Stress Detection from Photoplethysmogram

Mannika Garg, Guide: Dr.Sujay Deb
Indraprastha Institute of Information Technology, New Delhi
Independent Project

***Abstract*— With the advent of wearable technology, healthcare has been blessed with non-invasive methods to obtain biosignals. These signals along with machine learning techniques can be used for fast and low-cost computational diagnosis of various illness.**

In today's world, one of the major cause of physical and mental problems is stress. There is evidence of it being linked with cardiovascular diseases, diabetes and asthma. Chronic stress can also lead to anxiety and depression. Thus, reliable, automatic detection of stress can help in prognosis as well as the prevention of various diseases. However, the existing methods use multi-sensor data, for example, Electrocardiography(ECG), Electromyogram (EMG), Photoplethysmogram(PPG), body temperature, blood pressure, and electroencephalogram (EEG) for the detection of stress. For ECG alone, 12 or 15 electrodes are fitted to the chest, arms, hands, and legs[1].

In this paper, I aim to come up with a technique to reliably detect stress using only Photoplethysmogram (PPG) signals to avoid the constrain of a large number of sensors. The sensor obtained PPG signals have been preprocessed to remove noise.The data has been labelled using K-means clustering. Using the features extracted from the signal, different machine learning models have been compared for best accuracy.

I. Material and Methods

A. Dataset

The database was obtained from Physionet - Wrist PPG During Exercise[2]. It consists of body measurements conducted on 8 young participants 3 male, 5 female), aged 22–32 (mean 26.5) performing walking, running, low resistance bike riding and high resistance bike riding. Each activity is performed for a duration of 10 mins.
The dataset consists of signals of PPG, ECG, accelerometers and gyroscopes, each sampled at 256 samples/sec. The Physionet tool WFDB was used to read the data. The header names were manually cleaned. Only PPG signal were used for further analysis.

B. Pre-processing

The biosignals obtained from low-cost sensors are typically noisy due to motion artifacts and sensor distrosion. Preprocessing these signals is very essential in order to obtain reliable feature extraction. I have done 3 steps preprocessing.

1) Removal of undesired values: The first step involved removal of infinite (inf,-inf) values. Then all the missing data points were replaced with the mean signal value.

2) Detrending: The PPG signal in our dataset has baseline wander i.e. the mean of the signal is shifting over time. In order to remove such trends I have have used signal detrending.

3) Bandpass filtering: The above-obtained signal is passed through Butterworth bandpass filtering with a pass and stop frequency of 0.8Hz and 3 Hz respectively. This was done to remove the unwanted frequencies with less than 48 bpm and greater than 180 bpm.

C. Feature Extraction

After preprocessing, a window of size 4 seconds with an overlap of 0.1% is slidded over signal. The R-R peaks were found and 12 features were extracted for every 4 second window. Overlap was done to ensure a smooth transition in the features. Feature extraction has been done using the python library HeartPy. The final dataset consist of 1873 datapoints and 12 features. The features and their description are :-

- BPM : Beats per minute

	bpm	ibi	sdnn	rmssd	pnn20	pnn50	hr_mad	sd1	sd2	s	sd1/sd2
count	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000	1873.000000
mean	104.792459	577.489324	88.229201	138.507782	0.834945	0.697198	59.564577	82.651134	58.668025	21609.443733	2.099841
std	21.451990	124.585357	52.747951	87.417926	0.278888	0.324700	46.360029	60.858309	53.825928	26640.022492	4.804172
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	92.978208	502.790179	45.204337	71.424938	0.750000	0.500000	25.390625	36.239695	17.082867	2028.585379	0.751376
50%	106.666667	559.895833	82.633577	128.906250	1.000000	0.750000	46.875000	75.555398	41.075002	9670.438782	1.373168
75%	118.675497	640.625000	128.154373	194.186206	1.000000	1.000000	82.031250	124.591618	90.867133	34151.944915	2.214630
max	301.176471	1128.906250	287.109375	492.187500	1.000000	1.000000	287.109375	324.550964	288.167030	143831.751859	139.000000

Fig. 1. Description of extracted features

- IBI : Interbeat interval
- SDNN : Standard deviation if intervals between adjacent beats
- RMSSD : Root mean square of successive differences between adjacend R-R intervals
- PNN20 : Proportion of differences between R-R intervals greater than 20ms
- PNN50 : Proportion of differences between R-R intervals greater than 50ms
- MAD : Median absolute deviation
- SD1 : Standard deviation of Poincaré plot perpendicular to the line-of-identity
- SD2 : Standard deviation of the Poincaré plot along the line-of-identity
- S : Area of the Poincaré eclipse
- SD1/SD2 : Ratio of SD1 and SD2

