BOSTON COLLEGE

DEPARTMENT OF COMPUTER SCIENCE

HONORS THESIS

Exploratory Analysis and Predictive Modeling for Electrocardiogram (ECG) and Photoplethysmogram (PPG) Human Heart Activity Data

Written by:
Qiyuan Zhou

Advisor:
Prof. Sergio A. Alvarez

May 6th, 2023

ABSTRACT

Electrocardiography (ECG) and Photoplethysmography (PPG) are two widely used techniques for monitoring cardiovascular activity. ECG is a well-established method for detecting the electrical activity of the heart, while PPG utilizes optical technology to measure variations in blood volume in peripheral tissues. This thesis explores two applications of PPG and ECG signals, utilizing a PPG dataset with Human Activity Recognition labels and an ECG dataset labeled with various cardiac conditions. Preprocessing was carried out on the raw time-series data, through detrending, bandpass filtering, and outlier exclusion. Two reduced versions of the data were also considered, one using Heart Rate Variability (HRV) summary measures, and the other a spectral representation based on the Fast Fourier Transform (FFT). Exploratory Data Analysis and predictive data modeling using machine learning techniques were then performed on the preprocessed datasets. We comment on the predictive performance of the models, try to understand the results from a physiological perspective, and suggest possible directions for future work.

ACKNOWLEDGEMENT

This endeavor would not have been possible without Professor Alvarez's unwavering support and guidance, so huge thanks to Professor Alvarez for the past year and the classes I've taken with him. Additionally, I am appreciative of my parents for making this all possible and for their unrelenting support. Lastly, I would like to thank my friends for their emotional support and the happiness they brought.

Contents

1. Introduction

1.1. Motivation

In cardiology, Electrocardiography (ECG) and Photoplethysmography (PPG) are two of the most prevalent methods for analyzing and monitoring cardiovascular activity. ECG is a well-established method for detecting the electrical activity of the heart, whereas PPG uses light-based technology for measuring variations in blood volume in peripheral tissues, typically on fingertips and wristbands[1]. Both Electrocardiography (ECG) and Photoplethysmography (PPG) provide essential information about the functioning of the cardiovascular system and are gaining popularity in clinical research and practice.

To gain a greater understanding of these physiological signals, a simple PPG dataset with three labels (rest, squat, and step) was analyzed. After that, a second dataset containing more comprehensive ECG signals was utilized to classify numerous types of cardiac conditions.

In this study, our goal is to perform exploratory analysis on both datasets and to evaluate the capabilities of machine learning models in recognizing and differentiating between various human activities using PPG data and diagnosing various cardiac conditions using ECG data. The objectives include evaluating the efficacy of both types of signals and evaluating various machine learning models to identify the best performing algorithms for the provided data sets. In addition, the limitations of the datasets and models will be addressed.

1.2. Literature Review

Similar research has previously been conducted. Psathas et al.[2] examined a public PPG - DaLiA dataset containing fifteen individuals and nine activities. Twenty-four machine learning techniques were used in total. The greatest performance was obtained by the weighted k-Nearest Neighbors (k-NN), the Cubic Support Vector Machines (C-SVM), and the Bagged Trees (BGT), with respective results of 80%, 81.1%, and 92.8%. In Hnoohom et al.[3], a novel method, PPG-NeXt, for extracting relevant characteristics from the PPG signal using deep learning methods was used. The proposed model obtained a prediction F1-score of greater than 90% based on experimental results using only PPG data from the three benchmark datasets. In addition, the paper suggests that integrating PPG and acceleration signals can improve activity recognition. Rath et al[4] used two standard datasets consisting of ECG signals, MIT-BIH and PTB-ECG and applied deep learning models to detect heart diseases. The authors proposed an ensemble model using Long Short-Term Memory(LSTM) and Generative Adversarial Network(GAN) and achieved accuracy of 0.992 and area under curve(AUC) of 0.984. Zhang et al[5] proposed a 12 layer 1D CNN model to classify a single-lead

ECG signal into five distinct heart disease categories. The proposed model was tested on the MIT-BIH arrhythmia database and reached a positive predictive value of 0.977.

1.3. Background

1.3.1. PPG dataset description

The supplied data[6] was gathered by the electronics research team of the Department of Information Engineering at the Polytechnic University of Marche in Ancona, Italy. The dataset used in this study was collected from a convenience sample of 7 healthy participants (3 males and 4 females) with age between 20 and 52 years old. The data was recorded using a wrist-worn photoplethysmography (PPG) device that measures blood volume changes in the microvascular bed of tissue. Each participant was asked to complete a set of physical activities, including five series of ten squat exercises each, five series of ten stepper exercises each, and five series of resting for five minutes each. This dataset comprises 105 PPG signals (15 for each subject) along with the corresponding 105 tri-axial accelerometer signals, which were recorded at a sampling frequency of 400 Hz.

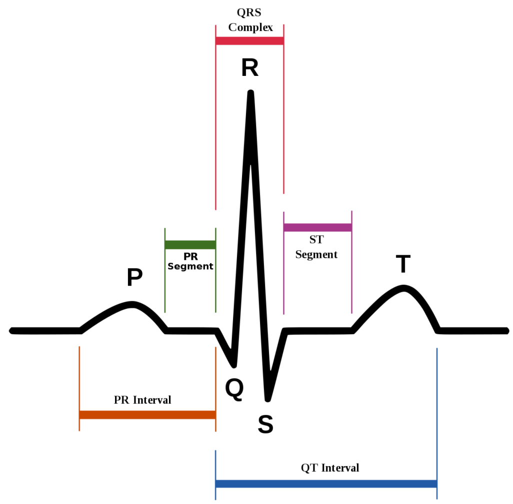
1.3.2. ECG dataset description

The PTB-XL (PhysioNet/Computing in Cardiology Challenge 2020) dataset is a large open-access electrocardiogram (ECG) dataset consisting of 21799 recordings from 18869 patients, 52% of whom are male and 48% of whom are female with ages range from 0 to 95. Each entry in the dataset is 10 seconds long.

The dataset includes ECG recordings from patients with various cardiac conditions as well as healthy individuals. The following describes the distribution of diagnoses: 9514 records have a normal ECG (NORM), 5469 records have Myocardial Infarction (MI), 5235 records have ST/T change (STTC), 4898 records have conduction disturbance (CD), and 2649 records have hypertrophy (HYP)[7]. Each of the cardiac conditions is explained below:

MI - Myocardial Infarction, also known as a heart attack, is mainly caused by coronary artery blockage. A prolonged lack of oxygen supply to the cardiac muscle can result in the death of cardiac muscle cells. Patients usually experience chest discomfort or discomfort in the neck, back, or arms[8].

STTC - ST/T Change, common in hypertensive adults, refers to the change in the ST segment. It describes the region between the conclusion of the QRS complex and the start of the T wave[9].

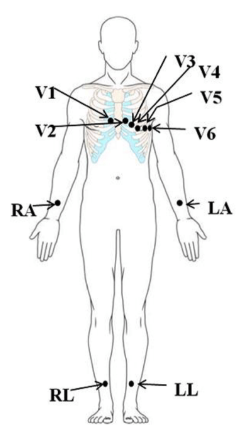


[10]

Figure 1. A normal waveform and some of its related points
Source: QRS Differentiation to Improve ECG Biometrics under Different Physical Scenarios Using Multilayer Perceptron

CD - Conduction Disturbance, also known as heart block, results from electrical signals not being produced effectively, not traveling through the heart as it should, or both[11].
HYP - Hypertrophy. Outflow obstruction due to asymmetric septal enlargement, resulting in sudden cardiac death.

The ECG signals were sampled at a rate of 500 Hz and are presented in the standard 12-lead format (I, II, III, aVL, aVR, aVF, V1–V6). Downsampled versions of the waveform data with a sampling frequency of 100 Hz are also available for the user's convenience, and they are the ones being used in this paper.



[12]

Figure 2. Graph showing the placement of electrodes that produce a 12-lead ECG
Source: Artificial intelligence methods for analysis of electrocardiogram signals for cardiac abnormalities: state-of-the-art and future challenges

2. Methods

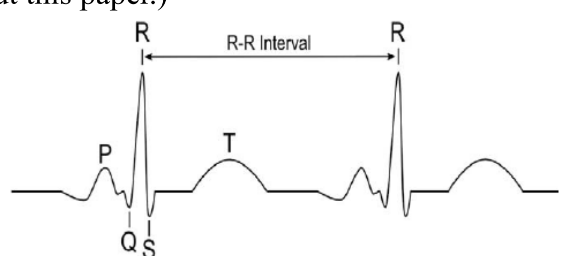
2.1. Datasets Preprocessing

2.1.1. Preprocessing for PPG dataset

Glob package was used to retrieve all of the PPG data files, and the heartpy package was used for filtering and extracting heart rate variability (HRV) variables. As a parameter for heartpy API functions, the specified sampling frequency of 400 Hz is used. After trial and error, I determined that it is difficult to preprocess all PPG data in order to keep it within a suitable range; hence, corrupted records were flagged and excluded from further analysis. To reduce the number of "corrupted" data records, I simply utilize the heartpy API function to set the cutoff threshold for the high pass Butterworth filter to 0.3 Hz and the cutoff level for the low pass Butterworth filter to 10 Hz. The frequency range of 0.5 Hz to 10 Hz is a commonly used bandpass, as cited in a number of other research literature[12–14]. After trial and error, it was determined that an order of 2 preserves the majority of samples while filtering out noise, where order is the order of an ordinary differential equation that can be used to generate the filter output using the original signal as the driving stimulus (input)[15]. Two data instances, S1/rest5 ppg and S2/squat3 ppg were not included for further analysis. Although the original dataset has 35 records for each of the categories rest, squat, and step, the following analysis is based on the uncorrupted data instances, which have 34 records for rest, 34 records for squat, and 35 records for step. Then, heartpy's process function is called, which generates ['bpm', 'ibi','sdnn','sdsd', 'rmssd', 'pnn20', 'pnn50', 'hr mad','sd1','sd2','s','sd1/sd2', 'breathingrate']. These HRV variables are explained below.

bpm: beats per minute.

ibi: inter-beat interval, also called the RR interval, refers to the variation in time between successive heartbeats. (Note that ECG and PPG signals typically use different terminology. In ECG signals, the RR interval is utilized, whereas in PPG signals, the PP (peak-to-peak) interval is employed[16]. For simplicity, we will use the RR interval throughout this paper.)



[17]

Figure 3. RR interval

Source: "Yoga Improves Autonomic Control in Males : A Preliminary Study Into the Heart of an Ancient Practice"

sdnn: standard deviation of NN intervals. NN intervals are derived from RR intervals, excluding unreliable RR intervals[18,19].

sdsd: standard deviation of successive differences in interbeat intervals[20], reflects the variability in the change of RR intervals from one beat to the next.

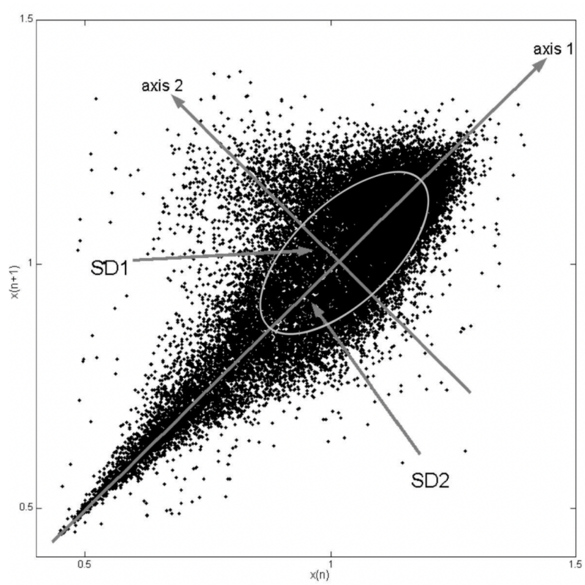
rmssd: the root mean square of successive RR interval differences[18], reflects the variability in the duration of the RR intervals.

pnn20/50: percentage of consecutive RR intervals that vary by more than 20/50 milliseconds[18].

hrmad: median absolute deviation of RR intervals[20].

sd1/sd2: related to Poincaré analysis. Here, RR intervals were plotted against one another in a scatter plot called the Poincaré plot, which enables us to visualize the data's variability. SD1 represents the standard deviation of distances between successive RR intervals from axis 1 and relates to short-term variability, while SD2 represents the standard deviation of distances between successive RR intervals from axis 2 and relates to long-term variability.

s: area of the ellipse.



