREE ROY (11700114089)

UNDER THE GUIDANCE OF

MR. SOMENATH NAG CHOUDHURY

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

SESSION 2017-2018

श्रमम् बिना न किमपि साध्यम्

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY
[Affiliated to Maulana Abul Kalam Azad University of Technology]
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY

श्रमम् बिना न किमपि साध्यम्

TO WHOM IT MAY CONCERN

I hereby recommend that the Project entitled **ENHANCED HUMAN ACTIVITY RECOGNITION USING ACCELEROMETER DATA FROM SMARTPHONES** prepared under my supervision by **Ashwini Dharewa (11700114020), Disha Roy Chowdhury(11700114031), Swagata Kundu (11700114085), Tanusree Roy (11700114089)** of B.Tech (7th Semester), may be accepted in partial fulfilment for the degree of **Bachelor of Technology in Computer Science & Engineering** under Maulana Abul Kalam Azad University of Technology (MAKAUT).

……………………………………………………..
Project Supervisor
Department of Computer Science and Engineering
RCC Institute of Information Technology

Countersigned:

……………………………………………………..
Head
Department of Computer Science & Engineering,
RCC Institute of Information Technology
Kolkata – 700015.

DEPARTMENT PF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY

CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT

1. ___________________________________

2.___________________________________

(Signature of Examiners)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY

ACKNOWLEDGEMENT

We would like to express our gratitude to our teacher Mr. Somenath Nag Chowdhury, for giving us a good guideline for our project. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in this project.

Our sincerest gratitude to many people, especially our classmates and team members itself, for making valuable comment suggestions which gave us an inspiration to improve our project. We thank all the people for their help directly and indirectly for this project.

Abstract

In the given Final Year Project "Human Activity Recognition using Accelerometer Data", we are implementing a hardware application (making use of Arduino Uno, Bluetooth module HC 05, Sensor MPU 6050) that is able to record data for different activities such as walking, jogging or sitting.

The very next task is to select one of the collected data and carry out the filtering process for the selected data record in order to remove the unwanted noises. Next we perform the activity recognition that detects which activity has been performed. It could be either walking, running or sitting. For that purpose we have implemented a software interface using MATLAB coding. Activity Recognition is carried out (comparing sample data with the base data with respect to gender and age) using some recognition features such as RMS measure, Automatic peak detection, Count Peak grouping etc. and further we carry out analysis part of activity recognition which checks the fitness.

Table of Contents

Table of figures

Chapter 1

INTRODUCTION

Human Activity Recognition and Analysis deals with the recognition of certain human activities and certainly manipulating the recorded data to check the fitness. Activities such as walking, running, jogging are some of the physical activities that a person performs in his day to day lives. Our Hardware is able to record the data for such activities. Later the recorded data is manipulated for certain purposes that we have described in our complete project work. The synergy of communication, computation and sensing capabilities in mobile systems-on-chip devices such as smartphones has made possible the development of wearable smart sensor systems for user activity monitoring and recognition. A human activity hierarchical recognition system based on neural networks without the need of the smartphone to be constrained to a single fixed position is presented. Experimental results on Android-capable smartphones on four on-body locations show that the recognition system achieves high classification rates, above 92%, for five activities including slow walking, fast walking, jogging, and up-down stairs walking, which outperforms other proposals.

REVIEW OF LITERATURE

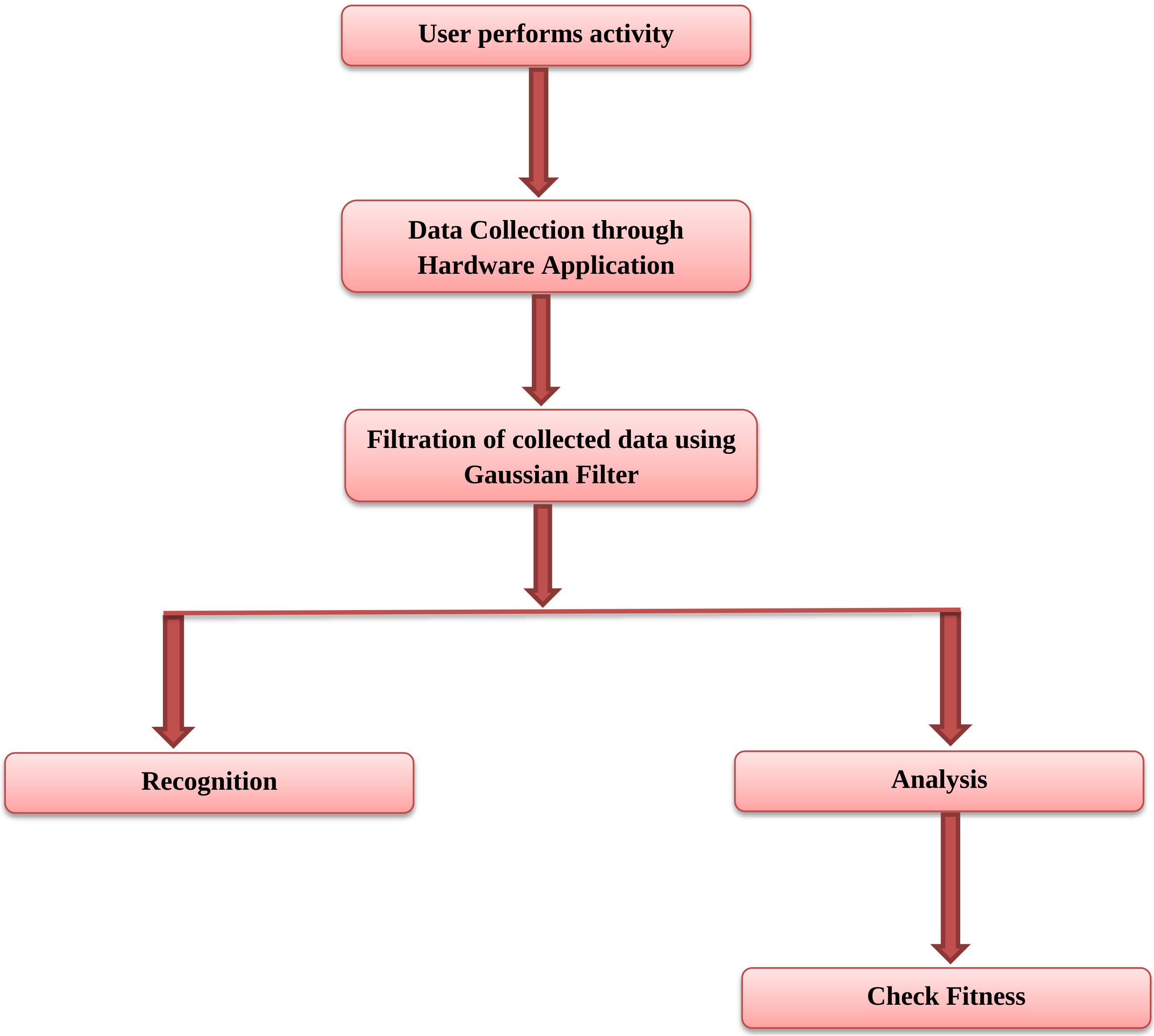
Reference Number	Author(s)	Published in	Year of publication
1.	Aminian,K and Robert, Ph and Buchser,EE and Rutschmann	Medical and Biological engineering and computing; 37:3-304	1991
2.	Bao,Ling and Intille,StephenS	Springer	2004
3.	Casale, Pierluigi and Pujol, Oriol and Radeva, Petia	289 pp, Springer.	2011
4.	Khan, Adil Mehmood and Lee, Young-Koo and Lee, Sungyoung Y and Kim, Tae-Seong	Information Technology in Biomedicine, IEEE	2010
5.	Lester, Jonathan and Choudhury, Tanzeem and Borriello, Gaetano	Springer	2006
6.	Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A	ACM SigKDD Explorations Newsletter	2011
7.	Ravi, Nishkam and Dandekar, Nikhil and Mysore, Preetham and Littman, Michael	AAAI	2005
8.	Mannini, Andrea and Sabatini, Angelo Maria	Sensors	2010
9.	Ruta, Dymitr and Gabrys, Bogdan	Informtion fusion	2005
10.	Gyllensten, Illapha Cuba and Bonomi, Alberto G	IEEE Transaction	2011
11.	Alelyani, Salem and Tang, Jiliang and Liu, Huan	IEEE	2013
12.	Fujiki, Yuichi	CHI'10 Extended Abstracts	2010
13.	Krishnan, Narayanan C and Colbry, Dirk and Juillard, Colin and Panchanathan, Sethuraman	, signals and information processing workshop	2008
14.	. Foerster, Friedrich and Fahrenberg, Jochen	instruments, & computers	2000
15.	Westerterp, Klaas R	European journal of applied physiology	2009
16.	Poppe, Ronald	Image and vision computing	2010
17.	Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A	Fourth IEEE International Conference	2010

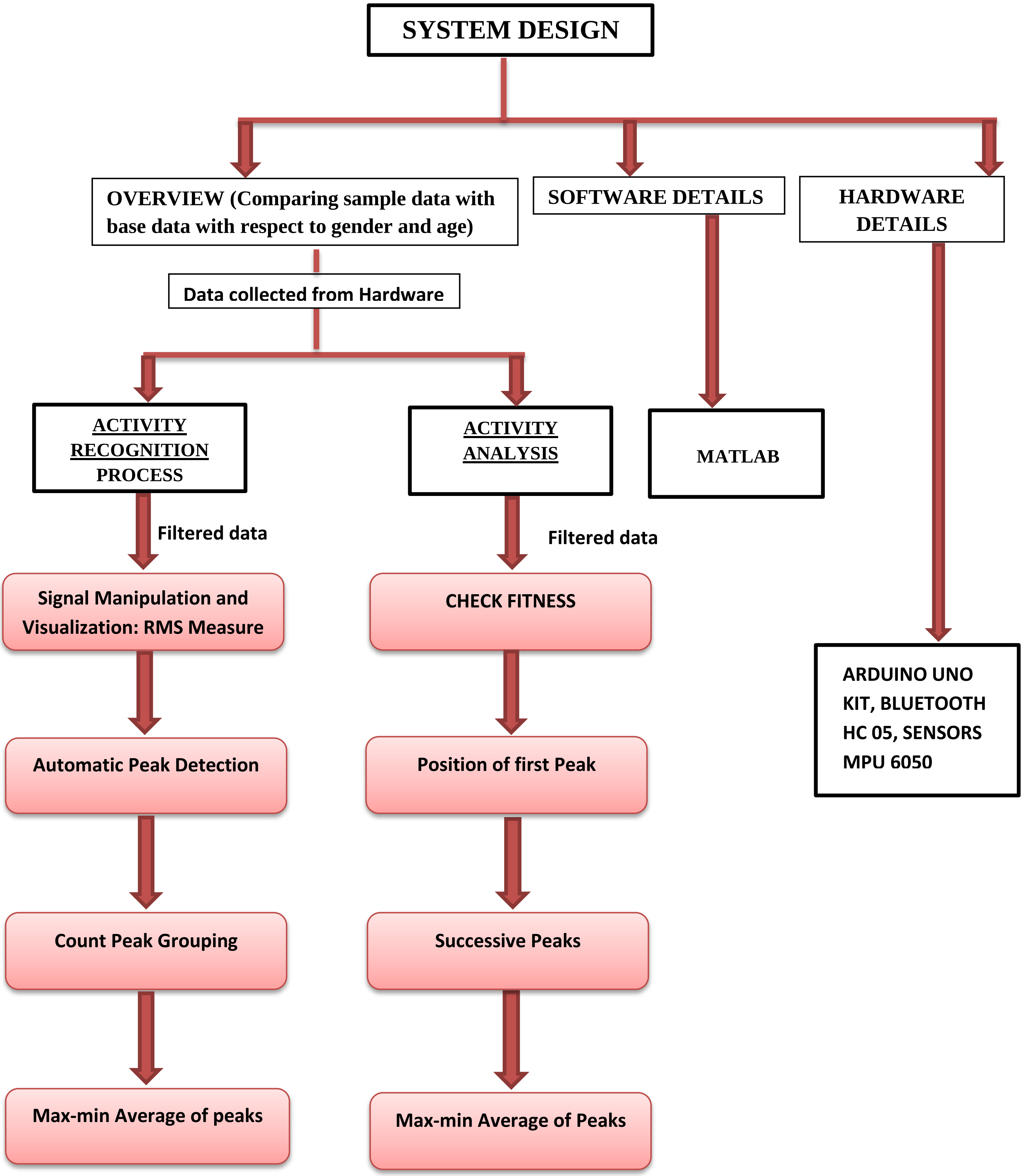
SPECIFIC PROJECT RELATED REFERENCES		
ACTIVITY RECOGNITION	**ABSTRACT**	**PUBLISHED IN**
Signal Manipulation and Visualization	The paper introduces activity detection to scientific simulations with respect to time and its utilization in scientific visualization.	IEEE Transactions on Visualization and Computer Graphics (Volume: 20,Issue: 3, March 2014)
Frequency Domain Analysis	The paper presents a smart phone position independent activity recognition model based on frequency domain using FFT curves.	Computer Science and Network Technology (ICCSNT),2013 at 3rd International Conference.
Feature Extraction of signals	Feature Extraction of EEG for emotion recognition using higher order crossings	Advances in Signal processing (CASP) 11th june 2010
Test and Train of neural Networks	Artificial neural networks in accelerometer based human activity recognition.	Mixed design of integrated circuits and sysytems (MIXDES),2015 22nd International Conference

