Detection of Physiological Stress using Photoplethysmography (PPG) Signals

Vishnu Kannan
vishnukanan@gmail.com

Ketki Ambekar
ambekar.ketki@gmail.com

Pradeep Ramadasan
pradeep.ramadasan@gmail.com

Abstract

Physiological Stress is one of the major causes of diseases in modern society. Empowering humans to know when they are stressed is crucial to help identify stressors in their life and come up with strategies to manage them. The goal of this paper is to improve our ability to detect stress using commonly available Blood Volume Pulse (BVP) signals from photoplethysmogram (PPG) sensors which are commonly found in most personal health wearable devices. BVP signals are often noisy depending on where the sensors are placed on the body, wrist being the most common location. Prior methods to detect stress involved complex signal processing and often needed two or more modalities including signals from accelerometers, electro dermal sensors, etc. In this paper, we propose detecting stress by employing just BVP signals with minimal preprocessing and a multi-layer convolutional neural network. We used a robust publicly available dataset, Wearable Stress and Affect Detection Dataset (WESAD) to train and test the efficacy of our model, which attained a AUC-ROC score of 0.70 on data from a test subject not previously encountered by the model. Our evaluation of several other non-neural models and autoencoder techniques are presented. Some of the challenges we faced while employing multi-layer convolutional networks and the strategies we tried are presented as well, paving way for future exploration.

1 Introduction

Psychological stress occurs when an individual perceives that environmental demands tax or exceed his or her adaptive capacity [9]. Stress is intuitively known to lead to various illness, both physical and emotional, but there is yet to be strong data to prove causal relationships [8]. Stress is known to impact functioning of prefrontal cortex [4] which can lead to sub-optimal decision making.

Cortisol levels detected typically via saliva samples are the canonical way for measuring stress [15]. However measuring cortisol in a laboratory setting isn't widely available and isn't possible on a continual basis. It is known that the heart function is related to mental stress which influences the sympathetic and para-sympathetic nervous systems [24].

Normally, an Electrocardiogram (ECG) is relied upon in a medical setting to obtain accurate heart readings. An ECG records electrical activity of the heart. Change of amplitude of this signal with respect to time, gives us insights into normality or abnormality of heart rhythms [6] However, ECG devices are not portable.

Photoplethysmogram (PPG) devices are portable, and are present in wrist-worn devices like smart watches, fitbits, etc. These wearables measure Blood volume pulse, which is a side effect of capillary dilation and constriction, in addition to some other indicators such as Electro-Dermal Activity (EDA), Accelerometer reading (ACC), etc. Their BVP signals however are distorted by Motion Artifacts (MA). To combat this several signal processing approaches have been proposed [14, 7] including employing deep neural networks that take inputs from an accelerometer to aid in cancelling noise [19] Figure 1 shows typical stressed and non-stressed examples of the BVP signal.

Even though there are several modalities to detect stress, recent research has centered around using bio-markers from wearables to detect stress [11]. This has led to several publicly available datasets that attempt to detect bio-markers while placing subjects in stressful situations [10, 22, 19, 17]. These datasets are useful in identifying rela-

