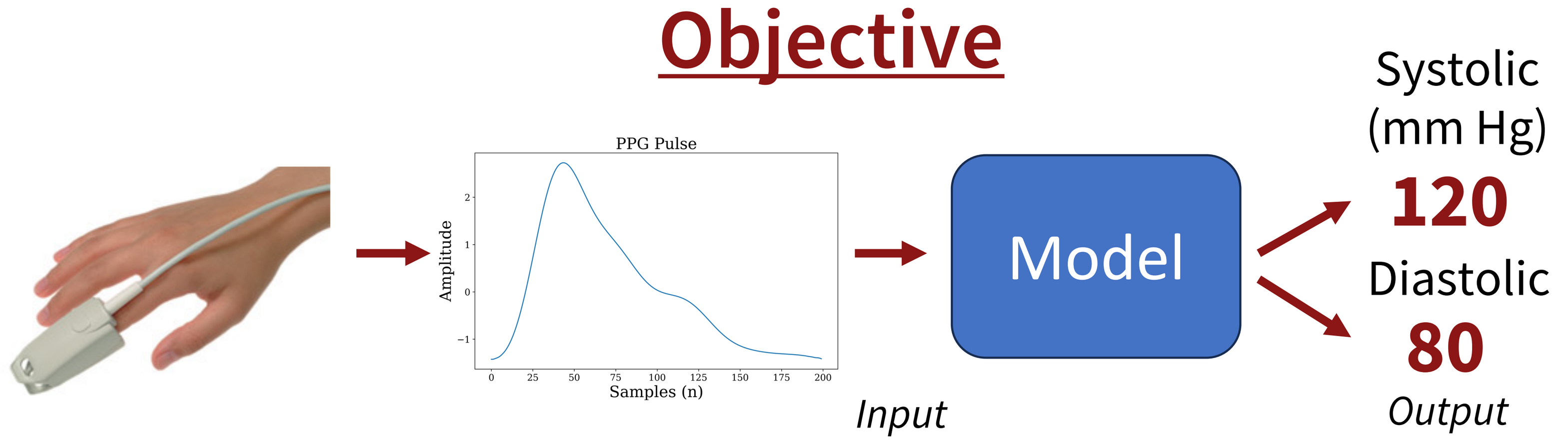
# Efficient Blood Pressure Prediction from Photoplethysmography Signals


Edward Kim[2] (edkim36@stanford.edu), Rohan Sanda[1] (rsanda@stanford.edu)

[1]Department of Electrical Engineering, Stanford University

[2]Department of Computer Science, Stanford University


## Objective



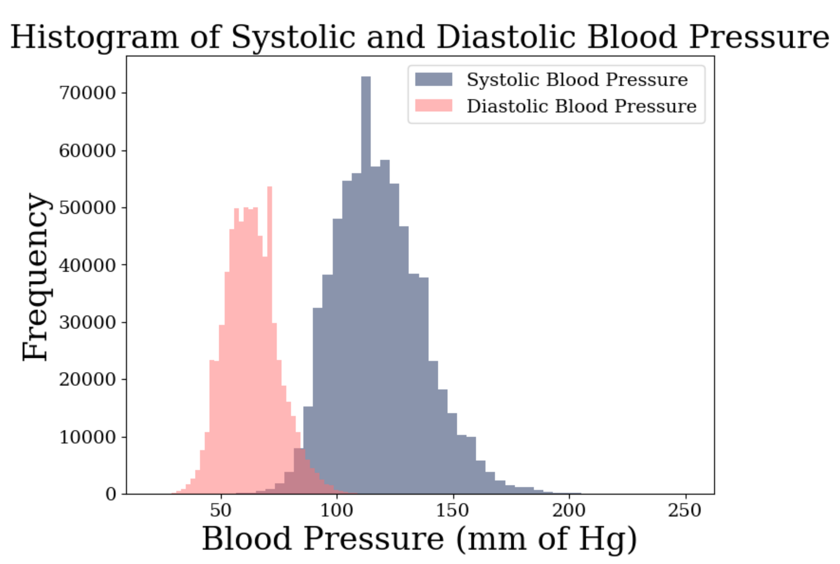

Continuous blood pressure (BP) measurements are crucial for monitoring and diagnosing disease. Photoplethysmography (PPG) signals correspond to volumetric variations in blood circulation and has been shown to be correlated with BP [1]. PPG (unlike ECG) is easy to accurately measure and incorporate into smart-watches, motivating research into PPG-based BP prediction [1], [2].