AUTOMATED SYSTEM DESIGN	ABSTRACT	PUBLISHED IN
Moving average filter	The paper presents a novel real-time QRS detection algorithm designed based on a simple moving average filter.	Computers in Cardiology,2003
Exponential Moving average filter	This paper studies MPPT (Maximum Power Point Tracking) control performance in the existence of noise, paying special attention to the output degradation problem.	Power and Energy (PECon),2012 IEEE International Conference

OBJECTIVE OF THE PROJECT

To design and develop a automated system application software to recognize, classify and analyze the different human activities and provide some informative message in return.




SYSTEM DESIGN
OVERVIEW (Comparing sample data with base data with respect to gender and age)
SOFTWARE DETAILS
HARDWARE DETAILS
Data collected from Hardware
ACTIVITY RECOGNITION PROCESS
ACTIVITY ANALYSIS
MATLAB
Filtered data
Filtered data
Signal Manipulation and Visualization: RMS Measure
CHECK FITNESS
ARDUINO UNO KIT, BLUETOOTH HC 05, SENSORS MPU 6050
Automatic Peak Detection
Position of first Peak
Count Peak Grouping
Successive Peaks
Max-min Average of peaks
Max-min Average of Peaks

Chapter 2

METHODOLOGY FOR IMPLEMENTATION

HARDWARE IMPLEMENTATION

1. User Performs activity (Jogging, Walking, Sitting)
2. Collection of data through a Hardware Application.

   Description: We have built a hardware application that could record data for different human activities and later those data could be used for the recognition as well as analysis process.

2.1 COMPONENTS USED FOR THE HARDWARE SETUP:

- **ARDUINO UNO KIT**

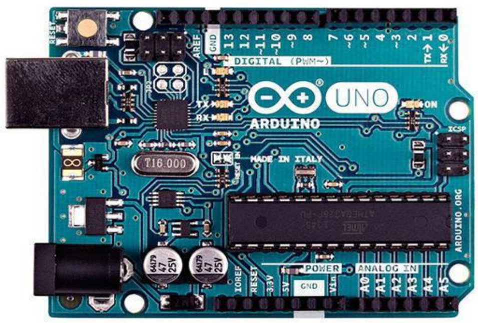


Fig 2.1(a) Arduino Kit

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory.

- **BLUETOOTH HC 05**

Fig 2.1(b) Bluetooth module HC 05

HC-05 module is an easy to use **Bluetooth SPP (Serial Port Protocol) module**, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port bluetooth module is fully qualified **Bluetooth V2.0+EDR (Enhanced Data Rate)** 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04**-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to any embedded project.

- **SENSOR MPU 6050**

Fig 2.1(c) Sensor MPU 6050

The MPU6050 has an embedded 3-axis MEMS gyroscope, a 3-axis MEMS accelerometer. It is very useful for some motion detecting. This small module integrate the logic level converter circuit（makes it compatible with 3.3V-5V voltage level) together with the MPU6050 sensor, you can integrate it to your project conveniently.

Features of MPU6050

- I2C interface
- Compatible with 3.3V-5.0V voltage level

2.2 BLOCK DIAGRAM FOR THE HARDWARE SETUP:

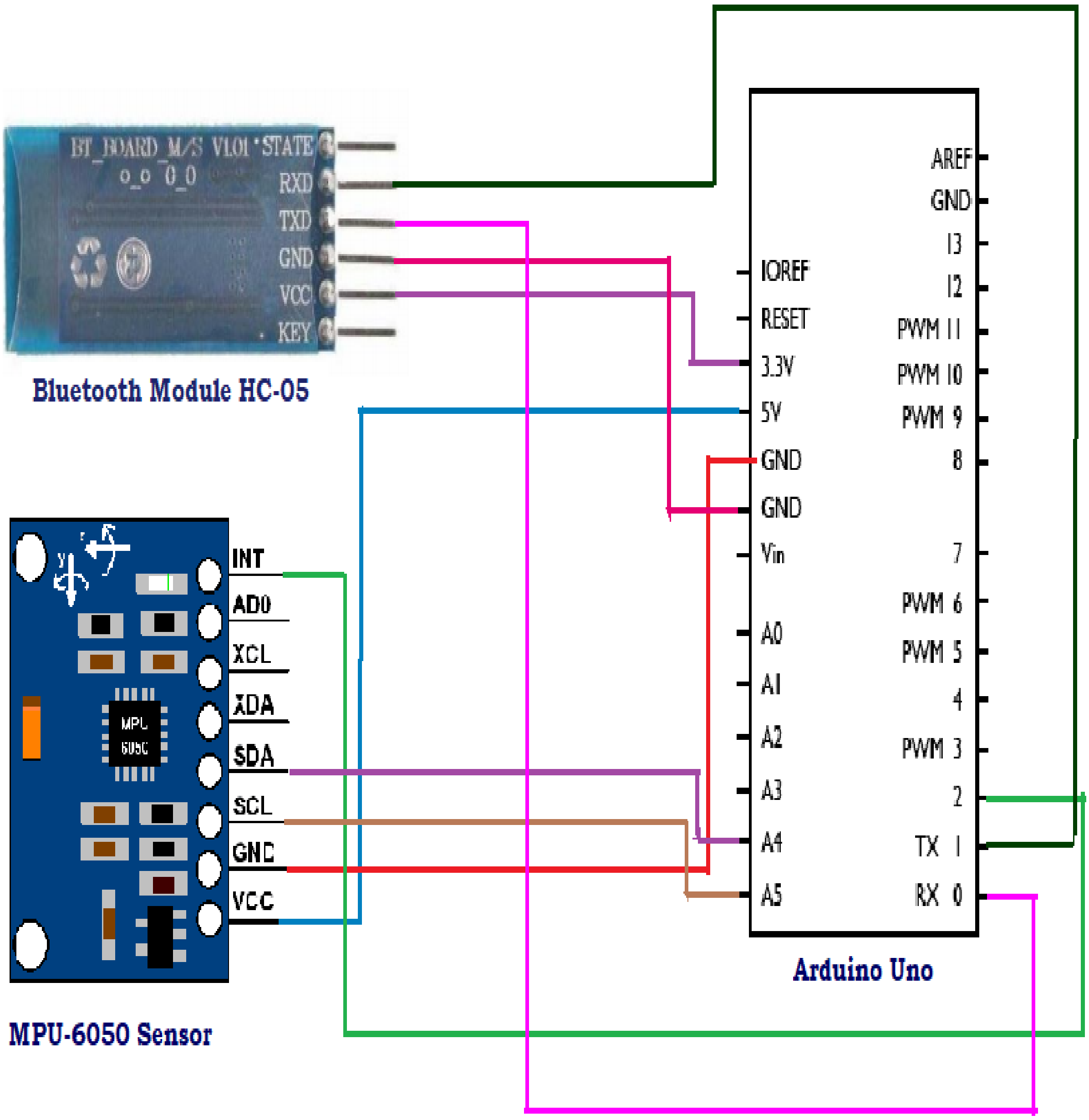


Fig 2.2 Block Diagram

2.3 OUR HARDWARE IMPLEMENTED KIT

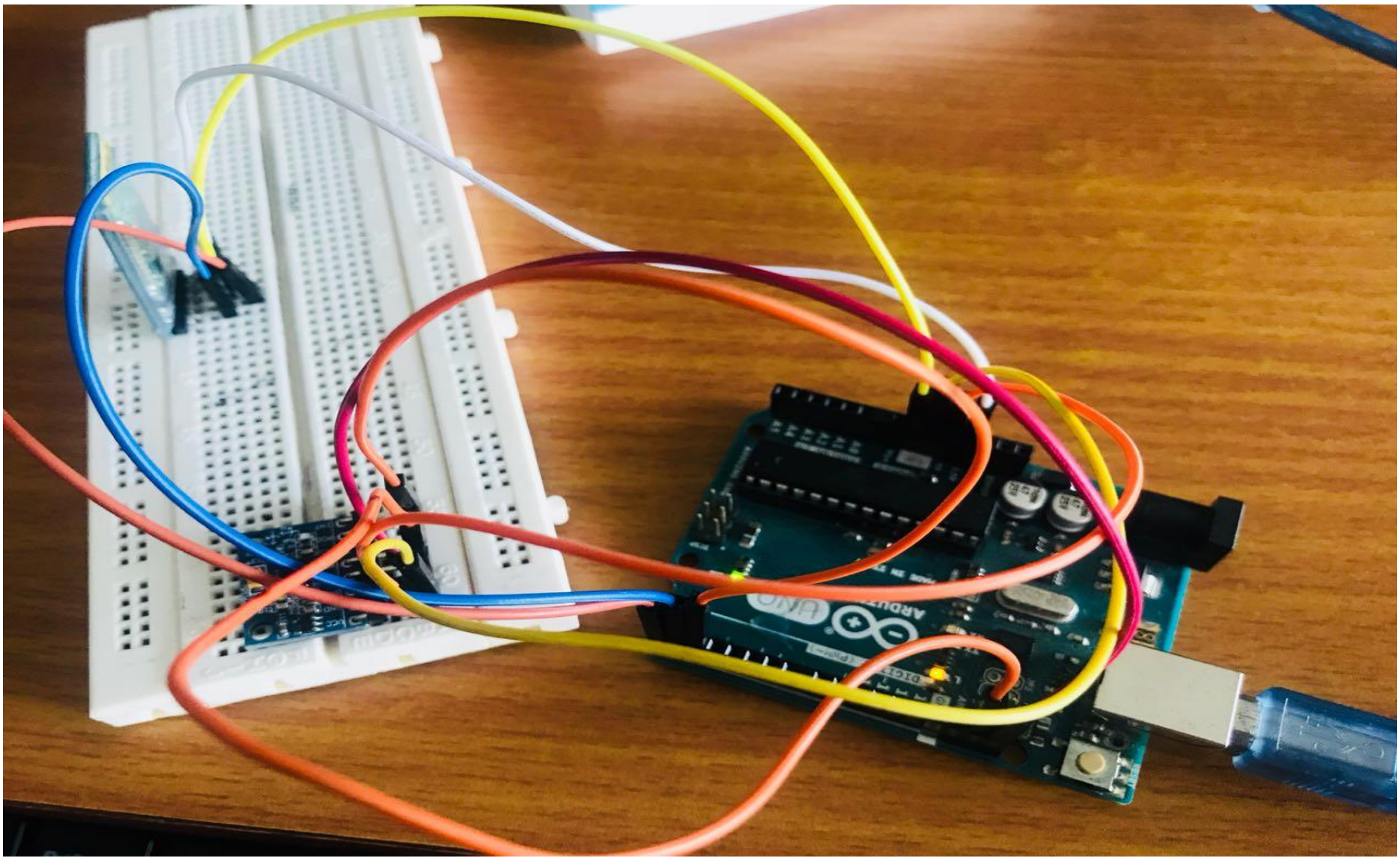


Fig 2.3 Hardware Kit

CONNECTIONS REQUIRED FOR THE HARDWARE IMPLEMENTATION

CONNECTING ARDUINO UNO TO MPU 6050 (SENSOR)