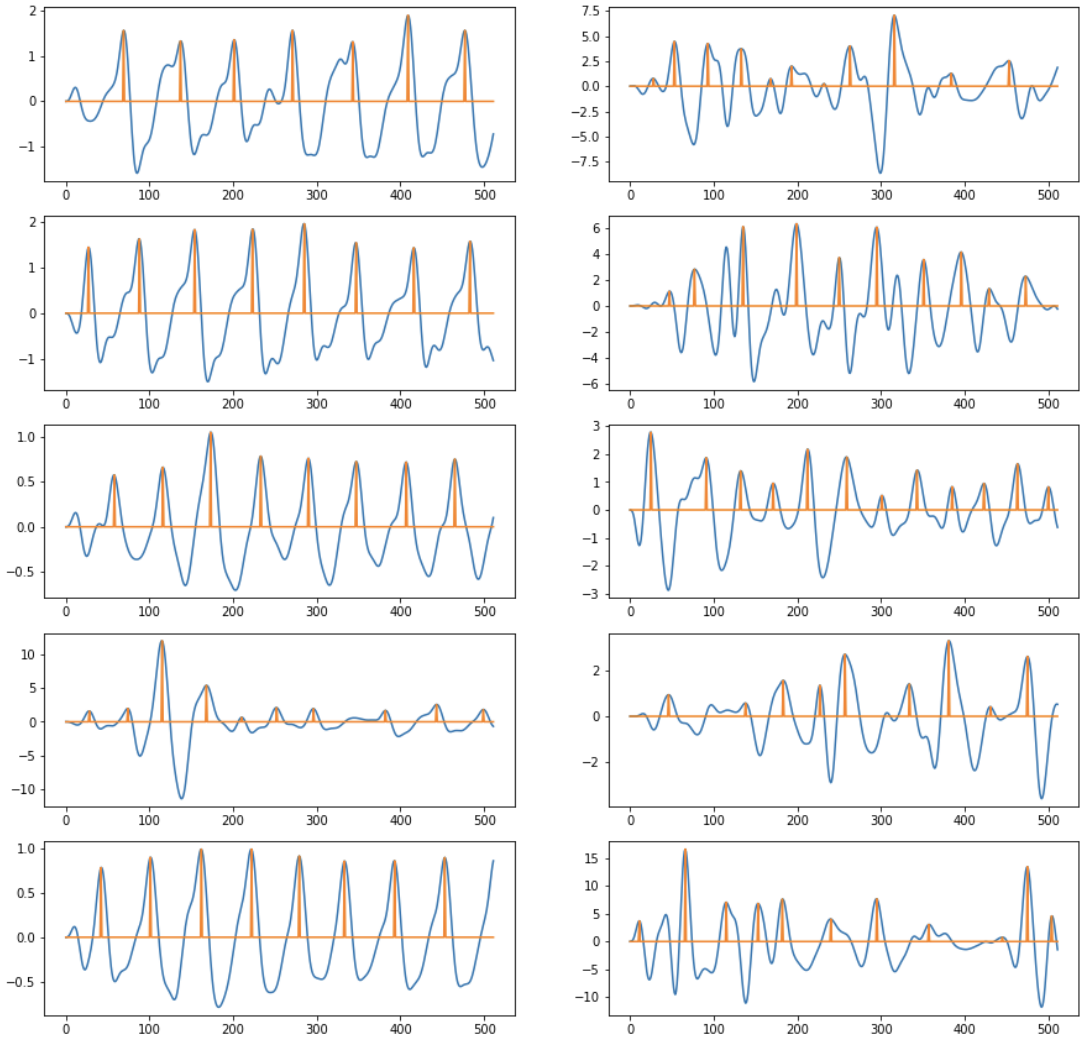


Figure 1: Left Column: Non-stressed BVP Samples with peaks identified, Right Column: Stressed BVP Samples with peaks identified

tionship between heart function and psychological stress.

In this paper, we attempt to model this relationship between heart function and mental stress using a publicly available dataset [22]. By limiting the modeling to just PPG signals, we intend for the model to be applicable in non wearable scenarios like remote PPG extraction via video signals from cameras [20] which can help identify speech patterns that are indicative of stress, thereby opening up more modalities for detecting stress.

In this paper we specifically look at identifying stressful moments for humans in their daily life based on Blood Volume Pulse (BVP) signals obtained via Photoplethysmography (PPG) sensors which have been a critical part of personal health wearable devices. These devices, typically worn in the wrist, generate multiple biomarkers like heart rate, heart rate variability, skin temperature, etc. Unlike ECG signals, which are considered the gold standard in clinical setting, signals from PPG sensors are inherently noisy due the influence of physical movement on the sensor's output [5].

In the rest of the paper, we first present past research on detecting stress. We then present the datasets we employed to model heart function in the presence of psychological stress. We then go over our modeling objectives and present the various Machine Learning techniques we employed to model heart function under stress. We present metrics that capture the efficacy of these techniques using the public datasets. Finally, we present some potential future work beyond what is presented in this paper.

2 Hypothesis

Our goal in this paper is to detect presence of stress from BVP signals with high precision, a binary classification problem. We prefer higher precision over recall to avoid potentially introducing additional stress by mis-classifying non-stressed state. The loss function we are attempting to optimize is as follows:

$$L = \sum_{i=1}^{N} [y_n * log(x_n) + (1 - y_n) * log(1 - x_n)],$$

where N is number of samples, y_n is binary target labels (stressed '1' vs non-stressed '0'), and x_n are features derived from the input BVP signal.

3 Understanding the Data

3.1 Data Acquisition and Preparation

3.1.1 WESAD

This is a multimodal dataset for wearable stress and affect detection. The dataset was recorded with a wrist-worn device (including the following sensors: PPG, accelerometer, electrodermal activity, and body temperature) and a chest-worn device (including the following sensors: ECG, accelerometer, EMG, respiration, and body temperature). 15 subjects (aged 24–35 years) participated in the data collection, each for approximately 100 min. The dataset was recorded with the goal to detect and distinguish different affective states (neutral, stress, amusement). Each subject was evaluated under three different scenarios, a baseline condition where they were sitting/standing at a table reading provided magazines, followed by an Amusement condition where subjects watched a set of eleven funny clips, and finally a stress condition where subjects were exposed to a well-studied Trier Social Stress Test (TSST) [15]. More details available in [22]. All subjects wore the Empatica E4 on their non-dominant hand. The E4 records BVP at 64 Hz. The datapoints were labelled as an integer between 0-7. Each label stands as thus: 0 = not defined / transient, 1 = baseline, 2 = stress, 3 = amusement,4 = meditation, 5/6/7 = ignored in this dataset. Labels provided match the task where the 'stress' condition has a label of '2' and 'amusement' condition has a label of '3' and the first 'baseline' task has a label of '1'. The rest of the labels were to be ignored. Labels (affective states) were provided to match the frequency of ECG signals (700 Hz). For the purposes of this paper, the labels were down sampled to match the frequency of PPG signal(64 Hz). We labelled data belonging to group 2 as stressed (1) and data belonging to other groups were labelled as non-stressed (0).