[21]

Figure 4. Poincaré plot fitted with an ellipse and descriptors SD1 and SD2
Source: Poincaré Plots in Analysis of Selected Biomedical Signals

Breathingrate: number of breaths taken per minute.

Fast Fourier Transform (FFT) is applied to the original data and used as input to evaluate the model's performance. FFT is a mathematical technique used for transforming a signal from the time-domain to the frequency-domain. To maintain a balance between the maximum number of timesteps and the maximum number of records, only the initial 15000 data points (37.5 seconds) in each record were chosen for FFT processing. Since FFT is symmetric, the first half of the FFT transformed data points were retained for future analysis. This FFT transformed dataset was then standardized using a standard scaler.

We make the following modifications to the string labels:

$$\text{rest} \rightarrow 0$$
$$\text{squat} \rightarrow 1$$
$$\text{step} \rightarrow 2$$

All classes 0 through 2 whose images or results appear below correspond to this relationship.

Correlations between the HRV variables were calculated, and some highly correlated variables were removed from the original dataset to generate a new dataset.

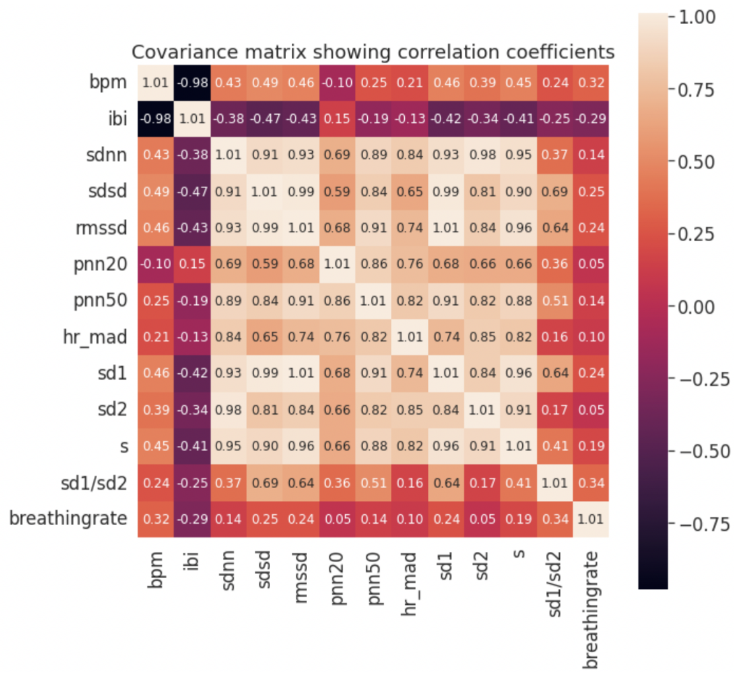


Figure 5. Correlation matrix of HRV variables

2.1.2. Preprocessing for ECG dataset

This data set was imported using the Waveform Database Python Package (wfdb). Labels were extracted from 'scp_statements.csv' and paired with raw ECG signals. Nan values in labels were removed, along with the corresponding raw ECG signals.

To extract HRV variables from the ECG records, the raw signals were first divided into five corresponding categories. An average was taken on 12-lead ECG data to make it 1-lead, and the package heartpy was then applied. Baseline wander and bandpass filters of [0.5 Hz, 40 Hz] were performed using functions in heartpy. A threshold of 130 bpm was determined, and records that generated bpm above 130 were regarded as corrupted. Similar to the PPG dataset, datasets of HRV variables of ['bpm', 'ibi', 'sdnn', 'sdsd', 'sdnn', 'sdsd', 'rmssd', 'pnn20', 'pnn50', 'hr mad', 'sd1', 'sd2', 's', 'sd1/sd2', 'breathingrate'] were generated. This dataset was further cleaned by removing NaN's and inf.

Fast Fourier Transform was applied to the original filtered dataset in terms of 12-leads and average 12-leads, and those data were saved as separate datasets for future use. Since ECG represents the electrical activity of the heart over time, FFT can be used to analyze the various frequency components of the ECG signal when applied to ECG data[22]. Since each data sample contains 1000 timesteps and the FFT is symmetric, only the first 500 FFT transformations were considered for computational efficiency.

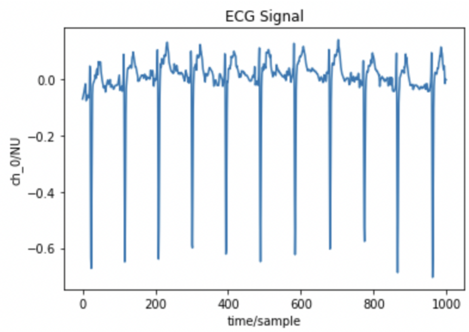


Figure 6. One example of ECG data in lead 6 (V1)

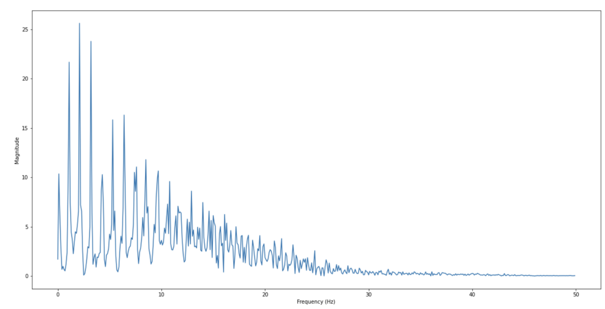


Figure 7. FFT transformed data on lead 6 (V1)

We make the following modifications to the string labels:

NORM → 0
MI → 1
STTC → 2
CD → 3
HYP → 4

All classes 0 through 4 that appear in the images or results below correspond to this relationship.

2.2. Exploratory Data Analysis

2.2.1. EDA for PPG data

To understand the PPG data better, the summary statistics for each category after preprocessing are printed below:

Summary Statistics for rest category after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50	hr_mad
count	34.000	34.000	34.000	34.000	34.000	34.000	34.000	34.000
mean	72.854	831.291	62.917	31.990	46.909	0.550	0.185	40.331
std	7.076	82.662	25.228	16.390	20.893	0.110	0.133	16.368
min	56.748	681.969	29.262	14.724	24.161	0.379	0.035	20.000
25%	67.653	782.797	41.926	17.438	29.069	0.464	0.075	27.188
50%	73.526	816.045	59.006	29.903	44.196	0.546	0.160	37.500
75%	76.648	886.882	76.634	40.567	57.761	0.659	0.285	47.500
max	87.981	1057.306	124.375	69.018	96.711	0.761	0.478	87.500

	sd1	sd2	s	sd1/sd2	breathingrate
count	34.000	34.000	34.000	34.000	34.000
mean	33.156	80.079	9513.885	0.417	0.149
std	14.756	29.532	7600.475	0.092	0.079
min	17.083	34.600	2239.504	0.262	0.000
25%	20.551	56.387	3542.619	0.347	0.126
50%	31.247	75.587	7819.865	0.399	0.133
75%	40.832	99.874	12460.373	0.476	0.167
max	68.381	152.739	31296.349	0.656	0.300

Summary Statistics for squat category after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50	hr_mad
count	34.000	34.000	34.000	34.000	34.000	34.000	34.000	34.000
mean	101.104	609.187	76.152	50.014	68.438	0.472	0.194	38.529
std	17.290	96.136	43.755	45.444	59.854	0.204	0.195	26.906
min	79.064	413.300	26.824	10.917	19.398	0.162	0.000	13.750
25%	89.806	532.970	44.719	16.508	26.353	0.317	0.046	21.250
50%	97.096	618.008	59.096	24.596	35.016	0.412	0.098	30.625
75%	112.584	668.123	99.340	78.281	102.454	0.661	0.353	42.812
max	145.173	758.881	176.900	154.189	225.442	0.909	0.662	131.250

	sd1	sd2	s	sd1/sd2	breathingrate
count	34.000	34.000	34.000	34.000	34.000
mean	47.980	91.214	19304.194	0.469	0.207
std	42.134	47.374	25163.726	0.198	0.069
min	13.715	34.709	1734.073	0.184	0.100
25%	18.546	56.043	3489.527	0.334	0.163
50%	24.475	76.900	6043.849	0.401	0.200
75%	72.418	116.477	24724.823	0.636	0.233
max	159.350	187.039	93634.108	0.890	0.387

Summary Statistics for step category after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50	hr_mad
count	35.000	35.000	35.000	35.000	35.000	35.000	35.000	35.000
mean	107.321	577.255	77.242	65.188	85.619	0.457	0.233	36.000
std	19.878	102.849	43.556	41.608	60.202	0.226	0.218	26.102
min	78.214	399.326	24.504	7.129	12.474	0.106	0.000	12.500
25%	92.863	507.550	44.369	24.305	32.561	0.262	0.070	20.000
50%	107.542	557.923	65.315	57.963	72.575	0.419	0.161	27.500
75%	118.216	646.121	103.119	96.119	120.832	0.627	0.372	38.750
max	150.253	767.125	177.547	139.191	250.029	0.973	0.892	130.000

	sd1	sd2	s	sd1/sd2	breathingrate
count	35.000	35.000	35.000	35.000	35.000
mean	60.191	85.355	21437.410	0.700	0.210
std	42.408	49.102	26270.385	0.350	0.079
min	8.818	33.397	925.144	0.186	0.000
25%	22.983	45.741	3829.003	0.487	0.167
50%	50.796	68.751	9691.215	0.621	0.200
75%	84.943	109.440	26127.518	0.858	0.267
max	176.544	204.882	113633.381	1.706	0.339

Figure 8. Summary statistics for HRV variables

Observe that the mean bpm and mean breathing rate for the rest category are lower than the respective values for the squat and step categories, which is intuitively expected. There is also a smaller standard deviation in the bpms for the rest category compared to the squat and step categories. This indicates that individuals generally have similar bpm at rest, but their bpm during exercise can vary depending on their physical abilities. Ibi tends to decrease during activity, which corresponds to an increase in heart rate. Breaths are taken more frequently during an exercise, which leads to lower breathing rates in the rest category than the other two. Higher sd1/sd2 means higher variabilities in consecutive RR intervals, meaning the step category has the highest variabilities in consecutive RR intervals.

More EDA methods were considered, such as 2D Multidimensional Scaling and t-SNE. MDS indicates a set of objects as points in a multidimensional space such that points corresponding to similar objects are near each other and those far apart objects are dissimilar[23]. t-SNE is predominantly employed to comprehend high-dimensional data and project it into low-dimensional space, 2D in this case. In the filtered unstandardized data, variables have different scales, ranges, and units (as shown in figure 8), which impacts the relative distance between graphed points. Standardization transforms the original data to have the same scale and range and ensures that all variables contribute equally to determining the distance between the points. Before standardization, as depicted on the left side of Figure 9, there appears to be a pattern among the various categories; however, on the right side of Figure 9, the points are dispersed, showing that standardization eliminates some patterns. The purple dots on the left have only a few points in common with other categories; therefore, if we wish to distinguish purple dots from the remaining, it is likely preferable to use unstandardized data.

