SIGNAL PROCESSING FOR PHYSIOLOGICAL SENSORS

Nazlı İrem Çamurlu

Faculty of Engineering and Natural Sciences,Computer Science and Engineering, Junior

cirem@sabanciuniv.edu

Deniz Küçükahmetler

Faculty of Engineering and Natural Sciences, Computer Science and Engineering, Junior

dkucukahmetler@sabanciuniv.edu

Ömer Said Öztürk

Faculty of Engineering and Natural Sciences,Electronics Engineering, Sophomore

omersaid@sabanciuniv.edu

Berk Açıkgöz

Faculty of Engineering and Natural Sciences, Electronics Engineering/Computer Science and Engineering, Junior

berkacikgoz@sabanciuniv.edu

Ozan Biçen

Faculty of Engineering and Natural Sciences, Electronics Engineering

Abstract

This project consists of two parts. In the first part, our dataset consisted of 26 experiments and each experiment was divided in itself to three different tasks. The ‘Word Generation’ task was taken into account when performing any analysis about the dataset. The aim of using correlation plots and histograms is to further expand the visualization of the data being studied and take further steps from that point. Another method that was implemented was to plot autocorrelation plots to see the correlation between channels in different lags. We have also tried to identify people by using their psychological signals. By using this dataset, we have tried to classify and identify people both in time domain and frequency domain, using machine learning (ML) and artificial neural network (ANN) techniques. In the second part of this project, the feasibility of cryptographic key generation for Body Area Network security was compared for two physiological signals, ECG and PPG. The resulting bit strings were compared in terms of two metrics, distinctiveness and error rates.

Keywords: Machine Learning, Deep Learning, Physiological Sensors, Cryptographic Key Generation

1. Introduction

"Functional Near-Infrared Spectroscopy (fNIRS) maps human brain function by measuring and imaging local changes in hemoglobin concentrations in the brain that arise from the modulation of cerebral blood flow and oxygen metabolism by neural activity."(Yücel et al., 2017).Meryem A.Yucel and her colleagues have analyzed the cruciality of using fNIRS for neuroimaging for many different studies that require movement in them. For the purposes of analyzing mobility studies,fNIRS and EEG were chosen to be perfect choices for neuroimaging.(Yücel et al., 2017). Signal processing of these neuroimaging techniques has been done using various machine learning and deep learning algorithms in different studies. In this manner, our project aims to make contributions to signal processing of fNIRS and EEG with our findings. In order to achieve this, datasets belonging to fNIRS and EEG were considered(*Motion Artifact Contaminated fNIRS and EEG Data* 2014)(refs) . Then, the most comprehensive dataset found was used (Shin et al., 2018). In this dataset, simultaneous acquisition of fNIRS and EEG data were recorded during different protocols. There are three main tasks performed and represented by their own datasets.

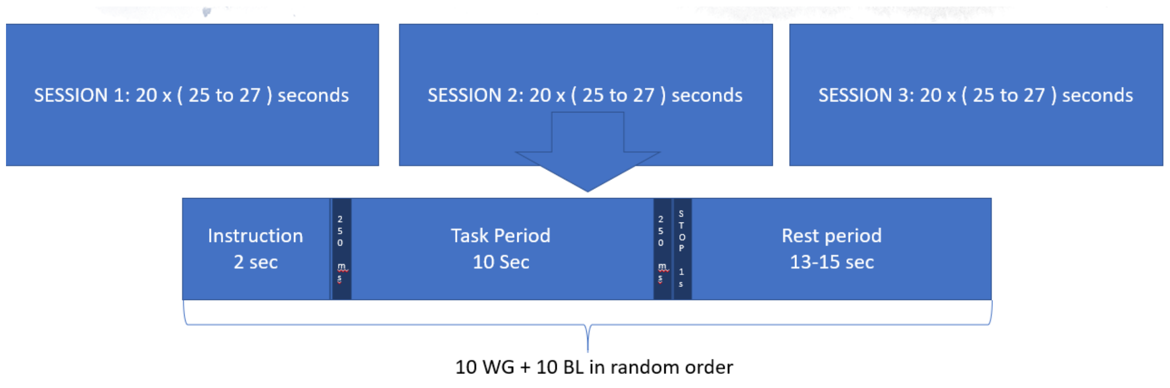
Three tasks in the dataset used are as follows:

1. Dataset A: N-back
2. Dataset B: Discrimination Selection Response (DSR)
3. Dataset C: Word Generation

   There are 2 kinds of trials in this dataset: word generation (WG) and baseline(BL). In WG trial participants were asked to think of a word starting with the letter shown on the screen. Participants were asked to think of a word starting with the letter shown on the screen. In BL trial, there was a fixation cross at the middle of the screen and participants were asked to look at this cross during the task period.

Dataset C which corresponds to the word generation task thought to be the most useful dataset to apply machine learning algorithms and ANN. Because there is a significant difference between two kinds of trials (BL and WG). However, in N-back task, the difficulty of task was varying and in DSR, during whole task period participants were doing a momentary and easy task which is pushing a button when seeing " O " sign on the screen.

Word generation task has been made for 24 participants in total. For each participant, three sessions were performed. Each session contains ten trials of the word generation (WG) and baseline (BL). For every 20 trials, WG and BL trials were randomly repeated. Therefore, there are 60 trials for each participant (3 x (10 WG + 10 BL)). Each trial takes 25-27 seconds.


SESSION 1: 20 x (25 to 27) seconds
SESSION 2: 20 x (25 to 27) seconds
SESSION 3: 20 x (25 to 27) seconds
Instruction 2 sec
250 ms
Task Period 10 Sec
250 ms
STOP 1 s
Rest period 13-15 sec
10 WG + 10 BL in random order

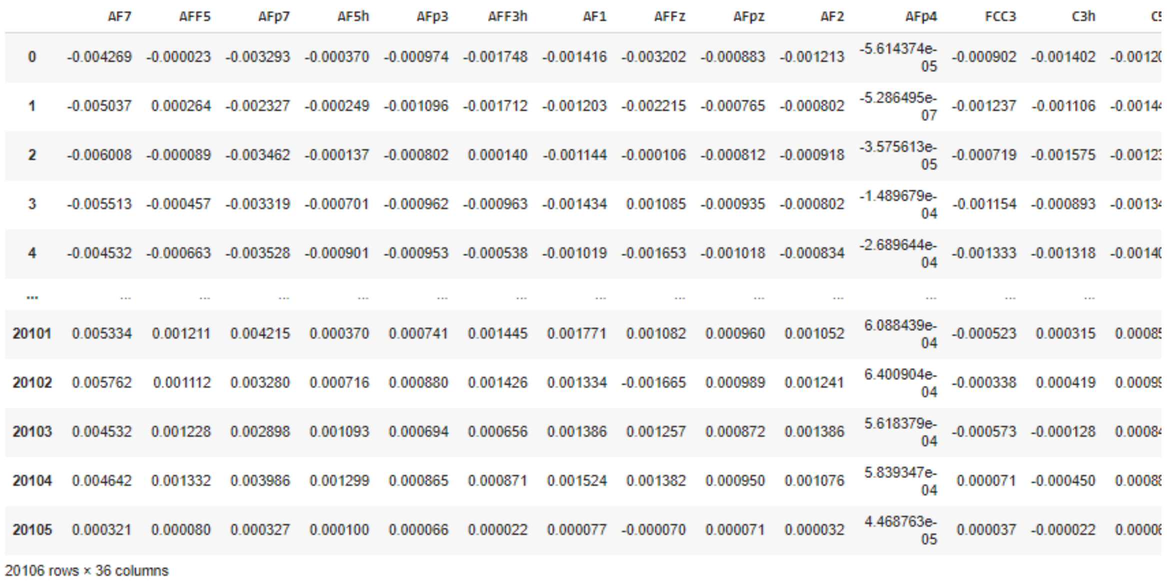
Machine learning algorithms, artificial neural network techniques, correlation graphs and data visualisation techniques are widely used in characterization of signals. According to the correlation graphs and visualisation of our data further steps were able to be taken such as machine learning algorithms and artificial neural network techniques.

According to IGI Global, pervasive healthcare is "the provision to health services based on the concepts of the pervasive computing paradigm with the purpose of the medical assistance anywhere and anytime", deploying wireless networks for this purpose (IGI-Global, n.a.). A wide application of this concept is the Body Area Networks (BANs), which are small, lightweight, and low-power wearables that constantly monitor physiological activities (Chen et al., 2010). However, since BANs store and transmit highly sensitive, person-specific information, they are vulnerable to targeted attacks (Karaoğlan Altop et al., 2017a). To establish the security and privacy of patients, person-specific cryptographic keys that govern the communication between wearables can be generated. The second part of this study aims to demonstrate the feasibility of using physiological signals for cryptographic key generation.

2. Signal Processing for Physiological Sensors, Part 1: fNIRS/EEG

2.1.Segmentation and Difference Data of WG and BL

One approach taken to process the signal was using segmentation for WG and BL. fNIRS continuous data was used for this purpose which is a 2D matrix. Each column represents one channel's time series of fNIRS data (density of oxy-hemoglobin and deoxy-hemoglobin) which is given with mmol/L.



	AF7	AFF5	AFp7	AF5h	AFp3	AFF3h	AF1	AFFz	AFpz	AF2	AFp4	FCC3	C3h	C
0	-0.004269	-0.000023	-0.003293	-0.000370	-0.000974	-0.001748	-0.001416	-0.003202	-0.000883	-0.001213	-5.614374e-05	-0.000902	-0.001402	-0.0012
1	-0.005037	0.000264	-0.002327	-0.000249	-0.001096	-0.001712	-0.001203	-0.002215	-0.000765	-0.000802	-5.286495e-07	-0.001237	-0.001106	-0.0014
2	-0.006008	-0.000089	-0.003462	-0.000137	-0.000802	0.000140	-0.001144	-0.000106	-0.000812	-0.000918	-3.575613e-05	-0.000719	-0.001575	-0.0012
3	-0.005513	-0.000457	-0.003319	-0.000701	-0.000962	-0.000963	-0.001434	0.001085	-0.000935	-0.000802	-1.489679e-04	-0.001154	-0.000893	-0.0013
4	-0.004532	-0.000663	-0.003528	-0.000901	-0.000953	-0.000538	-0.001019	-0.001653	-0.001018	-0.000834	-2.689644e-04	-0.001333	-0.001318	-0.0014
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
20101	0.005334	0.001211	0.004215	0.000370	0.000741	0.001445	0.001771	0.001082	0.000960	0.001052	6.088439e-04	-0.000523	0.000315	0.0008
20102	0.005762	0.001112	0.003280	0.000716	0.000880	0.001426	0.001334	-0.001665	0.000989	0.001241	6.400904e-04	-0.000338	0.000419	0.0009
20103	0.004532	0.001228	0.002898	0.001093	0.000694	0.000656	0.001386	0.001257	0.000872	0.001386	5.618379e-04	-0.000573	-0.000128	0.0008
20104	0.004642	0.001332	0.003986	0.001299	0.000865	0.000871	0.001524	0.001382	0.000950	0.001076	5.839347e-04	0.000071	-0.000450	0.0008
20105	0.000321	0.000080	0.000327	0.000100	0.000066	0.000022	0.000077	-0.000070	0.000071	0.000032	4.468763e-05	0.000037	-0.000022	0.0000

20106 rows × 36 columns

Figure 1.1

For each participant, the fNIRS continuous data was separated into segments of BL and WG. It is done by matching the sequence of WG and BL of the session with the marker times given in the dataset. Marker times specify when WG and BL starts during the session. The WG and BL sequence was given with 0's and 1's. For instance, a session has a sequence of:

0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1 where each 1 represents WG trial whereas each 0 represents BL trial. In another file, the starting times of those markers ie. 0's and 1's were given. After cutting the signal with respect to the markers, 30 segments of WG and 30 segments of BL were obtained.

After that, WG and BL segments were concatenated within themselves which resulted in two time series of WG and BL. Those time series can be seen below. Note that each color represents a channel.