In the binary case (stress vs. non-stress), performance of a benchmark classifier was published with accuracy of 85.83% and F1-score of 83.08% when just BVP signals were employed. Other relevant metrics like AUC-ROC, log likelihood, etc. were not published.

For this paper, the Data was scaled using the RobustScalar available as part of the scikit-learn package.

BVP Signal were recorded for approximately 1.6 hours for each subject over all activities. Given sampling rate of 64Hz, giving us on an average 370,602 samples per subject. Data values were distributed as 0 ± 4185.054413 on average. We further processed this BVP data, by creating windows of 8 seconds with a 2 second stride. resulting in a 512 ($8seconds \times 64Hz$) features per sample window. A 8 second windowing scheme was recommended based on past research [10, 21]. After windowing, on average 2900 data points was obtained for each of the 15 subjects. Our final count of samples for WESAD was 43,385 of which 37,207 are of label 0 (non-stressed) and 4981 (13%) are of label 1 (stressed).

3.1.2 PPG-DaLiA

In an attempt to develop general purpose representations of PPG data, the PPG-DaLiA dataset was also evaluated. This dataset is multi-modal, but does not include stress labels and so it was used only for experimentation with representation learning. Raw sensor data was recorded with two commercially available devices: a chest-worn device (RespiBAN Professional, and a wrist-worn device (Empatica E4). Since this paper focusses only on BVP signals, only data from Empatica E4 is relevant and is detailed here. The E4 device was worn on the subjects' non-dominant wrist. The PPG sensor output is the difference of light between oxygenated and non oxygenated peaks, provided with a sampling frequency of 64 Hz.

The pre-processed data is in the form of a dictionary data structure, where each subject is a separate key and EDA, BVP, ACC signal readings are present for each subject. We extracted the BVP signals for each subject. BVP signals are sampled at 64Hz by the device. We then applied 8 second

windowing (with 2 second strides) as we did in the WESAD dataset.

The dataset has BVP signal recordings of approximately 2.5 hours for each subject, over all activities. Sampled at 64Hz that's approximately 552,465 samples per subject. With windowing, we get approximately 4300 data points per subject (8 second windows, with 2 second strides). Data values were 0 ± 7117.928749 on average.

3.2 Dataset Balancing

Our primary dataset Wesad was imbalanced where only 11% of samples on average per subject belonged to the 'stressed' category. We attempted to employ class weights to influence the loss function presented earlier to give more importance to the 'stressed' class but that did not help our models fit the dataset. Alternatively, we used the SMOTE technique [3] to over-sample the 'stressed' class, which helped identify models that were able to fit the dataset. We used the implementation from the 'imblearn' library for the same. SMOTE generated new synthetic positive samples, such that our training dataset now has 31,832 samples of both classes. Since the PPG-DaLiA dataset was only used for representation learning, no balancing techniques were employed.

4 Methodology

Prior to attempting to test various model classes, we build a naive classifier which would classify all samples as 'non-stressed' which resulted in 89% accuracy. However since accuracy as a metric wasn't representative of our desire to favor higher precision over recall, we discarded this model and instead resorted to using the benchmark metrics published as part of the Wesad dataset as presented earlier.

All data was passed through a butterworth bandpass filter with 4 modes to extract frequencies in the range of [0.7, 3.5] which corresponds to normal human heart rate range of [42, 210].

In addition to employing scaled and filtered time domain signals, we also attempted to extract several features manually before evaluating potential model classes. These include,

a. Features automatically extracted by the 'HeartPy' python library [1], an open source framework for working with PPG and ECG signals,

b. power spectral density with the same windowing scheme as time domain signals,

c. Inter beat Interval based features: peak detection followed by computation of Heart Rate Mean and Standard Deviation, heart rate variability (HRV) mean, standard deviation and Root mean square of the HRV, Number and percent of HRV samples above 50ms, Energy in ultra low (0.01-0.04 Hz], low(0.04-0.15 Hz], high (0.15-0.4 Hz], and ultra high frequency component of the HRV (0.4-1.0 Hz], Ratio of LF and HF component, $\sum$ of the freq. components in ULF-HF

The raw time domain BVP signals and the various derived features were analyzed initially using Principal Component Analysis which showed no clear separation of the classes. In the graphs presented 'green' corresponds to 'non-stressed' state, while 'red' corresponds to 'stressed' state (See figures 2a 2c 2d)