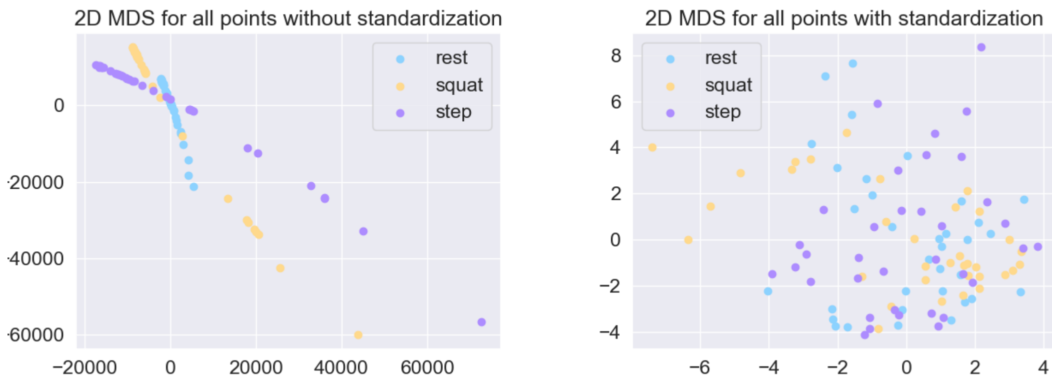


Figure 9. 2D MDS with (non)standardized dataset

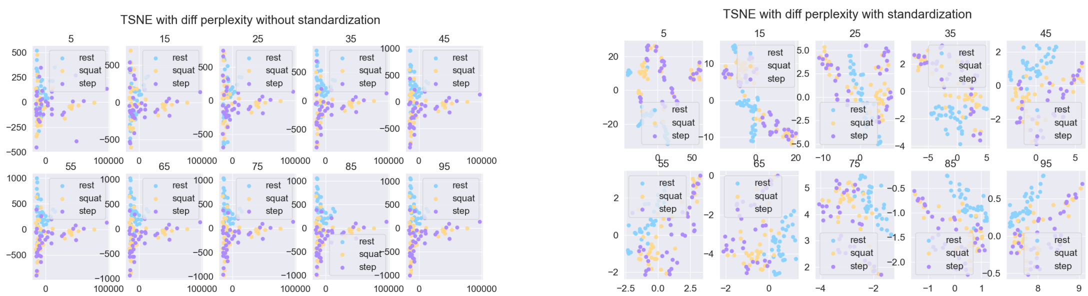


Figure 10. t-SNE with (non)standardized dataset with different perplexities

MDS and t-SNE are effective non-linear transformations for separating data visualizations, but they tend to deform the space in order to highlight the distinction. Thus, we also consider a linear transformation, PCA, for visualizations.

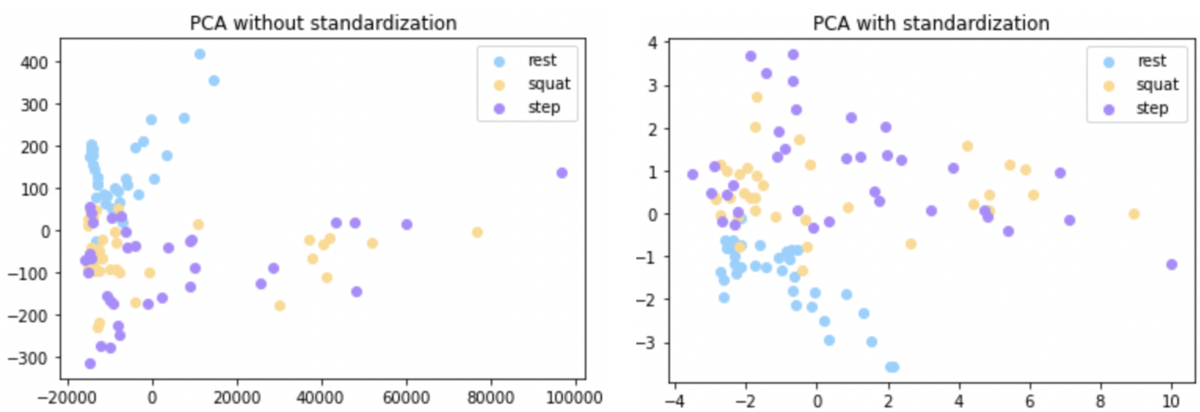


Figure 11. PCA with (non)standardized dataset

PCA was performed on both standardized and nonstandardized datasets, and we observe that the rest category is easily distinguishable from the squat and step categories. However, there is considerable overlap between squat and step classes in both datasets.

2.2.2. EDA for ECG data

To understand the ECG data better, the summary statistics for each category after preprocessing are as follows:

Summary Statistics for NORM after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50
count	7965.000	7965.000	7965.000	7965.000	7965.000	7965.000	7965.000
mean	71.415	865.713	27.801	17.791	31.680	0.302	0.102
std	12.778	147.351	26.125	20.699	36.551	0.291	0.199
min	37.618	461.667	0.000	0.000	0.000	0.000	0.000
25%	62.581	764.444	11.780	7.370	12.748	0.000	0.000
50%	69.767	860.000	19.645	11.547	20.976	0.250	0.000
75%	78.488	958.750	33.704	19.272	35.355	0.500	0.143
max	129.964	1595.000	283.507	315.000	515.024	1.000	1.000

	hr_mad	sd1	sd2	s	sd1/sd2	breathingrate
count	7965.000	7965.000	7.965e+03	7965.000	7965.000	7965.000
mean	18.276	21.431	2.934e+01	3417.153	0.860	0.197
std	18.116	25.055	2.701e+01	9110.242	1.633	0.073
min	0.000	0.000	2.274e-13	0.000	0.000	0.000
25%	10.000	8.478	1.183e+01	337.230	0.508	0.134
50%	10.000	13.997	2.108e+01	918.042	0.722	0.168
75%	20.000	23.908	3.658e+01	2653.528	1.000	0.252
max	200.000	364.160	3.083e+02	192052.821	103.000	0.538

Summary Statistics for STTC after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50
count	2581.000	2581.000	2581.000	2581.000	2581.000	2581.000	2581.000
mean	78.177	802.869	38.001	28.857	51.532	0.313	0.179
std	16.805	170.557	44.442	37.986	69.782	0.346	0.299
min	39.474	465.556	2.421	0.000	0.000	0.000	0.000
25%	65.753	677.692	9.166	6.556	10.690	0.000	0.000
50%	75.949	790.000	17.321	10.375	18.397	0.167	0.000
75%	88.536	912.500	49.355	30.067	57.096	0.600	0.286
max	128.878	1520.000	244.067	265.000	560.089	1.000	1.000

	hr_mad	sd1	sd2	s	sd1/sd2	breathingrate
count	2581.000	2581.000	2.581e+03	2581.000	2.581e+03	2581.000
mean	23.516	34.851	3.501e+01	8636.633	3.587e+10	0.211
std	32.427	47.624	4.016e+01	18402.972	1.357e+12	0.084
min	0.000	0.000	1.137e-13	0.000	0.000e+00	0.000
25%	5.000	7.071	8.898e+00	211.261	5.890e-01	0.136
50%	10.000	12.373	1.778e+01	694.655	8.866e-01	0.207
75%	25.000	38.198	4.443e+01	5104.338	1.291e+00	0.261
max	220.000	395.980	2.633e+02	148668.049	6.156e+13	0.634

Summary Statistics for MI after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50
count	1977.000	1977.000	1977.000	1977.000	1977.000	1977.000	1977.000
mean	75.629	826.498	28.943	21.020	37.206	0.252	0.113
std	15.838	163.785	34.971	30.494	54.305	0.303	0.239
min	40.472	461.579	0.000	0.000	0.000	0.000	0.000
25%	64.319	706.364	9.162	6.389	10.607	0.000	0.000
50%	72.464	828.000	15.612	9.428	16.903	0.143	0.000
75%	84.942	932.857	29.606	17.058	31.623	0.429	0.083
max	129.989	1482.500	232.086	207.771	409.176	1.000	1.000

	hr_mad	sd1	sd2	s	sd1/sd2	breathingrate
count	1977.000	1977.000	1.977e+03	1977.000	1977.000	1977.000
mean	17.466	24.806	2.716e+01	4846.515	1.024	0.212
std	24.614	36.671	3.130e+01	13546.925	1.760	0.080
min	0.000	0.000	1.137e-13	0.000	0.000	0.000
25%	5.000	7.071	8.819e+00	215.090	0.554	0.140
50%	10.000	11.158	1.558e+01	534.477	0.845	0.215
75%	20.000	20.817	3.053e+01	1865.622	1.183	0.261
max	200.000	284.753	2.541e+02	170520.778	67.060	0.565

Summary Statistics for CD after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50
count	3898.000	3898.000	3898.000	3898.000	3898.000	3898.000	3898.000
mean	76.976	812.839	32.334	24.525	43.413	0.277	0.137
std	16.043	165.735	39.989	35.328	64.322	0.319	0.259
min	39.867	462.632	0.000	0.000	0.000	0.000	0.000
25%	65.217	691.111	8.660	6.389	10.541	0.000	0.000
50%	74.720	803.000	15.947	9.798	17.744	0.143	0.000
75%	86.817	920.000	35.419	20.616	38.818	0.500	0.167
max	129.693	1505.000	354.714	350.000	717.161	1.000	1.000

	hr_mad	sd1	sd2	s	sd1/sd2	breathingrate
count	3898.000	3898.000	3.898e+03	3898.000	3.898e+03	3898.000
mean	18.729	29.447	2.971e+01	6350.549	4.627e+11	0.214
std	26.270	43.998	3.490e+01	16466.328	2.742e+13	0.083
min	0.000	0.000	1.137e-13	0.000	0.000e+00	0.000
25%	5.000	7.071	8.246e+00	192.859	5.947e-01	0.139
50%	10.000	11.934	1.618e+01	596.180	8.801e-01	0.215
75%	20.000	26.205	3.513e+01	2717.324	1.246e+00	0.263
max	330.000	498.446	2.682e+02	233374.227	1.710e+15	0.509

Summary Statistics for HYP after preprocessing

	bpm	ibi	sdnn	sdsd	rmssd	pnn20	pnn50
count	1536.000	1536.000	1536.000	1536.000	1536.000	1536.000	1536.000
mean	76.773	816.984	33.425	26.069	46.392	0.280	0.149
std	16.631	169.698	41.074	36.657	66.909	0.324	0.272
min	42.313	462.222	2.665	0.000	0.000	0.000	0.000
25%	64.777	693.333	8.765	6.325	10.801	0.000	0.000
50%	73.892	812.000	15.795	9.798	17.321	0.143	0.000
75%	86.538	926.250	34.956	22.226	41.503	0.500	0.200
max	129.808	1418.000	287.750	280.179	512.916	1.000	1.000

	hr_mad	sd1	sd2	s	sd1/sd2	breathingrate
count	1536.000	1536.000	1536.000	1536.000	1536.000	1536.000
mean	19.443	31.633	30.172	6930.646	1.195	0.212
std	28.167	46.361	35.522	17068.530	3.040	0.079
min	0.000	0.000	2.121	0.000	0.000	0.000
25%	5.000	7.266	8.246	186.813	0.629	0.140
50%	10.000	11.547	16.060	575.308	0.933	0.215
75%	20.000	28.252	34.982	2908.505	1.291	0.261
max	250.000	362.685	242.970	245253.110	92.064	0.522

Figure 12. Summary statistics on HRV variables

Note that the mean bpm and ibi are similar across categories, whereas sd2, which relates to long-term variability, and sd1/sd2 vary significantly across categories.

2D MDS and t-SNE are also performed on this dataset.

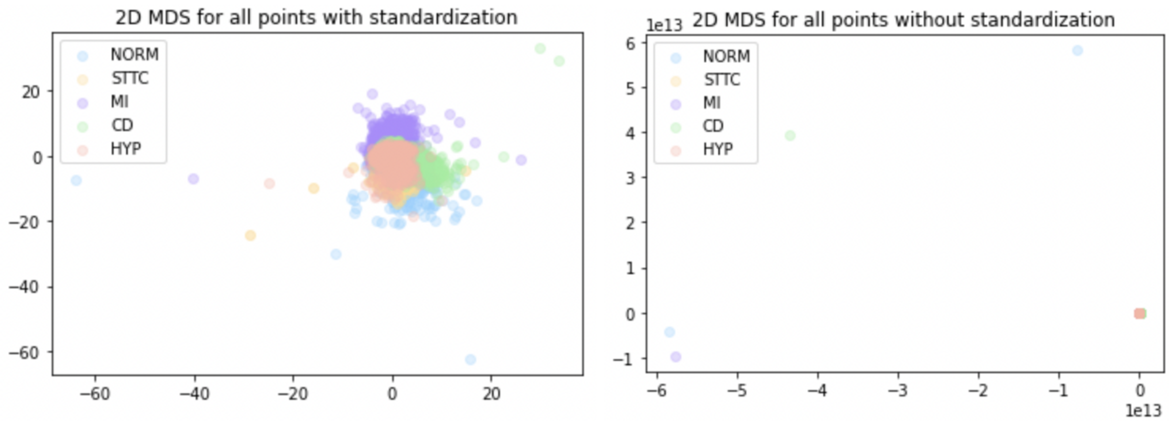


Figure 13. 2D MDS with (non)standardized dataset

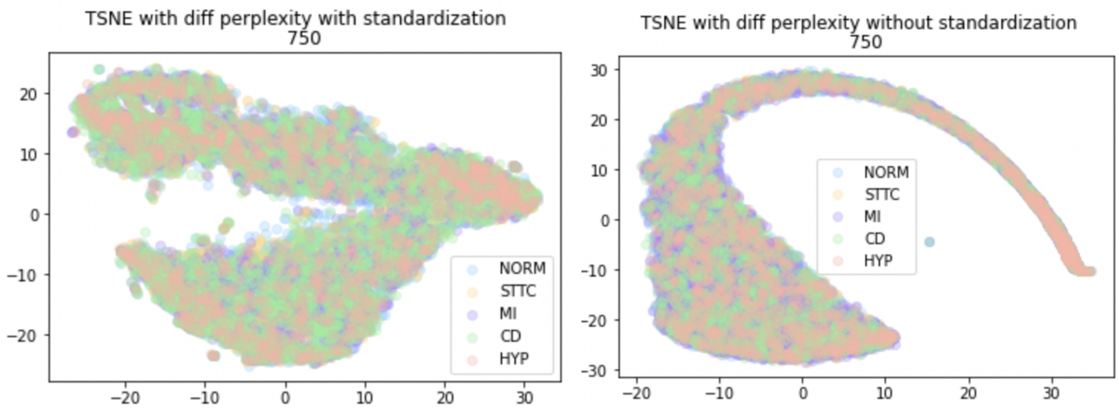


Figure 14. TSNE with (non)standardized dataset

Observe that almost all of the points overlap, indicating that it is difficult to distinguish between the categories using only the HRV variables.