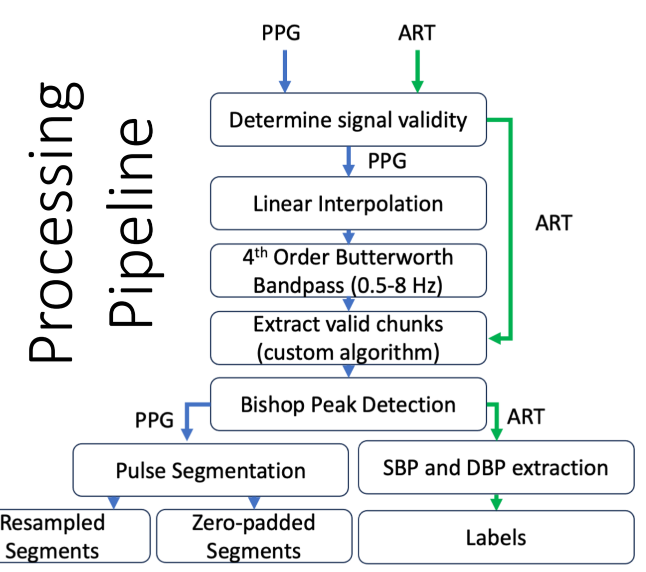
**Guiding Questions:**

1. How should we represent the PPG signal as input to our model?
2. How can we improve PPG-prediction models?

## Dataset and Processing

We obtained raw PPG and arterial BP waveforms (our ground truth) from the **VitalDB dataset** [3] which contains the vitals signs from 6,388 patients undergoing surgery in South Korea. We processed **300 patients' data** using a custom-built pipeline to generate two datasets each containing **718,035 normalized pulses** (200-vectors) and **corresponding SBP and DBP** (scalars).




## Methods

Guiding Question 1: How should we represent the input signal?

- Hand-crafted Mixed Features: Following the example of [1] we compute **45 features** for each PPG pulse. Features are morphological (temporal distances to critical points), frequency-dependent (Fourier coefficients), the top four PCA components of the pulse.
- Autoencoder Embeddings: We use the 45-dimensional embeddings from an off-the-shelf linear autoencoder. We could not get the LSTM autoencoder to converge
- Raw signal: We pass in the processed PPG pulse.

Guiding Question 2: How can we improve PPG-prediction models?

- Baseline models: Ridge Regression, Adaboost Regression, SVR Regression (RBF kernel), and Random Forest
- Multi-output Neural Networks: Fully-connected, 1D-CNN, 1D-CNN + RESNET

