# Secure Beat - Final Report

A report submitted as part of completion of
the Engineering Design course
(Elec Eng 4OI6)

April 2018

## Contents

1.0 Introduction ....... 3
2.0 Background....... 3
    2.1 Problem Definition ....... 3
    2.2 Solution....... 3
3.0 Design ....... 3
    3.1 The Electrocardiogram ....... 3
    3.2 Initial Digital Signal Processing ....... 4
    3.3 Signal De-noising and Feature Space Reduction....... 4
    3.4 Identification....... 4
    3.5 Hardware Enclosure ....... 5
4.0 Execution....... 5
    4.1 Implementation....... 5
    4.1.1 The Electrocardiogram ....... 5
    4.1.2 Initial Digital Signal Processing ....... 5
    4.1.3 Signal Denoising & Feature Space Reduction....... 6
    4.1.4 Identification....... 6
    4.1.5 Hardware Assembly ....... 6
    4.2 Simulation....... 7
    4.3 Problems Encountered....... 8
5.0 Future Plans ....... 8
6.0 Conclusion....... 8
7.0 References ....... 9
8.0 Appendix ....... 10
    8.1 Figures ....... 10
    8.2 Code....... 11

# 1.0 Introduction

The Secure Beat is an innovative solution to create a new level of security using the user's unique heartbeat. With our device, the user will be able to unlock their laptop (or other electronic devices) with their heart rhythm by simply wearing the electrodes. In this report, a first stage prototype is presented to the reader. Our vision for this technology is to see it embedded in a smart-watch. It will be further improved to track and analyze a variety of information related to the user's health; some of these include step-counting, sleep patterns, and fitness indices. All this information will be displayable on the smartwatch's screen or a phone using an integrated mobile app. Overall, the Secure Beat will provide the user with a new safe and convenient biometric security measure.

# 2.0 Background

## 2.1 Problem Definition

In today's world, our data is all over the internet and protecting ourselves, and our information is no longer an easy task. It is necessary to have strong passwords on all of our accounts and devices to protect ourselves from a wide variety of threats ranging from credit card fraud to intellectual property theft. Protecting devices that we use frequently with strong passwords can be cumbersome as increasing password strength reduces convenience. Our goal with Secure Beat is to make use of a unique physical identifier built into every person to strengthen security while maximizing ease of use.

## 2.2 Solution

Secure Beat is a hardware and software package that demonstrates how heart rhythm authentication could be integrated into a convenient consumer product. It creates a database of ECG signals for its users, uses machine learning algorithms that are continuously improving and achieving higher accuracy at identifying the right users. Secure Beat is also an open source device that can be further improved and packaged into a smaller circuit and integrated into existing technologies such as smart watches or Fitbit-like devices.

# 3.0 Design

## 3.1 The Electrocardiogram

During the first stage of our project, we looked at different biometric identifiers including fingerprints, DNA, brain signals (EEG) and the heart signal (ECG). The ECG signal was finally chosen for its uniqueness and ease of acquisition. The formation of an ECG signal is a function of the anatomy of the heart and its surrounding tissues, which make it unique for each individual.

To collect the ECG signal, the Einthoven triangle concept of measuring electrical heart activity was used, which makes the following assumptions:

- The heart is at the centre of an equilateral triangle formed by the two shoulders and the pubis.
- The chest is a volume conductor while the arms and legs are linear conductors
- The entire electrical activity of the heart can be represented by a dipole with a changing amplitude and direction.

In Figure 1, we can see the Einthoven triangle with the different lead configurations. Lead I configuration was chosen for our device as it requires the fewest electrodes while still providing a reliable signal. In Lead I configuration, three electrodes are needed: left arm, right arm and leg. This configuration gives us a very good view of what is going on from left to right in the heart, but a poor view of events moving up or down. After the signal is picked up, it is filtered and amplified using an ECG pre-processing circuit.

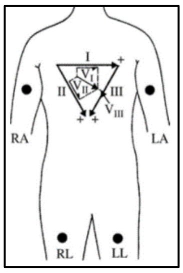


*Figure 1: Einthoven Triangle*

The result is a signal resembling Figure 2. This is called a PQRST complex.
The PQRST complex features in a Lead I signal can be broken down into the following electrical and mechanical processes:

- P: Represents the atrial depolarization and contraction
- QRS: Represents the ventricular depolarization and ventricular contraction
- T: Start of ventricular repolarization and relaxation.