2.3. Experimental setup

2.3.1. For PPG dataset

To gain a preliminary comprehension of the dataset, HRV variables were analyzed. To determine the amount of information contained in HRV features, non-Neural Network models were initially trained on summary statistics of HRV features alone, containing values of ['bpm', 'ibi','sdnn','sdsd', 'rmssd', 'pnn20', 'pnn50', 'hr mad','sd1','sd2','s','sd1/sd2', 'breathingrate']. The primary package used for analysis in this phase was sklearn, and pyplot from matplotlib was used for plotting. KNN, Random Forest, Naive Bayes Classifier, Linear Classifier, and the Multilayer Perceptron Model are non-NN machine learning models that were trained on this dataset. These non-NN machine learning models also utilized the transformed FFT dataset as inputs afterwards. Cross validation of five folds was considered in each of the models. Since the original dataset contains only 103 data samples, more than 5 folds would result in each test set being too small, and the final cross validation score would lose its persuasiveness. The values of the eigenvalues indicate that there are considerable variations between them. Due to the fact that Euclidean distance assigns equal weights to all attributes, resulting in a skewed distribution, Mahalanobis distance was considered in some of the models.

2.3.2. For ECG dataset

KNN, random forest, Naive Bayes classifier, linear classifier, multilayer perceptron model, and gradient boosting classifier were utilized to initiate the classification process. However, models based solely on HRV variables perform poorly. Due to the possibility of time differences when collecting 12-lead ECG data, the initial modification consisted of only considering chest-mounted 6-lead ECG data. However, it performed as badly as the average on 12-leads ECG data. Transformation based on principal component analysis (PCA) was then considered. After applying PCA transformations to the HRV variables dataset, models were applied; however, the performance was still inadequate.

Before training the original raw data, we considered fast fourier transformation (FFT) on the original filtered dataset. Multiple fundamental machine learning models were applied to 12-lead and 6-lead ECG data averages. We saw a slight improvement in performance, but not a substantial one. Following this, HRV variables and FFT data were combined and used as inputs for the models, but performance deteriorated.

An oversampler with parameter “distance_SMOTE” from the smote_variants Python module was used to circumvent the problem of imbalance in the dataset. SMOTE is short for Synthetic Minority Oversampling Technique, which oversamples the minority class by adding synthetic examples to the original data for each minority class sample. The “distance_SMOTE” parameter uses the weighted distance to locate the closest examples of the minority classes. The mean example was then obtained by averaging the k nearest neighbors, where k is a user-specified number (I set k to 5)[24]. Using this oversampler, each category was oversampled to achieve the same size as the "NORM" class, which is the most frequent class in the original dataset. The fundamental machine learning models were then applied once more, and satisfactory results were obtained. Cross validation of 5 folds were used in all basic machine learning models to calculate their performance.

However, we would still like to generate models from the original dataset that was not oversampled. Thus, we go further to build deep learning models on the original filtered dataset. Convolutional Neural Network (CNN), Inception, and Resnet were considered. All inputs have the format (21388, 1000, 12), where 21388 represents the number of samples, 1000 represents the number of time steps, and 12 represents the number of leads.

CNN was chosen as a starting point due to its simplicity of implementation; however, if we have deep structures of ECG data, it may suffice. ResNet and Inception are two state-of-the-art deep learning models that are more challenging to interpret and comprehend.

ResNet is designed to address the issue of vanishing gradients that arises during the training of extremely deep neural networks. Inception is a family of neural network architectures that prioritizes the cost of computation.

The structure of CNN is presented below. ResNet and Inception are harder to interpret, so structures are not provided.

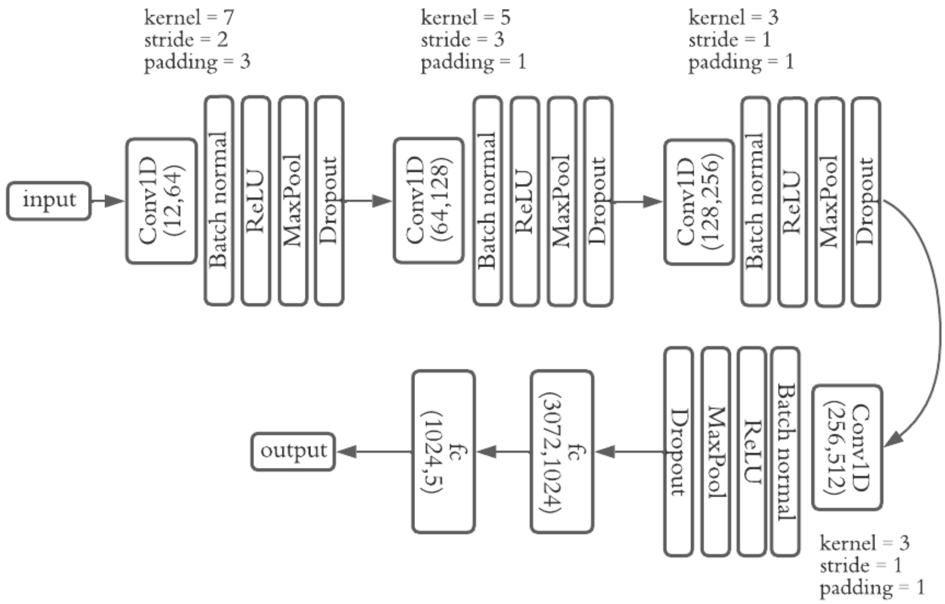


Figure 15. 1D CNN structure

3. Results

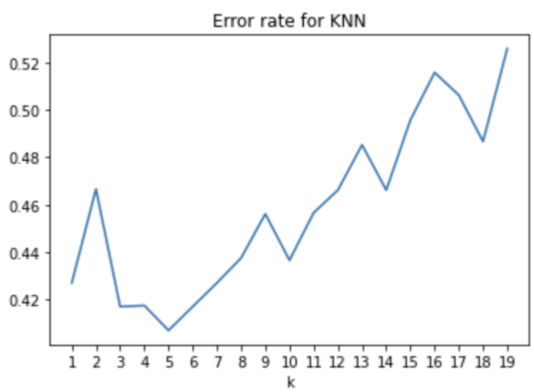
3.1. For PPG dataset

3.1.1. Basic Machine Learning models

3.1.1.1. K-Nearest Neighbors

For a given new sample, KNN examines the K nearest training samples and assigns the class label that occurs most frequently among these K samples as the predicted class label for the new sample.

```
mean cv scores: 0.5828571428571429
```



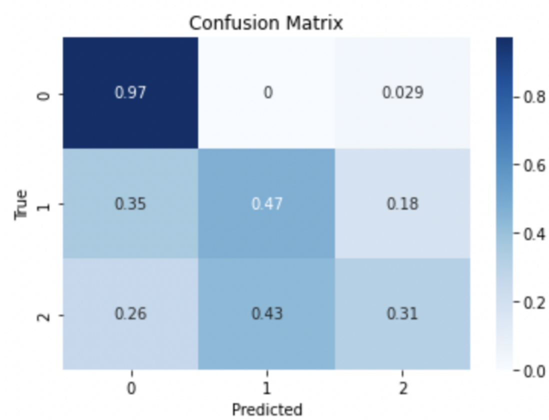
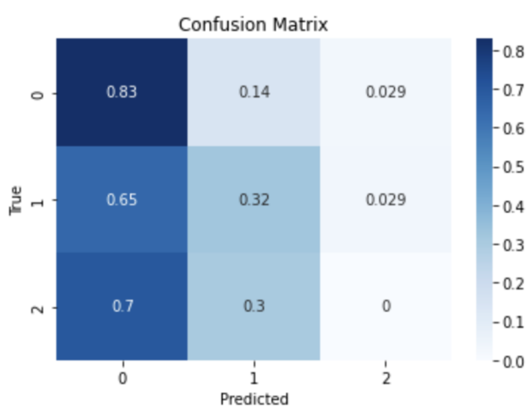
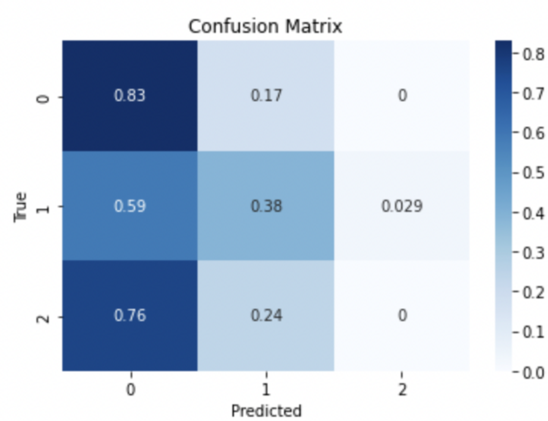


Figure 16. Error rate graph of KNN with respect to k with HRV variables as inputs
Figure 17. Mean 5-fold accuracy and confusion matrix with HRV variables as inputs

```
For original dataset with KNN
mean 5-fold accuracy: 0.3919047619047619
```

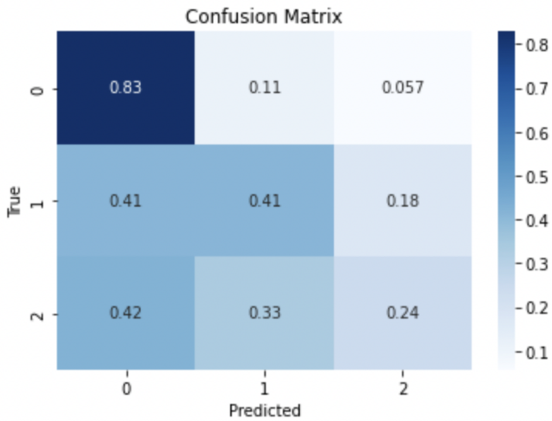



```
For standardized original dataset with KNN
mean 5-fold accuracy: 0.41238095238095235
```



```
For FFT dataset with KNN
mean 5-fold accuracy: 0.5014285714285714
```

```
For standardized FFT dataset with KNN
mean 5-fold accuracy: 0.3052380952380952
```



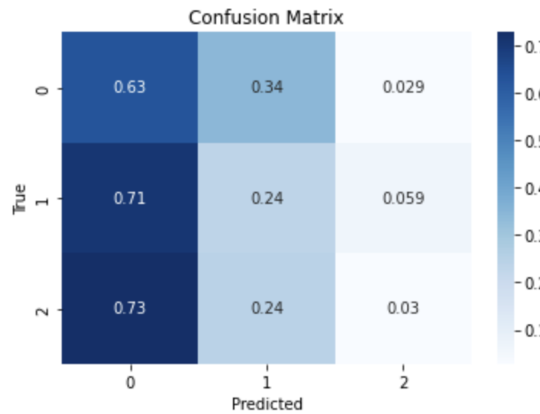


Figure 18. Confusion matrix for KNN

Notice that KNN with HRV variables has the highest accuracy, and the true positive rate for the rest category remains the highest for all inputs. Note that KNN barely gets the step class right and predominantly predicts all samples to be in the rest class. For the KNN model that takes HRV variables as inputs, we can break it down further to analyze the contribution of each HRV variable to the model. Using permutation, the importance of each variable is printed below. If a feature is important, permuting its values should significantly degrade the model's performance, whereas permuting the values of an unimportant feature should have little or no effect.

```
For KNN
permutation importance:
sd1 0.30194174757281556
ibi 0.27961165048543685
rmssd 0.2766990291262136
sd2 0.2087378640776699
sdnn 0.18058252427184468
bpm 0.12233009708737867
sdsd 0.11747572815533984
s 0.10000000000000002
pnn50 0.07378640776699029
pnn20 0.03398058252427184
sd1/sd2 0.024271844660194164
hr_mad 0.020388349514563097
breathingrate 0.014563106796116498
```

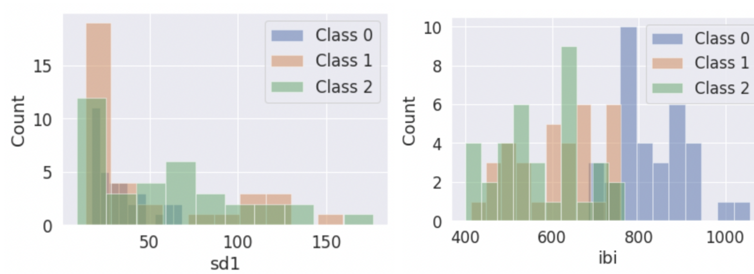


Figure 19. Permutation importance of variables and distributions of variables

Note here that the variables “sd1” and “ibi” have the highest importance for the performance of the model. However, the histogram plots show that there is lots of overlap between class 1 and class 2, which might lead to misclassification in the model.