5V → VCC
GND → GND
2 → INT
SDA → SDA
SLC → SLC

CONNECTING ARDUINO UNO TO MPU 6050 (SENSOR)

3.3 V → VCC
GND → GND
RX → TX
TX → RX

2.4 ARDUINO UNO CODE:

```
#include<Wire.h>

const int MPU6050_addr=0x68;

int16_t AccX,AccY,AccZ,Temp,GyroX,GyroY,GyroZ;
float t=0.000;
float i=0.002;

void setup(){

  Wire.begin();

  Wire.beginTransmission(MPU6050_addr);

  Wire.write(0x6B);

  Wire.write(0);

  Wire.endTransmission(true);

  Serial.begin(9600);
  Serial.println("Time(s), X ,Y ,Z ");

}

void loop(){

  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_addr,14,true);

  AccX=(Wire.read()<<8|Wire.read()) ;

  AccY=(Wire.read()<<8|Wire.read());

  AccZ=(Wire.read()<<8|Wire.read());
  Serial.print(t);
  Serial.print(",");
 Serial.print(AccX/16384.0000);
 Serial.print(",");
   Serial.print(AccY/16384.0000);
 Serial.print(",");
  Serial.println(AccZ/16384.0000);
```

```
 t=t+i;
 delay(10);

}
```

Data recorded for jogging activity

Time (s)	X	Y	Z
0	0.999603	5.999939	13.0419
0	2.143372	5.838425	9.56514
0	1.570297	5.883881	5.236511
0.016	-0.05203	6.92955	5.961548
0.036	-0.28415	7.045593	10.57372
0.054	1.204193	7.603134	14.16893
0.074	2.600403	6.669922	13.12805
0.093	2.056046	4.52356	8.733627
0.113	0.041275	2.780396	5.549973
0.131	-0.91226	2.03862	5.43512
0.15	0.434906	0.913986	6.397034
0.17	-0.51744	1.203522	5.498535
0.191	0.199203	1.312393	4.217178
0.207	0.304489	1.578003	0.633926
0.226	0.193222	1.95607	-0.46439
0.246	0.464813	2.38678	2.648682
0.265	0.182465	5.101425	6.456863
0.284	-0.80339	8.434631	8.974106
0.302	-1.30588	9.355865	9.926453
0.322	-3.44984	10.12636	12.03931
0.344	-0.54855	12.22366	16.49715
0.36	-1.26401	12.90682	17.32506

Fig 2.4(a) jog.xlsx

Data recorded for walking activity

	A	B	C	D
5	Time (s)	X	Y	Z
6	0	-0.30646	1.249786	13.92946
7	0.011	-0.91937	1.249786	11.4778
8	0.026	-1.53229	1.862701	8.719681
9	0.042	-1.83875	2.475616	7.187378
10	0.075	-2.1452	2.782074	7.800308
11	0.075	-1.83875	2.169159	9.332596
12	0.089	-0.91937	0.943329	11.78426
13	0.105	0	-0.58896	14.54237
14	0.121	0.919373	-1.50833	15.15529
15	0.138	0.919373	-1.20187	13.01009
16	0.154	0.919373	-0.2825	9.945511
17	0.17	0.612915	1.249786	7.800308
18	0.185	0.306458	2.169159	6.88092
19	0.201	0.612915	1.862701	7.493851
20	0.22	0.919373	0.943329	9.026138
21	0.232	0.919373	-0.2825	12.39717
22	0.248	0.612915	-1.81479	15.7682
23	0.265	-0.61292	-2.12125	16.38112
24	0.282	-1.22583	-1.20187	14.23592

Fig 2.4(b) walk.xlsx

Data recorded for sitting activity

	A	B	C	D
5	Time (s)	X	Y	Z
6	0	-0.61292	0.943329	9.639053
7	0	-0.61292	0.943329	9.332596
8	0.017	-0.61292	1.249786	9.332596
9	0.032	-0.61292	1.249786	9.639053
10	0.061	-0.61292	1.249786	9.639053
11	0.063	-0.61292	1.249786	9.945511
12	0.08	-0.61292	1.249786	9.639053
13	0.095	-0.61292	1.249786	9.639053
14	0.114	-0.61292	1.249786	9.639053
15	0.129	-0.61292	0.943329	9.639053
16	0.143	-0.61292	0.943329	9.639053
17	0.159	-0.61292	0.943329	9.945511
18	0.175	-0.61292	0.943329	9.945511
19	0.19	-0.61292	0.943329	9.639053
20	0.207	-0.61292	0.943329	9.639053
21	0.222	-0.61292	1.249786	9.639053
22	0.237	-0.61292	1.249786	9.639053
23	0.254	-0.61292	1.249786	9.639053
24	0.27	-0.61292	1.249786	9.945511

Fig 2.4(c) sit.xlsx

Chapter 3

METHODOLOGY FOR IMPLEMENTATION

3. ACTIVITY RECOGNITION

Here we will be selecting out a random collected data through browsing the excel sheets and further the selected data will be recognised or detected to be walking, sitting or jogging activity.

3.1 FILTERING OF THE SELECTED DATA

Here we are making use of Gaussian Filter for the purpose of filtering.
Filtering is basically done to limit the noise in the selected data.

Gaussian Filter is a filter whose impulse response is a Gaussian Function.

MATLAB CODE FOR THE FILTERING PROCESS:

```
global walk1;
global sit1;
global jog1;
global selectedfile1;
global walk;
global sit;
global jog;
global selectedfile;
global T1;
global T2;
global T3;
global T4;
global T5;

T1=xlsread(selectedfile1,'Sheet1','A1:A1845')          //Read time value of selected data
X = xlsread(selectedfile1,'Sheet1','B1:B1845')         //Read AccX  value of selected data
Y = xlsread(selectedfile1,'Sheet1','C1:B1845')         //Read AccY value of selected data
for i=1:length(X)
   V1(i)=sqrt((X(i)*X(i))+(Y(i)*Y(i)))                //Calculate |V| from AccX and AccY
End

[selectedfile, window] = smoothdata(V1,'gaussian');    //Filtering the data and store it in
                                                       selectedfile
```

```
T2=xlsread(walk1,'Sheet1','A1:A1845')
walk = xlsread(walk1,'Sheet1','E1:E1845')

T3=xlsread(sit1,'Sheet1','A1:A1845')
sit = xlsread(sit1,'Sheet1','E1:E1845')

T5=xlsread(jog1,'Sheet1','A1:A1845')
jog = xlsread(jog1,'Sheet1','E1:E1845')
```

//Read Base Data for walking,sitting and jogging for the corresponding age and gender

```
plot(T1,V1)
hold on
plot(T1,selectedfile)
legend('Original Data','Filtered Data')
```

//Plot original vs. filtered data

PICTORIAL RERESENATION FOR THE FILTERED DATA:

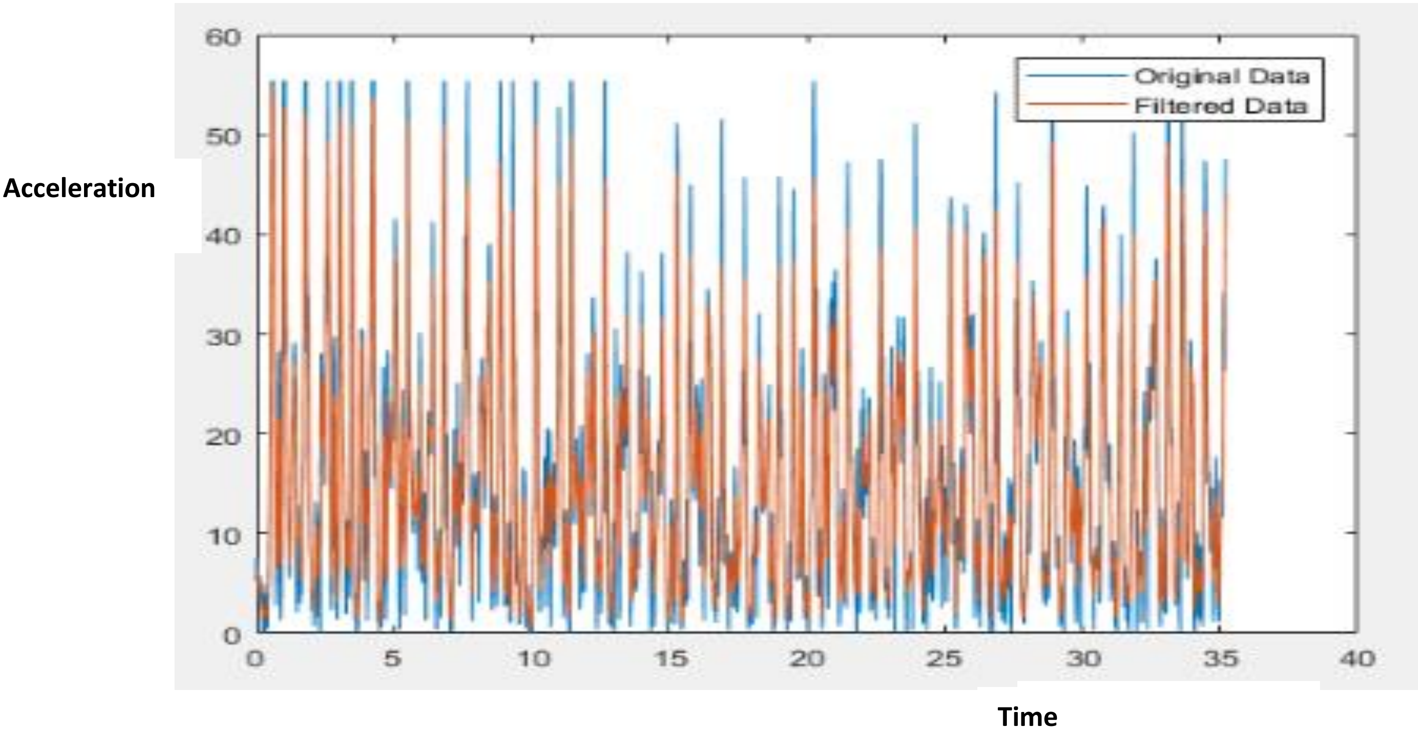


Fig 3.1 Filtered Data

3.2 SIGNAL MANIPULATION AND VISUALISATION: RMS MEASURE

Question: What is RMS Measure?

Answer: RMS is abbreviated as Root Mean Square is mainly defined as the square root of the mean square (arithmetic mean of the squares of a set of numbers).

Pictorial Representation for the calculation of RMS Measure for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Jogging
Description: Here we have initially collected 1845 data records for the respective activity and taking 450 records per interval, the RMS value is calculated and finally a bar chart is consecutively plotted.