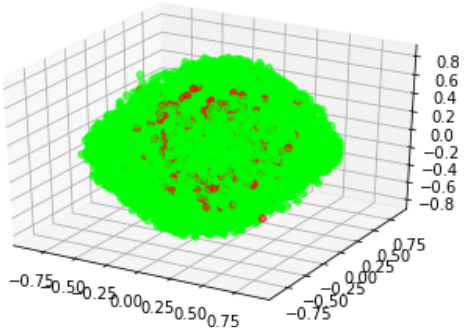
4.1 Cross Validation

Leave one group out cross validation (LGOCV) was employed for all linear models evaluated where data from each subject was picked as the test set only once across several training runs. A slightly modified cross validation methodology was employed for neural models where data from a randomly selected subject was picked as the test set, and amongst the remaining, one randomly picked subject's data served as the validation set while the rest serving as test set. As opposed to LGOCV, the models' ability to generalize to all subjects wasn't analyzed in the case of neural models.

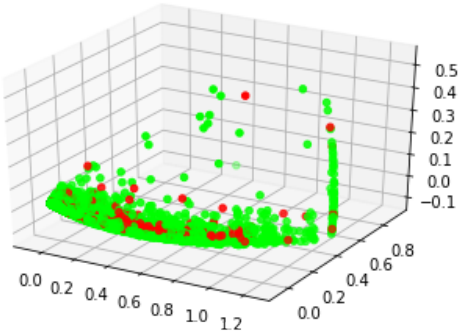
4.2 Linear Models

All features mentioned above were first attempted to be modeled using linear models. Stochastic Gradient Descent (SGD) was employed with "Hinge" loss and "Log" loss functions. SGD was set to run for $1e4$ iterations with a loss tolerance of $1e{-}4$.

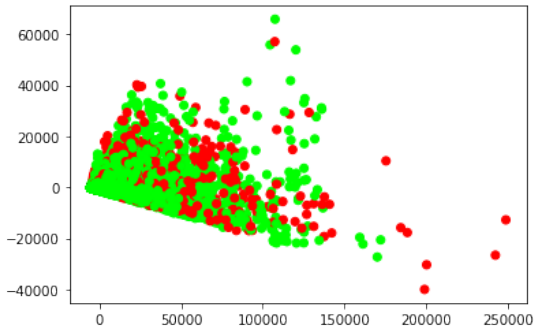
With the "hinge" loss an approximation of RBF Kernel using Monte Carlo Approximation of the Fourier Transform as implemened in 'sklearn' python package [2] was employed as an variation to linear SVM with the following hyperparameters: 50 components and 'gamma' value of '0.2'. We found that the number of components in the RBF kernel had little influence on the ability of SVMs to identify a desired soft margin.



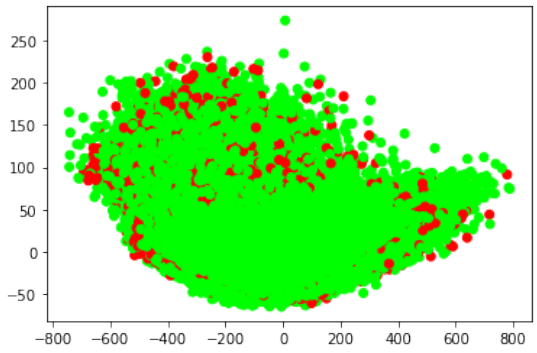
(a) First three principal components of BVP signals across test subjects. (Green: non-stressed, Red: stressed)



(b) First three principal components of features computed by HeartPy library. (Green: non-stressed, Red: stressed)



(c) First two principal components of features based on heart rate and LF/HF ratio. (Green: non-stressed, Red: stressed)



(d) First two principal components of Power Spectral Density of BVP signal. (Green: non-stressed, Red: stressed)

Figure 2: Principal Component Analysis

Results from the Linear Models are presented in Table 1.

As is evident in Table 1, the data isn't linearly separable even with the introduction of several derived hand crafted features. When presented with a choice between fine tuning features, possibly experimenting with alternate SVM kernels or employing neural networks, we chose the latter to see if we can rid ourselves of the need for hand crafted features and instead have the network identify and learn latent features by itself. We made this choice based on past research presented in [19] and [23] where neural networks with many convolutional layers were shown to produce state of the art accuracy on heart rate estimation and identifying Atrial Fibriliation based on BVP signals for PPG sensors. In addition [18] also suggested the potential for representation learning to help identify stressed vs non-stressed affective state. The rest of this paper presents the neural architectures we tried and metrics to understand how well the model fared during our experiments.

4.3 Neural Networks

We attempted to employ two kinds of networks. The first one employed 3 to 5 convolutional layers followed by 2 to 3 dense layers prior to a sigmoid layer for binary classification. The second one employed a ResNeXt [26] style grouped convolutional layers which was described as being effective in [23]. These two model architectures were experimented with both pretraining using an autoencoder and direct end to end learning settings.