Some improvements in the accuracy of the model were made when removing highly correlated variables from the HRV variables, especially for the rest and squat categories.

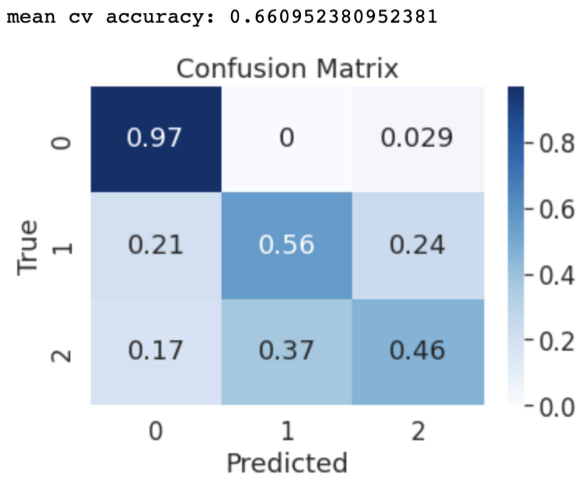


Figure 20. Mean 5-fold accuracy and confusion matrix with selected HRV variables as inputs

After removing some of the highly correlated variables ('sd1','sd1','sd2','s','sd1/sd2'), we obtain a slightly better result for KNN.

3.1.1.2. Random Forest

sklearn.model_selection.RandomizedSearchCV was used to find an optimized combination of hyperparameters for random forests.

Since a random forest involved a lot of randomization, the result kept changing even when the parameters remained the same. As a result, both the output of the RandomizedSearchCV and the accuracy provided by the best parameters chosen by the RandomizedSearchCV were constantly changing. Obtaining the parameter combination from RandomizedSearchCV, cross validation of 5 folds was then applied to the best estimate out of the sample accuracy. However, no matter how the parameter combination changed, the accuracy of random forest was always between 0.60 and 0.70.

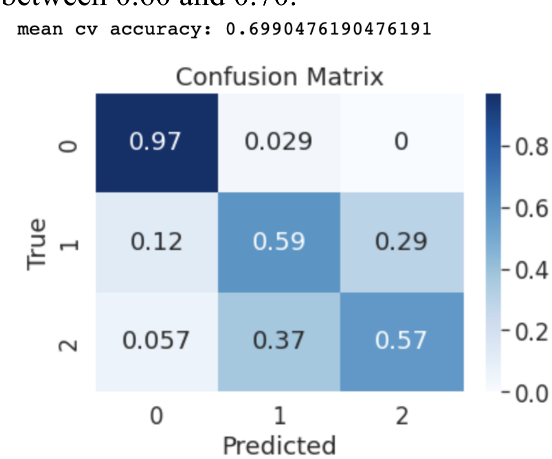


Figure 21. Mean 5-fold accuracy and confusion matrix with HRV variables as inputs

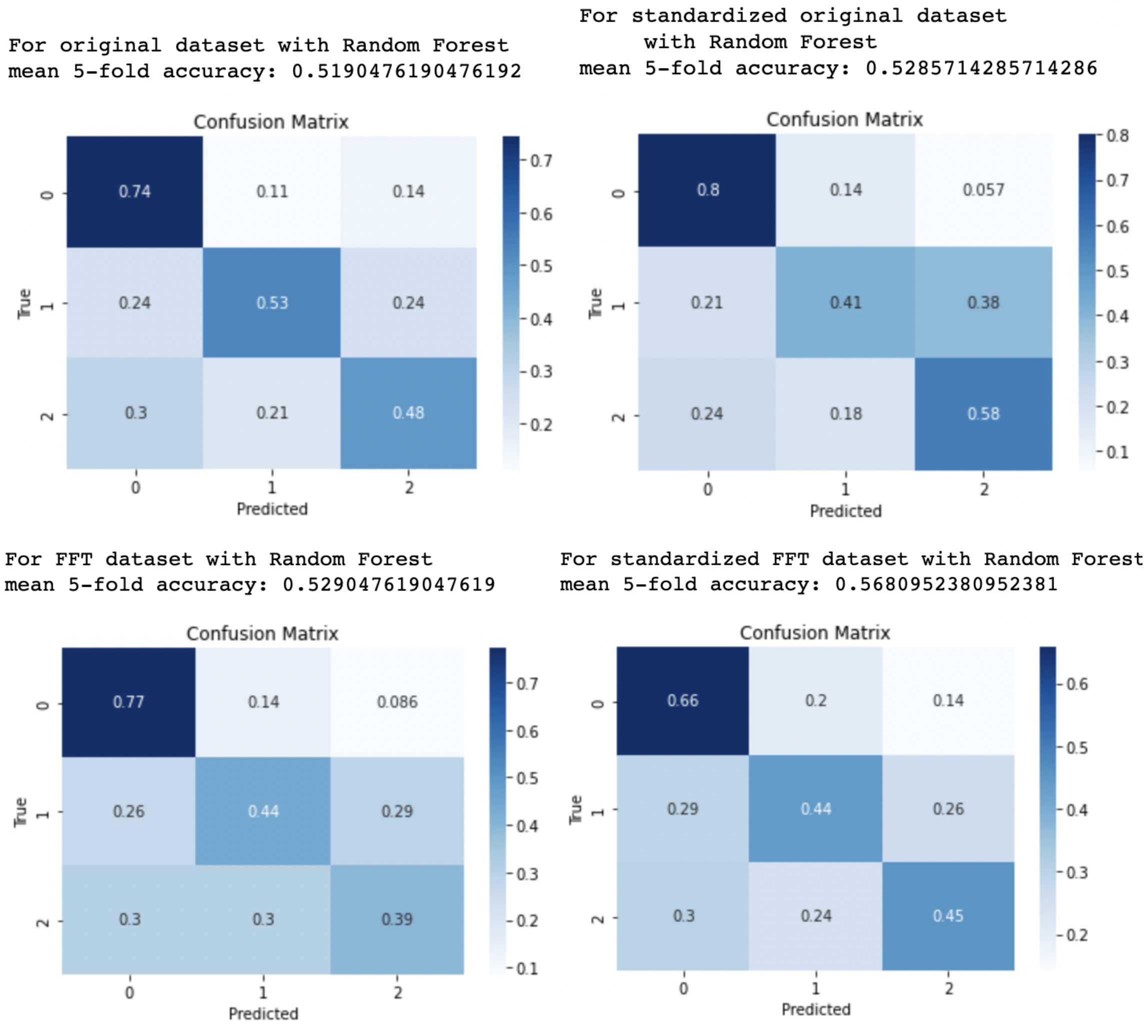


Figure 22. Accuracy and confusion matrix for Random Forest

Random forest with HRV variables as inputs had the highest accuracy, with some improvements in classifying the squat and step classes. However, misclassifications between the step and squat classes were still common. Selected HRV variables were also taken as inputs, but the accuracy was about the same.

```
bpm: 0.1830
ibi: 0.2026
sdnn: 0.0469
sdsd: 0.0592
rmssd: 0.0606
pnn20: 0.0830
pnn50: 0.0430
hr_mad: 0.0335
sd1: 0.0406
sd2: 0.0469
s: 0.0515
sd1/sd2: 0.1024
breathingrate: 0.0468
```

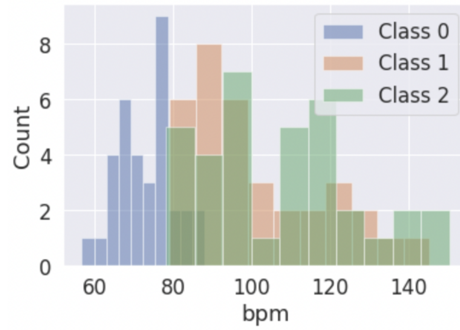


Figure 23. Permutation importance of variables and distributions of variables

Similarly, there is lots of overlap in the distribution of bpm for class 1 and 2, which might cause trouble for random forest to differentiate between class 1 and class 2.

3.1.1.3. Naive Bayes Classifier

The Naive Bayes algorithm is a Bayesian probabilistic machine learning algorithm. Given the class label, Naive Bayes assumes that the features are conditionally independent, which means that the presence of one feature does not impact the probability of the presence of another feature. The naive assumption can lead to suboptimal performance. In addition, Naive Bayes assumes that the features are categorical, which is not true in this case. This classifier also assumes a linear relationship between the features and the label, which may not be true in practice.

```
mean cv scores: 0.5514285714285714
```

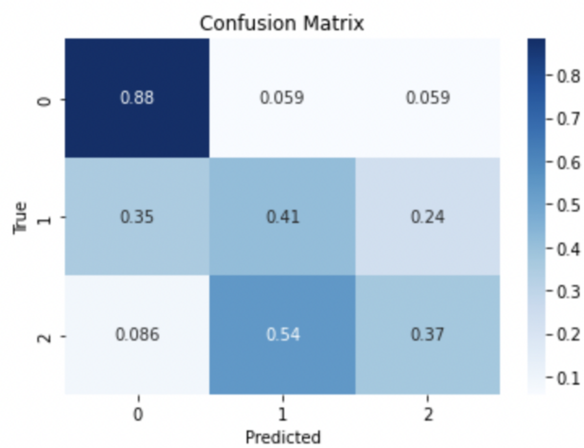


Figure 24. Mean 5-fold accuracy and confusion matrix with standardized HRV variables as inputs