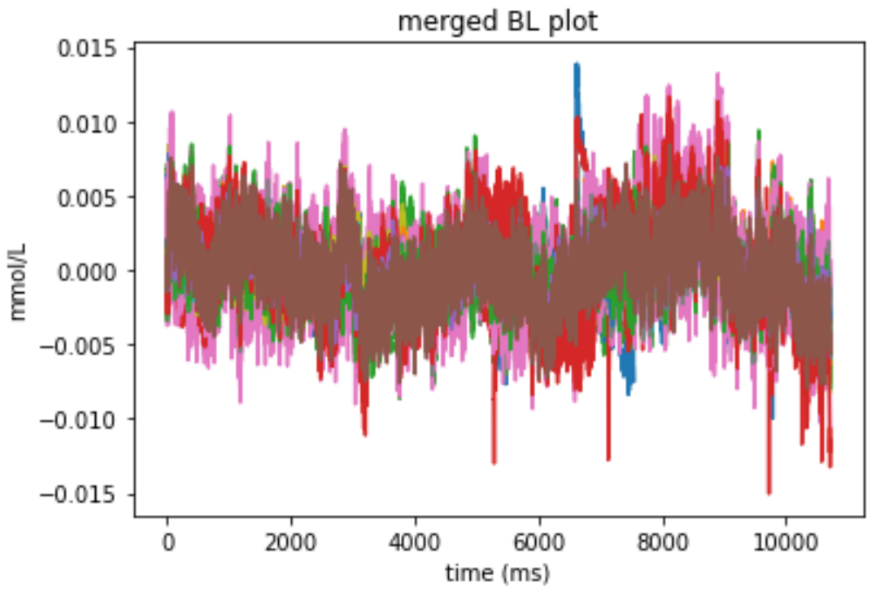


Figure 1.2

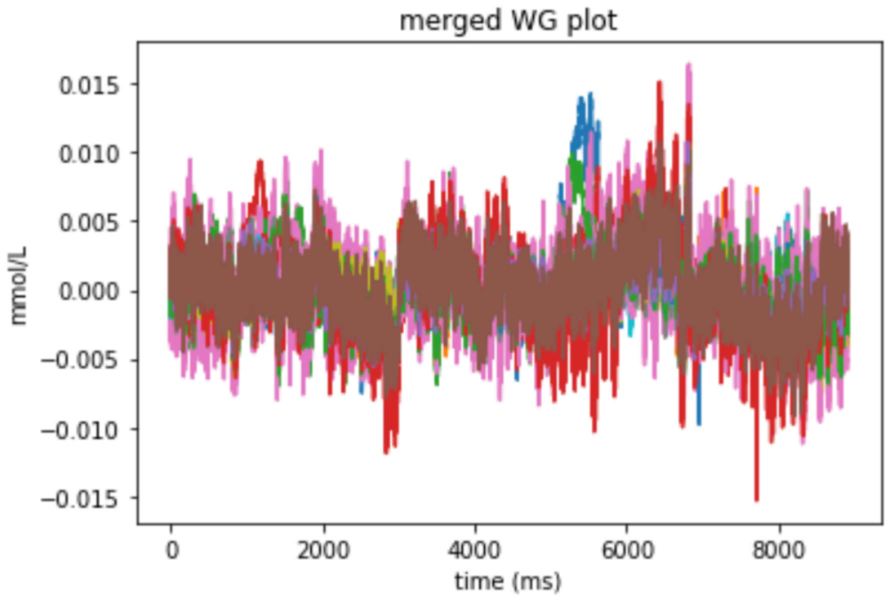


Figure 1.3

It can be seen that the plot gets more complex with different behaviors of different channels.. To make a better understanding of the concatenated time series, the histogram of them were also produced:

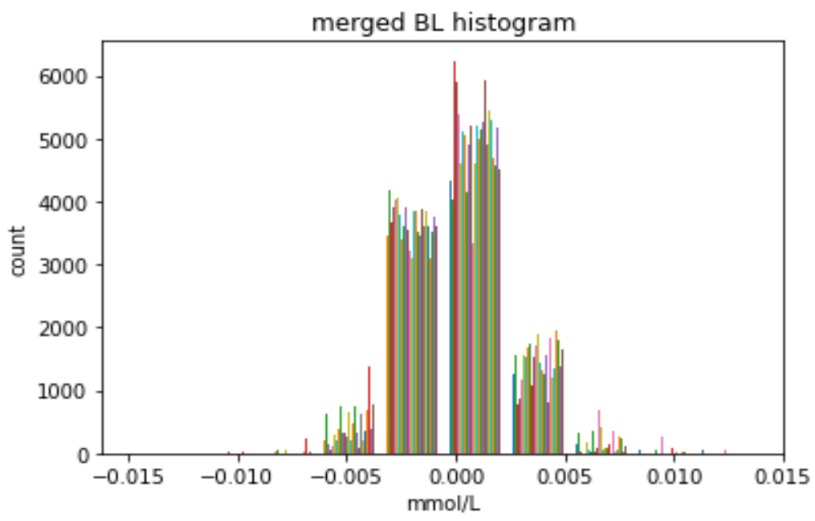


Figure 1.4

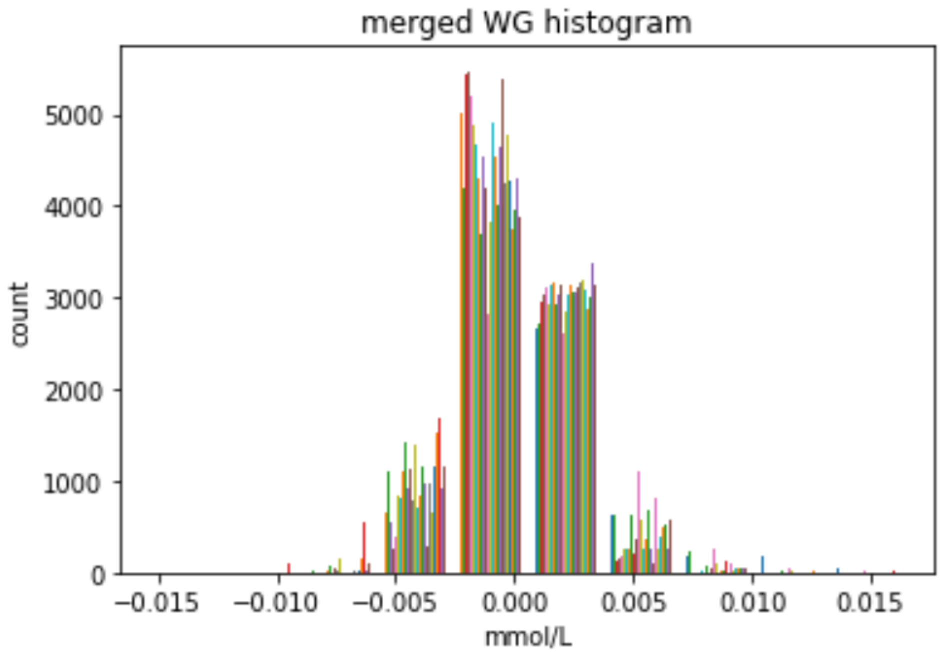


Figure 1.5

After that, for the purpose of understanding the difference between the WG and BL, the time series data difference was taken and visualized. In addition, the histogram of the difference data was also plotted.

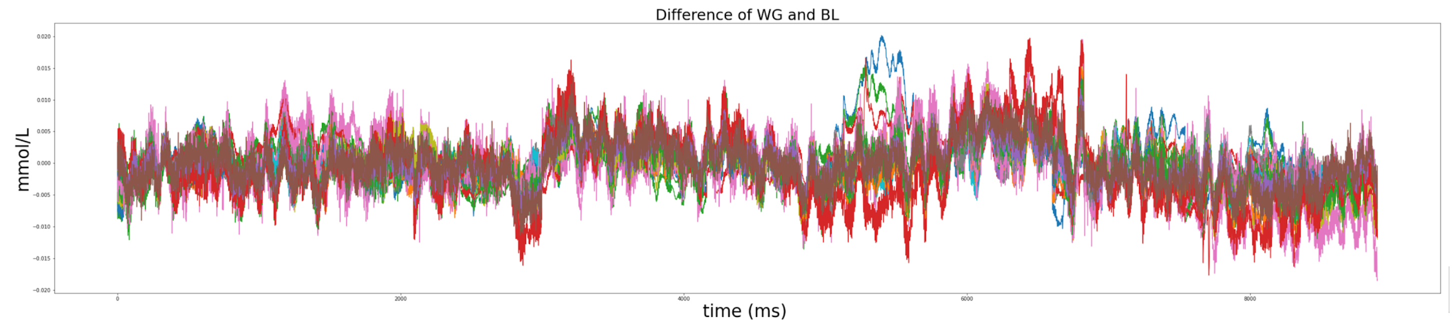


Figure 1.6

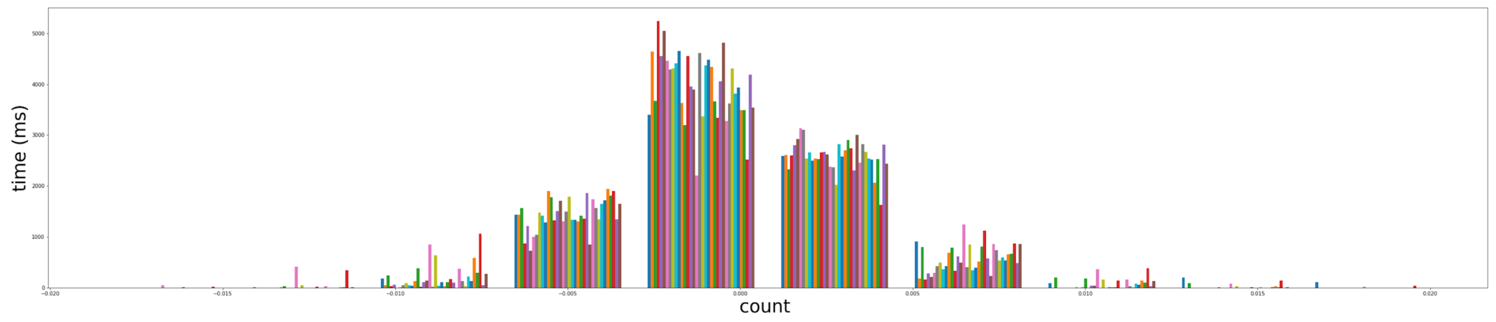


Figure 1.7

2.2.1 Correlations

Correlations between channels was found for both WG and BL tasks combined in order to observe whether the fNIRS channels were behaved similar to each other.36 fNIRS channels was taken into consideration in total and the figures shown below are scatter plots of different channels with each other which have direct correlation.To be able to get further information on the correlation between different channels, correlation matrix table was implemented below for BL and WG tasks individually for each experiment.

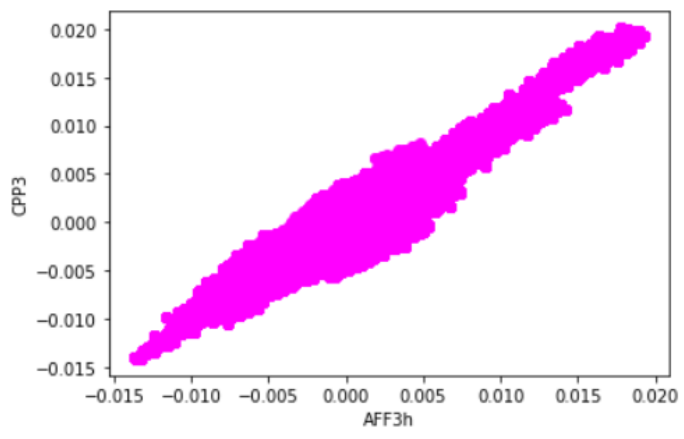


Figure 2.1

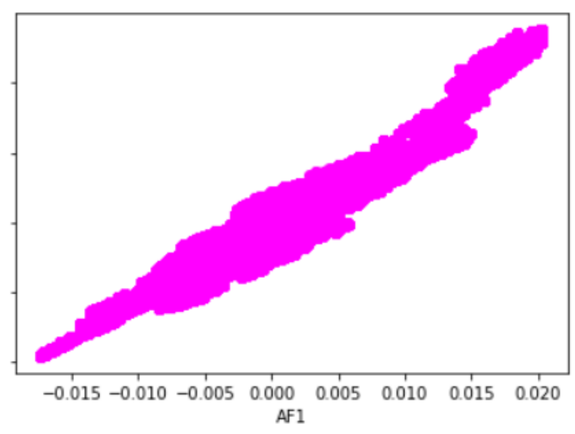
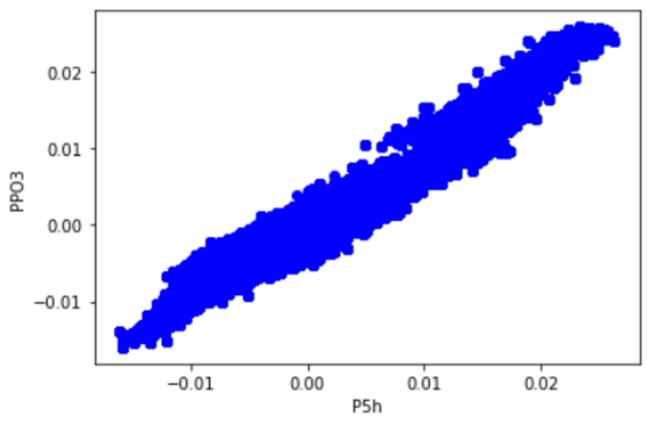