We attempted to employ Variational Autoencoders with a Gaussian Prior [12] with the following architecture where kernel size, stride and padding were constant at '4', '2' and '1' respectively across all layers:

Conv1d (10 output channels)
→ ReLU
→ Conv1d (20 output channels)
→ ReLU
→ Dense (10 latent dimensions each to estimate Mu and Sigma separately)
→ Sampling
→ ReLU
→ ConvTranspose1d (10 output channels)
→ ConvTranspose1d (1 output channel)

When trained with Binary Cross Entropy loss,

Features	Model	AUC	Accuracy
Scaled PPG signals	SGD with Log Loss	0.49	0.83
Scaled PPG signals	SGD with Hinge Loss	0.50	0.81
Scaled PPG signals	SGD with Log Loss and RBF Kernel	0.50	0.88
HeartPy derived Features	SGD with Log Loss	0.49	-
HeartPy derived Features	SGD with Hinge Loss	0.47	-
HeartPy derived Features	SGD with Log Loss and RBF Kernel	0.71	0.88
Power Spectral Density	SGD with Log Loss and RBF Kernel	0.53	-
Inter Beat Interval Features	SGD with Log Loss and RBF Kernel	0.6	0.57
Inter Beat Interval Features	Random Forests (max_depth=10, min_samples_split=4) and class weights of [0.9, 0.1]	0.66	0.52
Inter Beat Interval Features	Linear Discriminant Analysis	0.64	0.57

Table 1: Results from the Linear Models

this model had Posterior Collapse [16] where a local maximum was hit early on and training got stuck there.
We abandoned this approach and instead tried a normal autoencoder where the variational sampling was discarded and a Smooth L1 Loss function as described below was employed.

$$l(x, y) = (1/n) \sum_i z_i$$
$$where, z_i = 0.5(x_i - y_i)^2$$
$$if\ |x_i - y_i| < 1,$$
$$or\ |x_i - y_i| - 1\ otherwise$$

This model was able to learn latent representations of the PPG data well where the reconstruction error was '0.002' on the training set and '0.0125' on the validation set. However since the number of the latent dimensions were quite high in this model, we attempted to reduce that by employing a Dense layer to output 10 dimensional latent representations from the encoder with which the model did not converge.
To work around this, we employed a much deeper ResNeXt type model which outputted a '16' dimensional latent variable which was then attempted to be decoded by a simple two layer transpose convolutional network with a dense layer before and after the two transpose convolution operations, to reconstruct the input. The hyperparameters for the ResNeXt model were 'cardinality' of '32' and base-channels of size '16' with 3, 4, 6 and 3 blocks of Bottleneck Convolutional blocks one after another. This resulted in a model with 11,026,673 parameters.

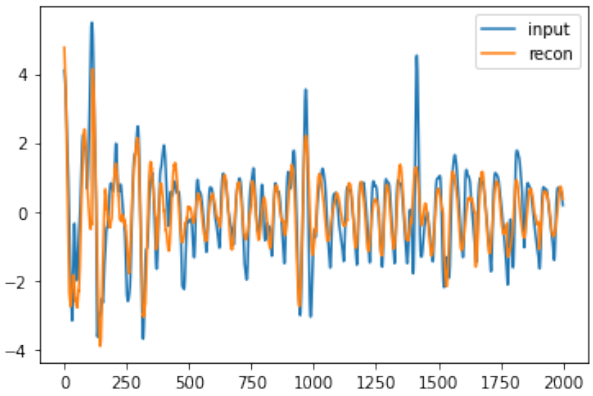


Figure 3: Input and Output from ResNeXt based auto encoder showing reconstruction loss

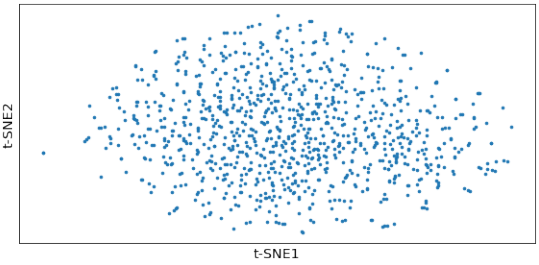


Figure 4: tSNE 2D representation of encoder outputs

As shown in Figure 3 the output from the autoencoder matched the input signal well with some noise filtered and mostly in phase.
A two dimensional visualization of the latent representation by the encoder was generated using tSNE [25] as shown in Figure 4 with a perplexity of '40'. The data wasn't readily separable which gave us an intuition that the learned representation may not help with binary classification.
This was confirmed when we attempted to employ

the encoder portion of the ResNeXt based AutoEncoder with pretrained weights and attempted to optimize for the binary classification challenge using Binary Cross Entropy Loss [2]. Two Dense layers were added after the encoder with a final sigmoid activation function to generate binary probabilities.
When the encoder weights were frozen, training loss was stuck in a local minima. If the encoder weights were unfrozen, the effect on training loss was not statistically significant across several runs.
This prompted us to ignore representation learning and attempt direct end to end learning with deep convolutional networks.