D. Stress Labels

The database has been classified based on the extracted features into two classes reflecting stressed (labelled as 1) and normal (labelled as 0) conditions. For the classification, K-means clustering with 600 iterations has been used. The algorithm was tested on all the 12 features to select the one with the best results. Figure 3 compares the mean value of each of the features obtained during stress and normal conditions. It is visible that the value of PNN50 shows the highest variation between the 2 classes, thus proving to be a good measure to detect stress. It's value is high(0.88 for our data) during normal conditions and low (0.26 for our data) under stress[5]. The results have been theoretically verified from previously done researches in the same area[4].

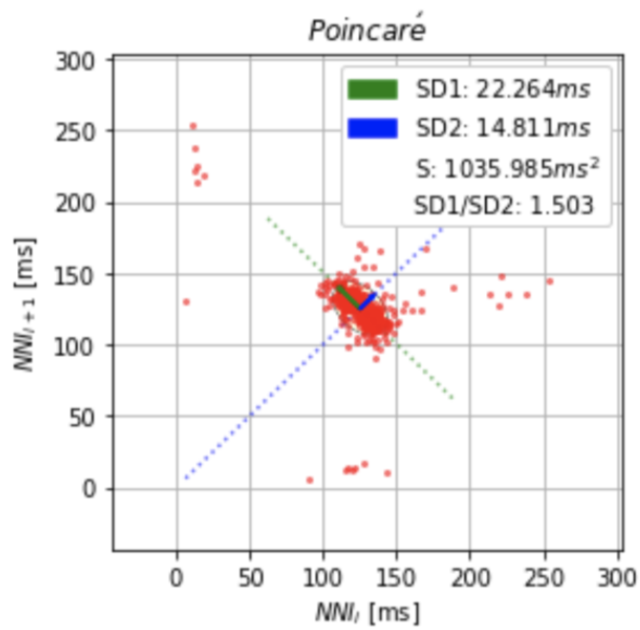


Fig. 2. Poincaré plot of the signal obtained from participant 3 while running.The Poincaré plot is a scattergram, which is constructed by plotting each RR interval against the previous one[3]. The area of the elipse and standard deviation along and perpendicular to the line of identity are features of time domain analysis of PPG signal.

II. Training of data

I have split the dataset in the ratio 2:8 for testing and training, respectively. I have used 5-fold cross-validation to fine-tune the best hyper parameters of the proposed models.

A. Feature Reduction

For cleaning our data set, correlation between the feature vectors was analysed. I have set the threshold as 0.95. As shown in Figure 4, the high-

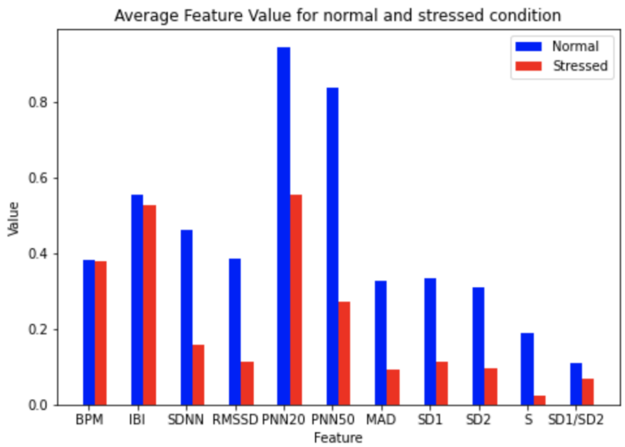


Fig. 3. Comparision of features for stressed and normal condition

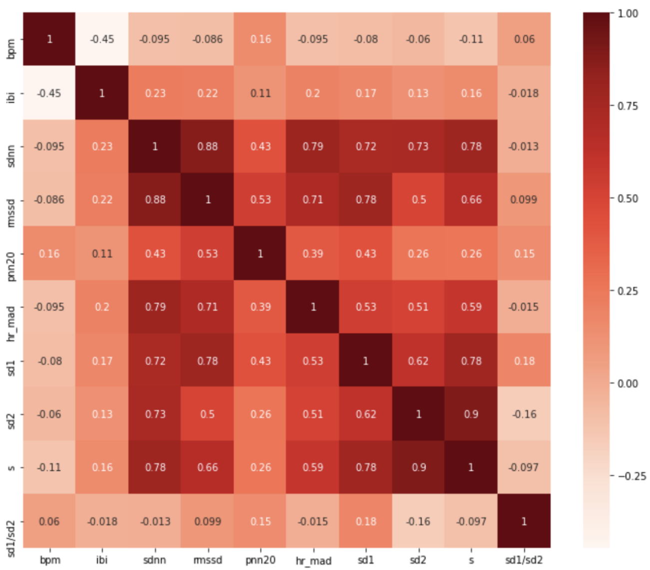


Fig. 4. Correlation Matrix

est correlation was found in SDNN and RMSSD of 88%. Thus, none of the features were removed. Further the features were checked for constant and quasi constant values. I have also employed PCA and LDA to compare results derived from different models.