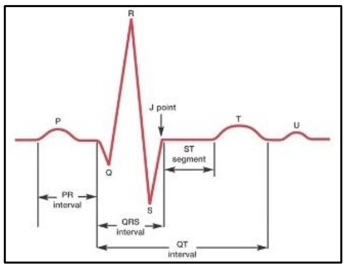


*Figure 2: PQRST Complex*

## 3.2 Initial Digital Signal Processing

Once the ECG signal has been acquired and digitized (Figure 3), there are three key preprocessing stages required to prepare the signal for comparison. The first stage is identifying peaks and using those to split the signal into its PQRST complexes. The second stage is vertical drift correction as shown in Figure 4. This aligns the signal allowing each PQRST complex in a sample to be directly compared. The third step is to compute the average of all the complexes in a signal and use that in the signal de-noising and Feature Space Reduction stages.

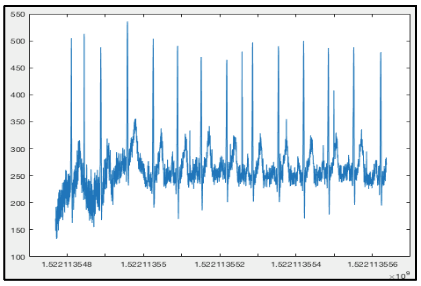


*Figure 3: Acquired ECG Signal*

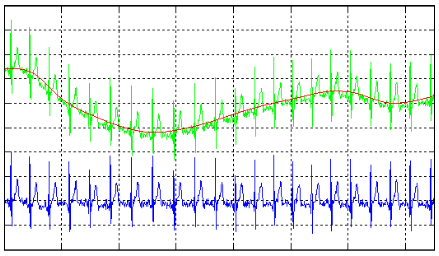


*Figure 4:Vertical Drift Correction*

## 3.3 Signal De-noising and Feature Space Reduction

As opposed to other data analysis tool like the Fourier transform, the Wavelet transform captures significant information from abrupt changes in certain signals (for example an ECG) over a finite duration of time. In relation to the project, the discrete wavelet transform was used for feature space reduction as it is favorable for de-noising and signal compression.

## 3.4 Identification

Linear Discriminant Analysis (LDA) is a simple to use and mathematically robust classification method that often produces models highly accurate models. LDA is based upon the concept of searching for a linear combination of variables (predictors) that best separates two classes (targets).
In layman's terms the LDA is a linear separation between two clusters of different points on a 2d graph. The line separates yes and no clusters of PQRST entries, if yes you are logged in, if no you are not.

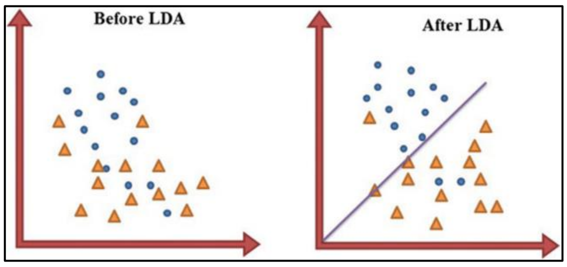


*Figure 5: LDA Clusters*

### 3.5 Hardware Enclosure

The initial motivation for the case design came from the layout of an oven interior. The idea was to stack the battery, the Thing microcontroller and the ECG AD8232 board to minimize space requirements. The initial sketch (Figure 6). Due to problems encountered during data acquisition with the Thing, we had to use an Arduino UNO instead which substantially changed the dimensions of the case and another design had to be chosen. The final design used is called the Molebox and was adopted from the Thingiverse (Figure 8). This design consists of compartments for both the Arduino UNO and the ECG AD8232 board. With this design, there is enough room for flow of extra wires, the Lead I ECG electrodes and the serial cord to the computer.