Figure 2.2



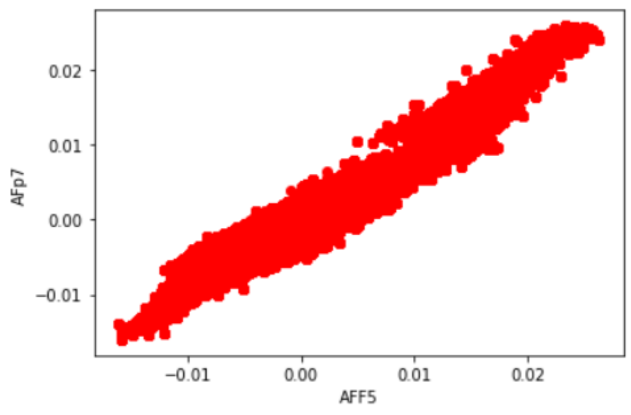
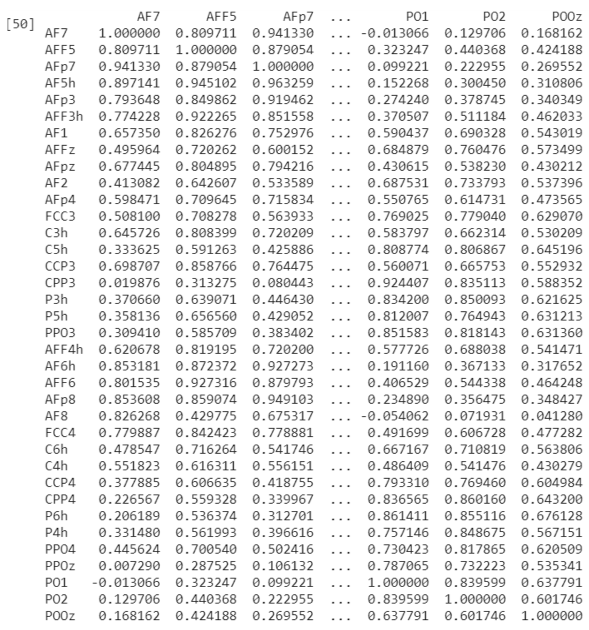


Figure 2.4

Figure 2.3

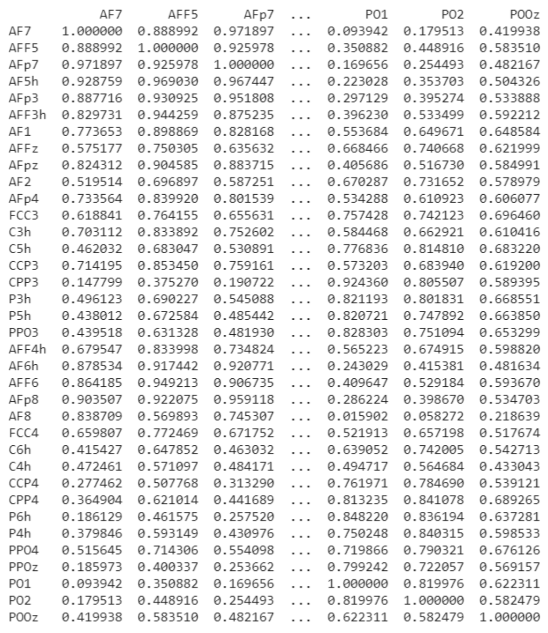
2.2.2 Correlation matrix

A correlation matrix is a table which shows the correlation coefficients of different variables with each other.Below the correlation coefficients were calculated for 36 fNIRS channels.After separating the data into WG and BL respectively the correlation matrix for both WG and BL periods was found for each participant.The correlation matrix below in figure 2.7 is the correlation matrix of the first thousand rows of the BL period of the third experiment.Since it would be harder to calculate the correlation of all the data, a portion of it was chosen to calculate them.The same correlation matrix was implemented for the WG period seen in figure 2.8 to be able to observe whether the rest and task period are similar to each other.



```
[50]         AF7      AFF5      AFp7  ...       PO1       PO2      POOz
AF7     1.000000  0.809711  0.941330  ... -0.013066  0.129706  0.168162
AFF5    0.809711  1.000000  0.879054  ...  0.323247  0.440368  0.424188
AFp7    0.941330  0.879054  1.000000  ...  0.099221  0.222955  0.269552
AF5h    0.897141  0.945102  0.963259  ...  0.152268  0.300450  0.310806
AFp3    0.793648  0.849862  0.919462  ...  0.274240  0.378745  0.340349
AFF3h   0.774228  0.922265  0.851558  ...  0.370507  0.511184  0.462033
AF1     0.657350  0.826276  0.752976  ...  0.590437  0.690328  0.543019
AFFz    0.495964  0.720262  0.600152  ...  0.684879  0.760476  0.573499
AFpz    0.677445  0.804895  0.794216  ...  0.430615  0.538230  0.430212
AF2     0.413082  0.642607  0.533589  ...  0.687531  0.733793  0.537396
AFp4    0.598471  0.709645  0.715834  ...  0.550765  0.614731  0.473565
FCC3    0.508100  0.708278  0.563933  ...  0.769025  0.779040  0.629070
C3h     0.645726  0.808399  0.720209  ...  0.583797  0.662314  0.530209
C5h     0.333625  0.591263  0.425886  ...  0.808774  0.806867  0.645196
CCP3    0.698707  0.858766  0.764475  ...  0.560071  0.665753  0.552932
CPP3    0.019876  0.313275  0.080443  ...  0.924407  0.835113  0.588352
P3h     0.370660  0.639071  0.446430  ...  0.834200  0.850093  0.621625
P5h     0.358136  0.656560  0.429052  ...  0.812007  0.764943  0.631213
PPO3    0.309410  0.585709  0.383402  ...  0.851583  0.818143  0.631360
AFF4h   0.620678  0.819195  0.720200  ...  0.577726  0.688038  0.541471
AF6h    0.853181  0.872372  0.927273  ...  0.191160  0.367133  0.317652
AFF6    0.801535  0.927316  0.879793  ...  0.406529  0.544338  0.464248
AFp8    0.853608  0.859074  0.949103  ...  0.234890  0.356475  0.348427
AF8     0.826268  0.429775  0.675317  ... -0.054062  0.071931  0.041280
FCC4    0.779887  0.842423  0.778881  ...  0.491699  0.606728  0.477282
C6h     0.478547  0.716264  0.541746  ...  0.667167  0.710819  0.563806
C4h     0.551823  0.616311  0.556151  ...  0.486409  0.541476  0.430279
CCP4    0.377885  0.606635  0.418755  ...  0.793310  0.769460  0.604984
CPP4    0.226567  0.559328  0.339967  ...  0.836565  0.860160  0.643200
P6h     0.206189  0.536374  0.312701  ...  0.861411  0.855116  0.676128
P4h     0.331480  0.561993  0.396616  ...  0.757146  0.848675  0.567151
PPO4    0.445624  0.700540  0.502416  ...  0.730423  0.817865  0.620509
PPOz    0.007290  0.287525  0.106132  ...  0.787065  0.732223  0.535341
PO1    -0.013066  0.323247  0.099221  ...  1.000000  0.839599  0.637791
PO2     0.129706  0.440368  0.222955  ...  0.839599  1.000000  0.601746
POOz    0.168162  0.424188  0.269552  ...  0.637791  0.601746  1.000000
```

Figure 2.5



```
             AF7      AFF5      AFp7  ...       PO1       PO2      POOz
AF7     1.000000  0.888992  0.971897  ...  0.093942  0.179513  0.419938
AFF5    0.888992  1.000000  0.925978  ...  0.350882  0.448916  0.583510
AFp7    0.971897  0.925978  1.000000  ...  0.169656  0.254493  0.482167
AF5h    0.928759  0.969030  0.967447  ...  0.223028  0.353703  0.504326
AFp3    0.887716  0.930925  0.951808  ...  0.297129  0.395274  0.533888
AFF3h   0.829731  0.944259  0.875235  ...  0.396230  0.533499  0.592212
AF1     0.773653  0.898869  0.828168  ...  0.553684  0.649671  0.648584
AFFz    0.575177  0.750305  0.635632  ...  0.668466  0.740668  0.621999
AFpz    0.824312  0.904585  0.883715  ...  0.405686  0.516730  0.584991
AF2     0.519514  0.696897  0.587251  ...  0.670287  0.731652  0.578979
AFp4    0.733564  0.839920  0.801539  ...  0.534288  0.610923  0.606077
FCC3    0.618841  0.764155  0.655631  ...  0.757428  0.742123  0.696460
C3h     0.703112  0.833892  0.752602  ...  0.584468  0.662921  0.610416
C5h     0.462032  0.683047  0.530891  ...  0.776836  0.814810  0.683220
CCP3    0.714195  0.853450  0.759161  ...  0.573203  0.683940  0.619200
CPP3    0.147799  0.375270  0.190722  ...  0.924360  0.805507  0.589395
P3h     0.496123  0.690227  0.545088  ...  0.821193  0.801831  0.668551
P5h     0.438012  0.672584  0.485442  ...  0.820721  0.747892  0.663850
PPO3    0.439518  0.631328  0.481930  ...  0.828303  0.751094  0.653299
AFF4h   0.679547  0.833998  0.734824  ...  0.565223  0.674915  0.598820
AF6h    0.878534  0.917442  0.920771  ...  0.243029  0.415381  0.481634
AFF6    0.864185  0.949213  0.906735  ...  0.409647  0.529184  0.593670
AFp8    0.903507  0.922075  0.959118  ...  0.286224  0.398670  0.534703
AF8     0.838709  0.569893  0.745307  ...  0.015902  0.058272  0.218639
FCC4    0.659807  0.772469  0.671752  ...  0.521913  0.657198  0.517674
C6h     0.415427  0.647852  0.463032  ...  0.639052  0.742005  0.542713
C4h     0.472461  0.571097  0.484171  ...  0.494717  0.564684  0.433043
CCP4    0.277462  0.507768  0.313290  ...  0.761971  0.784690  0.539121
CPP4    0.364904  0.621014  0.441689  ...  0.813235  0.841078  0.689265
P6h     0.186129  0.461575  0.257520  ...  0.848220  0.836194  0.637281
P4h     0.379846  0.593149  0.430976  ...  0.750248  0.840315  0.598533
PPO4    0.515645  0.714306  0.554098  ...  0.719866  0.790321  0.676126
PPOz    0.185973  0.400337  0.253662  ...  0.799242  0.722057  0.569157
PO1     0.093942  0.350882  0.169656  ...  1.000000  0.819976  0.622311
PO2     0.179513  0.448916  0.254493  ...  0.819976  1.000000  0.582479
POOz    0.419938  0.583510  0.482167  ...  0.622311  0.582479  1.000000
```

Figure 2.6

2.2.3 Autocorrelation

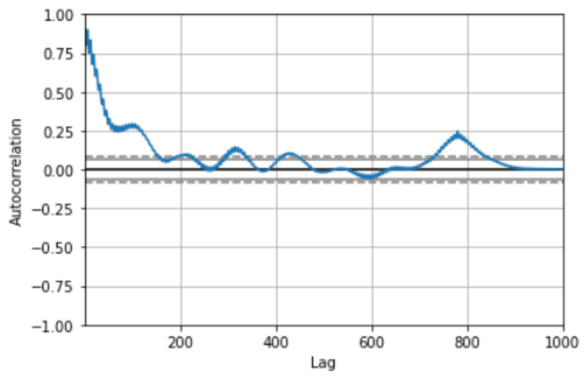
The signals in different fNIRS channels were able to be compared within themselves using a simple yet an effective method called autocorrelation.Autocorrelation is a method in which the correlation between signals are found with time lags between them.As we wanted to further analyse the mmol/L within the time series per channel, autocorrelation function was implemented for each channel and for WG(task) and BL(rest) periods separately.