B. Models

I have compared eight models - Stochastic Gradient Descent, Support Vector Classifier, Linear SVC, Gradient Boosted Decision Tree, Random Forrest , K-Nearest Neighbours , Gaussian Naive Bayes and X Gradient Boosting. Later, the Auto ML library TPOT was also tested. The top performing models have been described below.

1) Random Forest Classifier: I have used the ”RandomForestClassifier” and ”VotingClassifier” implementation from the Scikit-learn library. In my model, the voting classifier consists of 3 Randomforest classifiers. I have used hard voting criteria. The number of trees in the forest (n_estimators) is 50,80 and 100, respectively. I have kept oob_score as True, and all the other parameters are set as default.

2) Gradient boosted decision tree: In this model, I have used a voting estimator (with hard voting) consisting of 3 Gradient boosting classifiers, with learning rate as 0.1, from the Scikit-learn library. The number of boosting stages (n_estimator) used is 20 ,25 and 20, respectively. max_depth is 10, the minimum number of samples required to split the internal node is 2, the minimum number of samples needed to be a leaf is three, and the subsample is tuned as 0.5 for all the three classifiers.

3) X Gradient Boosting: In this model, I have used learning rate 0.1 and n_estimators as 100.

4) Tree-Based Pipeline Optimization Tool (TPOT): TPOT is python automated machine learning. It compares several machine learning pipelines to provide the best results.The number of generations and population size are set to 10 and 20 respectively. The tool predicted model consisting of a pipeline of GaussianNB, LinearSVC, XGBClassifier to give optimal results.

III. Results

The best accuracy was obtained through Tree-Based Pipeline Optimization Tool followed by Gradient boosted decision trees, Random Forest and X Gradient Boosting . Figure 5 shows the accuracy obtained from different models.

Accuracy is the most intuitive metric to identify the best model, but it is suitable for a uniform dataset with the same number of positives and negatives. In my training dataset, the number of stressed and normal labels are not equal, and

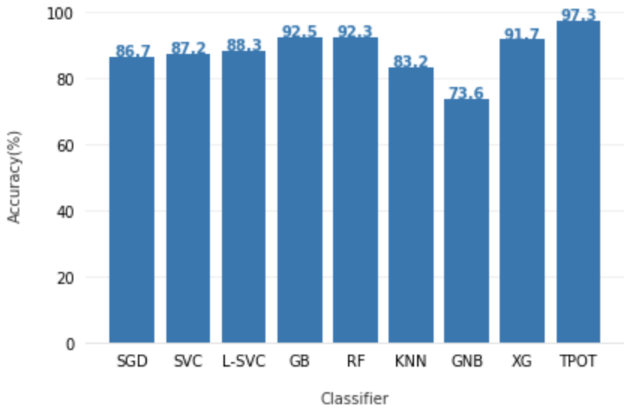


Fig. 5. Accuracy Comparision

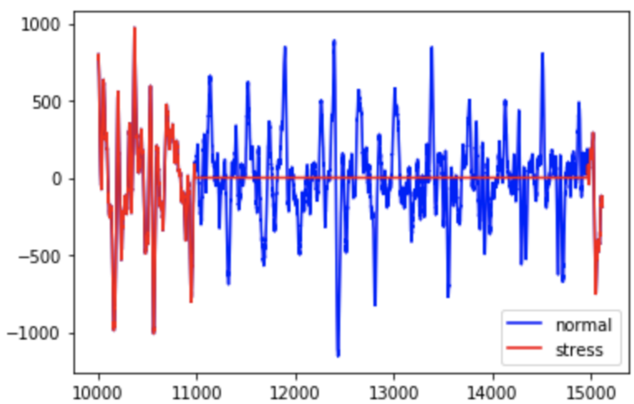


Fig. 6. Predicted labelled Signal

thereby, I will use the F1 score matrix. From table

N or S	RF	GB	Xboost	TPOT
Normal	0.940	0.944	0.938	0.979
Stress	0.879	0.887	0.874	0.961

TABLE I: F1 Score

I, it is evident that Tree-Based Pipeline Optimization Tool is best for detection of stress because the F1 score for Normal and Stressed conditions is obtained the highest using TPOT. Figure 6 shows a 20 second window of a PPG with predicted labels. Stress is marked with 0 value where the person is calm.