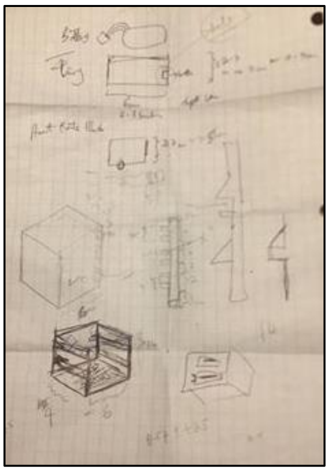


*Figure 6: Initial Case Sketch*

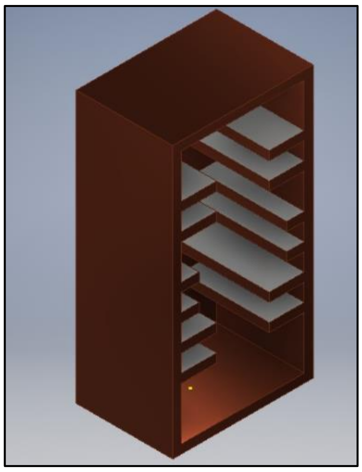


*Figure 7: Initial Case CAD Model*

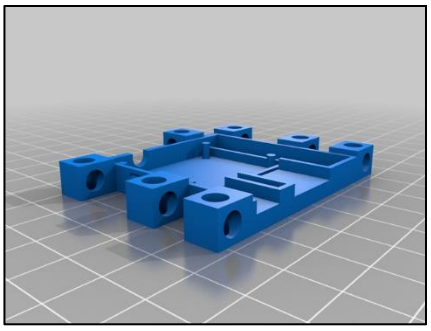


*Figure 8: CAD Model of Molebox (Final Design)*

## 4.0 Execution

### 4.1 Implementation

#### *4.1.1 The Electrocardiogram*

As discussed earlier, the Lead I configuration was used to obtain the ECG signal. Sticky electrodes were placed on the right wrist, left wrist and right ankle. Analog processing was performed on the signal on board the AD8232 SparkFun Single Lead Heart Rate Monitor. The amplified filtered signal was then transmitted to the microcontroller (Arduino) for analog to digital conversion. Then the digitized signal was serially transmitted to a computer for digital signal processing. A Python script was run on the computer to receive the data and record it in a CSV file to be read into Matlab for the following steps.

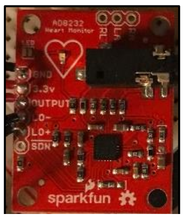


*Figure 9: The AD8232 board*

#### *4.1.2 Initial Digital Signal Processing*

The first stage in our digital signal processing is to separate each PQRST complex in one recording. This was done by identifying the R-peak in each of the signals and collecting 80 samples to the left of the peak followed by 170 samples to the right. This standardizes the complexes so that each one is the same size and the peak occurs at the same location. The number of samples taken to the left and the right of the peak was determined by trial and error. The second processing stage is to remove the vertical drift of the ECG. This was done by finding the mean of each PQRST complex and subtracting it from each sample point in the complex. Then all the complexes in one recording are averaged together to form the signal used in the rest of the process.

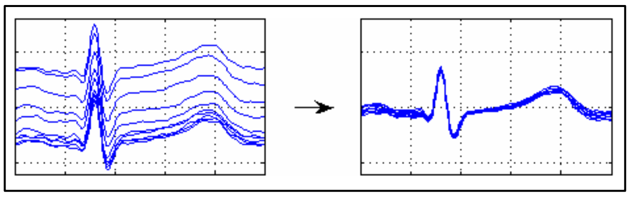


Figure 10: Vertical Drift Correction

### 4.1.3 *Signal Denoising & Feature Space Reduction*

A wavelet transform was used to analyze the 250 samples in the resulting signal from 4.1.2. Daubechies wavelet (db3) in the Matlab wavelet toolbox was implemented as it most closely resembles a PQRST complex. After applying the wavelet transform the output indicated that the first 45 sample points were the most relevant for comparing PQRST complexes from different recordings.

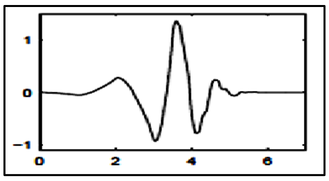


Figure 11: Daubechies Wavelet

### 4.1.4 *Identification*

A database of processed signals containing recordings from the user and from others was created to train an instance of Matlab's Linear Discriminant Analysis (LDA) tool. The LDA tool was then used to classify a newly processed signal as the user or not the user based on the training database. The result of the classification is then output to a file for a Python script to read and display an image of an unlocked or locked lock based on whether or not the user was authenticated.

### 4.1.5 *Hardware Assembly*

The initial case and final case designs were 3D printed and can be seen in Figures 12 and 13 below.