The autocorrelation can be usually calculated with the following formula where h is the lag:

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-|h|} (x_{t+h} - \bar{x})(x_t - \bar{x})$$

Tugay, K., Koray TugayKoray Tugay 25311 gold badge22 silver badges88 bronze badges, & Kevin LiKevin Li 89844 silver badges1111 bronze badges. (1967, August 01). What does (pandas) autocorrelation graph show? Retrieved December 24, 2020, from https://stats.stackexchange.com/questions/357300/what-does-pandas-autocorrelation-graph-show

Figure 2.7 shows the autocorrelation of all channels for task period (WG) for experiment 3. The figure shows us that correlation between different time series vary a lot since it becomes negative in short periods and again positive in short periods. Figure 2.8 is the autocorrelation for the rest period(BL) for experiment 3. Again, the correlation within the channel goes positive and negative in short time periods.



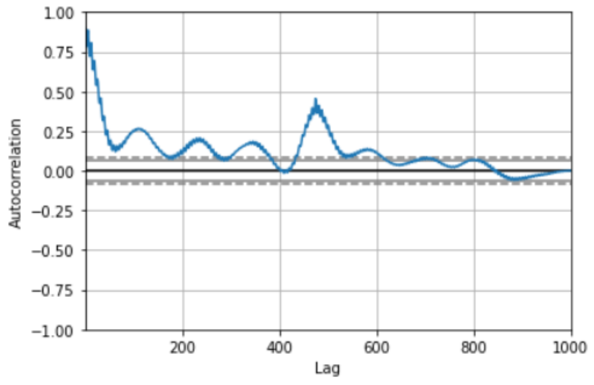
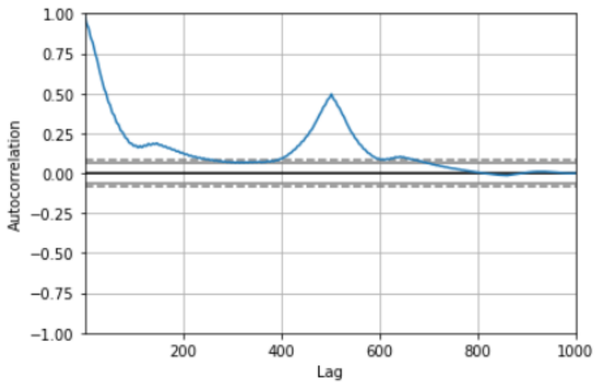


Figure 2.7

Figure 2.8



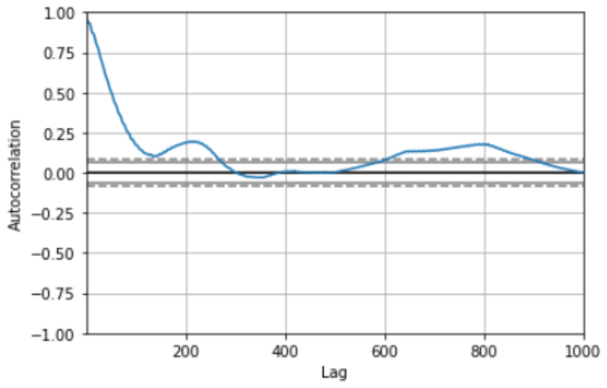


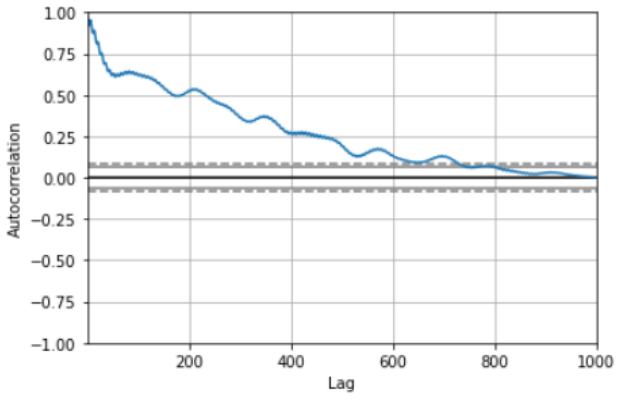
Figure 2.9

Figure 2.10

Figure 2.9 is the depiction of autocorrelation of the BL(rest) period for experiment 5 and the figure 2.10 is the depiction of autocorrelation of the WG(task) period for experiment 5 again.The correlation seen in 2.9 between the lag 0-200 can be seen as negative from the figure.The time period between lag 400-600 shows us that the correlation is positive for the first half of this period and goes negative for the second half.

In Figure 2.10, the lag between 0-200 is predominantly negative but the positiveness and negativeness varies within the time series a lot.

The figures shown below (2.11 and 2.12) have the same characteristics for the WG and BL periods as the figure 2.11 is the autocorrelation of the rest period(BL) of experiment 12 and the figure 2.12 is the autocorrelation of the task period (WG).



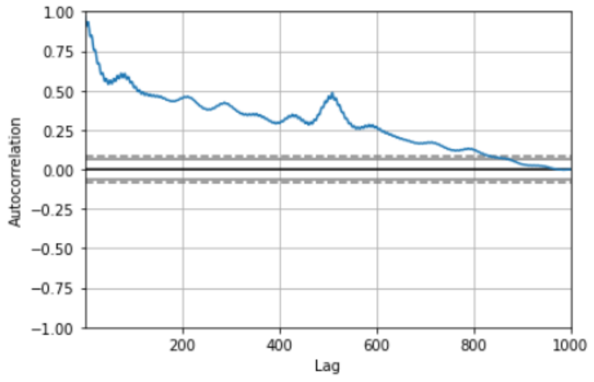
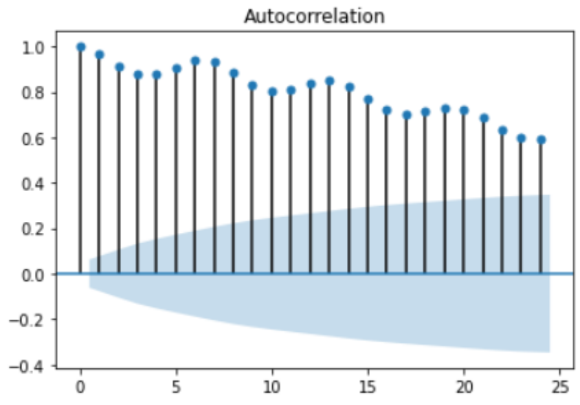


Figure 2.11

Figure 2.12

Another autocorrelation depiction within the time series is shown below in a different format. In this autocorrelation graph the autocorrelation is calculated for each channel separately for WG and BL tasks separately. Figure 2.13 is the depiction of autocorrelation of the channel 'AFF5' for the BL period for experiment 3. Figure 2.14 is the plot of the autocorrelation of the channel 'AF7' for the BL period again for experiment 3.



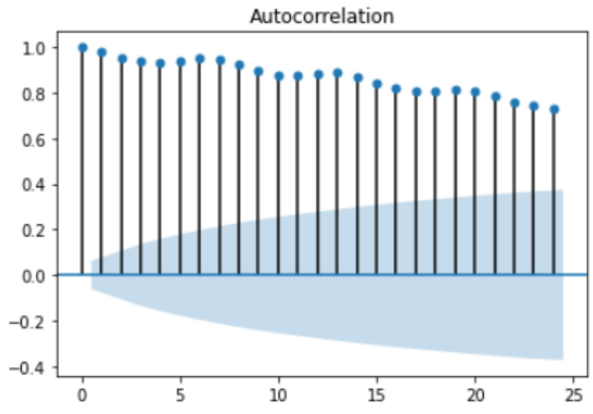


Figure 2.13

Figure 2.14

2.2.4 Histogram for fNIRS channels

For the data overall,histograms can be shown to be able to see the frequency distribution difference between different fNIRS channels. Figure 2.15 shows the difference of frequencies of data between channels ‘AF7’ and ‘AFF5’. Figure 2.16 shows the difference of frequencies of data between channels ‘AFp7’ and ‘AF5h’.

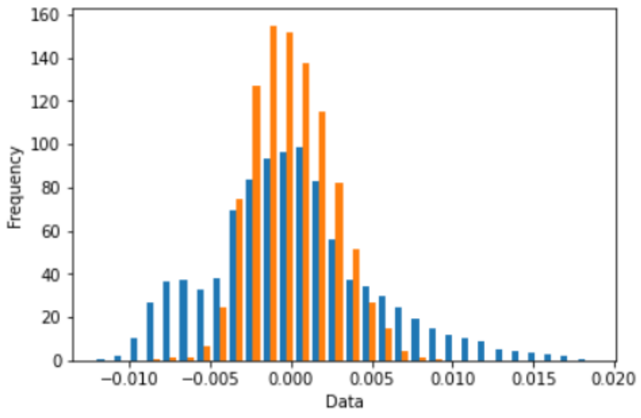


Figure 2.15

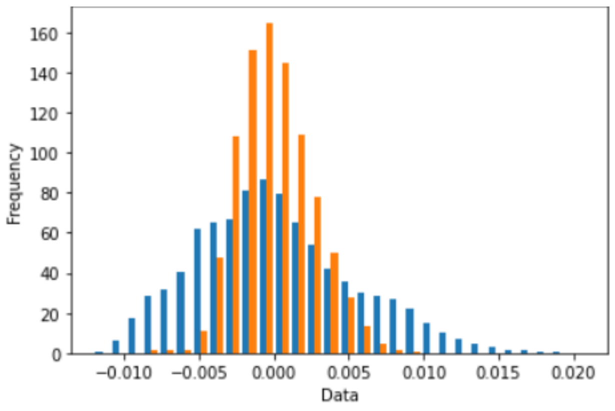


Figure 2.16

2.3 Classification of Signals With ANN

In this section of the project, we have tried to classify our fNIRS signals with artificial neural networks (ANN). For the sake of simplicity, we have used feed forward neural networks. To obtain better results we have applied the following process.

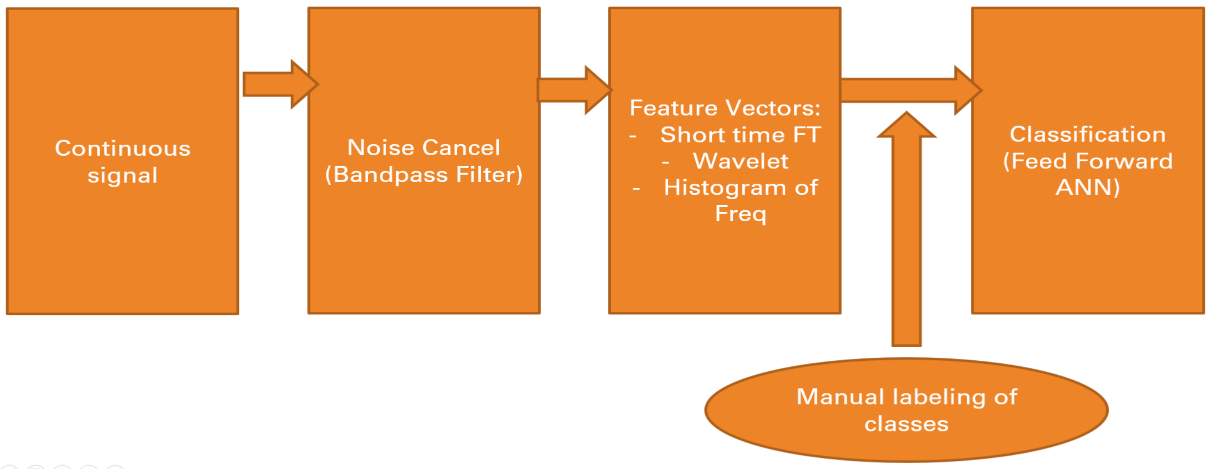

Continuous signal
Noise Cancel (Bandpass Filter)
Feature Vectors:
- Short time FT
- Wavelet
- Histogram of Freq
Classification (Feed Forward ANN)
Manual labeling of classes

Figure 2.17

2.3.1 Preprocessing of the signal

For getting better results, firstly we preprocessed signals. The preprocessing section consists of four main parts: segmentation of signal, noise cancellation, feature extraction and manual labeling of features (classes).

2.3.1.1 Segmentation of signal

As we have stated before, the dataset contains a continuous signal from the beginning to the end. For making trial based analysis, we have splitted the continuous signal from the points where marker data stated. At the blow, you can see the plot of the begging of each trial.