IV. GUI

Two Graphic User Interface have been proposed. Both the GUI are in early stage of development. A

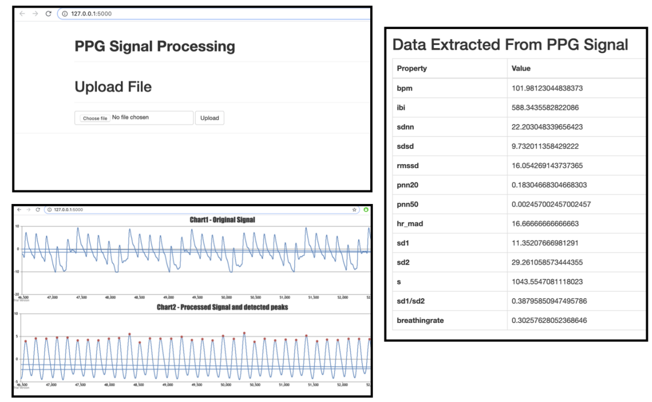
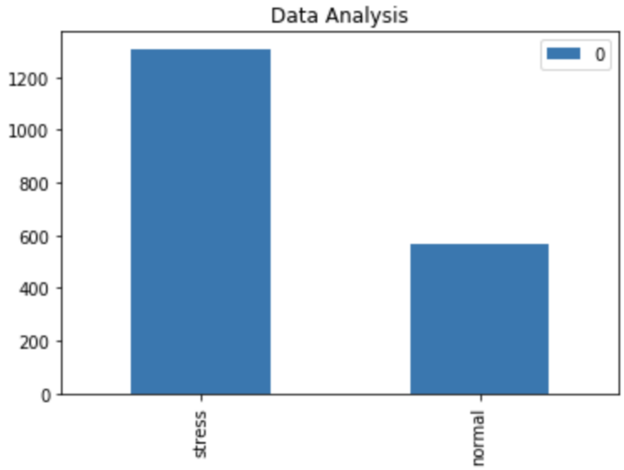


Fig. 7. Web App

web app has been constructed using flask, which allows user to input the PPG data in the form of .csv file and provides a visualization of the input, processed and peak detected signal.The app comes with a zoom in facility, where the user can select and drag the mouse over the portion to be zoomed. The app also reports the average extracted features for the provided signal. The other gui is developed in python which would allow the user to visualise the data in realtime. Here we have assumed that the PPG data coming from the sensor is getting updated to a .csv file in realtime. Both the GUI's can further be integrated with the proposed machine learning model to detect stress.

V. Discussion

In the implementation, I have first labelled the data. PNN50 was found to be a good measure of stress. Then I have compared eight classification models, i.e., Stochastic Gradient Descent, Support Vector Classifier, Linear SVC, Gradient Boosted Decision Tree, Random Forrest , K-Nearest Neighbours, Gaussian Naive Bayes and X Gradient Boosting using Accuracy. Then the top four models that gave the highest accuracy were compared based on F1 score. I have used 5-fold cross-validation to select the best hyperparameters for the following model. The utilization of PCA on the feature vector has no significant effect on the accuracy.
There is no online PPG data available with the stress labels. I believe that self generated dataset



with known labels could help to improve the accuracy further. The study is a prototype and can be implemented worldwide across the globe. The difference in F1 score and Accuracy is the result of non-uniformity in the dataset i.e. high disparity in number of normal and stress labels (Fig.8). It is, therefore, essential to have a better representation of the data for better results.

In this paper, we have established that PPG signals can be reliably used for stress detection. The TPOT Auto ML library obtained the best results using a pipeline of Gaussian Naive Base,Linear SVC and XGBoost. It should be noted that the prediction can be made more robust by including dataset of people across ages.

References

[1] https://www.sciencedirect.com/topics/engineering/electrocardiography

[2] 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.

[3] https://www.researchgate.net/publication/290416554_NONLINEAR_DYNAMICS_METHODS_FOR_ASSESSING_HEART_RATE_VARIABILITY_IN_PATIENTS_WITH_RECENT_MYOCARDIAL_INFARCTION

[4] https://www.researchgate.net/publication/319158240 HRV_based_Stress_Level_Assessment_Using_Very_Short_Recordings settings

[5] ”Stress and Heart Rate Variability during University Final Examination among Lebanese Students” https://www.mdpi.com/2076-328X/9/1/3/htm