**Q3:**

**Proposal:** Recently, the wearable devices such as Jawbone Up, Nike FuelBand, Fitbit, and Apple Watch become more and more popular. There is a large amount of data about personal physical activity that has been collected automatically and inexpensively. These data are used to quantify self-movement, find patterns in behavior, and improve their health. However, there is a lack of a commonly used, standard dataset and established benchmarking problems for these physical activity data. The final goal of this proposal is to take advantage of large amount dataset collected from normal people and patients, to distinguish different behavior including normal activity (eg. walking, running, etc.) and pathological activity (eg. seizure, heart attack, etc.), in order to quantify how well they live. Once detected a pathological activity, the device will give alarm signal to avoid late treatment. In this proposal, for a preliminary study, we used machine learning techniques on a public physical activity data set that was collected from 3 inertial measurement units and a heart rate monitor in 9 participate while they were performing18 different activities.

**Goal:** To predict activities based on data from IMU sensory data on the hand, chest, and ankle, as well as hear rate of 9 participants.

**Data Source:**

<https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring>

Data set:

The 54 columns in the data files are organized as follows:   
1. timestamp (s)   
2. activityID (see below for the mapping to the activities)   
3. heart rate (bpm)   
4-20. IMU hand   
21-37. IMU chest   
38-54. IMU ankle 

The IMU sensory data contains the following columns:   
1. temperature (Â°C)   
2-4. 3D-acceleration data (ms-2), scale: Â±16g, resolution: 13-bit   
5-7. 3D-acceleration data (ms-2), scale: Â±6g, resolution: 13-bit   
8-10. 3D-gyroscope data (rad/s)   
11-13. 3D-magnetometer data (Î¼T)   
14-17. orientation (invalid in this data collection)   
  
List of activityIDs and corresponding activities:   
1 lying   
2 sitting   
3 standing   
4 walking   
5 running   
6 cycling   
7 Nordic walking   
9 watching TV   
10 computer work   
11 car driving   
12 ascending stairs   
13 descending stairs   
16 vacuum cleaning   
17 ironing   
18 folding laundry   
19 house cleaning   
20 playing soccer   
24 rope jumping   
0 other (transient activities)

**Preliminary Code:**

%% Step 1: Loading the Raw Data

data0{1}=importdata('PAMAP2\_Dataset/Protocol/subject101.dat');

data0{2}=importdata('PAMAP2\_Dataset/Protocol/subject102.dat');

data0{3}=importdata('PAMAP2\_Dataset/Protocol/subject103.dat');

data0{4}=importdata('PAMAP2\_Dataset/Protocol/subject104.dat');

data0{5}=importdata('PAMAP2\_Dataset/Protocol/subject105.dat');

data0{6}=importdata('PAMAP2\_Dataset/Protocol/subject106.dat');

data0{7}=importdata('PAMAP2\_Dataset/Protocol/subject107.dat');

data0{8}=importdata('PAMAP2\_Dataset/Protocol/subject108.dat');

data0{9}=importdata('PAMAP2\_Dataset/Protocol/subject109.dat');

%% Step 2: Preprocess data

data=[];

for i=1:9

data=[data;preprocess(data0{i})];

end

%% Step 3: Exploratory Data Analysis

plotsignal(data(1:size(data0{1},1),:));

%% Step 4: Data Splitting

% seperate xdata and ydata for training and testing dataset

t=data(:,1);

Xdata=data(:,3:end);

Ydata=data(:,2);

%Split traing data into training, validation and test.

cvpart = cvpartition(Ydata,'holdout',0.3);

Xtrain0 = Xdata(training(cvpart),:);

Ytrain0 = Ydata(training(cvpart),:);

Xtest = Xdata(test(cvpart),:);

Ytest = Ydata(test(cvpart),:);

cvpart2 = cvpartition(Ytrain,'holdout',0.3);

Xtrain = Xtrain0(training(cvpart2),:);

XValid = Xtrain0(test(cvpart2),:);

Ytrain = Ytrain0(training(cvpart2),:);

YValid = Ytrain0(training(cvpart2),:);

%% Step 5: Random Forest Model

bag = fitensemble(Xtrain,Ytrain,'Bag',40,'Tree',...