The usage of the deep ResNeXt architecture did not yield any noticeable improvement on our model metrics AUC-ROC and accuracy as compared to a simple 5 layer convolutional network and so we stopped exploring that model.

```
----------------------------------------------------------
        Layer (type)               Output Shape
==========================================================
            Conv1d-1              [-1, 10, 256]
              ReLU-2              [-1, 10, 256]
       BatchNorm1d-3              [-1, 10, 256]
           Dropout-4              [-1, 10, 256]
            Conv1d-5              [-1, 20, 128]
              ReLU-6              [-1, 20, 128]
       BatchNorm1d-7              [-1, 20, 128]
           Dropout-8              [-1, 20, 128]
            Conv1d-9               [-1, 30, 64]
             ReLU-10               [-1, 30, 64]
      BatchNorm1d-11               [-1, 30, 64]
          Dropout-12               [-1, 30, 64]
           Conv1d-13               [-1, 40, 32]
             ReLU-14               [-1, 40, 32]
        AvgPool1d-15                [-1, 40, 4]
          Flatten-16                  [-1, 160]
           Linear-17                   [-1, 10]
             ReLU-18                   [-1, 10]
          Dropout-19                   [-1, 10]
           Linear-20                    [-1, 2]
             ReLU-21                    [-1, 2]
           Linear-22                    [-1, 1]
==========================================================
Total params: 9,895
Trainable params: 9,895
Non-trainable params: 0
----------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.24
Params size (MB): 0.04
Estimated Total Size (MB): 0.28
```

Figure 5: CNN Model Structure

Our best performing model had the structure as shown in Figure 5.

The model was trained for 1000 epochs with a learning rate of 0.001 and batch size of 512. The model parameters with the best validation set AUC score after 100 epochs was checkpointed and used for evaluation.

We experimented with the following variations to the model architecture and hyperparameters.
1. Dropout varied between 0 to 25% in increments of 5%
2. Convolutional layers increased from 4 to 5 and 6 with a similar doubling of number of filters
3. Max pooling prior to average pooling
None of these variations resulted in improving our model evaluation scores.

5 Results

Our model was trained for 500 epochs with average of per mini batch total Binary Cross Entropy loss being used as the optimization criterion. Model evaluation scores are shown in Table [2].

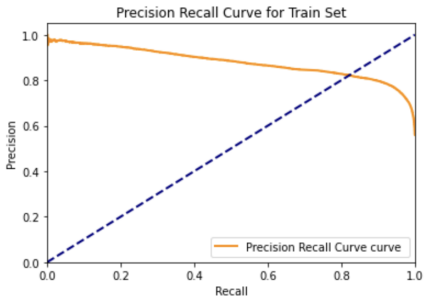
Dataset	AUC	Accuracy
Train	0.9	0.83
Validation	0.82	0.69
Test	0.7	0.6

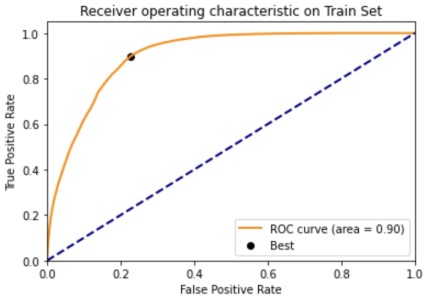
Table 2: Results from the Conv Net

The ROC and Precision/Recall graphs for this model are shown in Figure 6.
Our model suffered from overfitting as is shown in the graphs. We attempted to employ several regularization strategies including varying Dropout, adding pooling and/or batch normalization layers. However these did not help

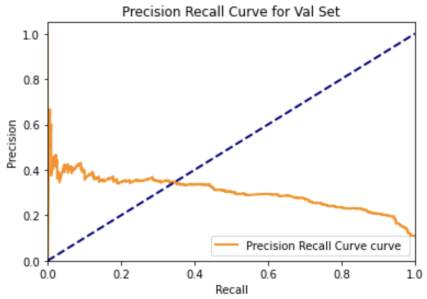
6 Discussion

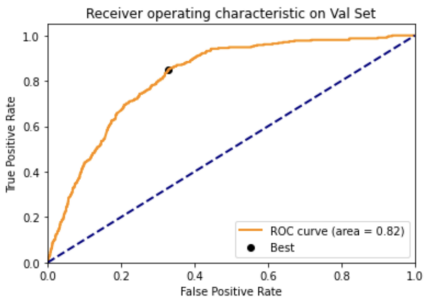
We demonstrated that it is possible to extract meaningful features to detect stress from Blood Volume Pulse signals obtained via PPG sensors. However, we did not succeed in finding an optimal model architecture that would generalize well to sample data found in Wesad dataset.
We could not identify suitable features that would help separate the classes in the dataset using a linear model. It may be possible to consider more complex kernels with Support Vector Machines to try and identify a linear separation, but we chose to not go down that path. Additional signal processing may have helped too, but past research [19] showed that signal processing techniques