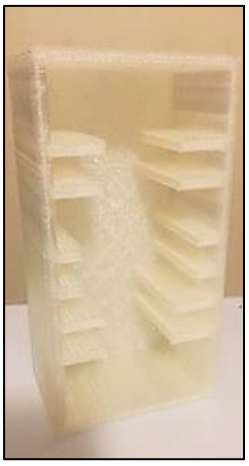


Figure 12: 3D Printed Initial Case

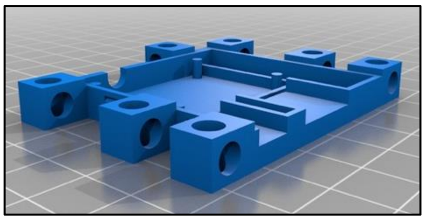


Figure 13: CAD Model of Molebox (Final Design)

Figure 15 displays the final hardware setup fully assembled:

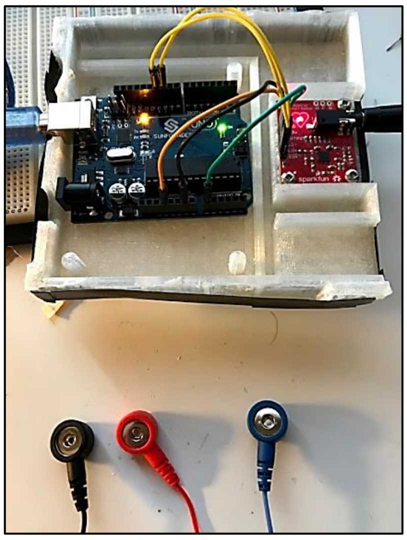


*Figure 15: The Arduino and The ECG Shield Assembled in 3D Printed Box*

## 4.2 <u>Simulation</u>

To allow for parallel progression of the software and hardware development of the project, an ECG database from MIT was used to help develop the algorithm within Secure Beat. Thirty ECG sets from different users were downloaded, all having regular heart rhythms, with up to ten ECG files for each; an additional ten sets were downloaded from users with irregular rhythms. This data was used as the training set for the LDA classification, as well as for testing purposes. After the hardware was able to acquire the project teams' ECG files, we were able to add it to the ECG sets from MIT. After testing extensively, the accuracy of our algorithm for all three datasets was 94.37%. The following procedure was used to determine the accuracy:

1. A test-subject is chosen in which their ECGs are labelled as being "<u>correct</u>" and is then loaded into the training set.
2. All other users are considered "<u>incorrect</u>" and are labelled as such, then added into the training set.
3. Once the set is created, one file from the test-subject ("correct" user) is removed from the training set.
4. The removed ECG is now tested against the training set. If it is successful, it will output a '1' indicating the algorithm classifies the unknown ECG as the test-subject.
5. Repeat steps 3 and 4 until all test-subject's files have been tested individually against the data set.
6. One file from any other subject ("incorrect" user) is removed from the training set.
7. The removed ECG is now tested against the training set. If the algorithm works as intended, it will output a '0' indicating the algorithm classifies the unknown ECG as not the test-subject.
8. Repeat steps 6 and 7 until all other subjects' files have been tested individually against the data set.
9. Select a new test-subject as the "correct" and repeat steps 1 to 8 until all users have been tested.

In Figure 14, we have separated our testing into regular and irregular heart rhythms, in which the statistics for correct user and incorrect user identification accuracy is shown. Although there were fewer users for irregular rhythms, it identified the user correctly 10% more often than regular. In the available dataset, each irregular set has more ECG files than the regular sets, which affects the training set that is used within the machine learning algorithm. The more ECG files used in the set will increase the likeliness to identify correctly, hence the higher overall accuracy for irregular heart users.