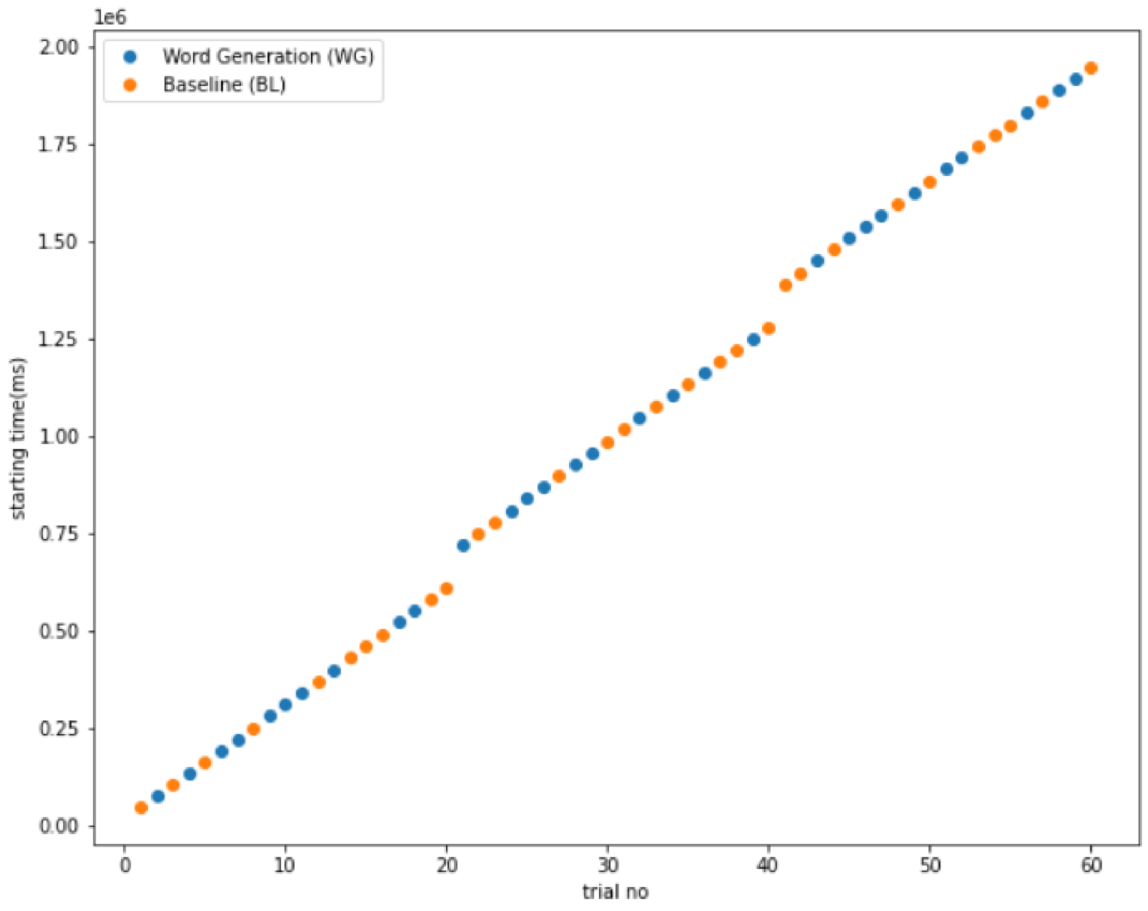


Figure 2.18

At the manual labeling of classes section, this data will be used again.

2.3.1.2 Noise cancellation

The raw fNIRS signal contains some coefficients from unwanted frequencies. That is why we have band-pass filtered the signal between 0.001 Hz and 0.4 Hz. Below you can see some filtered signal output and original signal of the channel **AF2** for the participant 1 .

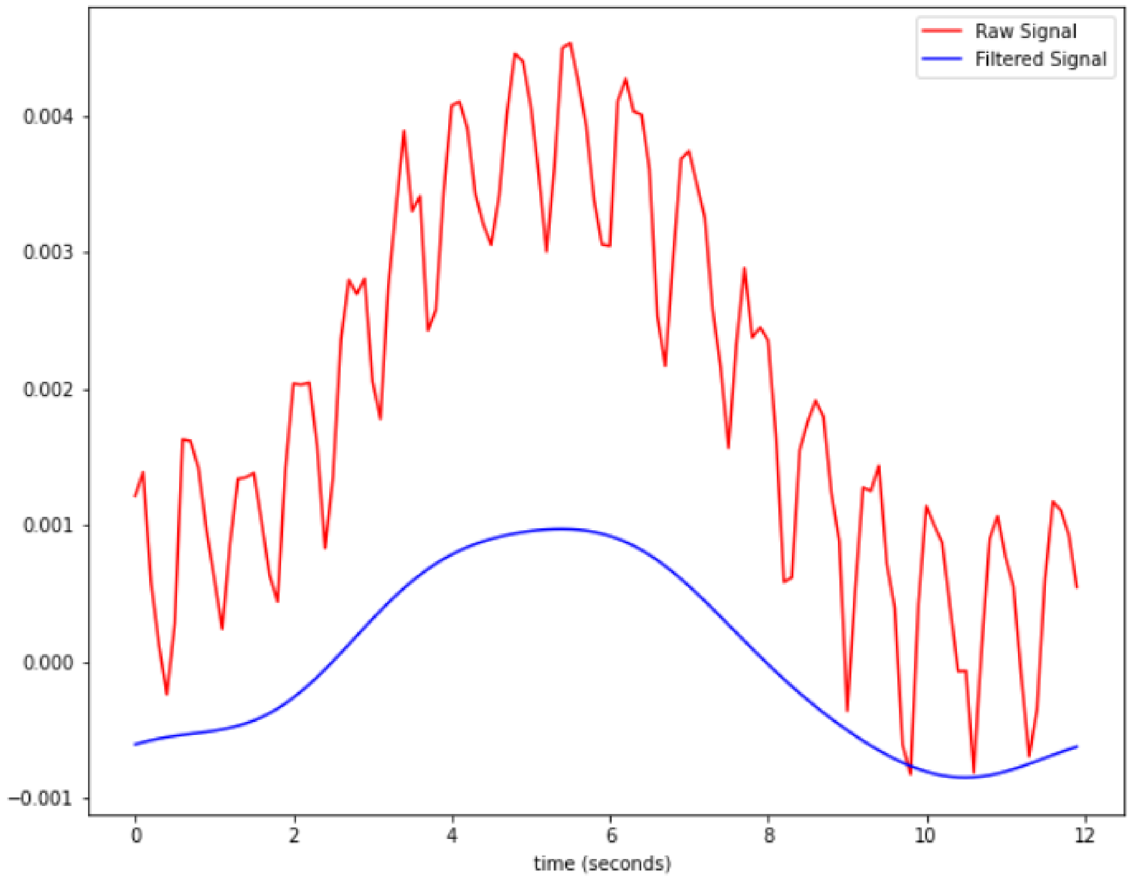


Figure 2.19

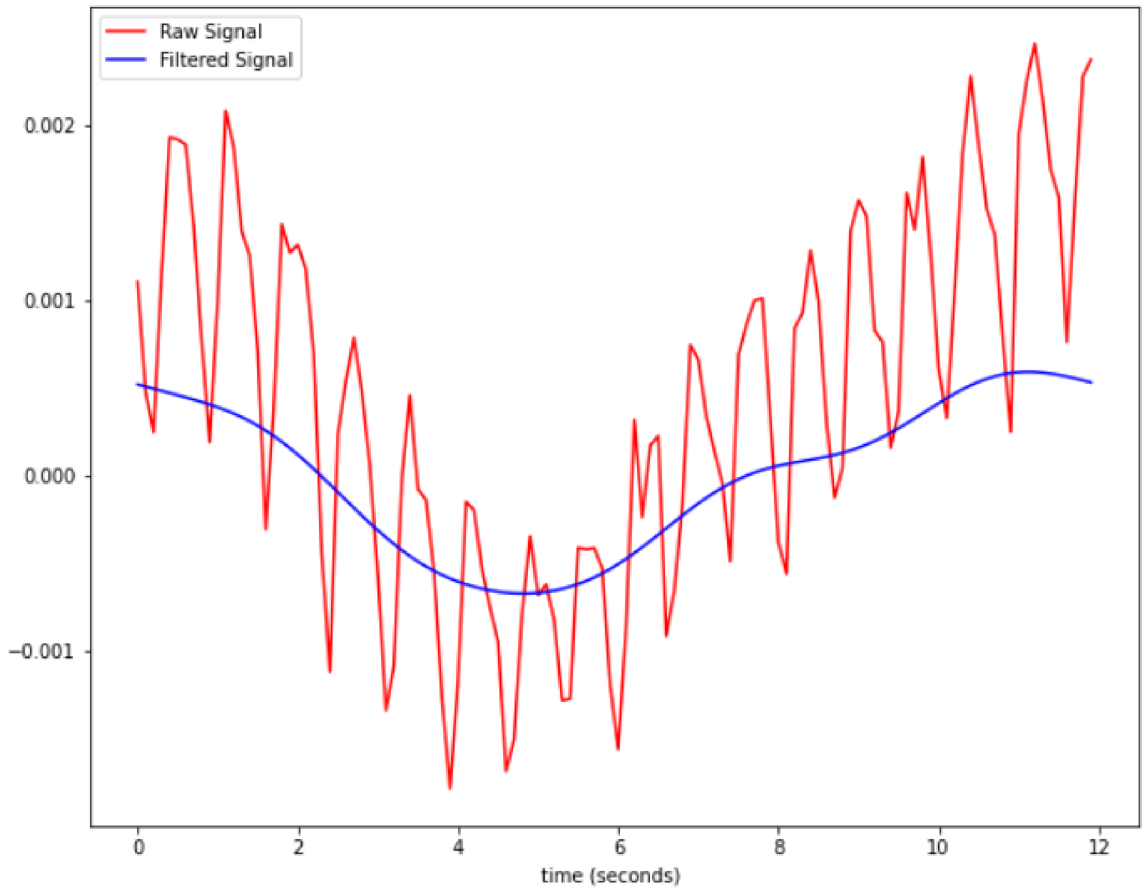


Figure 2.20

2.3.1.3 Feature Extraction

If we feed the neural network with the signals that are shown above we get 66% accuracy. However this accuracy is not as good as we wanted. So we have extracted some features in the frequency domain which are short time Fourier transform(STFT) coefficients, wavelet responses and frequency domain histograms. We have extracted the features in the frequency domain because we may have some synchronization problems in the time domain. We did not train the model in the frequency domain yet. However our aim is getting more than 80% consistent accuracy.

2.3.2 Training the artificial neural network

We have labeled our data according to the marker data as word generation task or baseline and trained our ANN with this labeled data. Since we gave our filtered and labeled data directly, our ANN had 120 input dimensions (1 for each sampling in the first 12 seconds of the task period). This input layer uses relu as an activation function. The model has one hidden layer with dimension 60 and with the activation function relu. Finally we have one output layer with the activation function sigmoid. Because we are making binary classification. Since word generation tasks labeled as 0 and baseline tasks labeled as 1, if the output is bigger than 0.5 it is predicted as baseline. Else it is predicted as word generation.The optimizer is determined as adam. This model gave 66% accuracy. It is planned to do frequency domain analysis after feature extraction.

3) Signal Processing for Physiological Sensors, Part 2: ECG and PPG

The features to be used for cryptographic key generation are interpeak intervals (IPIs) that are extracted from ECG and PPG signals. The practice of using physiological signals for cryptographic key generation has been studied in the past (Karaoğlan Altop et al, 2015; Karaoğlan Altop et al, 2017a; Karaoğlan Altop et al, 2017b). These studies have used different types of physiological signals such as ECG, PPG and ABP as a whole to generate physiological parameters, whereas in this study, we will compare the performances of physiological parameters generated by IPI extraction for ECG and PPG separately and investigate whether if they can be used interchangeably for BAN security.

3.1) Dataset and Pre-Processing

The database for the PhysioNet Computing in Cardiology Challenge (CinC) 2015, which aimed to detect false arrhythmia alarms was used in this study (Clifford et al., 2015) (Goldberger et al., 2000). The data was collected from patients in ICUs and included at least one electrocardiogram (ECG) lead and two of photoplethysmogram (PPG), arterial blood pressure (ABP) and respiration recordings. Each signal has a duration of 5 or 5.5 minutes. At least one ECG lead is present in every patient file, with the data collected from the II lead configuration being the most common one. As for the other signals, up to two out of the three were included for each patient. For the scope of this research, only the ECG and PPG signals were used.

As the dataset has many errors due to movement artifacts, sensor disconnections and other sources of noise such as pacemakers (PhysioNet, 2015), a certain degree of pre-processing was required in order to eliminate useless signals and clean up the remaining ones. Out of 750 entries, 102 ‘useful’ signals were manually selected for use. The selected signals were filtered with band-pass FIR filters with Kaiser windows and passband 0.75-25 Hz. This removed the baseline wander that especially affected the ECG recordings (Kher, 2019).