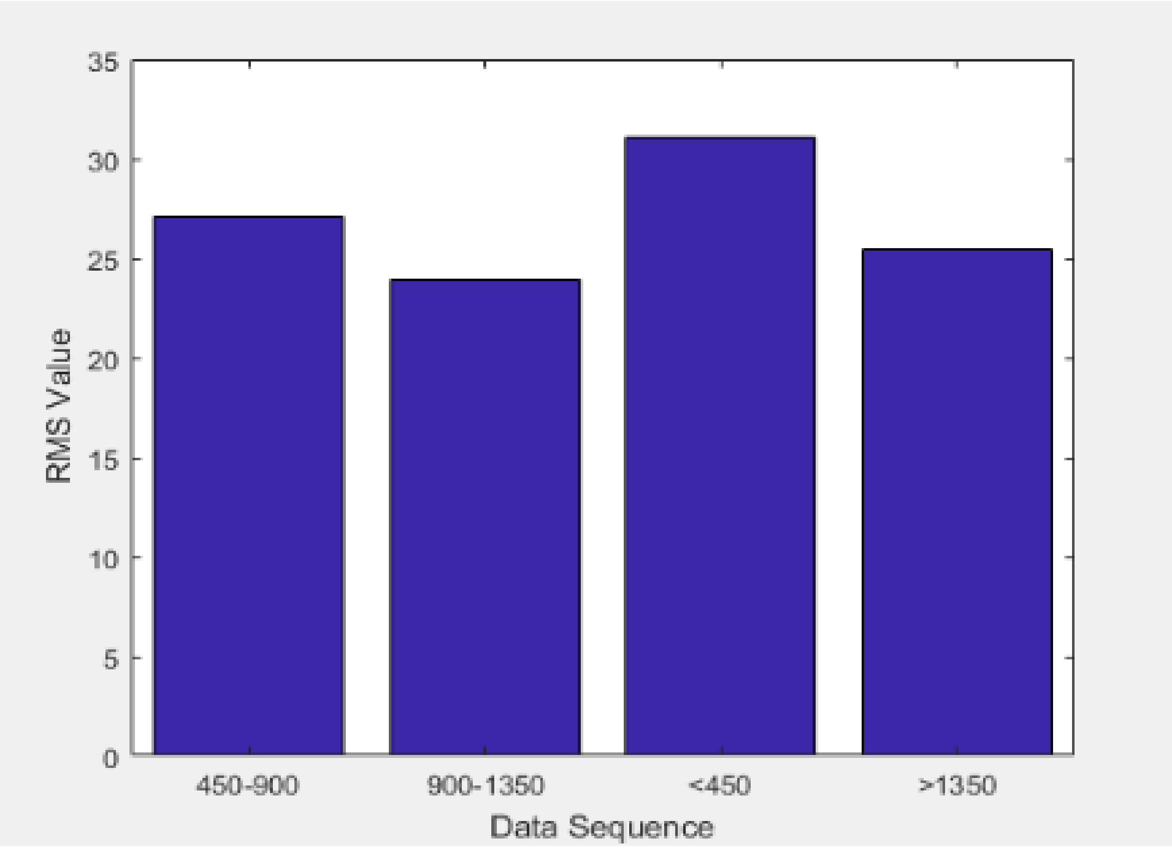


Fig 3.2 RMS Measure

MATLAB CODE for RMS Measure

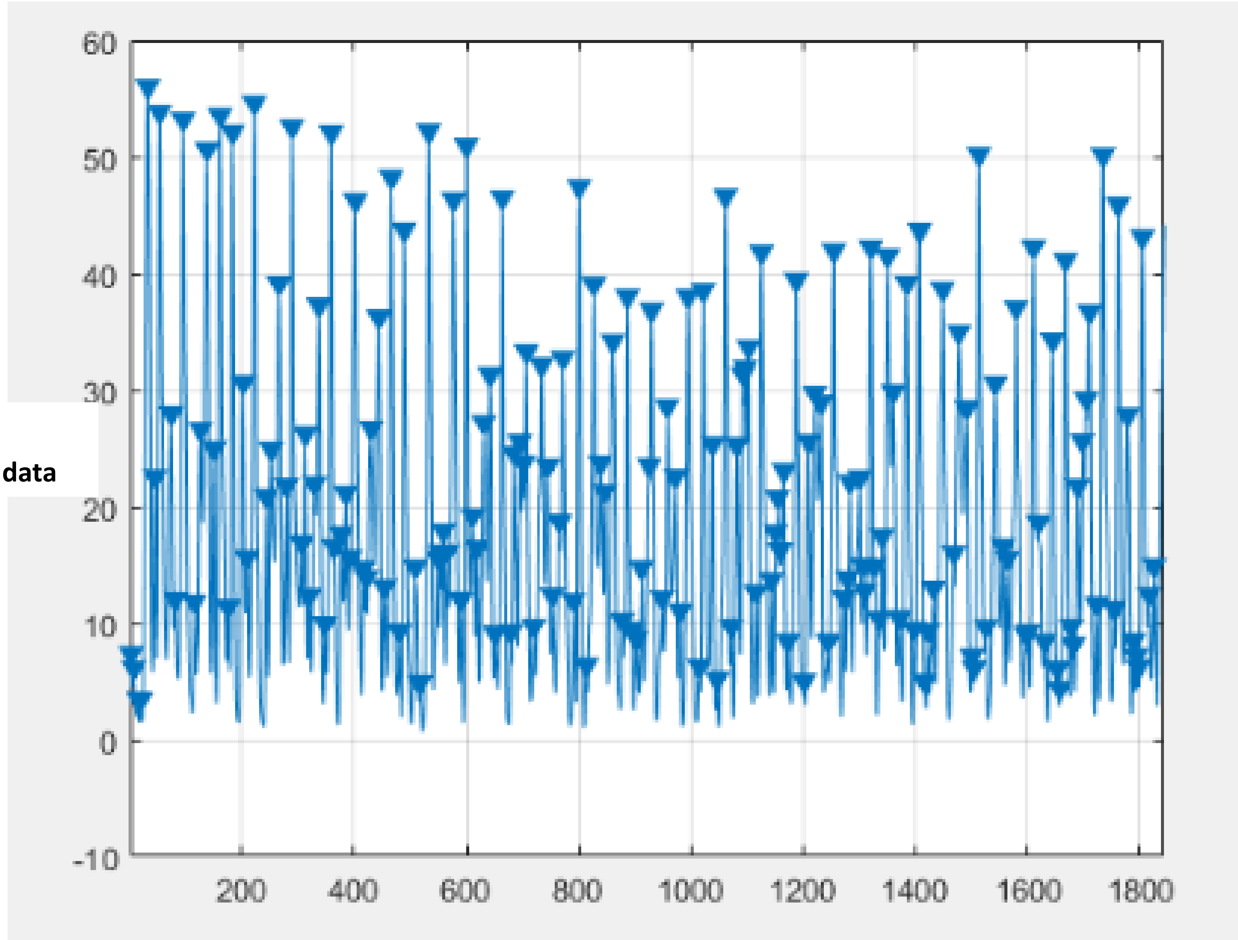
```
global selectedfile;

global T1;

y=selectedfile(1:450)

x=T1(1:450)

s=findpeaks(y)

z1=rms(s)

y=selectedfile(450:900)

x=T1(450:900)

s=findpeaks(y)

z2=rms(s)

y=selectedfile(900:1350)

x=T1(900:1350)

s=findpeaks(y)

z3=rms(s)

y=selectedfile(1350:1800)

x=T1(1350:1800)

s=findpeaks(y)

z4=rms(s)

grp=[z1 z2 z3 z4]

c = categorical({'<450','450-900','900-1350','>1350'});

bar(c,grp)

xlabel('Data Sequence')

ylabel('RMS Value') global selectedfile;

global T1;

y=selectedfile(1:450)                         //Take first 450 data

x=T1(1:450)

s=findpeaks(y)                                //Find peaks and store it in s

z1=rms(s)                                     //Find rms of the peaks

y=selectedfile(450:900)
```

```
x=T1(450:900)

s=findpeaks(y)

z2=rms(s)

y=selectedfile(900:1350)

x=T1(900:1350)

s=findpeaks(y)

z3=rms(s)

y=selectedfile(1350:1800)

x=T1(1350:1800)

s=findpeaks(y)

z4=rms(s)

grp=[z1 z2 z3 z4]                                  //Taking  rms value of 4 groups in an array

c = categorical({'<450','450-900','900-1350','>1350'});

bar(c,grp)

xlabel('Data Sequence')

ylabel('RMS Value')
```

3.3 AUTOMATIC PEAK DETECTION

Pictorial Representation of Automatic Peak Detection for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Jogging
Description: Here we are plotting the peaks of the complete recorded data.

Y axis- Selected data



(x axis- Time)

Fig 3.3 Automatic Peak Detection

MATLAB CODE for Automatic Peak Detection:

```
global selectedfile;
global T1;
y=selectedfile
x=T1
plot(x,y)
xlabel('x')
ylabel('Y')
findpeaks(y)                                    //Find peaks
```

3.4 COUNT PEAK GROUPING

Pictorial Representation for the Count Peak Grouping for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Walking
Description: Here we are initially calculating the 30% and 60% of the highest peak value and respectively counting the number of peaks that are
1. Below 30% 2. Between 30%-60% 3.More than 60%

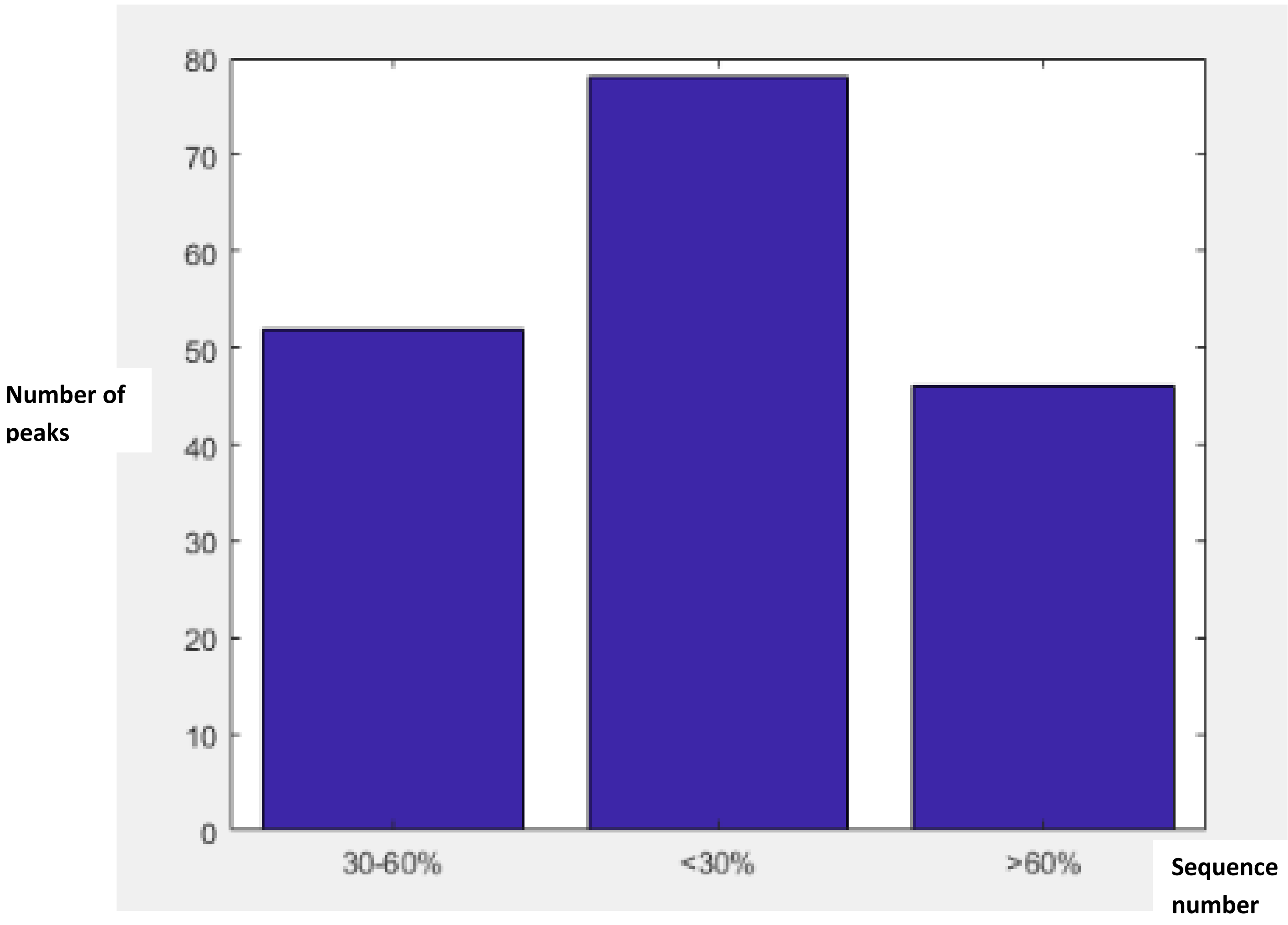


Fig 3.4 Count Peak Grouping

MATLAB CODE for Count Peak Grouping:

```
global selectedfile;
global T1;
y=selectedfile
x=T1

max_peak=max(y)                    //Find maximum peak value
j=1
k=1
l=1
m_30=max_peak*0.3                  //Calculating 30% of maximum peak value
m_60=max_peak*0.6                  //Calculating 60% of maximum peak value

peak=findpeaks(y)
for i=1:length(peak)
   if peak(i)<m_30
      grp1(j)=peak(i)              //In grp1 array store peak values which are <  30%  of
                                   maximum peak

      j=j+1
   end
   if peak(i)>=m_30 && peak(i)<=m_60
      grp2(k)=peak(i)              //In grp2 array store peak values which are between 30%
                                   and 60%  of  maximum peak


      k=k+1
   end
   if peak(i)>m_60
      grp3(l)=peak(l)              //In grp3  array store peak values which are>60%  of
                                   maximum peak


      l=l+1
   end
end
grp=[j k l]
c = categorical({'<30%','30-60%','>60%'});
bar(c,grp)
hold on
```