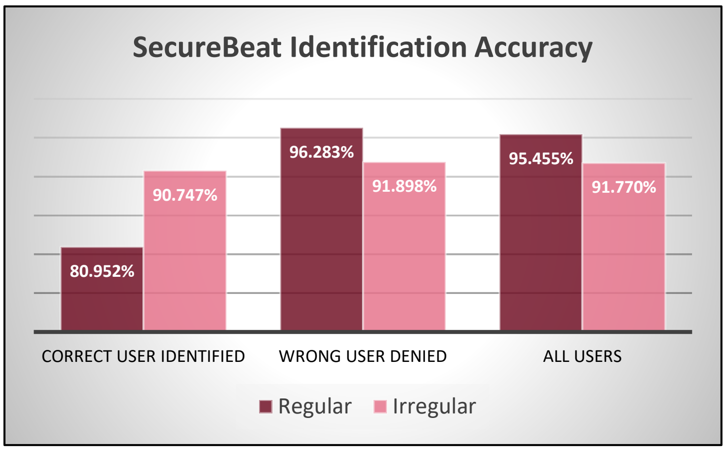


*Figure 14: Statistics Chart Explaining Identification Results*

### 4.3 <u>Problems Encountered</u>

- Wireless communication and data acquisition of ECG signals with the Thing microcontroller
- Inability to get rid of the third electrode (right ankle ground) for easier signal acquisition
- Dry electrodes were not integrated as they are either prohibitively expensive for the scope of the project or require too much research to design in time.
- Due to the problems above, some of our original goals such as designing the fitness app and smart watch were not met because of the time constraint.

## 5.0 Future Plans

We aim to integrate our product into smartwatches providing the user with a unprecedented level of biometric authentication and secure convenience. Enabling users to log in to their devices and services by simply touching their watch. The following steps will need to be taken to achieve this goal:

1. Design a circuit (comparator) to replace third electrode
2. Replace the sticky electrodes with affordable and reliable dry electrodes.
3. Implement wireless communication with the device that is being secured.
4. Moving the digital signal processing and identification algorithm from the computer onto the smartwatch.

## 6.0 Conclusion

As the world's technology grows exponentially, it is increasingly important for everyone to keep their information secure. This project has demonstrated that an ECG signal can be used as a secure and convenient biometric identifier. The completed first stage prototype can acquire a signal and serially transmit it to a computer for processing and identification. Moving forward the project aims to shrink the signal acquisition and processing hardware to package it in a wearable form factor to simplify the ever-important task of keeping private information secure yet convenient to access.

## 7.0 References

- Elec Eng 4BD4 Lectures, Dr. Hubert deBruin.
- Cardiology Explained, Conquering the ECG, https://www.ncbi.nlm.nih.gov/books/NBK2214/
- Lugovaya T.S. Biometric human identification based on electrocardiogram. [Master's thesis] Faculty of Computing Technologies and Informatics, Electrotechnical University "LETI", Saint-Petersburg, Russian Federation; June 2005.
- Effective Feature Extraction of ECG for Biometric Application, Kiran Kumar Patro, P.Rajesh Kumar
- Thingiverse Molebox, https://makerware.thingiverse.com/thing:2648692
- http://chem-eng.utoronto.ca/~datamining/dmc/classification.htm
- http://chem-eng.utoronto.ca/~datamining/dmc/flash/LDA_flash.html
- https://www.youtube.com/watch?v=moqPyJQHR_s
- https://www.youtube.com/watch?v=t-V6y5jwcsA
- https://www.researchgate.net/figure/Linear-discriminant-analysis_fig10_2880025280
- Carreiras C., Lourenço A., Silva H., Fred A. (2013) A Unifying Approach to ECG Biometric
- Recognition Using the Wavelet Transform. In: Kamel M., Campilho A. (eds) Image Analysis and Recognition. ICIAR 2013. Lecture Notes in Computer Science, vol 7950. Springer, Berlin, Heidelberg