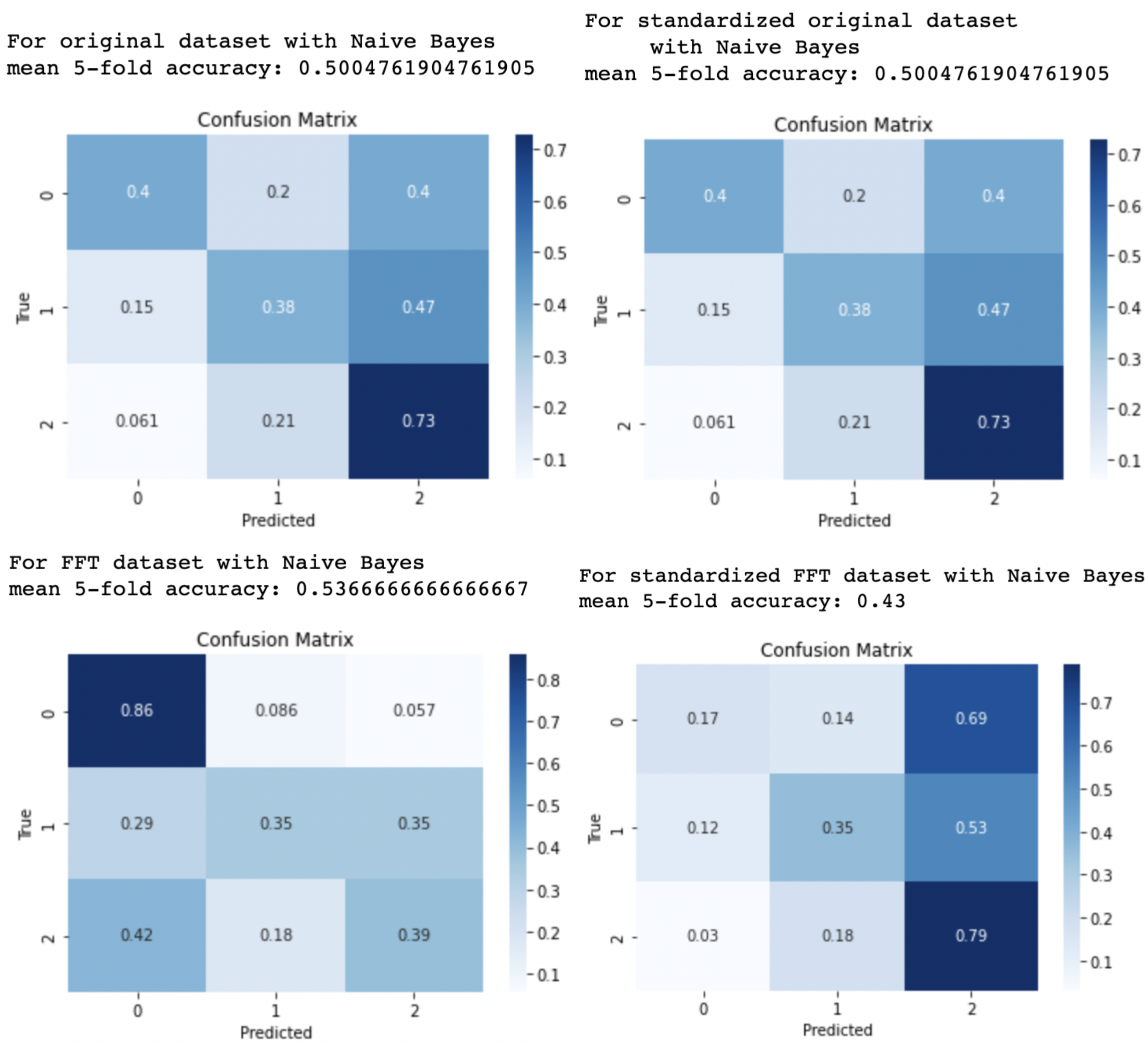


Figure 25. Accuracy and confusion matrix for Naive Bayes Classifier

3.1.1.4. Linear Classifier

In a linear classifier, a linear boundary is used to separate different classes. Here, sklearn.linear_odel.SGDClassifier was used. By default, it fits a linear support vector machine (SVM) and employs stochastic gradient descent (SGD) as the optimization algorithm for determining the linear model's weights. SVMs and other linear classifiers inherently perform binary classification, which might result in reduced performance.

```
mean cv scores: 0.6414285714285715
```

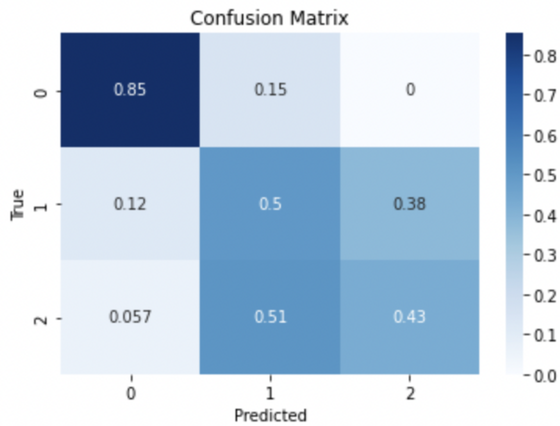
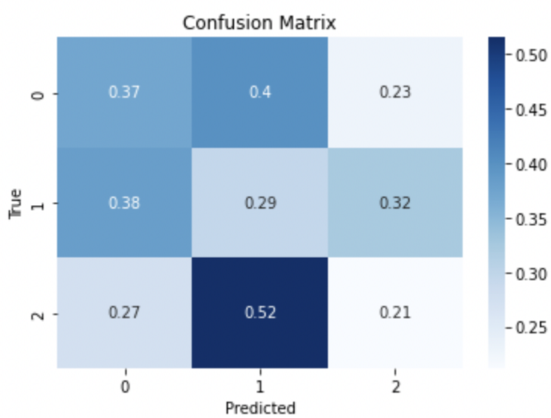
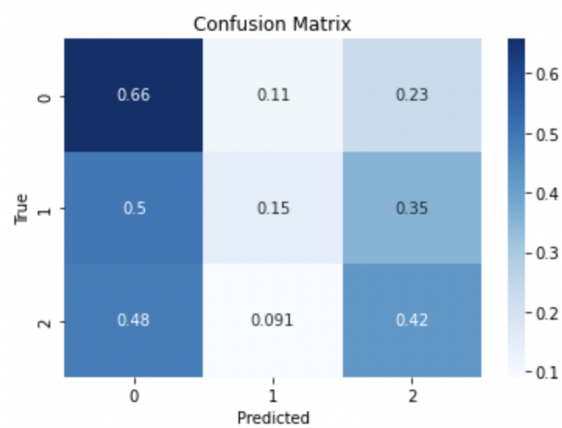


Figure 26. Mean 5-fold accuracy and confusion matrix with standardized HRV variables as inputs

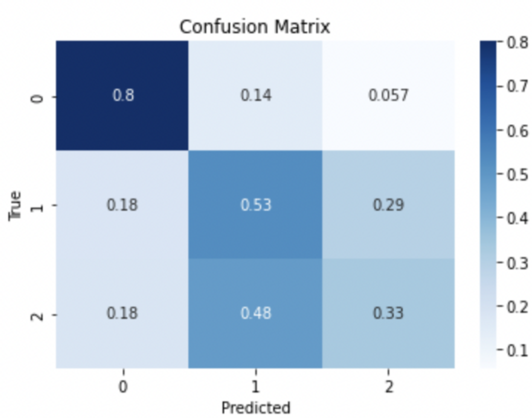
```
For original dataset with Linear Classifier
mean 5-fold accuracy: 0.3433333333333334
```




```
For standardized original dataset
     with Linear Classifier
mean 5-fold accuracy: 0.4419047619047619
```




```
For FFT dataset with Linear Classifier
mean 5-fold accuracy: 0.5766666666666668
```




```
For standardized FFT dataset
     with Linear Classifier
mean 5-fold accuracy: 0.44761904761904764
```

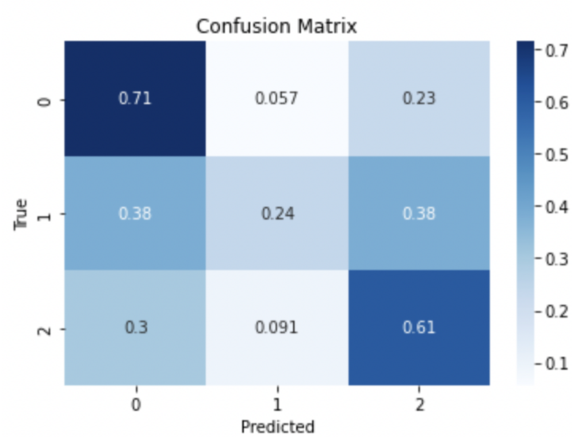


Figure 27. Accuracy and confusion matrix for Linear Classifier

3.1.1.5. Multilayer Perceptron Model

For the Multilayer perceptron Model, I used sklearn.neural_network.MLPClassifier. Similarly, cross validation and standardized data were used. Among the layer values I tried, the layer [128,64,32,8] gave the highest accuracy value, which reached 0.6805.

```
mean 5 fold accuracy: 0.6804761904761906
```

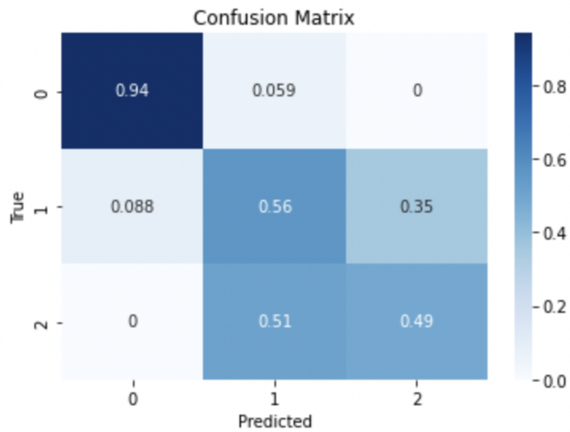
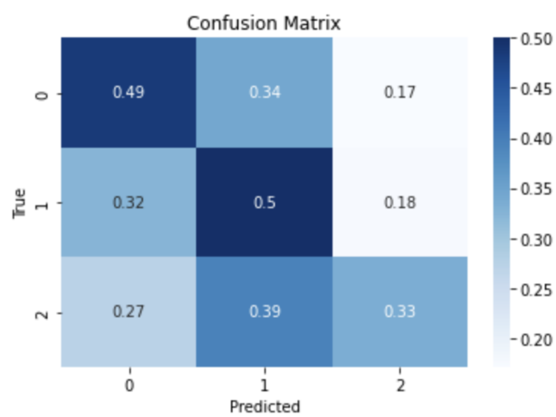
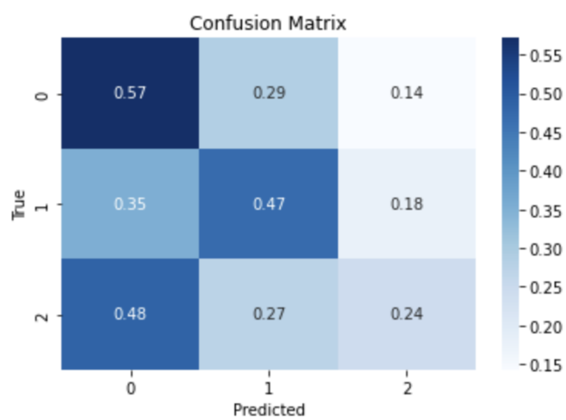


Figure 28. Mean 5-fold accuracy and confusion matrix with standardized HRV variables as input

```
For original dataset with MLP
mean 5-fold accuracy: 0.4323809523809524
```




```
For standardized original dataset with MLP
mean 5-fold accuracy: 0.4023809523809524
```



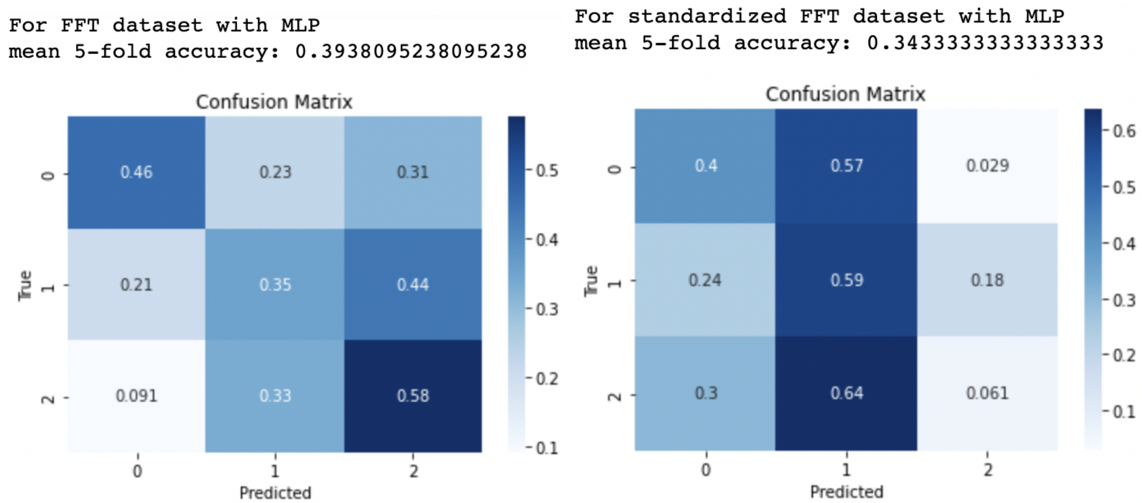


Figure 29. Accuracy and confusion matrix for MLP

3.1.2. Discussion

Summary of basic machine learning models with 5-fold mean accuracy:

	KNN	Random Forest	Naive Bayes	Linear	MLP
HRV variables	0.5829	0.6990	0.5514	0.6414	0.6805
Original	0.3919	0.5190	0.5005	0.3433	0.4323
Standardized original	0.4123	0.5286	0.5005	0.4419	0.4023
FFT	0.5014	0.5290	0.5367	0.5767	0.3938
Standardized FFT	0.3052	0.5681	0.43	0.4476	0.3433

Table 1. Accuracy for different models and inputs

Among all, a random forest classifier with HRV variables as inputs had the best performance, and MLP with HRV variables had similar accuracy. The arrays of accuracy values from the same data fold generated by the cross-validation procedure were examined to determine whether the difference between these two results is significant. Since the dataset was comparatively small, a paired t-test was utilized. As shown in figure 30, a p-value of 0.4493 was obtained for 10 CV folds; thus, there appears to be no statistically significant performance difference between the random forest classifier and the MLP classifier when using standardized HRV variables as inputs if we use an alpha of 0.05.

```
model1:RandomForestClassifier()
model2:MLPClassifier(hidden_layer_sizes=[128, 64, 32, 8], max_iter=1000)
Model 1 accuracy for each fold: [0.72727273 0.90909091 0.81818182 0.6        0.7        0.6
 0.6        0.6        0.7        0.5       ]
Model 2 accuracy for each fold: [0.45454545 1.         0.72727273 0.7        0.9        0.6
 0.5        0.6        1.         0.7       ]
Paired t-test: t-statistic = -0.7909250552164026, p-value = 0.44932721587350444
There is no significant difference in the performance of the two models.
```

Figure 30. Paired t-test results for Random Forest Classifier and MLP using standardized HRV

```
model1:RandomForestClassifier()
model2:KNeighborsClassifier(metric='mahalanobis',
                     metric_params={'V': array([[ 4.71367835e+02, -3.09230564e+03,  3.60607743e+02,
         4.10967614e+02,  5.19084836e+02, -4.05771373e-01,
         9.74747297e-01,  1.07817770e+02,  3.62550983e+02,
         3.62709172e+02,  2.14304745e+05,  1.40811363e+00,
         5.45717483e-01],
       [-3.09230564e+03,  2.15798298e+04, -2.14594325e+03,
        -2.65156812e+03, -3.26442595e+03,  4.0...
         7.13775201e+00,  8.92183233e+00,  1.79019514e-02,
         2.51423566e-02,  1.01829577e+00,  6.27966632e+00,
         1.91407088e+00,  2.38659262e+03,  7.17333329e-02,
         7.18613200e-03],
       [ 5.45717483e-01, -3.40456508e+00,  4.22493552e-01,
         7.67188446e-01,  1.01937339e+00,  7.57521118e-04,
         2.04913406e-03,  1.91302886e-01,  6.95049154e-01,
         1.79222045e-01,  3.36795767e+02,  7.18613200e-03,
         6.41306217e-03]])})
Model 1 accuracy for each fold: [0.72727273 1.         0.81818182 0.5        0.9        0.6
 0.5        0.6        0.7        0.5       ]
Model 2 accuracy for each fold: [0.63636364 0.81818182 0.72727273 0.7        0.6        0.7
 0.5        0.6        0.6        0.4       ]
Paired t-test: t-statistic = 1.268037731171391, p-value = 0.23660386455685378
There is no significant difference in the performance of the two models.
```

Figure 31. Paired t-test results for Random Forest Classifier and KNN using standardized HRV

The p-value for random forest classifier and KNN when using 10 CV folds and standardized HRV variables as inputs is 0.237, which shows that there is no statistically significant difference between the two models if we use an alpha of 0.05.

```
model1:RandomForestClassifier()
model2:GaussianNB()
Model 1 accuracy for each fold: [0.63636364 1.         0.81818182 0.7        0.9        0.5
 0.6        0.6        0.8        0.5       ]
Model 2 accuracy for each fold: [0.81818182 0.90909091 0.45454545 0.5        0.5        0.6
 0.1        0.3        0.7        0.6       ]
Paired t-test: t-statistic = 2.118085072027459, p-value = 0.06323398959613402
There is no significant difference in the performance of the two models.
```

Figure 32. Paired t-test results for Random Forest and Naive Bayes classifiers using standardized HRV

The p-value for random forest classifier and naive bayes classifier when using 10 CV folds and standardized HRV variables as inputs is 0.0632, which shows that there is no statistically significant difference between the two models if we use an alpha of 0.05. However, there is a statistically significant difference between random forest classifier and Naive Bayes classifier if we use an alpha of 0.1.

```
model1:RandomForestClassifier()
model2:SGDClassifier()
Model 1 accuracy for each fold: [0.72727273 1.         0.90909091 0.7        0.9        0.5
 0.4        0.7        0.7        0.5       ]
Model 2 accuracy for each fold: [0.54545455 1.         0.72727273 0.6        0.8        0.4
 0.3        0.6        0.6        0.7       ]
Paired t-test: t-statistic = 2.206455361476754, p-value = 0.054760975339837856
There is no significant difference in the performance of the two models.
```

Figure 33. Paired t-test results for Random Forest and Linear classifiers using standardized HRV

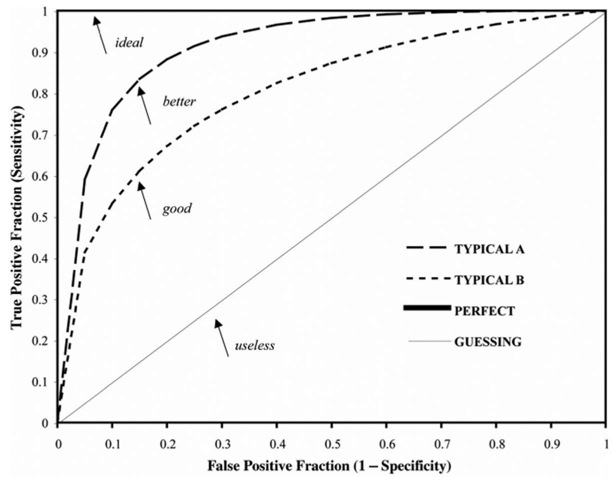
The p-value for random forest classifier and linear classifier classifier when using 10 CV folds and standardized HRV variables as inputs is 0.0548, which shows that there is no statistically significant difference between the two models if we use an alpha of 0.05. However, there is a statistically significant difference between random forest classifier and linear classifier if we use an alpha of 0.1.

Notice that the performance of all models with HRV variables as inputs was superior to that of the same model with other data as inputs. The majority of rest-labeled recordings in the original dataset were significantly longer than the other two categories. To ensure that all records in the original dataset and the FFT-transformed dataset had the same duration, only the initial 15000 timesteps of each data sample were considered. The majority of recordings must be abridged, yielding only 37.5 seconds of data per record, which might not be sufficient for classification models. In contrast, HRV variables incorporated every piece of information in the original dataset, making them more informative than the FFT dataset.

3.2. For ECG dataset

3.2.1. Basic Machine Learning models

Accuracy is a metric that measures the proportion of correct predictions made by the model relative to the total number of predictions. Unlike the PPG dataset, this dataset is extremely unbalanced. Since accuracy does not consider the distribution of classes, a model can obtain a high accuracy score by constantly predicting the majority class. Thus, we considered ROC-AUC scores when training with basic machine learning models for the ECG dataset. ROC-AUC is short for Receiver Operating Characteristic Area Under the Curve. ROC curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) at different probability thresholds and captures the trade off between these two values[25]. AUC refers to the area under the ROC curve. The larger the AUC, the more accurately the model distinguishes between classes. All of the models utilized in this study employed the One-Versus-Rest (OvR) method, which compares each class to the others simultaneously.



[25]

Figure 34. Typical receiver operating characteristic curves along with the upper (perfect) and lower (guessing) bounds
Source: Receiver Operating Characteristic Analysis: Basic Concepts and Practical Applications

Since the difference between the 12-lead and 6-lead datasets was negligible, the following results were based on the 12-lead dataset.

As stated previously, basic machine learning models based solely on HRV variables performed inadequately, and only the model with greatest performance is demonstrated below, which is gradient boosting (random forest has similar performance, and since running a random forest model is faster than running a gradient boosting model, I mainly used a random forest model when using FFT datasets as inputs).

```
ROC AUC Scores:
Class 0: 0.616
Class 1: 0.618
Class 2: 0.605
Class 3: 0.616
Class 4: 0.597
```

Figure 35. Gradient boosting ROC AUC scores on HRV variables
Mean ROC AUC score = 0.6104

The outcomes of PCA transformations with component numbers between 12 and 6 were similar for different PCA components, so I will only provide one example.

```
ROC AUC Scores:
Class 0: 0.607
Class 1: 0.614
Class 2: 0.606
Class 3: 0.613
Class 4: 0.591
```

Figure 36. Gradient boosting ROC AUC scores on HRV variables with PCA component = 10
Mean ROC AUC score = 0.6062

Applying FFT on average 12-lead data to naive bayes and random forest classifier yielded the following performance:

```
ROC AUC Scores:
Class 0.0: 0.632
Class 1.0: 0.623
Class 2.0: 0.639
Class 3.0: 0.632
Class 4.0: 0.627
```

Figure 37. Naive Bayes ROC AUC scores on FFT average 12-lead data
Mean ROC AUC score = 0.6306

```
ROC AUC Scores:
Class 0.0: 0.687
Class 1.0: 0.695
Class 2.0: 0.701
Class 3.0: 0.697
Class 4.0: 0.696
```

Figure 38. Random forest ROC AUC scores on FFT average 12-lead data
Mean ROC AUC score = 0.6952

Using FFT data and HRV variables as inputs on a random forest classifier generated the following performance:

```
ROC AUC Scores:
Class 0: 0.588
Class 1: 0.583
Class 2: 0.578
Class 3: 0.584
Class 4: 0.570
```

Figure 39. Random forest ROC AUC scores on FFT data + HRV variables
Mean ROC AUC score = 0.5806

Using oversamplers from smote_variants python module on average of the original 12-lead dataset, the best performance has been reached:

```
ROC AUC Scores:
Class 0.0: 0.899
Class 1.0: 0.954
Class 2.0: 0.969
Class 3.0: 0.989
Class 4.0: 0.988
```

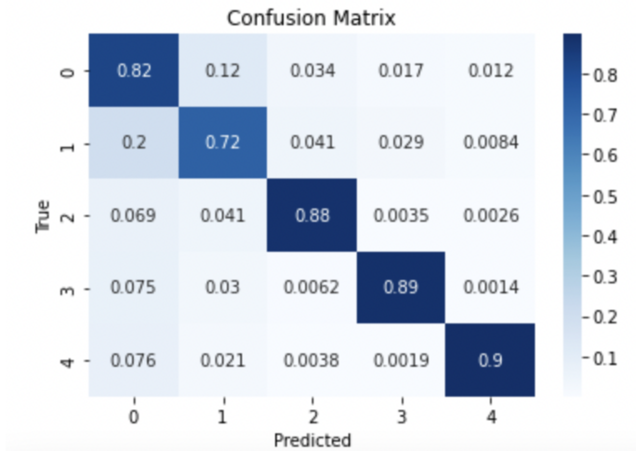


Figure 40. Random forest ROC AUC scores and confusion matrix on oversampled data

It was also attempted to use oversampled FFT datasets as inputs for random forest classifiers, but the outcomes were slightly inferior.