3.5 MAXIMUM-MINIMUM AVERAGE OF PEAKS

Pictorial Representation for Max-min-average of peaks for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Jogging
Description: Here we are calculating the maximum, minimum and average of the peaks for a collected sample data.

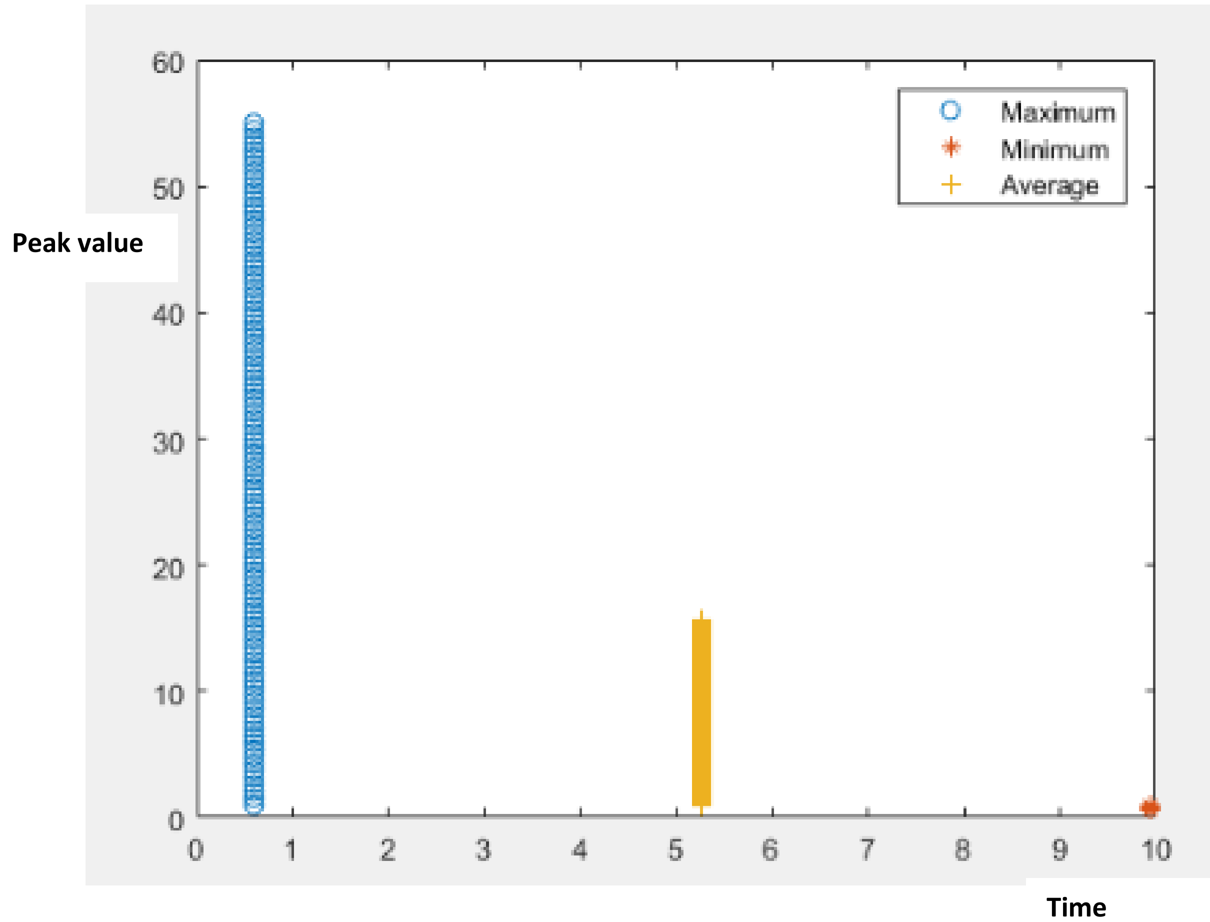


Fig 3.5 Max-min avg of Peaks

MATLAB CODE for Maximum Minimum Average of Peaks:

```
global selectedfile;
global T1;
y=selectedfile
x=T1
z=max(y)                                        //Calculate max value
acc=linspace(1,z,100)
for a=1:length(x)
   if y(a)==z
      t=x(a)
   end
```

```
end
for i=1:100
  time(i)=t
end
plot(time,acc,'o')
hold on


r=min(y)                                    //Calculate min  value
acc2=linspace(1,r,100)
for b=1:length(x)
   if y(b)==r
       s=x(b)
   end
end
for j=1:100
  time2(j)=s
end
plot(time2,acc2,'*')
hold on

r2=mean(y)                                  //Calculate average value
acc3=linspace(1,r2,100)
m=(t+s)/2
for k=1:100
  tme3(k)=m
end
plot(tme3,acc3,'+')
hold on
legend('Maximum','Minimum','Average')
```

Chapter 4

METHODOLOGY FOR IMPLEMENTATION

4. ACTIVITY ANALYSIS

In Activity Analysis, we are recording sample data for three different activities (jogging, walking and sitting) of a single person, next we are filtering out the collected data using a filtering process.

CODE FOR FILTERING PROCESS:

```
global walking;
global sitting;
global jogging;
global wx;
global wy;
global sx;
global sy;
global jx;
global jy;
wx=xlsread(walking,'Sheet1','A1:A1845')
X = xlsread(walking,'Sheet1','B1:B1845')
Y = xlsread(walking,'Sheet1','C1:B1845')          //Filtering the browsed walking data
for i=1:length(X)
   V1(i)=sqrt((X(i)*X(i))+(Y(i)*Y(i)))
end
[wy, window] = smoothdata(V1,'gaussian');

sx=xlsread(sitting,'Sheet1','A1:A1845')
X = xlsread(sitting,'Sheet1','B1:B1845')
Y = xlsread(sitting,'Sheet1','C1:B1845')          //Filtering the browsed sitting data
for i=1:length(X)
   V1(i)=sqrt((X(i)*X(i))+(Y(i)*Y(i)))
end
[sy, window] = smoothdata(V1,'gaussian');

jx=xlsread(jogging,'Sheet1','A1:A1845')
X = xlsread(jogging,'Sheet1','B1:B1845')
Y = xlsread(jogging,'Sheet1','C1:B1845')          //Filtering the browsed jogging data
for i=1:length(X)
   V1(i)=sqrt((X(i)*X(i))+(Y(i)*Y(i)))
end
[jy, window] = smoothdata(V1,'gaussian');
```

4.1 CHECKING POSITION OF THE FIRST PEAK

Checking position of the first peak for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Walking

MATLAB CODE:

```
global wx;
global wy;
global walk;
global T2;
global accuracy1;
x1=T2
y1=walk

x2=wx
y2=wy

p1=findpeaks(y1)          //Calculate peaks of base data
n1=p1(1)                  //First peak of base data
p1=findpeaks(y2)          //Calculate peaks of browsed data
n2=p1(1)                  //First peak of browsed data
d=abs(t1-t2)              //Difference in first peak
                          between base data and browsed data

diff=(d/t1)*100
accuracy1=100-diff        //calculating accuracy of an activity
                          based on this step
```

4.2 SUCCESSIVE PEAKS

Successive peaks for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Walking

MATLAB CODE:

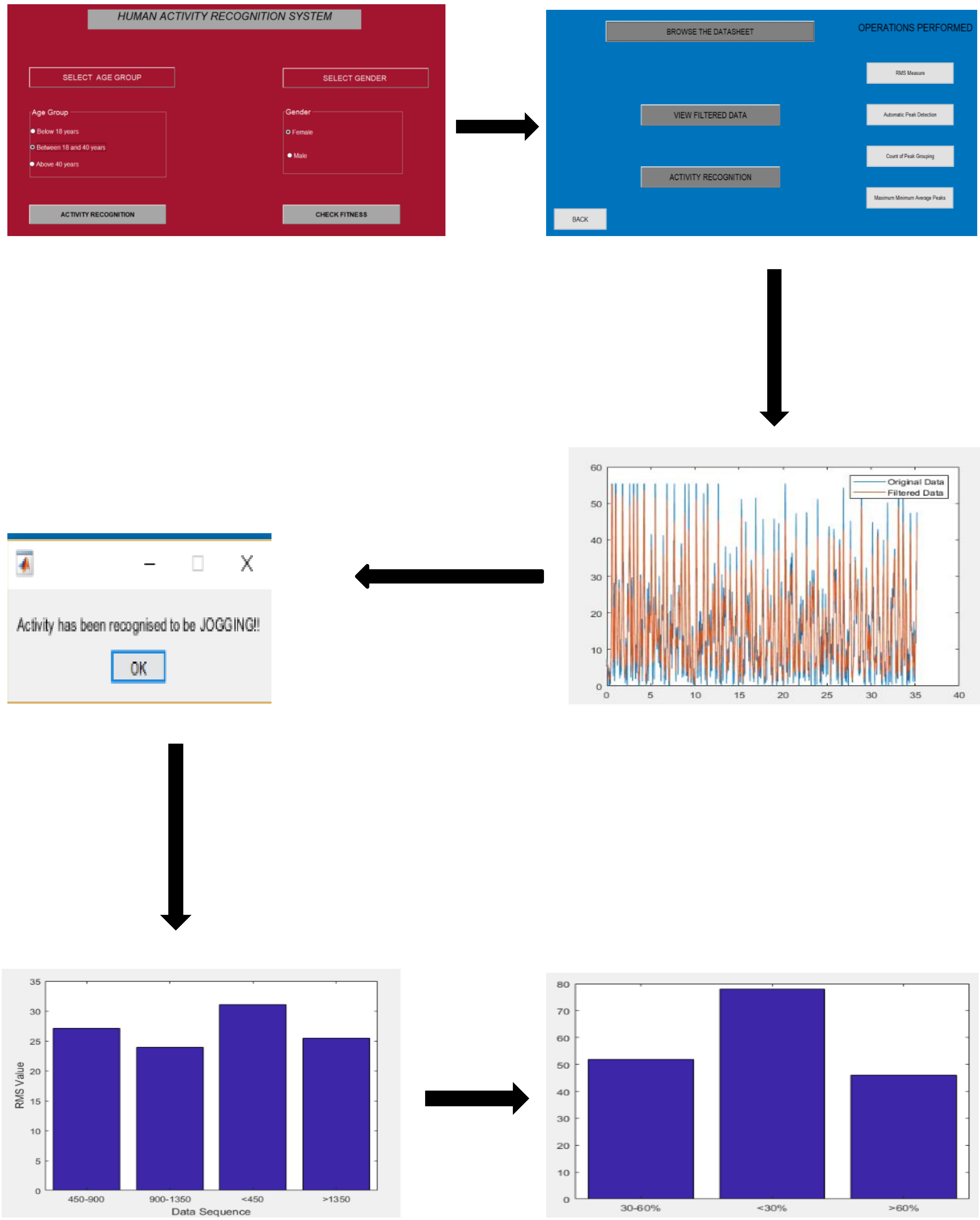
```
global wx;
global wy;
global walk;
global T2;
global accuracy2;
x1=T2
y1=walk

x2=wx
y2=wy
p1=findpeaks(y1)                    //find peaks of base data
p2=findpeaks(y2)                    //find peaks of browsed data
n1=length(p1)
n2=length(p2)
sum=0
if(n1>n2)
   for i=2:n2
      d1(i-1)=abs(p1(i)-p1(i-1))
      d2(i-1)=abs(p2(i)-p2(i-1))
   end
else                                //Difference between successive peaks
   for i=2:n1
      d1(i-1)=abs(p1(i)-p1(i-1))
      d2(i-1)=abs(p2(i)-p2(i-1))
   end
end
for i=1:length(d1)
   diff=abs(d1(i)-d2(i))            //Difference between base data and sample data
   sum=sum+diff
end
len=length(d1)
res=sum/len
accuracy2=100-res                   //calculating accuracy of an activity
                                      based on this step
```