# 8.0 Appendix

## 8.1 Figures

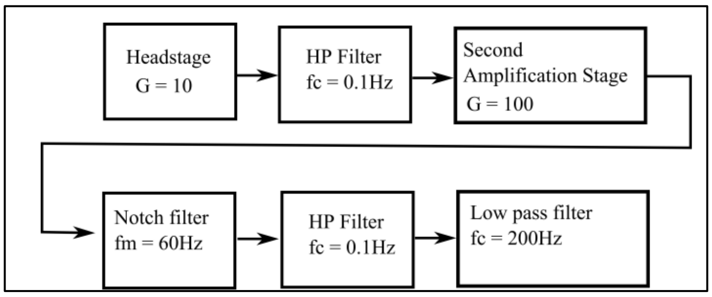


*Figure A1: ECG Analog Signal Processing Stage*

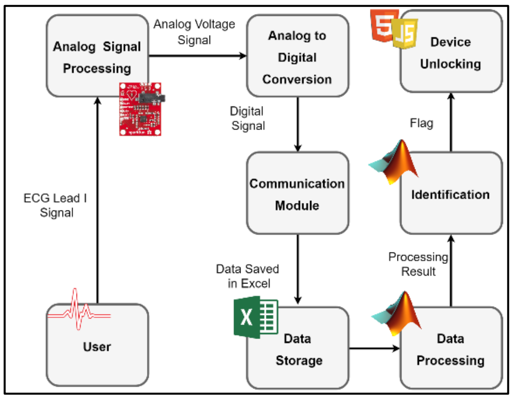
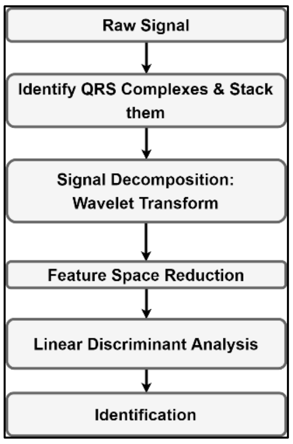


*Figure A2: System Overview*



Raw Signal
Identify QRS Complexes & Stack them
Signal Decomposition: Wavelet Transform
Feature Space Reduction
Linear Discriminant Analysis
Identification