3.2) Heartbeat/IPI Detection

In order to find IPIs, the peaks of the physiological signals need to be detected beforehand. We used a MATLAB implementation of the Pan-Tompkins R-peak detection algorithm for detecting peaks in the ECG (Pan, J. & Tompkins, WJ., 1985; Seghamidz, 2020). An exemplary result is shown in Figure 3.1. For peak detection in PPG signals, a peak detection algorithm that detected the systole peaks (Figure 3.2) was written from scratch, which made use of the findPeaks method in MATLAB and then utilized thresholding in order to distinguish the systole peaks from diastole peaks and other artifacts/anomalies. The peaks of the PPG signal taken from the same patient in Figure aa is shown in Figure 3.3.

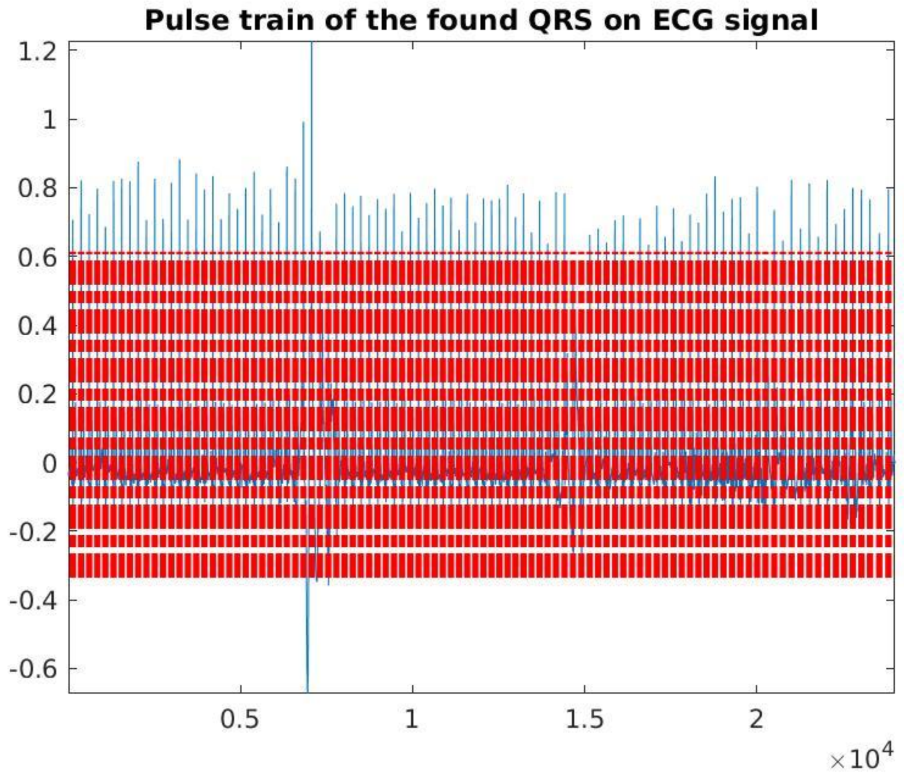


Figure 3.1: Pulse train detected on an ECG signal by the MATLAB implementation of the Pan-Tompkins R-peak detection algorithm.

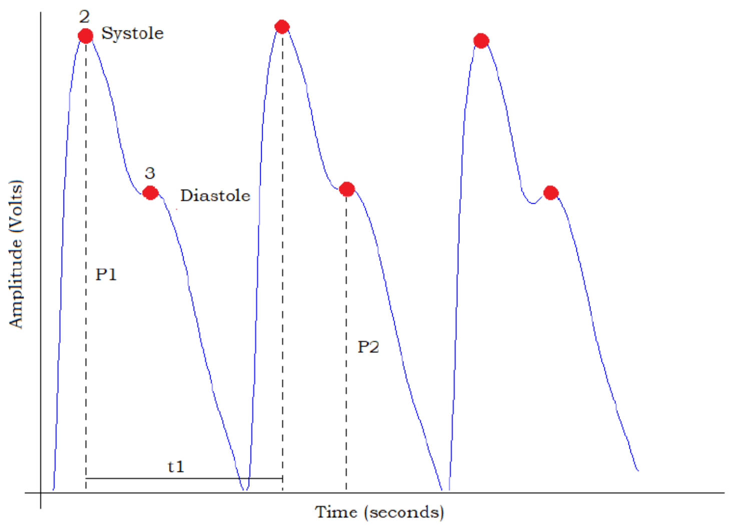


Figure 3.2: Graph of a PPG signal taken from a healthy person (Moraes et al., 2018).

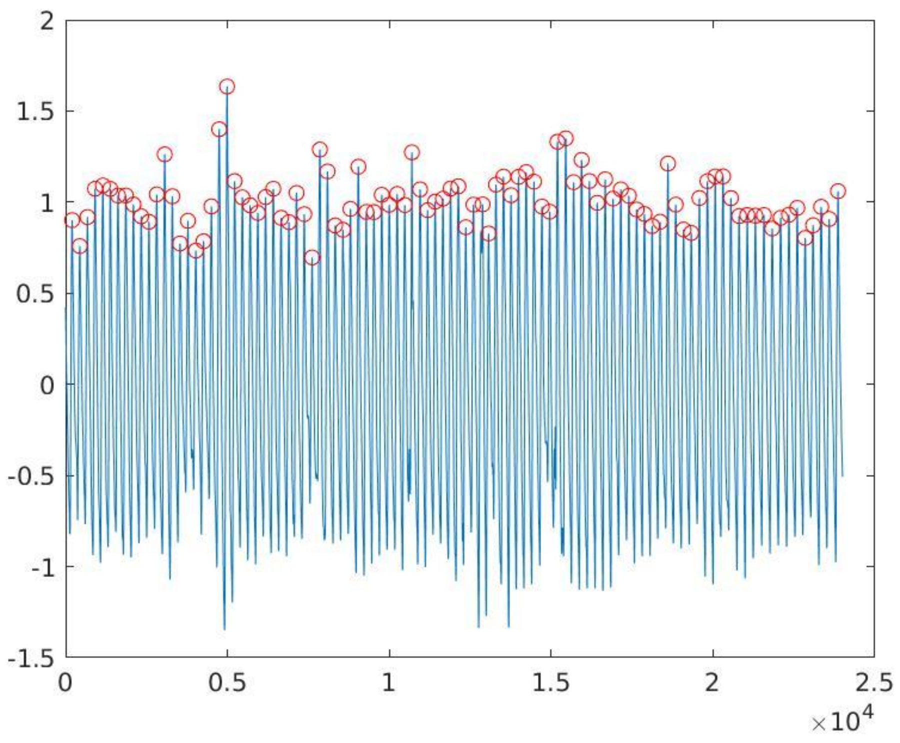


Figure 3.3: Systole peaks detected by the peak detection algorithm on the PPG signal that belongs to the patient whose ECG signal is shown in Figure 3.2.1.

3.3) Bit String Generation

The methodology used by Karaoğlan Altop et al (2015) was utilized for converting IPI values into bit strings. Our adaptation of this methodology used IPI sequences of different lengths $l \in \{32, 64\}$ taken from different starting points $p \in \{0.8, 32, 64, 96\}$ (in terms of seconds) for each part (explained in 2..4). These IPI sequences were put through circular uniform quantization, which divides the IPI range (using the minimum and maximum values found during the IPI generation process) into partitions with respect to step size $s \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and assigns a value to each partition from $\{0, 1, \ldots, 2^{n-1}\}$, where n is the number of bits that the value of each partition will be mapped to, chosen as n = 4 for l = 32, n = 2 for l = 64. The ultimate goal is to generate a 128-bit binary string, hence the aforementioned bit choices. An example is as follows: if l = 32, min = 450, max = 839, min $\leq f_i \leq$ max, where $1 \leq i \leq$ noPatients, s = 6, partitions are $\{450 - 455, 456 - 461, \ldots., 834 - 839\}$. Since there are $2^4 = 16$ values possible, the 1st, 17th... partitions are mapped to 0, the 2nd, 18th... partitions are mapped to 1, ..., the 16th, 32nd... partitions are mapped to 15. The result of the mapping is a codebook that is used in the quantization process (Karaoğlan Altop et al, 2015). The final step is binarizing all quantized values into the specified number of bits using Gray encoding (Gray, 1953).

3.4) Performance Analysis and Comparison

The research problem was a synthesis of last summer's PURE research (Açıkgöz et al, 2020) and the previous studies, which were done by two of our instructors in the summer project, that investigated the feasibility of using physiological signals in cryptographic key generation (Karaoğlan Altop et al, 2015; Karaoğlan Altop et al, 2017a; Karaoğlan Altop et al, 2017b). Instead of fitting probability distribution functions to physiological signals, the previously investigated IPI method is used and the biometric authentication model that was developed initially in the summer is used. Each signal is divided into three parts and the bit strings are generated as explained in Section 2..3. The key for the first part would be the template/registration key, kept in a (hypothetical) server/cloud for authentication purposes. The bit strings of other parts were compared with the bit string of the first part, as if the user was trying to authenticate themselves to the system. Two evaluation metrics were used for performance analysis; distinctiveness and Equal Error Rate (EER).

3.4.a) Distinctiveness

Hamming distance is used to evaluate the distinctiveness of the keys. A matrix of Hamming distances is created (and visualized as gray scale in the end). The diagonals, where the row and column indices are equal, the distinctiveness for the signals taken from the same user is evaluated as follows: the Hamming distances between part 1-part 2 and part1-part3 are calculated and averaged, which makes the final result. We compared part 1 with part 2 and part 3 since the research problem investigates if the key from a person (part 2 and part 3) is similar enough with the registration key (part 1) to be authenticated. For the rest, the distinctiveness for the signals taken from the different users is evaluated as follows: for the entry in M[a][b], the Hamming distances between (user a, part 1)-(user b, part 2), (user a, part1)-(user b, part3), (user b, part 1)-(user a, part 2), (user b, part 1)-(user a, part 3) are calculated and averaged, which makes the final result, investigating if the key from a person is different enough with the registration key of a different person. The resulting grayscale plots are given below in Figures 3.4-3.5. The bit strings with |IPI| = 64 appear to perform very poorly as the plots are of mostly similar shades of gray (especially for lower step sizes), indicating that the distinctiveness between the diagonal (same users) and the remaining is not different. On the other hand, the diagonal is only barely visible for |IPI| = 32. For ECG, the diagonal is most visible for |IPI| = 32, step sizes starting from 4 whereas for PPG, it is most visible for |IPI| = 32, step sizes starting from 6. For both, the diagonal seems to be less visible for step size = 10. It is observed that ECG performs slightly better than PPG in terms of distinctiveness, although it should be noted that neither perform at the required level.