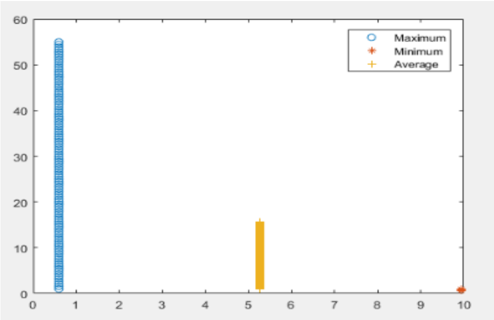
4.3 MAX-MIN AVERAGE OF PEAKS

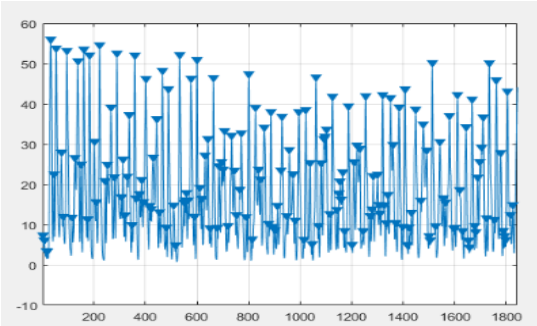
Max-min average of peaks for a sample data
Age: Between 18-40 years
Gender: Female
Activity: Walking
MATLAB CODE:

```
global selectedfile;
global T1;
y=selectedfile
x=T1
z=max(y)                                //Calculate max value
acc=linspace(1,z,100)
for a=1:length(x)
    if y(a)==z
        t=x(a)
    end
end
for i=1:100
   time(i)=t
end
plot(time,acc,'o')
hold on
r=min(y)                                //Calculate min  value
acc2=linspace(1,r,100)
for b=1:length(x)
    if y(b)==r
        s=x(b)
    end
end
for j=1:100
   time2(j)=s
end
plot(time2,acc2,'*')
hold on
r2=mean(y)                              //Calculate average value
acc3=linspace(1,r2,100)
m=(t+s)/2
for k=1:100
   tme3(k)=m
end
plot(tme3,acc3,'+')
hold on
legend('Maximum','Minimum','Average')
```

5. PICTORIAL WORKFLOW FOR THE ACTIVITY RECOGNITION AND ANALYSIS




Maximum
Minimum
Average




FITNESS ANALYSIS
BROWSE WALKING DATASHEET
BROWSE JOGGING DATASHEET
BROWSE SITTING DATASHEET
BACK
NEXT

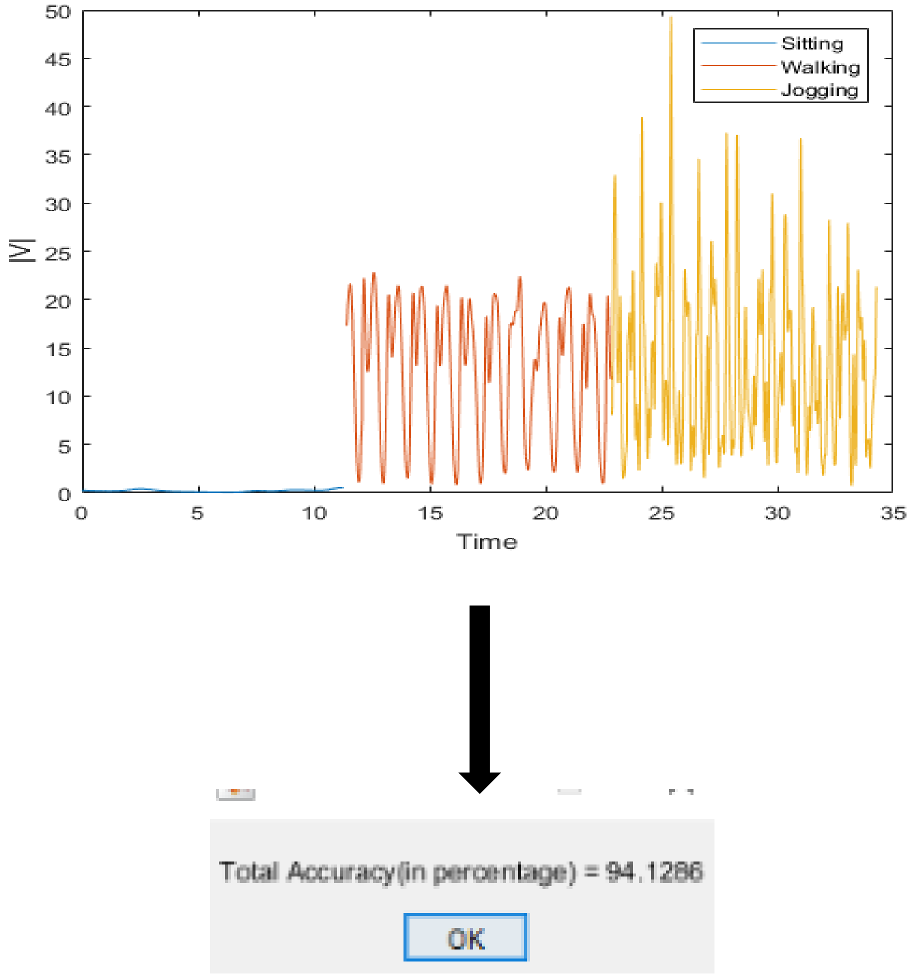


Fig 5 Pictorial workflow for recognition and analysis

Chapter 6

6. IMPLEMENTATION DETAILS

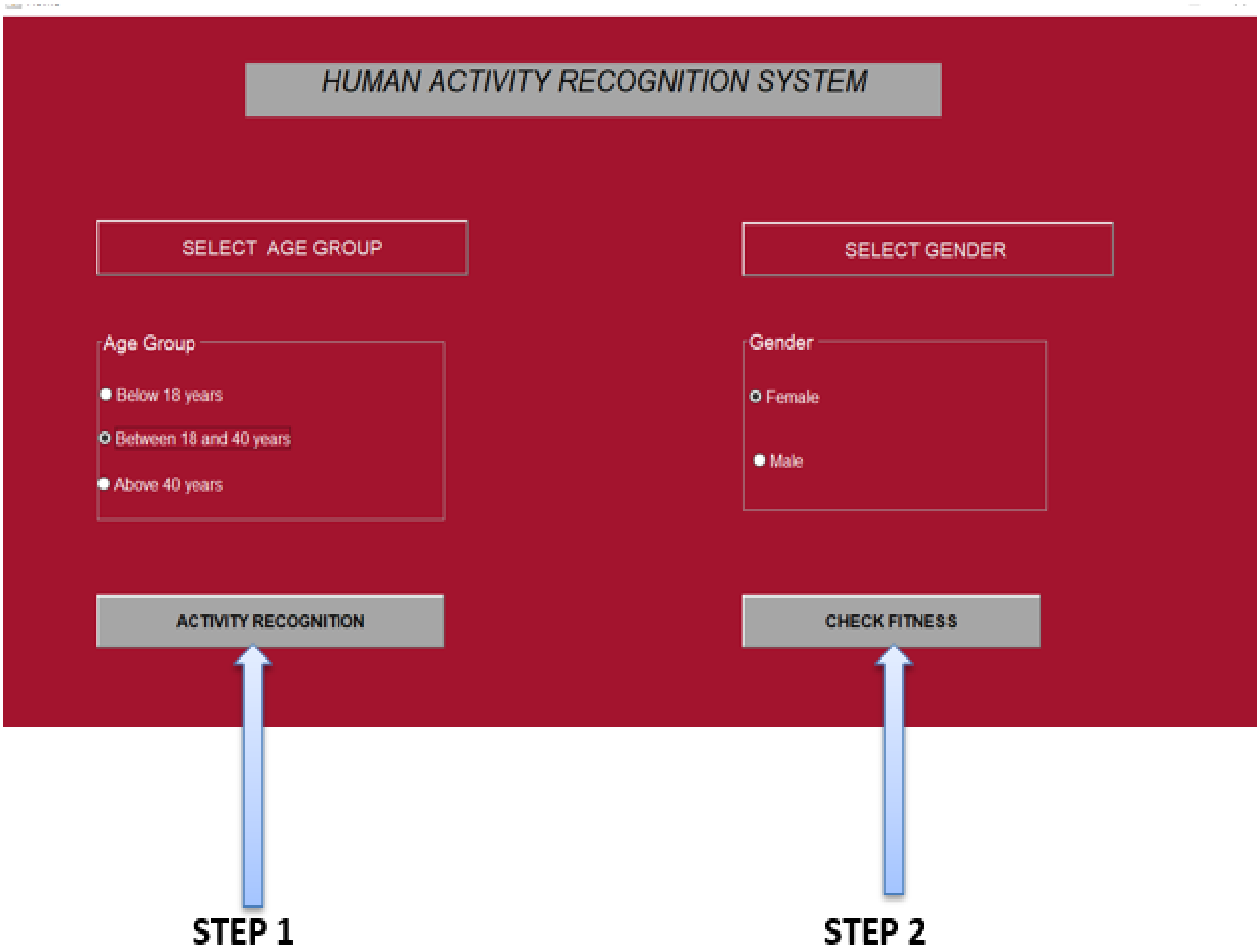


Fig 6(a) Home Page

Step 1: Activity will be recognized.

Step 2: Fitness will be checked.