*Figure A3: Software Algorithm*

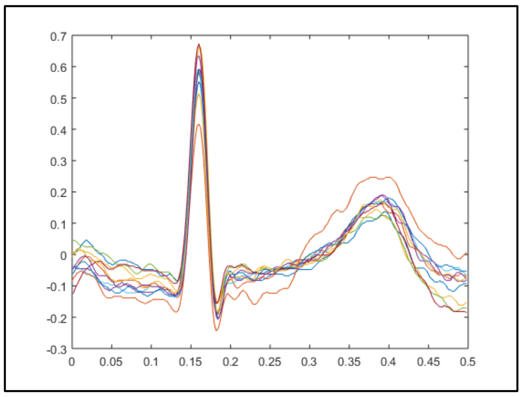


*Figure A4: Stacked and aligned PQRST Complexes*

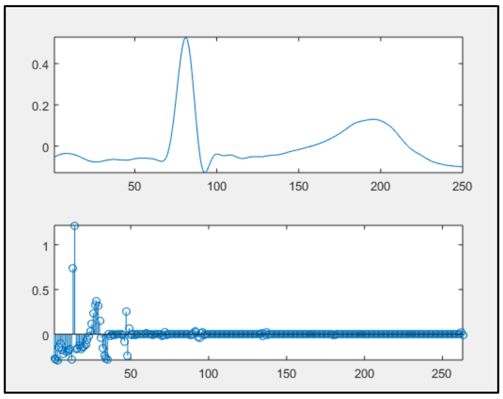


*Figure A5: Wavelet Transform of Mean Signal*

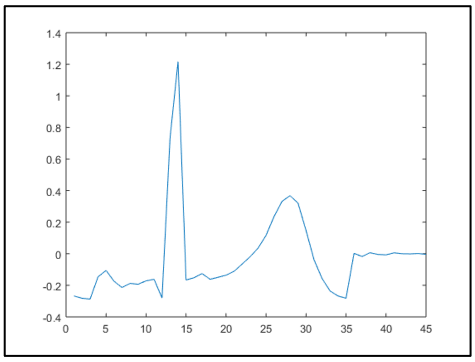


*Figure A6: Mean Wavelet Transformed Signal With Feature Space Reduction*

## 8.2 Code

```
                     Matlab Code:
   Below is the script used in Capstone Demo, createTrain() and
                   ecgFiltWav() functions.


%% For Capstone Demo
% For demonstration purposes, the entire MIT database was created
into a
% training set. To test data to see accuracy of identication
algorithm we
% first need to create the data set using createTrain(ID); where
ID is the
% person who is correct - This will change per user
%
% After the set is created, you will remove different files from
the
% training set and test them against the data to see if it is
correct.

%% Step 1: Create Training Set

% NOTE: After step 1 has been completed once, comment out this
from the code
% to allow for faster computation speed. However, everytime a new
user is
% being identified, it will need to run again

% %ID Must be a string.
% Takes roughly 3 seconds to run for 100 ecg files
tic
ID = '1';
[trainingSet,label,amtCorrect] = createTrain(ID);
time_1 = toc;

%% Step 2: Test Training Set

% % We will now remove one from the set test it within the data
set.
% % NOTE: The correctID value must be <= amtCorrect
clc;
j = 0;
tic
correctID = 8;
Answer = testCorrectPerson(trainingSet,label,correctID);
time_2 = toc;
% testCorrectPerson() will also output .txt file for python
%
% This will find the accuracy of denying users that are not wanted
for i = 1:116
    tic
    correctID = i;
    Answer                                                       =
testInCorrectPerson(trainingSet,label,correctID+amtCorrect);
    if (Answer == 1)
        j = j + 1;
    end
    time_3 = toc;
end
Percent = 100*(j/116)
```

```matlab
function [set,label,correct,complete] = createTrain(ID);

%% Find Folder
tic
% Specify the folder where the files live.
myFolder = 'data'; %This is a relative folder of where your data
set is -
% Change 'data' to your path ie. 'info/ecg'
% Check to make sure that folder actually exists.  Warn user if it
doesn't.
if ~isdir(myFolder)
  errorMessage = sprintf('Error: The following folder does not
exist:\n%s', myFolder);
  uiwait(warndlg(errorMessage));
  return;
end
% Get a list of all files in the folder with the desired file name
pattern.
filePattern = fullfile(myFolder, '*.csv'); % Change pattern to
obtain csv's
theFiles = dir(filePattern); % Finds amount of files in folder
%% Initial Structure of data
i = 0; j=0;
for k = 1 : length(theFiles)
  baseFileName = theFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName);
  ecg(k).data = csvread(fullFileName,2,0); % copy ecg data to
structure
  ecg(k).name = baseFileName; % copy ecg name to structure

end
%% Creating LDA Training set
% Error check based on name
if (length(ID)==1)
    A = ID(1);
    B = '_';
    ID = strcat(A,B);
end

for h = 1 : length(theFiles)
    % Change based on which value you want to identify
    % All correct ecgs will be stored in this set
  if ((ecg(h).name(4) == ID(1)) && (ecg(h).name(5) == ID(2)))
      i = i + 1;
      wavlet =  ecgFiltWav(ecg(h).data);
      training_1(i, 1:45) = wavlet;
      lbl_1(1,i) = {'correct'};
      % All other peoples will be stored into this data set as
incorrect
  else
      j = j + 1;
      wavlet = ecgFiltWav(ecg(h).data);
      training_2(j, 1:45) = wavlet;
      lbl_2(1,j) = {'incorrect'};
  end
end

%% Finalizing LDA Training set for output
%Concatenate person 1 and not person 1 data
set = vertcat(training_1,training_2);
label = horzcat(lbl_1,lbl_2);
correct = i;
end
function [output,time] = ecgFiltWav(ecg1);
```

```
%peaks are all greater than 0.4 so if we just store the time/data
pairs
%where the data is above 0.4 we can find the peaks then collect
samples 80
%before the peak 170 after the peak then plot those on top of each
other

%findPeaks locates where the peak is within the segment and then
centralized the data around it
[pks,locs]                                                         =
findpeaks(ecg1(:,2),ecg1(:,1),'MinPeakProminence',0.27,'MinPeakDi
stance',0.5 ,'Annotate','extents');
segments = zeros(length(pks),250);
emeandist = zeros(length(pks),1);

for i = 1:length(pks)
    j = find(ecg1(:,1) == locs(i));
    k = j - 80;
    if (k+249 > length(ecg1(:,1)))

        break;
    elseif (k < 1)
        k = 1;
    end
    segments(i, :) = ecg1(k:k+249,2);
    segments(i, :) = segments(i, :) - mean(segments(i, :));
    %plot(ecg1(1:250,1), ecg1(k:k+249,3))
    %plot(ecg1(1:250,1), segments(i,:))
    %hold on;

    %% wavelet
    data=(segments(i,:)).';
    waveletName = 'db3';
    waveletLevel = 3;
    dwtmode('ppd');
    [C,L] = wavedec(data,waveletLevel,waveletName);
    wav(i,:)=C(1:45);
end

%plot(ecg1(1:250,1), mean(segments))

for l = 1:length(pks)
     emeandist(l) = norm(mean(segments) - segments(l,:));
end

output = mean(wav);
time = ecg1(1:250,1);

end
```