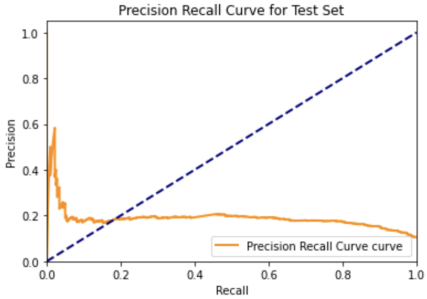


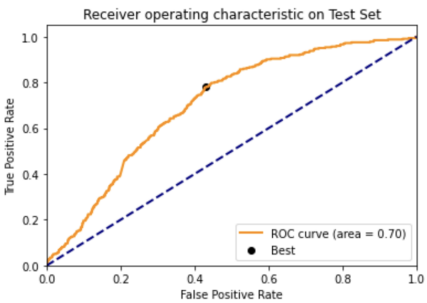
(a) ROC graph on Train Set





(b) ROC graph on Val Set





(c) ROC graph on Test Set

Figure 6: ROC curves

required some engineering beyond feature extraction to account for variations in the data. Hence we still believe identifying suitable non-linear neural networks are best suited for our hypothesis. Visual analysis of the data did not yield any obvious clues. Our attempts at developing various derived features did not help much either. We were not able to reproduce the results published in [22] where an F1 score of 83% was reported for predicting stress vs non-stress. We could not find robust software libraries that could compute the most commonly features typically extracted from PPG and ECG signals. HeartPy [1] library wasn't able to process many samples leading to further reduction of the data and required a fair deal of tuning filters and signal shaping techniques to obtain features in acceptable range, like heart rate falling in the range or [30, 250] for example. Power spectral density over time wasn't helpful in itself for classification purposes either. Inter beat interval and heart rate variability computed through that was expected to help with classification, but that wasn't strongly indicative of stress either. Overall, we had little choice but to attempt to model directly from PPG signals after simple filtering and scaling.

BVP signals from PPG sensors were inherently noisy. Their amplitude isn't meaningful by itself and it is the second derivative that carries the most useful information. We didn't get around to test if our models were able to successfully identify the second derivative automatically in any of it's intermediate layers. Past research in [23] demonstrated CNNs' ability to identify the derivatives. One possibility is to explicitly compute the second derivatives and use that as inputs to a simpler neural network and evluate it's performance.

Another issue we faced was that of an heavily imbalanced dataset. This coupled with the fact that our dataset was relatively small (less than 4x model parameters) as made it challenging to develop a model that could generalize well. Our oversampling strategy with SMOTE helped increase the dataset size, but that added samples likely did not match the statistical distributions found in the validation or test set.
Since we identified a model architecture that is able to overfit the training data well, our challenge

was to identify optimal regularization parameters to offset the over-fitting and have the model generalize adequately. Our exploration did not yet yield a favorable set of model parameters with precision ideally greater than 0.9.

It might be useful to further evaluate variational autoencoders to generate synthetic data for developing a more robust classifier as labeled data is scarce typically in healthcare problems. Our attempts to develop general purpose latent representations of the data were unsuccessful as well which highlighted the general complexity of our hypothesis.

We did not score our model against a separate dataset which would help with identifying if the model learned meaningful representation or it learned to classify based on noise in the dataset. There is at least one other dataset [17] which could be used for this purpose. We could have randomly sampled subjects from both datasets to help our model learn from a more diverse dataset too.
Since we noticed generalization issues even within the same dataset, it would be useful to run a more comprehensive crossvalidation (LOGCV) across the entire dataset which would provide a more robust measure of the fficacy of the model.

We did not run an exhaustive search on other hyperparameters like kernel size, stride length, padding, output channels, learning rate, batch size, etc., which could have helped us identify a more optimal model architecture or optimization strategy.

We picked our model parameters based on validation set AUC score. This may not be the ideal choice. We could have picked a preset precision and recall threshold and instead chosen model parameters that meet or exceed our threshold in retrospect, or learn that our model fails to generalize through that process. Employing the Precison-Recall curve was useful in understanding the model characteristics.

To conclude, our core findings were as follows:
1. Detecting Stress from PPG signals is a non-linear classification problem.
2. Extracting common features found in the literature doesn't readily help detect stress.
3. Convolutional networks with simple pre-processing of PPG signals show ability to detect stress, but will require careful model optimization, cross-validation and generalization tests with separate datasets to develop robust models.
4. Autoencoder (or GANs) are a useful avenue to explore to develop more synthetic data to train larger neural networks effectively. Our work demonstrated that autoencoder are capable of developing representations of PPG data for the purposes of reconstruction.