MATLAB CODE FOR THE HOME PAGE

```
function varargout = Home(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Home_OpeningFcn, ...
                   'gui_OutputFcn',  @Home_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end

function Home_OpeningFcn(hObject, eventdata, handles, varargin)
global walk1;
global sit1;
global jog1;

a=get(handles.age,'SelectedObject')
age1=get(a,'Tag')
b=get(handles.female,'Value')
c=get(handles.male,'Value')

if age1=='a1'                      //If selected age is below 18 years
   if b==1.0                       //and gender is female,
      walk1='a1b1walk.xlsx'        //corresponding base data is selected
      sit1='a2b1sit.xlsx'
      jog1='a1b1jog.xlsx'
   end
end

if age1=='a1'                      //If selected age is below 18 years and
   if c==1.0                       //gender is male
      walk1='a1b2walk.xlsx'        //corresponding base data is selected
      sit1='a2b1sit.xlsx'
      jog1='a1b2jog.xlsx'
   end
end

if age1=='a2'                      //If selected age is between 18 and 40  years
   if b==1.0                       //and gender is female,
      walk1='a2b1walk.xlsx'        //corresponding base data is selected
      sit1='a2b1sit.xlsx'
      jog1='a2b1jog.xlsx'
   end
end

if age1=='a2'                      //If selected age is between 18 and 40
                                   //Years and gender is male,
                                   //corresponding base data is selected
```

```
    if c==1.0
      walk1 = 'a2b2walk.xlsx'
      sit1 ='a2b1sit.xlsx'
      jog1 ='a2b2jog.xlsx'
    end
end

if age1=='a3'
    if b==1.0
       walk1='a3b1walk.xlsx'
       sit1='a2b1sit.xlsx'
       jog1='a3b1jog.xlsx'
    end
end

if age1=='a3'
    if c==1.0
       walk1='a3b2walk.xlsx'
       sit1='a2b1sit.xlsx'
       jog1='a3b2jog.xlsx'
    end
end
run('Interface.m')

function checkfitness_Callback(hObject, eventdata, handles)

run('InterfaceFitness.m')
```

//If selected age is above 40
//Years and gender is female,
//corresponding base data is selected

//If selected age is above 40 years
//and gender is male,
//corresponding base data is selected

INTERFACE FOR THE ACTIVITY RECOGNITION

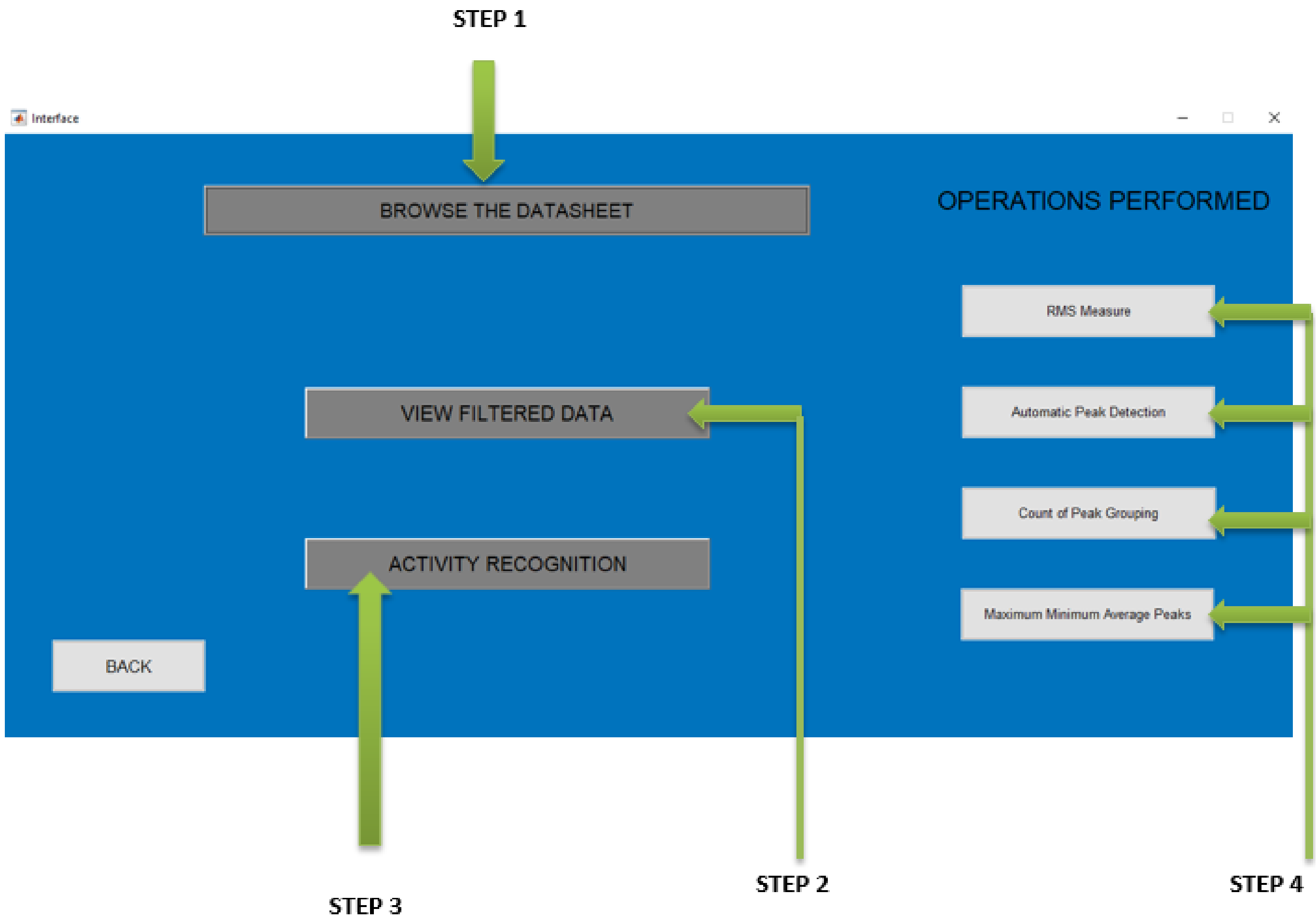


Fig 6(b) Activity analysis and recognition

Step 1: The sample datasheet needs to be selected.

Step 2: Graphical representation of the selected data is displayed.

Step 3: The recognized activity is displayed.

Step 4: Graphical representation of the mentioned operations is performed.

MATLAB CODE FOR "BROWSE THE DATASHEET"

```
global selectedfile1;
[file,path] = uigetfile('C:\Users\USER\Desktop\2.4.2018',...
'Select an Excel File')
selectedfile1 = fullfile(path,file);
```

INTERFACE FOR THE ACTIVITY ANALYSIS

Fig 6(c) Selection of activities to be analyzed

Step 1: The sample datasheet for activity:'walking' needs to be selected.

Step 2: The sample datasheet for activity:'jogging' needs to be selected.

Step 3: The sample datasheet for activity:'sitting' needs to be selected.

MATLAB CODE FOR BROWSING WALKING DATASHEET

```
global walking;

[file,path] = uigetfile('C:\Users\USER\Desktop\2.4.2018\BrowseExcelSheet',...
'Select an Excel File')

walking = fullfile(path,file);
```

//browse the excel sheet from this path

MATLAB CODE FOR BROWSING JOGGING DATASHEET

```
global jogging;

[file,path] = uigetfile('C:\Users\USER\Desktop\2.4.2018',...
'Select an Excel File')

jogging = fullfile(path,file);
```

MATLAB CODE FOR BROWSING SITTING DATASHEET

```
global sitting;

[file,path] = uigetfile('C:\Users\USER\Desktop\2.4.2018',...
'Select an Excel File')

sitting = fullfile(path,file);
```

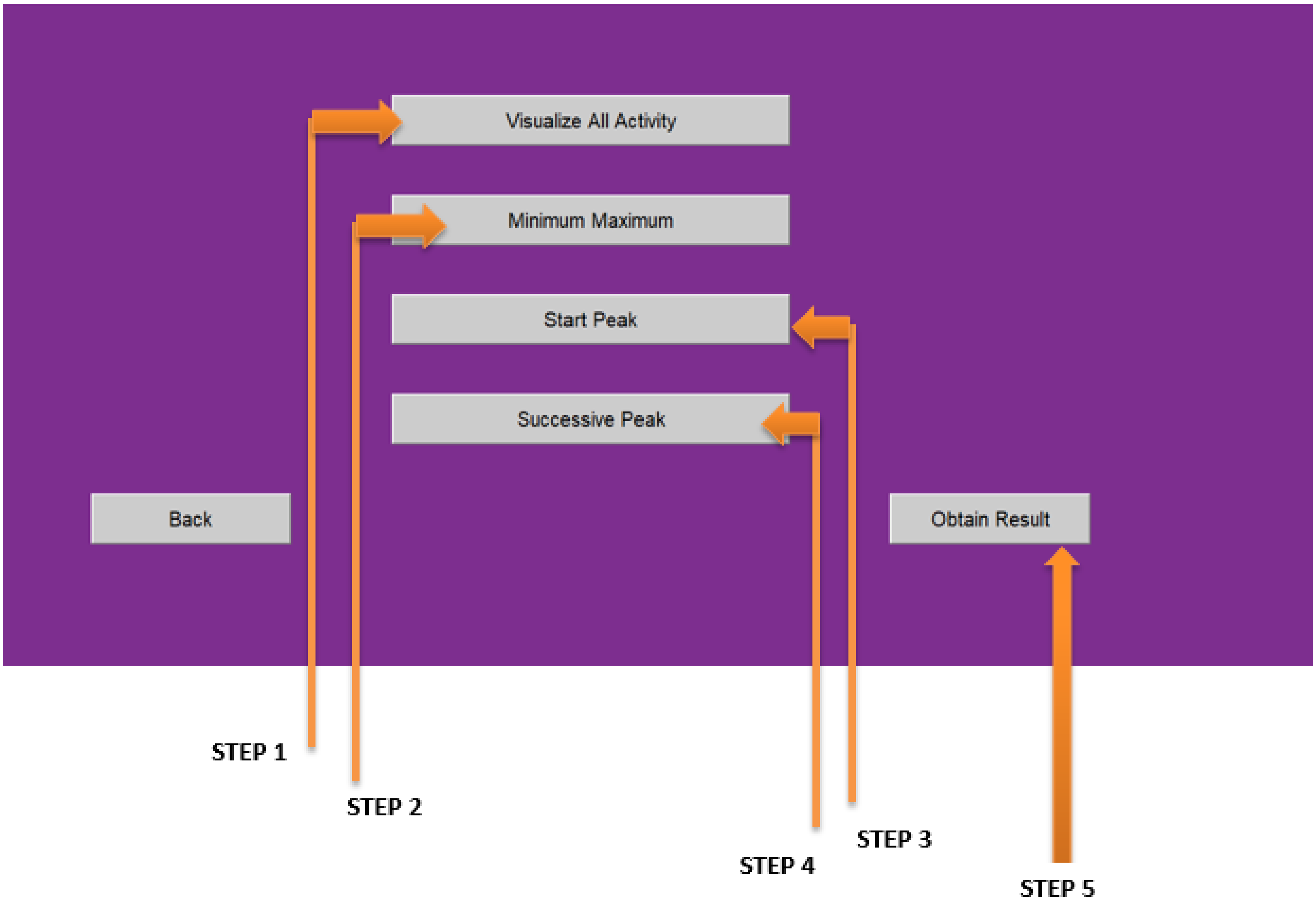


Fig 6(d) Fitness Analysis

Step 1: Graphical representation of all activities.

Step 2: Accuracy based on maximum-minimum peak.

Step 3: Accuracy based on the first peak.

Step 4: Accuracy based on successive peak.

Step 5: The final accuracy is displayed based on all operations performed.