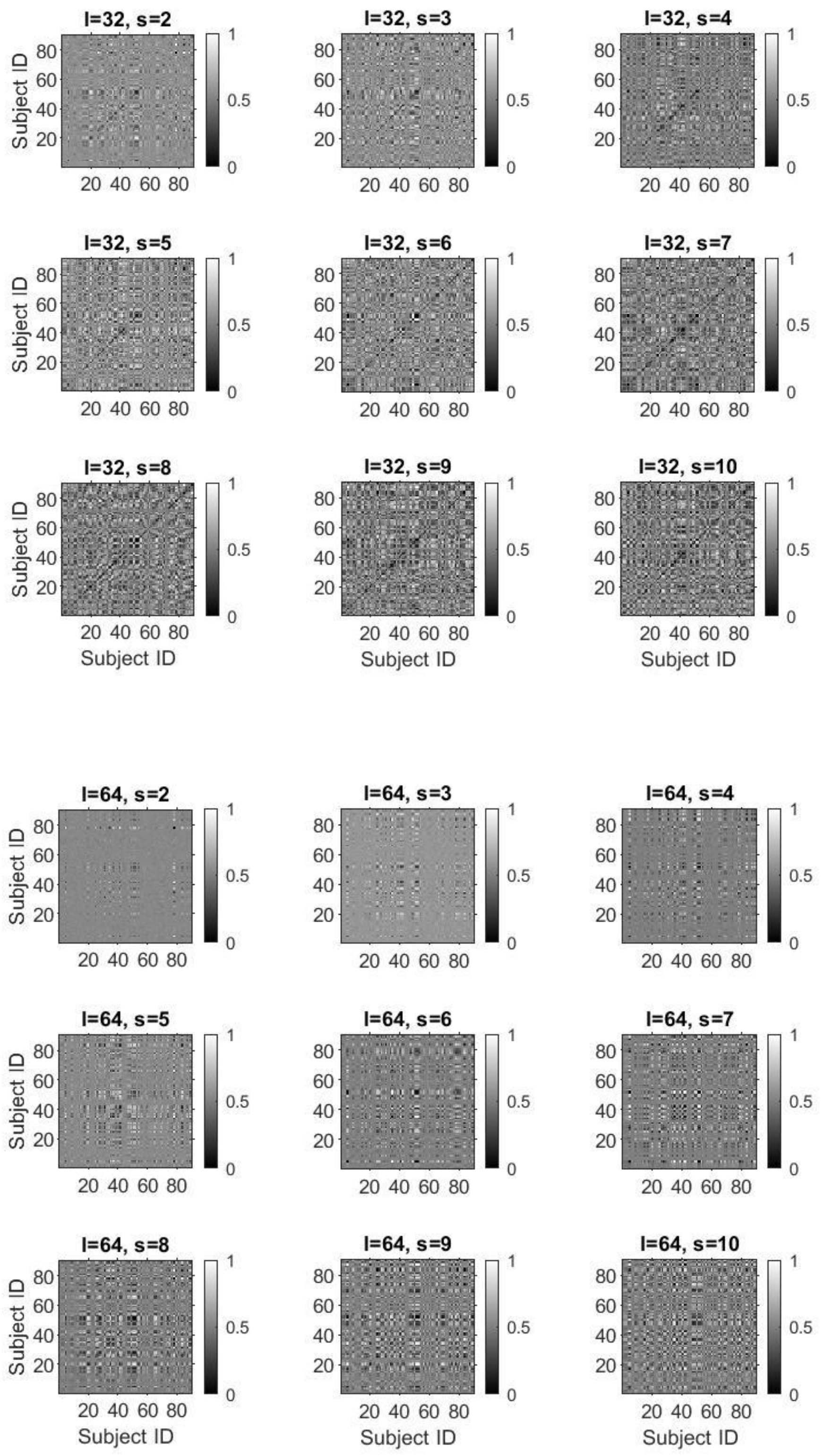


Figure 3.4: Distinctiveness matrices for ECG signals (l = no of IPIs, s = step size).

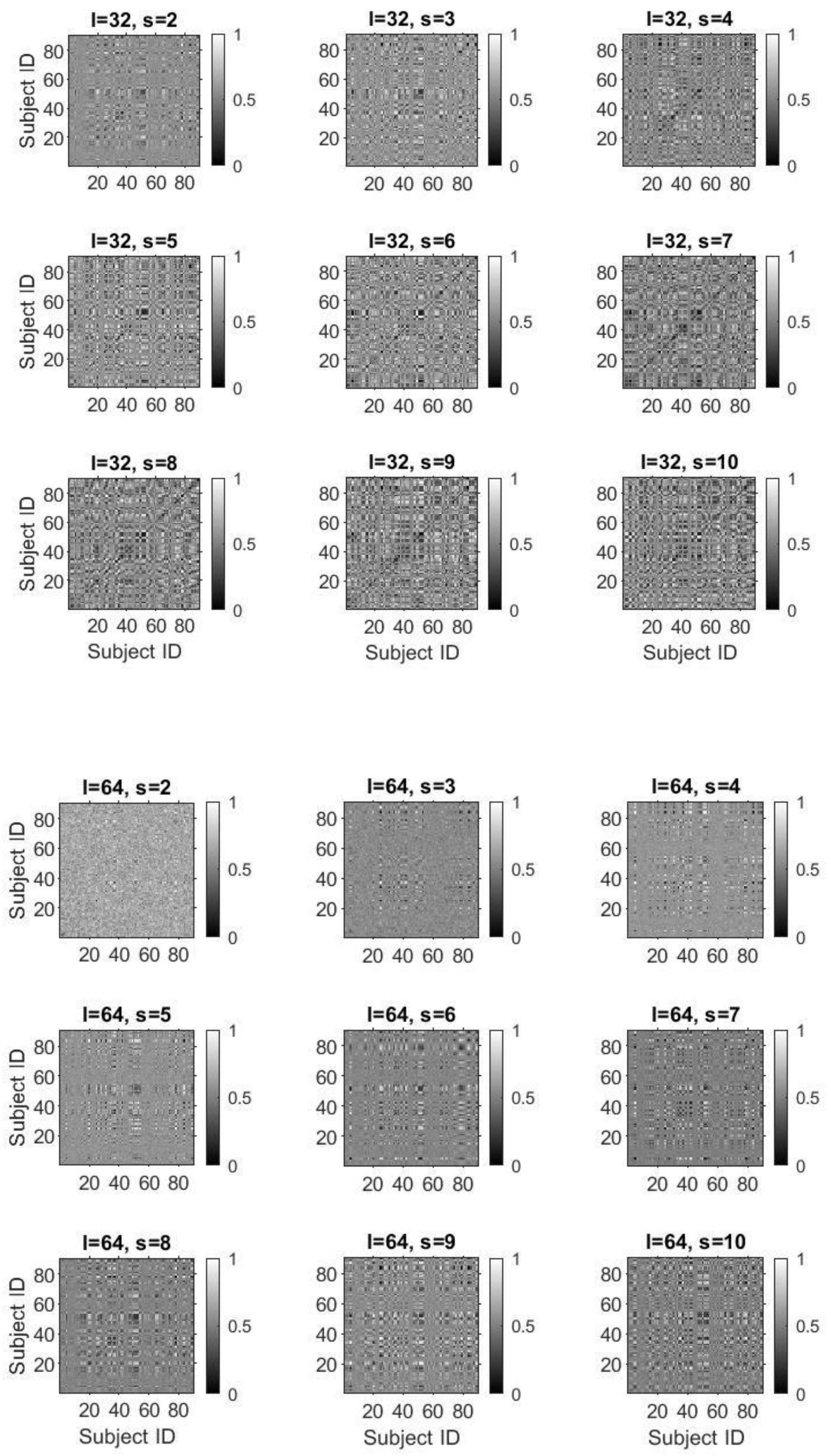


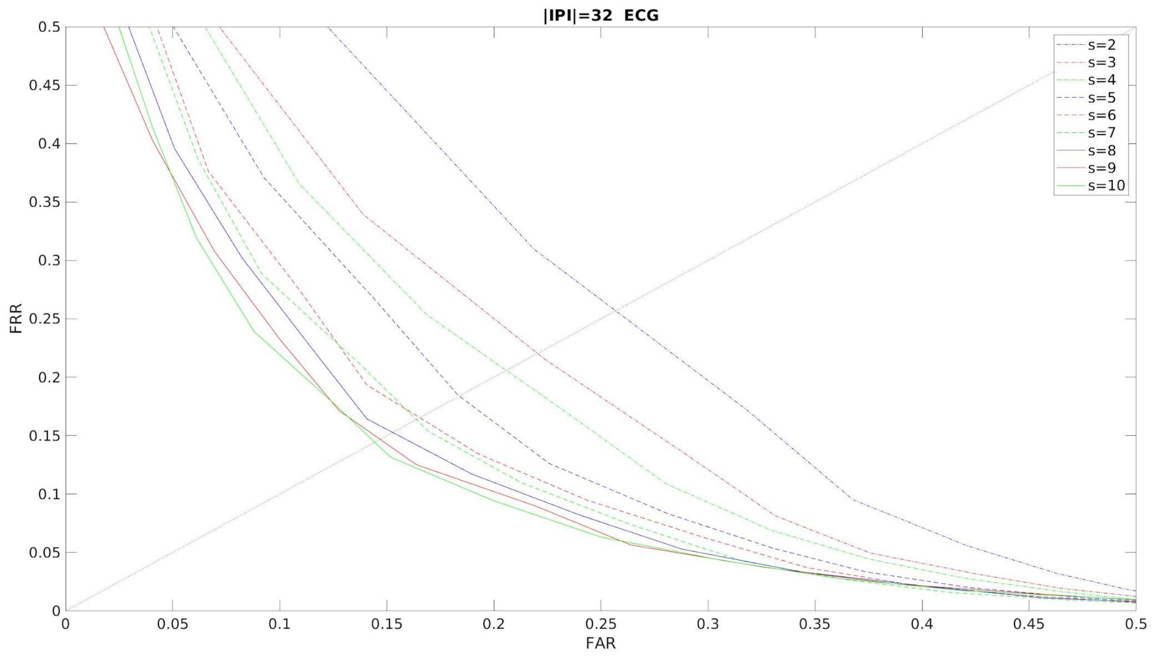
Figure 3.5: Distinctiveness matrices for PPG signals (l = no of IPIs, s = step size).

3.4.b) Error Rates

Two types of error rates are evaluated. The first one, False Accept Rate (FAR) is the rate of bit strings that belong to different signals, falsely classified as similar. This is implemented by calculating the Hamming distance between the part 1 of one user with the part 2 and 3 of the other person, and vice versa, making a total of 4 distances. Each resulting Hamming distance is compared to a threshold value t ($0 \leq t \leq 1$, $t = 5n/128$, $n \in \mathbb{N}$, t is divided by 128 since there are 128 bits in each string, therefore the maximum number of different bits between two strings can be 128, for which the Hamming distance is 1) and the bit strings are considered similar if the distance is below that threshold, hence, a false acceptance. The average of the false acceptances for different values of t is taken for each user pair. The average of those is the overall FAR.

The second type of error, False Rejection Rate (FRR), is the rate of bit strings that belong to the same signal, falsely classified as different. This is implemented by calculating the Hamming distance between (part1, part2) and (part1, part3) pairs where all parts belong to the same signal, making a total of 2 distances. Each resulting Hamming distance is compared to a threshold value t (same as in FAR) and the bit strings are considered different if the distance is above that threshold, hence, a false rejection. The average of the false rejections for different values of t is taken for each user. The average of those is the overall FFR.

As stated in (Karaoğlan Altop et al., 2015), “the Equal Error Rate (EER), is the rate where the FAR and FRR are equal to each other”, which being low indicates more accuracy. The Receiver Operating Curve (ROC) for different IPI lengths and signals is shown in Figure 3.6-3.7, displaying the tradeoff between FAR and FRR.



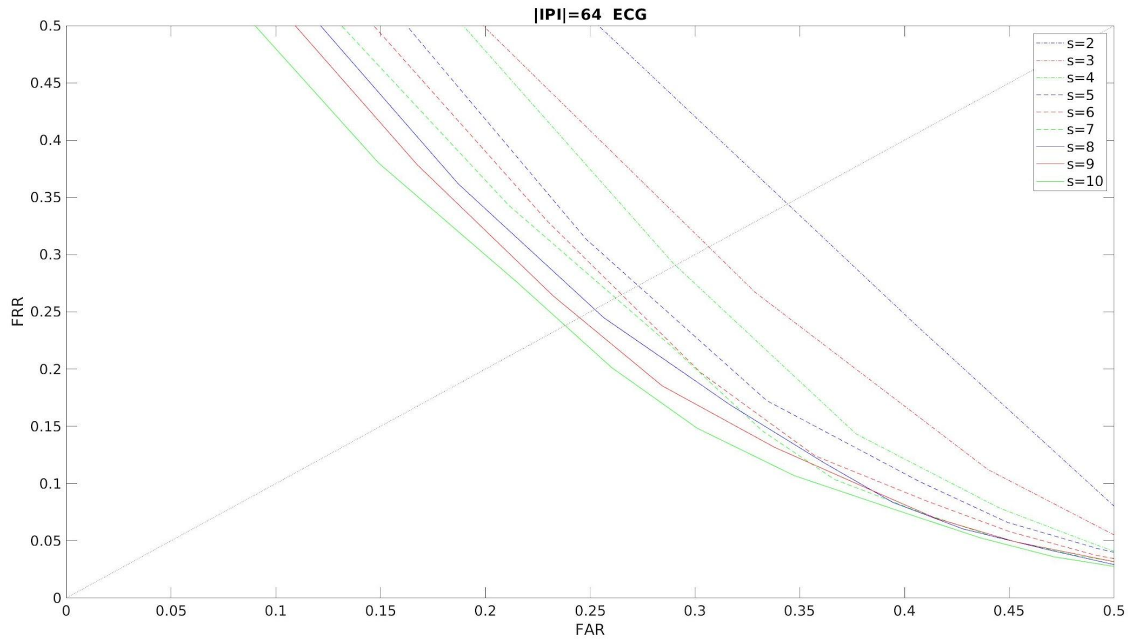
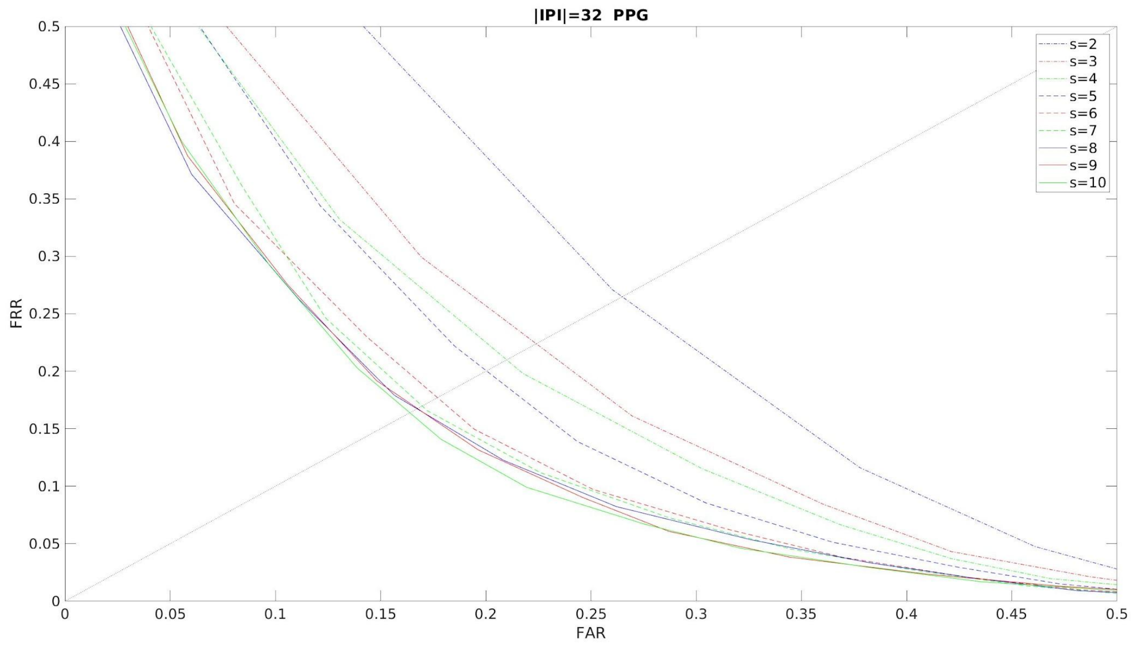


