**­­CODING**

clc;

clear;

% addpath(genpath('X:\XX\XX'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set parameters %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% filter window size

n=3;

% gravitational acceleration

g=9.8;

% detector length (~3s)

D=800;

% detector tail length (fall detection)

T=80;

% detection threshold

th=120;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Acceleration data include 'accel\_x, accel\_y,%

% accel\_z' on x, y and z axis %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% read the data file

f=tsvread('XXX.tsv');

% calibration data

t0=f(:,2);

t1=t0-t0(1);

t2=datetime(t1./1000,'ConvertFrom','posixtime','Format','mm:ss.SSS');

t=seconds(timeofday(t2));

accel\_x=f(:,3)./g;

accel\_y=f(:,4)./g;

accel\_z=f(:,5)./g;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Extract the features of the signal %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% signal filter

ax=medfilt1(accel\_x,n);

ay=medfilt1(accel\_y,n);

az=medfilt1(accel\_z,n);

% SVM - signal vector magnitude

svm=sqrt(ax.^2+ay.^2+az.^2);

% Power

pow0=abs(svm).^2;

pow=[t pow0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Accident detection %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y=[t accel\_y];

detected=[];

% slide window across the signal

v=length(y)-D;

k=1;

while v~=0

v=v-1;

% find the anomalous targets (Fall detection)

if y(D,2)>-0.1 & y(D,2)<0.5

% calculate the signal power

for i=1:length(pow)

ifpow(i,1)==y(D,1)

% save the maximum power value in the set of samples (T)

pmax=max(pow(i-T:i,2));

end

end

% accident detection

ifpmax>th

detected(k,:)=y(D,:);

power\_value(k,:)=pmax;

k=k+1;

% anomaly detector (Window)

y=circshift(y,-(1:length(y)-D));

else

y=circshift(y,-(1:length(y)-D));

end

else

y=circshift(y,-(1:length(y)-D));

end

end

% output detection result

ifisempty(detected)==0

for i=1:length(pow)

for j=1:length(power\_value)

ifpow(i,2)==power\_value(j,1)

% save the time series of the accident

at(j,1)=pow(i,1);

end

end

end

% find duplicate values

accid\_t=unique(at);

% print the warning text

format='WARNING: Accident detected at around %6.3f s.\n';

fprintf(format,accid\_t);

% output result graph

figure;

subplot(2,1,1)

plot(t,accel\_y);

hold on

plot(detected(:,1),detected(:,2),'r+');

title('Accident detection on Y-axis');

xlabel('Time(s)');

ylabel('Acceleration(g)');

legend('Y-axis acceleration signal','Accident target');

subplot(2,1,2)

plot(t,pow0);

hold on

plot(at,power\_value,'ro');

title('Power of Accident detected');

xlabel('Time(s)');

ylabel('Power');

legend('Acceleration pow','Accident target');

else

% output power graph

figure;

plot(t,pow0);

title('Aceleration Power');

xlabel('Time(s)');

ylabel('Power');

fprintf('NO accident detected.\n');

end

clc;

clear;

addpath('function');

addpath('data');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set parameters %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% filter window size

n=3;

% gravitational acceleration (g)

g=9.8;

% threshold

th=6.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Acceleration data include 'accel\_x, accel\_y,%

% accel\_z' on x, y and z axis %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% read the data file

f=tsvread('XXX.tsv');

% calibration data

t0=f(:,2);

t1=t0-t0(1);

t2=datetime(t1./1000,'ConvertFrom','posixtime','Format','mm:ss.SSS');

t=seconds(timeofday(t2));

accel\_x=f(:,3)./g;

accel\_y=f(:,4)./g;

accel\_z=f(:,5)./g;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Preprocess acceleration data %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% signal filter for x, y, z axis (median filter)

ax=medfilt1(accel\_x,n);

ay=medfilt1(accel\_y,n);

az=medfilt1(accel\_z,n);

% output raw signal

figure;

subplot(2,1,1);

plot(t,accel\_x,'r');

hold on

plot(t,accel\_y,'g');

hold on

plot(t,accel\_z,'b');

title('Raw data');

xlabel('Time(s)');

ylabel('Acceleration(g)');

legend('Axis X','AxisY','Axis Z');

% output filtered signal

subplot(2,1,2);

plot(t,ax,'r');

hold on

plot(t,ay,'g');

hold on

plot(t,az,'b');

title('Filtered data');

xlabel('Time(s)');

ylabel('Acceleration(g)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Extract signal features %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SVM - signal vector magnitude

svm=sqrt(ax.^2+ay.^2+az.^2);

% output filtered signal

figure;

plot(t,svm);

title('Accel Signal');

xlabel('Time(s)');

ylabel('Acceleration(g)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Accident detection %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% anomaly detection

tag=find(svm>=th);

% output anomaly targets

figure;

subplot(2,1,1);

plot(t,svm);

hold on

plot(t(tag),svm(tag),'r+');

title('Anomaly detection');

xlabel('Time(s)');

ylabel('Acceleration(g)');

legend('Acceleration signal','Anomaly target');

% accident detection

for i=1:length(tag)

data{i}=svm(tag(i)-3:tag(i));

med(i,:)=mean(data{1,i});

%stand(i,:)=std(data{1,i});

end

for j=1:length(med)

if med(j)<6

tag(j)=0;

end

end

tag(tag==0)=[];

% output accident targets

subplot(2,1,2);

plot(t,svm);

hold on

plot(t(tag),svm(tag),'ro');

title('Accident detection');

xlabel('Time(s)');

ylabel('Acceleration(g)');

legend('Acceleration signal','Accident target');