3.2.2. Deep Learning models

After all these basic machine learning models were tried, deep learning models were applied and tuned. Both the original datasets and FFT transformed datasets were used as inputs for these models, but the original datasets performed better within the same models.

```
Epoch [16/50], Loss: 0.7502, Train Accuracy: 0.7245, Test Accuracy: 0.7373
 - AUC-ROC for class 0: Train 0.8602, Test 0.8649
 - AUC-ROC for class 1: Train 0.8114, Test 0.7945
 - AUC-ROC for class 2: Train 0.7361, Test 0.7688
 - AUC-ROC for class 3: Train 0.7137, Test 0.7541
 - AUC-ROC for class 4: Train 0.6902, Test 0.6631
```

Figure 41. CNN accuracy and AUC-ROC scores for each class

The best test accuracy for a CNN model is 0.7373, with mean AUC-ROC scores of 0.7691.

```
Epoch [11/50], Loss: 0.6389, Train Accuracy: 0.7669, Test Accuracy: 0.7683
 - AUC-ROC for class 0: Train 0.8856, Test 0.8882
 - AUC-ROC for class 1: Train 0.8530, Test 0.8517
 - AUC-ROC for class 2: Train 0.7670, Test 0.8045
 - AUC-ROC for class 3: Train 0.7565, Test 0.7556
 - AUC-ROC for class 4: Train 0.6945, Test 0.7343
```

Figure 42. Inception accuracy and AUC-ROC scores for each class

The best test accuracy for an Inception model is 0.7683, with mean AUC-ROC scores of 0.8069.

```
Epoch [15/50], Loss: 0.4903, Train Accuracy: 0.8142, Test Accuracy: 0.7511
 - AUC-ROC for class 0: Train 0.9050, Test 0.8799
 - AUC-ROC for class 1: Train 0.8871, Test 0.8316
 - AUC-ROC for class 2: Train 0.8135, Test 0.7899
 - AUC-ROC for class 3: Train 0.8211, Test 0.7712
 - AUC-ROC for class 4: Train 0.7677, Test 0.7159
```

Figure 43. Resnet accuracy and AUC-ROC scores for each class

The best test accuracy for a Resnet model is 0.7511, with mean AUC-ROC scores of 0.7977.

3.2.3. Discussion

Even though HRV variables generated by heartpy are more human-comprehensible, they result in inadequate model performance. The FFT-transformed dataset, which incorporates the dataset's frequency information, produces slightly better outcomes than the HRV variables alone. However, combining the FFT information with the HRV variables as inputs yielded poorer results, which may have been due to the disparity and scaling between the FFT dataset and the HRV variables. Among the tests performed, random forest classifiers with oversampled original data yielded the best results, with about 90% ROC AUC scores for all categories. The confusion matrix shows that the MI category has the lowest true positive rate (72%), which might be caused by MI occurring in areas of the heart that are not well represented by ECG data[26].

Deep learning models performed better than all non-NN machine learning models except the random forest classifier that used the oversampled original dataset as inputs. Notice that for all models, class 0 (the class that contains normal ECG data) has the highest accuracy, and class 4 (the class that contains HYP data) has the lowest accuracy. There are several physiological reasons behind this. Hypertrophy can be difficult to detect among other cardiac diseases because it often has no symptoms in the early stages[27]. Thus, records labeled as HYP might not be significantly different from others. In addition, hypertrophy can be caused by a variety of factors and can present in different ways depending on the location of the thickened heart muscle. Thus, patients might have different symptoms and different test results.

Out of the papers that cited the PTB-XL ECG dataset, two research papers could be used as benchmarks: “Bimodal CNN for cardiovascular disease classification by co-training ECG grayscale images and scalograms”[28] and “Estimating critical values from electrocardiogram using a deep ordinal convolutional neural network”[29]. The first paper transformed the original 1D ECG data into two-dimensional grayscale images and scalograms that were simultaneously supplied as dual input images to the bimodal CNN model. The bimodal CNN model used contains Inception-V3, which was pre-trained on the ImageNet database and reached a final accuracy of 95.74% on all leads. The second paper also modified the original dataset. Instead of the labels provided by the original PTB-XL dataset, the second paper mapped the diagnostic conclusions to critical values, which served as a threshold for determining the severity of health-related conditions. After that, a 61-layer deep convolutional neural network named CardioV was built and trained, eventually reaching a mean ROC-AUC score of 0.8735. Due to the fact that the datasets used in both publications were slightly modified variants of the original dataset, the provided performance scores are merely for reference.

4. Conclusions

PPG and ECG are two extensively used methods for monitoring cardiovascular activity, and their implications in the real world are spreading. They have a wide range of applications in healthcare, fitness monitoring, sleep monitoring, and biometric authentication. With the increasing availability of wearable devices and the development of advanced algorithms for data analysis, these technologies have significant potential to improve health outcomes and enhance daily life.

This research applies EDA to PPG data and ECG data and evaluates the ability of machine learning models to recognize and differentiate human activities using PPG data and to diagnose cardiac conditions using ECG data. The results of our models indicate that for both the PPG and ECG datasets, the normal or rest class has the highest true positive rate, while the other categories perform worse. The frequent misclassification of squat and step categories in the PPG dataset may be due to the small size and short duration of the recordings. The classification of hypertrophy is mildly hindering performance of the models, which may be due to the absence of symptoms of hypertrophy in the early stages. It is worth noting that some of the HRV variables are highly correlated with each other, and this should be taken into account when developing machine learning models for HRV analysis. Another PPG dataset that has longer durations should be examined, as it could provide additional insights and improve the predictive performance. It is possible to construct deeper deep learning models for the ECG dataset, which may lead to improved accuracy while avoiding overfitting. Transfer learning should also be considered by pretraining models on higher-quality ECG data before applying them to PPG data. In addition, different compression methods besides FFT can be considered, such as Discrete Wavelet Transform and Discrete Cosine Transform.

5. References

1. Castaneda, D., Esparza, A., Ghamari, M., Soltanpur, C., & Nazeran, H. (2018). A review on wearable photoplethysmography sensors and their potential future applications in health care. *International Journal of Biosensors & Bioelectronics*, *4*(4), 195–202. https://doi.org/10.15406/ijbsbe.2018.04.00125
2. Psathas, A.P., Papaleonidas, A., Iliadis, L. (2020). Machine Learning Modeling of Human Activity Using PPG Signals. In: Nguyen, N.T., Hoang, B.H., Huynh, C.P., Hwang, D., Trawiński, B., Vossen, G. (eds) Computational Collective Intelligence. ICCCI 2020. Lecture Notes in Computer Science(), vol 12496. Springer, Cham. https://doi.org/10.1007/978-3-030-63007-2_42
3. Hnoohom, N., Mekruksavanich, S., & Jitpattanakul, A. (2023). Physical Activity Recognition Based on Deep Learning Using Photoplethysmography and Wearable Inertial Sensors. *Electronics*, *12*(3), Article 3. https://doi.org/10.3390/electronics12030693
4. Rath, A., Mishra, D., Panda, G., & Satapathy, S. C. (2021). Heart disease detection using deep learning methods from imbalanced ECG samples. *Biomedical Signal Processing and Control*, *68*, 102820. https://doi.org/10.1016/j.bspc.2021.102820
5. Zhang, W., Yu, L., Ye, L., Zhuang, W., & Ma, F. (2018). ECG Signal Classification with Deep Learning for Heart Disease Identification. *2018 International Conference on Big Data and Artificial Intelligence (BDAI)*, 47–51. https://doi.org/10.1109/BDAI.2018.8546681
6. Biagetti, G., Crippa, P., Falaschetti, L., Saraceni, L., Tiranti, A., & Turchetti, C. (2020). Dataset from PPG wireless sensor for activity monitoring. *Data in Brief*, *29*, 105044. https://doi.org/10.1016/j.dib.2019.105044
7. Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., & Schaeffter, T. (2020). PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, *7*(1), 154. https://doi.org/10.1038/s41597-020-0495-6
8. Ojha, N., & Dhamoon, A. S. (2023). Myocardial Infarction. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK537076/
9. Kashou, A. H., Basit, H., & Malik, A. (2023). ST Segment. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK459364/
10. Tirado-Martin, P., Liu-Jimenez, J., Sanchez-Casanova, J., & Sanchez-Reillo, R. (2020). QRS Differentiation to Improve ECG Biometrics under Different Physical Scenarios Using Multilayer Perceptron. *Applied Sciences*, *10*(19), Article 19. https://doi.org/10.3390/app10196896
11. *Arrhythmias—Conduction Disorders | NHLBI, NIH*. (2022, March 24). https://www.nhlbi.nih.gov/health/conduction-disorders
12. Saini, S., & Gupta, Dr. R. (2022). Artificial intelligence methods for analysis of

electrocardiogram signals for cardiac abnormalities: State-of-the-art and future challenges. *Artificial Intelligence Review*, *55*, 1–47. https://doi.org/10.1007/s10462-021-09999-7

13. Bagha, S., & Shaw, L. (2011). A Real Time Analysis of PPG Signal for Measurement of SpO2 and Pulse Rate. *International JournalOf Computer Application*.
14. Park, J., Seok, H. S., Kim, S.-S., & Shin, H. (2022). Photoplethysmogram Analysis and Applications: An Integrative Review. *Frontiers in Physiology*, *12*. https://www.frontiersin.org/articles/10.3389/fphys.2021.808451
15. Butterworth filter. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Butterworth_filter&oldid=1152211897#Transfer_function
16. Selvaraj, N., Jaryal, A., Santhosh, J., Deepak, K. K., & Anand, S. (2008). Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of Medical Engineering & Technology*, *32*(6), 479–484. https://doi.org/10.1080/03091900701781317
17. *Figure 1. A diagrammatic representation of the interbeat interval in an...* (n.d.). ResearchGate. Retrieved April 25, 2023, from https://www.researchgate.net/figure/A-diagrammatic-representation-of-the-interbeat-interval-in-an-electrocardiogram-signal_fig1_257748654
18. Shaffer, F., & Ginsberg, J. P. (2017). An Overview of Heart Rate Variability Metrics and Norms. *Frontiers in Public Health*, *5*, 258. https://doi.org/10.3389/fpubh.2017.00258
19. *Difference between RR interval and NN interval*. (2020, December 17). Hexoskin Support Community. https://hexoskin.zendesk.com/hc/en-us/articles/360045123314-Difference-between-RR-interval-and-NN-interval
20. Jeongwhan Lee, Keesam Jeong, Jiyoung Yoon, & Myoungho Lee. (1997). A simple real-time QRS detection algorithm. *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, *4*, 1396–1398. https://doi.org/10.1109/IEMBS.1996.647473
21. Golińska, A. K. (2013). Poincaré Plots in Analysis of Selected Biomedical Signals. *Studies in Logic, Grammar and Rhetoric*, *35*(1), 117–127. https://doi.org/10.2478/slgr-2013-0031
22. Kumar M., A., & Chakrapani, A. (2022). Classification of ECG signal using FFT based improved Alexnet classifier. *PLoS ONE*, *17*(9), e0274225. https://doi.org/10.1371/journal.pone.0274225
23. Zhang, Z., & Takane, Y. (2010). Multidimensional Scaling. In P. Peterson, E. Baker, & B. McGaw (Eds.), *International Encyclopedia of Education (Third Edition)* (pp. 304–311). Elsevier. https://doi.org/10.1016/B978-0-08-044894-7.01348-8
24. de la Calleja, J., & Fuentes, O. (2007). *A Distance-Based Over-Sampling Method for Learning from Imbalanced Data Sets.* 634–635.

25. Tourassi, G. (2018). Receiver Operating Characteristic Analysis: Basic Concepts and Practical Applications. In E. Samei & E. A. Krupinski (Eds.), *The Handbook of Medical Image Perception and Techniques* (2nd ed., pp. 227–244). Cambridge University Press. https://doi.org/10.1017/9781108163781.015
26. Izumi, C., Iga, K., Kijima, T., Himura, Y., Gen, H., & Konishi, T. (1995). Limitations of electrocardiography in the diagnosis of acute myocardial infarction—Comparison with two-dimensional echocardiography. *Internal Medicine (Tokyo, Japan)*, *34*(11), 1061–1063. https://doi.org/10.2169/internalmedicine.34.1061
27. *Hypertrophic cardiomyopathy—Symptoms and causes*. (n.d.). Mayo Clinic. Retrieved April 26, 2023, from https://www.mayoclinic.org/diseases-conditions/hypertrophic-cardiomyopathy/symptoms-causes/syc-20350198
28. Yoon, T., & Kang, D. (2023). Bimodal CNN for cardiovascular disease classification by co-training ECG grayscale images and scalograms. *Scientific Reports*, *13*(1), Article 1. https://doi.org/10.1038/s41598-023-30208-8
29. Wei, G., Di, X., Zhang, W., Geng, S., Zhang, D., Wang, K., Fu, Z., & Hong, S. (2022). Estimating critical values from electrocardiogram using a deep ordinal convolutional neural network. *BMC Medical Informatics and Decision Making*, *22*(1), 295. https://doi.org/10.1186/s12911-022-02035-w