'type','classification');

%% Step 6: Validation

figure;

plot(loss(bag,XValid,YValid,'mode','cumulative'));

xlabel('Number of trees');

ylabel('Test classification error');

pred=predict(bag,XValid);

figure;

subplot(2,1,1)

plot(YValid);

ylabel('Actual activityID','k')

subplot(2,1,2)

plot(pred,'r');

ylabel('Predicted activityID')

xlabel('Time (s)')

activityID=unique(data(:,2));

ActualTime=zeros(1,length(activityID));

EstTime=zeros(1,length(activityID));

for i=1:length(activityID)

ActualTime(i)=length(find(YValid==activityID(i)))\*0.01;

EstTime(i)=length(find(pred==activityID(i)))\*0.01;

end

%% Step 7:Predicting the Test Sets

predtest=predict(bag,Xtest);

figure;plot(Ytest);hold on;plot(predtest,'r');

figure;

subplot(3,1,1)

plot(Ytest,'k');

ylabel('Actual activityID')

subplot(3,1,2)

plot(predtest,'r');

ylabel('Predicted activityID')

xlabel('Time (s)')

ActualTime=zeros(1,length(activityID));

EstTime=zeros(1,length(activityID));

for i=1:length(activityID)

ActualTime(i)=length(find(Xtest==activityID(i)))\*0.01;

EstTime(i)=length(find(predtest==activityID(i)))\*0.01;

end

subplot(3,1,3)

plot(activityID,ActualTime)

hold on

plot(activityID,EstTime,'r')

% handle missing values NaN

function data=handleNaN(data)

ind=find(isnan(data)==0);

for i=1:length(ind)

if i==length(ind)

data((ind(i)+1):length(data))=data(ind(i));

else

data((ind(i)+1):(ind(i+1)-1))=data(ind(i));

end

end

function plotsignal(data)

figure('position',[500 300 700 660])

subplot(4,3,1)

plot(data(:,1),data(:,2),'k')

axis tight

ylabel('activityID')

subplot(4,3,3)

plot(data(:,1),data(:,3),'k')

ylabel('Heart Rate')

axis tight

subplot(4,3,4)

% plot(data1(:,1),data1(:,5),'k')

% hold on

plot(data(:,1),data(:,5),'r')

% legend('Original signal','Filtered signal')

title('IMU band accel X')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,5)

% plot(data1(:,1),data1(:,6),'k')

% hold on

plot(data(:,1),data(:,6),'r')

% legend('Original signal','Filtered signal')

title('IMU band accel Y')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,6)

% plot(data1(:,1),data1(:,7),'k')

% hold on

plot(data(:,1),data(:,7),'r')

% legend('Original signal','Filtered signal')

title('IMU band accel Z')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,7)

% plot(data1(:,1),data1(:,15),'k')

% hold on

plot(data(:,1),data(:,15),'r')

% legend('Original signal','Filtered signal')

title('IMU chest accel X')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,8)

% plot(data1(:,1),data1(:,16),'k')

% hold on

plot(data(:,1),data(:,16),'r')

% legend('Original signal','Filtered signal')

title('IMU chest accel Y')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,9)

% plot(data1(:,1),data1(:,17),'k')

% hold on

plot(data(:,1),data(:,17),'r')

% legend('Original signal','Filtered signal')

title('IMU chest accel Z')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,10)

% plot(data1(:,1),data1(:,25),'k')

% hold on

plot(data(:,1),data(:,25),'r')

% legend('Original signal','Filtered signal')

title('IMU ankle accel X')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,11)

% plot(data1(:,1),data1(:,26),'k')

% hold on

plot(data(:,1),data(:,26),'r')

% legend('Original signal','Filtered signal')

title('IMU ankle accel Y')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

subplot(4,3,12)

% plot(data1(:,1),data1(:,27),'k')

% hold on

plot(data(:,1),data(:,27),'r')

% legend('Original signal','Filtered signal')

title('IMU ankle accel Z')

xlabel('Time (s)')

ylabel('Acceleration')

axis tight

function data=preprocess(data0)

% remove uselessdata from IMU sensory 5-7,and 14-17

notselectindx=[8:10 17:20 25:27 34:37 42:44 51:54];

indx=setxor(notselectindx,1:size(data0,2));

data1=data0(:,indx);

% handle missing values NaN

data1(:,3)=handleNaN(data1(:,3));

for i=4:size(data1,2)

data1(:,i)=handleNaN(data1(:,i));

end

ind=find(isnan(data1(:,5))==0&isnan(data1(:,15))==0&isnan(data1(:,25))==0&isnan(data1(:,3))==0);

data1=data1(ind,:);

% Signal Noise Removal:

% The Hampel filter helps to remove outliers from a signal without overly smoothing the data.

fs = 100;

data=data1;

for i=4:size(data1,2)

data(:,i)=medfilt1(data1(:,i),3);

end

% % remove the linear effect of time on heart rate

% data(:,3)=detrend(data(:,3),'linear');