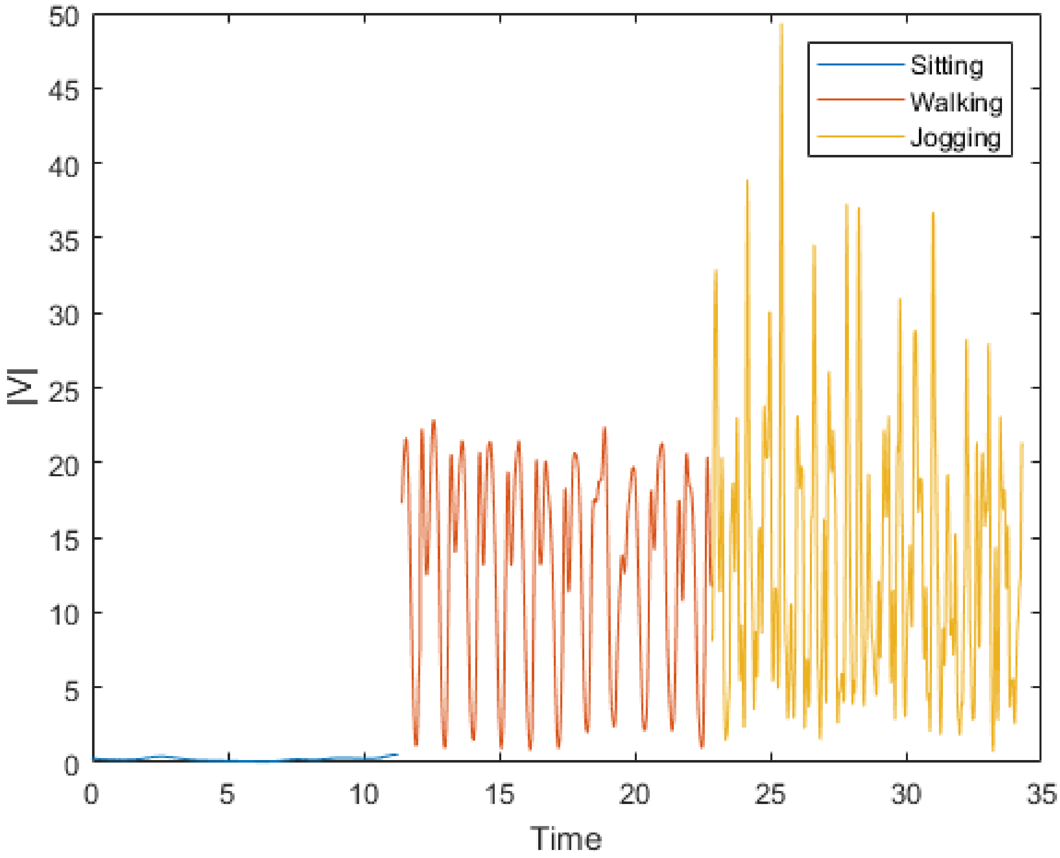


Fig 6(e) Combined visualization of all activities

MATLAB CODE:

```
global walking;
global jogging;
global sitting;
vSit=xlsread(sitting,'Sheet1','E1:E600');
vWalk=xlsread(walking,'Sheet1','E601:E1200');
vJog=xlsread(jogging,'Sheet1','E1201:E1800');
tSit=xlsread(sitting,'Sheet1','A1:A600');
tWalk=xlsread(walking,'Sheet1','A601:A1200');
tJog=xlsread(jogging,'Sheet1','A1201:A1800');

plot(tSit,vSit)
hold on
plot(tWalk,vWalk)
hold on
plot(tJog,vJog)
xlabel('Time')
ylabel('|V|')
legend('Sitting','Walking','Jogging')
```

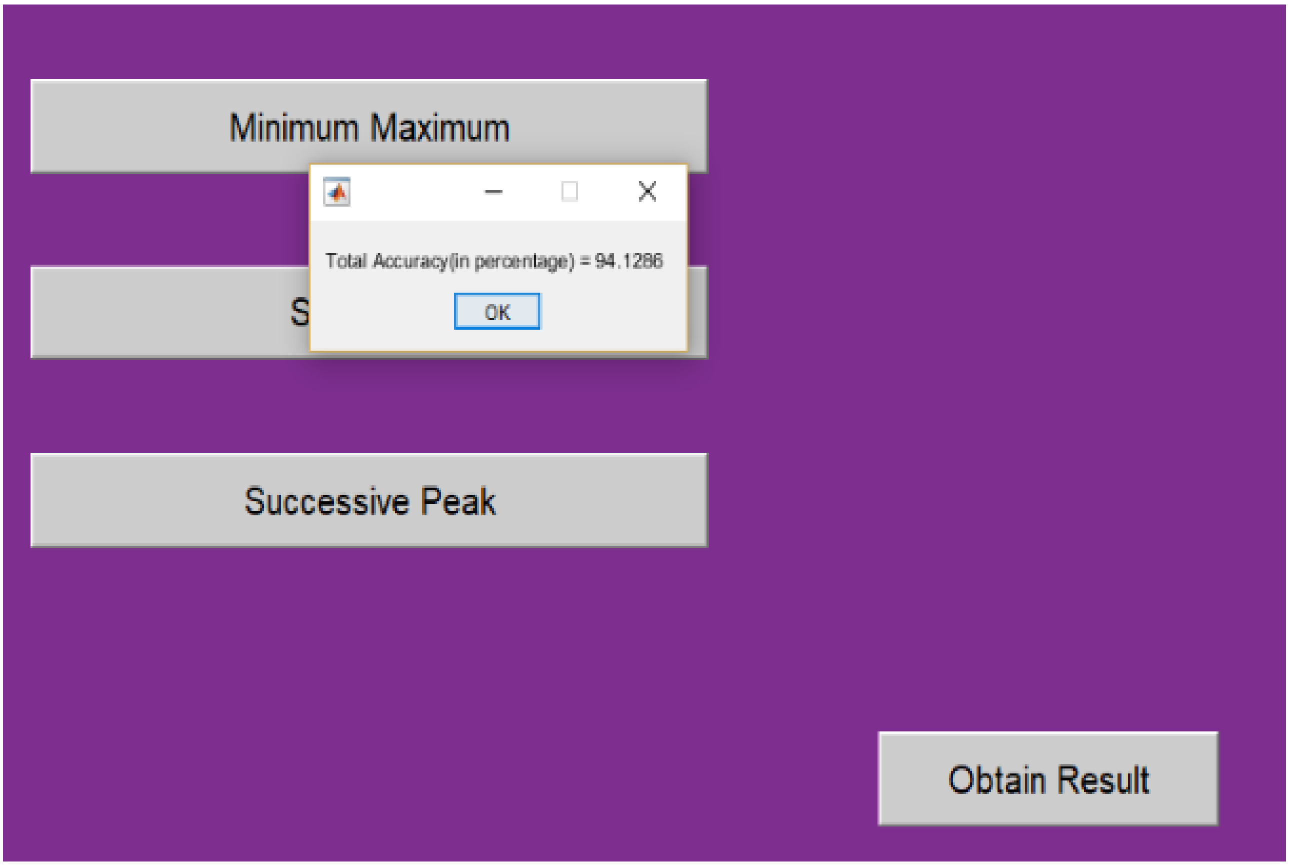


Fig 6(f) Obtaining fitness accuracy

MATLAB CODE FOR OBTAINING RESULT

```
function varargout = Comparision(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Comparision_OpeningFcn, ...
                   'gui_OutputFcn',  @Comparision_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Comparision_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);


function varargout = Comparision_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```

```
function minmax_Callback(hObject, eventdata, handles)
global acc1;
global acc2;
global acc3;                                    //Accuracy of 3 activities based on
run('MaxMinCompare.m')                          maximum,minimum peaks
run('MaxMinCompareJog.m')
run('MaxMinCompareSit.m')
acc1=(accuracy3+accuracySit3+accuracyJog3)/3
msgbox(sprintf('Accuracy(in percentage) = %g',acc1))


function startpeak_Callback(hObject, eventdata, handles)
global acc1;
global acc2;                                    //Accuracy of 3 activities based on start
global acc3;                                    peak
run('StartPeakCompare.m')
run('StartPeakCompareJog.m')
run('StartPeakCompareSit.m')
acc2=(accuracy1+accuracySit1+accuracyJog1)/3
msgbox(sprintf('Accuracy(in percentage) = %g',acc2))

function successive_Callback(hObject, eventdata, handles)
global acc1;                                    //Accuracy of 3 activities based on
global acc2;                                    successive peak  difference
global acc3;
run('SuccessivePeakCompare.m')
run('SuccessivePeakCompareJog.m')
run('SuccessivePeakCompareSit.m')
acc3=(accuracy2+accuracySit2+accuracyJog2)/3
msgbox(sprintf('Accuracy(in percentage) = %g',acc3))

function result_Callback(hObject, eventdata, handles)
global acc1;                                    //Total Accuracy of 3 activities
global acc2;
global acc3;
accuracy=(acc1+acc2+acc3)/3
msgbox(sprintf('Total Accuracy(in percentage) = %g %',accuracy))

function back_Callback(hObject, eventdata, handles)
run('InterfaceFitness.m')
```

CONCLUSION

- We have successfully collected the data through hardware.
- Implemented appropriate visualisation of the collected data.
- Filter chosen correctly filters out the noise in the channel and noise due to gravitational acceleration.
- The used time-window correctly compares the data.
- Expected feature analysis has been successfully implemented.(peak detection)
- Expected performance outcome has been achieved.
- An easy access GUI has been created.
- The developed system can classify the activities like sitting, jogging and walking.
- An informative message with percentage about the overall fitness can be made.
- We have ultimately reached our basic aim of the project to classify and analyse the different human activities and to produce performance result using the GUI.

FUTURE SCOPE

- Clear distinction between almost similar types of activities like StairCaseDown-and-Walking and Jogging-and-Running.
- Implementation of complete automated on-chip system for data collection and analysis.
- Implementation of wireless client-server architecture.
- Constant monitoring and implementation of variable-length time-window for better analysis.
- Invariant system analysis with respect to the position of the hardware.
- Different other human physical analysis like heartbeat, pressure, specific disease like asthma and other medical issues.
- Accounting other prospective of data classification, application of digital filters with variable filter size.
- Analysis over the frequency domain may improve the accuracy of the system.

REFERENCES

1. Aminian, K and Robert, Ph and Buchser, EE and Rutschmann, B and Hayoz, D and Depairon, M Physical activity monitoring based on accelerometry: validation and comparison with video observation, Medical & biological engineering & computing, 1999; 37:3–304.

2. Bao, Ling and Intille, Stephen S, Pervasive computing, Activity recognition from user-annotated acceleration data, 2004, Springer.

3.Casale, Pierluigi and Pujol, Oriol and Radeva, Petia, Human activity recognition from accelerometer data using a wearable device, Pattern Recognition and Image Analysis,2011;289, Springer.

4.Khan, Adil Mehmood and Lee, Young-Koo and Lee, Sungyoung Y and Kim, Tae-Seong, A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer, Information Technology in Biomedicine, IEEE Transactions on,2010;14:5–1166.

5.Lester, Jonathan and Choudhury, Tanzeem and Borriello, Gaetano, A practical approach to recognizing physical activities, Pervasive Computing, 2006; Springer.

6.Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter, 2011; 12:2–74.

7. Ravi, Nishkam and Dandekar, Nikhil and Mysore, Preetham and Littman, Michael L, Activity recognition from accelerometer data, AAAI, 2005; 5–1541.

8.Mannini, Andrea and Sabatini, Angelo Maria, Machine learning methods for classifying human physical activity from on-body accelerometers, Sensors, 2010; 10:2–1154.

9. Ruta, Dymitr and Gabrys, Bogdan, Classifier selection for majority voting, Informtion fusion, 2005; 6:1–63.

10. Gyllensten, Illapha Cuba and Bonomi, Alberto G, Identifying types of physical activity with a single accelerometer: evaluating laboratory-trained algorithms in daily life, Biomedicl Engineering, IEEE Trnsaction on, 2011; 58:9–2656.

11. Alelyani, Salem and Tang, Jiliang and Liu, Huan, Feature selection for clustering: A review, Data Clustering: Algorithms and Applications, 2013; 29.

12. Fujiki, Yuichi, iPhone as a physical activity measurement platform, CHI'10 Extended Abstracts on Human Factors in Computing Systems, 2010; 4315.

13.Krishnan, Narayanan C and Colbry, Dirk and Juillard, Colin and Panchanathan, Sethuraman, Real time human activity recognition using tri-axial accelerometers, Sensors, signals and information processing workshop, 2008.

14. Foerster, Friedrich and Fahrenberg, Jochen, Motion pattern and posture: correctly assessed by calibrated accelerometers, Behavior research methods, instruments, & computers, 2000; 32:3–450.

15. Westerterp, Klaas R, Assessment of physical activity: a critical appraisal, European journal of applied physiology, 2009; 105:6–823.

16. Poppe, Ronald, A survey on vision-based human action recognition, Image and vision computing, 2010

17. Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A, Cell phone-based biometric identification, Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on, 2010.