6.1 Direction of Future Research

For this paper, We assumed the class labels provided by the Wesad dataset authors as being valid. We did not test to see if our models were more correlated with the self-reported stress scores provided by test subjects than the ground truth. It would be useful to identify any other modalities that are more indicative of stress like ECG signals, skin temperature, electro dermal activity (influenced by sweat), respiration rate, etc., and use them to evaluate the accuracy of the class labels. An Autoencoder could be employed to identify clusters that may exist between the different affective states studied in the dataset.

Given the scarcity of data, it may be useful to explore multi-task learning where a model trained to identify heart rate could be used as an input to the stress prediction model [13].

Future research around stress related data collection should ideally include some clinical analysis of Cortisol (in sweat or saliva) to establish a better personalized ground truth on the level of stress per test subject.

References

[1] In: (). URL: https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/.

[2] In: (). URL: https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.RBFSampler.html.

[3] Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal Of Artificial Intelligence Research* 16 (2002), pp. 321–357. URL: https://arxiv.org/abs/1106.1813.

[4] Amy FT Arnsten. "Stress signalling pathways that impair prefrontal cortex structure and function". In: *Nature reviews neuroscience* 10.6 (2009), pp. 410–422.

[5] Dwaipayan Biswas et al. "Heart Rate Estimation From Wrist-Worn Photoplethysmography: A Review". In: (2019).

[6] Denisse Castaneda et al. "A review on wearable photoplethysmography sensors and their potential future applications in health care". In: *International journal of biosensors & bioelectronics* 4.4 (2018), p. 195.

[7] Sayeed Shafayet Chowdhury et al. "Real-time robust heart rate estimation from wrist-type PPG signals using multiple reference adaptive noise cancellation". In: *IEEE journal of biomedical and health informatics* 22.2 (2016), pp. 450–459.

[8] Sheldon Cohen, Denise Janicki-Deverts, and Gregory E Miller. "Psychological stress and disease". In: *Jama* 298.14 (2007), pp. 1685–1687.

[9] Sheldon Cohen, Ronald C Kessler, Lynn Underwood Gordon, et al. "Strategies for measuring stress in studies of psychiatric and physical disorders". In: *Measuring stress: A guide for health and social scientists* (1995), pp. 3–26.

[10] "Cup I.S.P. Heart Rate Monitoring During Physical Exercise using Wrist-Type Photoplethysmographic (PPG) Signals." In: (2015). URL: https://sites.google.com/site/researchbyzhang/ieeespcup2015.

[11] Giorgos Giannakakis et al. "Review on psychological stress detection using biosignals". In: *IEEE Transactions on Affective Computing* (2019).

[12] Irina Higgins et al. "beta-vae: Learning basic visual concepts with a constrained variational framework". In: (2016).

[13] Dong-Jin Kim et al. "Disjoint multi-task learning between heterogeneous human-centric tasks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1699–1708.

[14] Sang Hyun Kim, Dong Wan Ryoo, and Changseok Bae. "Adaptive noise cancellation using accelerometers for the PPG signal from forehead". In: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2007, pp. 2564–2567.

[15] Clemens Kirschbaum, Karl-Martin Pirke, and Dirk H Hellhammer. "The 'Trier Social Stress Test'–a tool for investigating psychobiological stress responses in a laboratory setting". In: *Neuropsychobiology* 28.1-2 (1993), pp. 76–81.

[16] James Lucas et al. "Understanding posterior collapse in generative latent variable models". In: (2019).

[17] Rita Meziatisabour et al. "UBFC-Phys: A Multimodal Database For Psychophysiological Studies Of Social Stress". In: *IEEE Transactions on Affective Computing* (2021).

[18] Ali Oskooei et al. "Destress: deep learning for unsupervised identification of mental stress in firefighters from heart-rate variability (hrv) data". In: *Explainable AI in Healthcare and Medicine*. Springer, 2021, pp. 93–105.

[19] Attila Reiss et al. "Deep ppg: Large-scale heart rate estimation with convolutional neural networks". In: *Sensors* 19.14 (2019), p. 3079.

[20] Philipp V Rouast et al. "Remote heart rate measurement using low-cost RGB face video: a technical literature review". In: *Frontiers of Computer Science* 12.5 (2018), pp. 858–872.

[21] Seyed Salehizadeh et al. "A novel time-varying spectral filtering algorithm for reconstruction of motion artifact corrupted heart rate signals during intense physical

activities using a wearable photoplethysmogram sensor". In: *Sensors* 16.1 (2016), p. 10.

[22] Philip Schmidt et al. "Introducing wesad, a multimodal dataset for wearable stress and affect detection". In: *Proceedings of the 20th ACM international conference on multimodal interaction*. 2018, pp. 400–408.

[23] Yichen Shen et al. "Ambulatory atrial fibrillation monitoring using wearable photoplethysmography with deep learning". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1909–1916.

[24] J. Taelman et al. "Influence of Mental Stress on Heart Rate and Heart Rate Variability". In: 2009.

[25] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[26] Saining Xie et al. "Aggregated Residual Transformations for Deep Neural Networks". In: *arXiv preprint arXiv:1611.05431* (2016).