**Preliminary Result:**

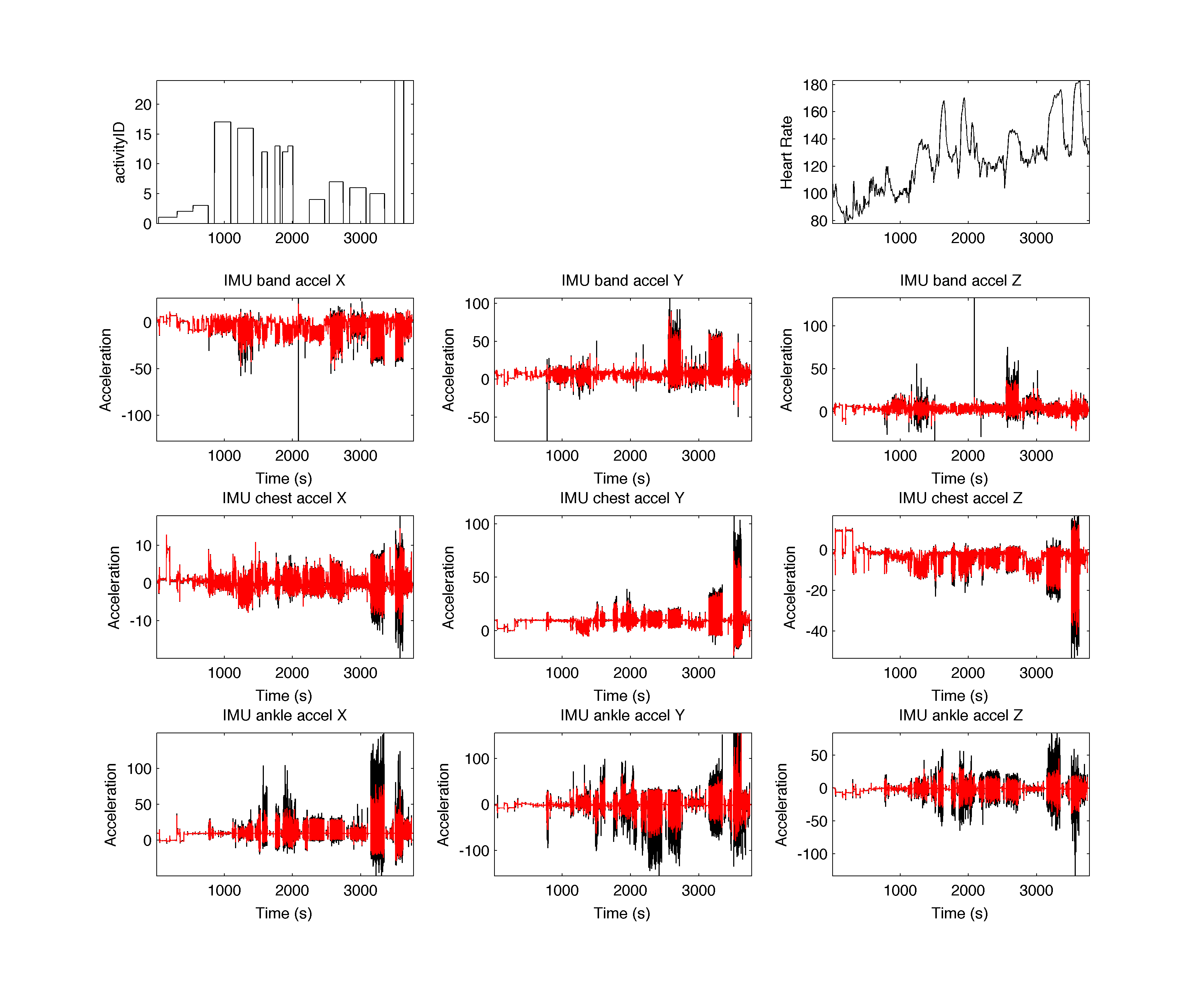
Figure1: Example of data set from subject 101. Black: original signal; Red: filtered signal after noise removal. 

Figure2:

**Future Improvement:**

1. **Feature extraction**

Due to the limitation of time, I only used preprocessed signal for input to the model. In the future, we should extract more features from the data to improve the performance. Eg. power in different frequency band, combining sensors of different placements, etc.

1. **Explore different machine learning techniques**

There are many classification models, et. neural network. In the future, we should study their disadvantage and advantage on the data, or combine different models to improve the performance.

Reference:

[1] A. Reiss and D. Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. The 16th IEEE International Symposium on Wearable Computers (ISWC), 2012.   
[2] A. Reiss and D. Stricker. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. The 5th Workshop on Affect and Behaviour Related Assistance (ABRA), 2012.