Figure 3.6: ROCs for ECG.



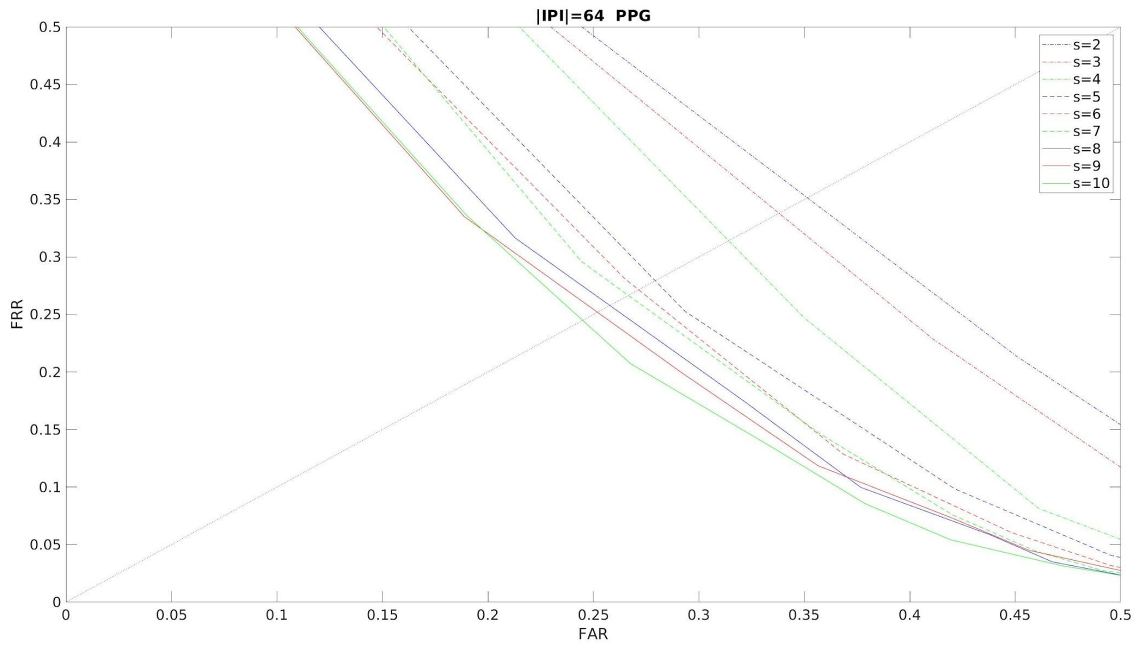


Figure 3.7: ROCs for PPG.

Table 3.1: EER for all IPI lengths and signal types.

	|IPI| = 32, ECG	|IPI| = 64, ECG	|IPI| = 32, PPG	|IPI| = 64, PPG
step size = 2	0.256889	0.344227	0.264829	0.351481
step size = 3	0.220399	0.306647	0.223965	0.338035
step size = 4	0.205853	0.290654	0.209935	0.314169
step size = 5	0.183853	0.272880	0.200331	0.279356
step size = 6	0.165402	0.265060	0.177054	0.271496
step size = 7	0.163806	0.262143	0.169764	0.266623
step size = 8	0.152754	0.252171	0.167340	0.257848
step size = 9	0.146887	0.244930	0.167560	0.251977
step size = 10	0.144392	0.238024	0.163905	0.244947

For both signals, EER is lower for |IPI| = 32 and decreases as step size increases. ECG is observed to perform better than PPG for every IPI length and step size. The best performance for both is at |IPI| = 32 and step size = 10, with 0.144 for ECG and 0.164 for PCG. Both are slightly larger than the EER values reported at (Karaoğlan Altop et al, 2015), which is a good starting point for the continuation of this research.

4. Discussion and Conclusion

4.1 EEG/fNIRS

It can be seen that in the results obtained by the segmentation and concatenation of WG and BL trials, the characteristics of channels used in the trials have different behaviors. Some oscillate more whereas some don't have strong fluctuations. Therefore, in order to do further feature extraction and key generation in the future, channels have to be chosen. Meaning, channels which reflect the fNIRS data better should be taken into account and other channels should be eliminated. Choosing channels would give more reliable results and decrease the complexity of the computations since there will be less channel data to process. In order to choose the channels, correlation plots between different channels have been found. The correlation matrix has been produced to see the correlations between channels. In addition, histograms for channels have been produced. For further investigation and elimination of the channels, various techniques will be used such as t-test. Also to be able to further see if the channels have behaved differently between WG and BL periods, random experiments were chosen out of all the experiments and autocorrelation between the word generation and baseline periods was plotted.Some experiments suggested that there was no visible difference in autocorrelation plots between the WG and BL periods whereas some suggested the opposite. After determining the channel(s), it is planned to use the random forest and SVD to do the classification of WG and BL. ANN implementation was carried out in parallel to those. The time domain ANN classification was already done. It is planned to do frequency domain analysis and classification after extracting the feature vectors in frequency domain. After extracting the feature vectors per recording per subject, it is planned to do statistics calculations such as convergence verification. Then, to improve the algorithm further, the noise model development needs to be carried out. Afterwards, the added noise feature vectors will be formed to improve the ANN algorithm. In addition, it can be used to generate continuous fNIRS signals. Another approach to be taken is planned as, after constructing the feature vectors per recording per subject, the sample covariances will be checked in order to truncate SVD to find the essential parameters of fNIRS signal. After finding the essential parameters, compression of fNIRS signals may be possible, or in other words, continuous fNIRS signals may be defined with some parameters.

4.2 ECG/PPG

It was observed that both signals did not perform well in terms of distinctiveness but they performed considerably well in terms of EER. For both metrics, ECG performed slightly better than PPG. If we were to choose the best method for generating bit strings for cryptographic key generation solely based on EER, |IPI| = 32 and step size = 10 would be good candidates for both ECG and PCG. However, as stated in 3.4.a, both signals appeared to perform worse for step size = 10, which makes it not an optimal candidate.

As EER decreases with increasing step size, it would be a logical next step to investigate EER for larger step sizes in search for an even smaller EER. The signals should also be investigated in terms of other metrics such as randomness and temporal variance (Karaoğlan Altop et al, 2015; Karaoğlan Altop et al, 2017a; Karaoğlan Altop et al, 2017b). If it is observed that PPG is not far from ECG in terms of reliability for cryptographic key generation, it can be used by itself in securing BANs as it is easier to measure and record PPG compared to ECG for the equipment that the average user can use in everyday life.

References

Açıkgöz, B., Abdulla, V., Yazıcı, Ö., Sayan, A., Biçen, O., Levi, A., Karaoğlan Altop, D. (2020). PURE Summer 2019-2020, Signal Processing for Physiological Signals.

Chen, M. et al (2010). Body Area Networks: A Survey. Mobile Networks and Applications, 16 (2), 171-193, doi: 10.1007/s11036-010-0260-8.

Clifford, GD., Silva, I., Moody, B., Li, Q., Kella, D., Shahin, A., Kooistra, T., Perry, D., Mark, R. (2015). The PhysioNet/Computing in Cardiology Challenge 2015: Reducing False Arrhythmia Alarms in the ICU.

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

Gray, F. (1953). Pulse Code Communication . New York, USA: Bell Telephone Laboratories, Incorporated. U.S. Patent 2,632,058. Serial No. 785697. Archived from the original on August 5, 2020. Retrieved from https://web.archive.org/web/20200805094312/https://patentimages.storage.googleapis.com/a3/d7/f2/0343f5f2c0cf50/US2632058.pdf on January 3, 2020.

Hu, X., Zhuang, C., Wang, F., Liu, Y., Im, C., & Zhang, D. (2019, March 21). FNIRS Evidence for Recognizably Different Positive Emotions. Retrieved January 01, 2021, from https://www.frontiersin.org/articles/10.3389/fnhum.2019.00120/full

IGI-Global (n.d.). What is pervasive healthcare. Retrieved from https://www.igi-global.com/dictionary/pervasive-multiplatform-health-care-support/33267 on January 3, 2021.

Karaoğlan Altop, D., Levi, A., Tuzcu, V. (2015). Towards Using Physiological Signals as Cryptographic Keys in Body Area Networks. Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. doi:10.4108/icst.pervasivehealth.2015.260074

Karaoğlan Altop, D., Levi, A., Tuzcu, V. (2017a). SU-PhysioDB: A Physiological Signals Database for Body Area Network Security. 2017 IEEE BlackSeaCom, Turkey.

Karaoğlan Altop, D., Levi, A., Tuzcu, V. (2017b). Deriving cryptographic keys from physiological signals. Pervasive and Mobile Computing, 39, 65-79. doi:10.1016/j.pmcj.2016.08.004

Kher, R. (2019). Signal Processing Techniques for Removing Noise from ECG Signals. Journal of Biomedical Engineering and Research, 3 (101), 1-9. doi:10.17303/jber.2019.3.101

Moraes, J., Rocha, M., Vasconcelos, G., Filho, J.V., Albuquerque, V. D., Alexandria, A. (2018). Advances in Photoplethysmography Signal Analysis for Biomedical Applications. Sensors, 18(6), 1894. doi:10.3390/s18061894

Motion Artifact Contaminated fNIRS and EEG Data. (2014, March 03). Retrieved January 01, 2021, from https://physionet.org/content/motion-artifact/1.0.0/

Pan, J., Tompkins, W. J. (1985). A Real-Time QRS Detection Algorithm. IEEE Transactions on Biomedical Engineering, BME-32(3), 230-236. doi:10.1109/tbme.1985.325532.

PhysioNet (2015). Reducing false arrhythmia alarms in the ICU - The PhysioNet Computing in Cardiology Challenge 2015. Retrieved from https://physionet.org/content/challenge-2015/1.0.0/ on January 3, 2021.

Sedghamiz, H. (2020). Complete Pan Tompkins Implementation ECG QRS detector, MATLAB Central File Exchange. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector on August 25, 2020.

Shin, J., Lühmann, A. V., Kim, D., Mehnert, J., Hwang, H., & Müller, K. (2018). Simultaneous acquisition of EEG and NIRS during cognitive tasks for an open access dataset. *Scientific Data, 5*(1). doi:10.1038/sdata.2018.3

Yücel, Meryem A., et al. “Functional Near Infrared Spectroscopy: Enabling Routine Functional Brain Imaging.” *Current Opinion in Biomedical Engineering*, Elsevier, 6 Oct. 2017, www.sciencedirect.com/science/article/abs/pii/S2468451117300697.