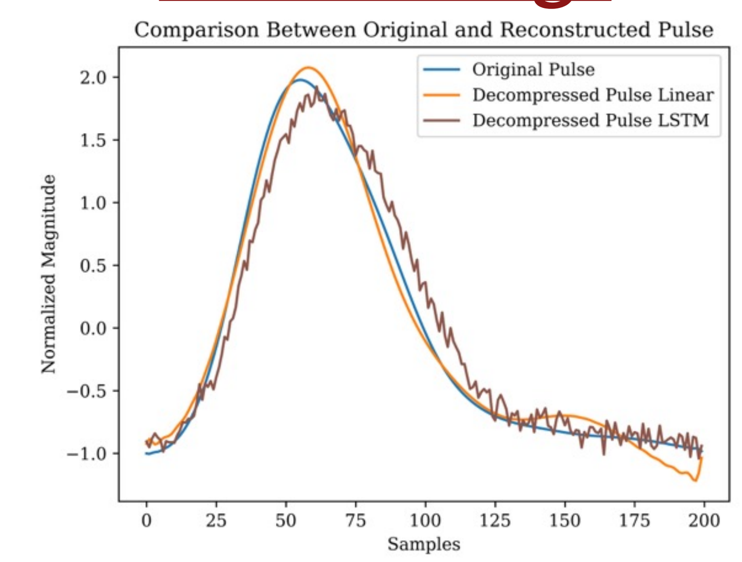
## Evaluation Metrics

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

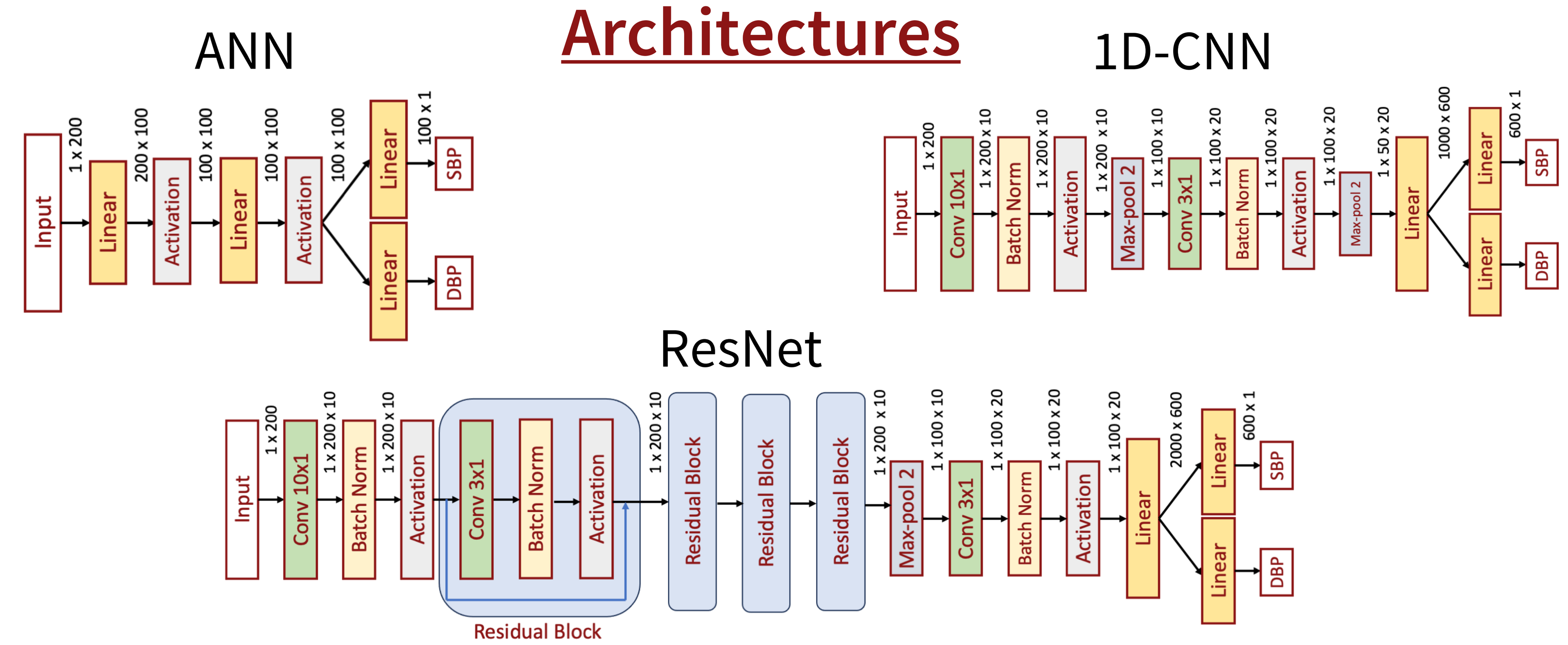
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

## Embeddings




## Architectures




As shown below, the raw signal, zero-padded dataset performed best on the baselines. We implemented three different neural network architectures above on this dataset to improve prediction performance from baselines. Note that the loss function for the multi-output neural networks was as follows:

$$Combined\ Loss = MSE(SBP) + MSE(DBP)$$

There is an inherent weighting towards the SBP loss due to its larger magnitude. This weighting is desirable as SBP has higher variance than DBP.

## Results

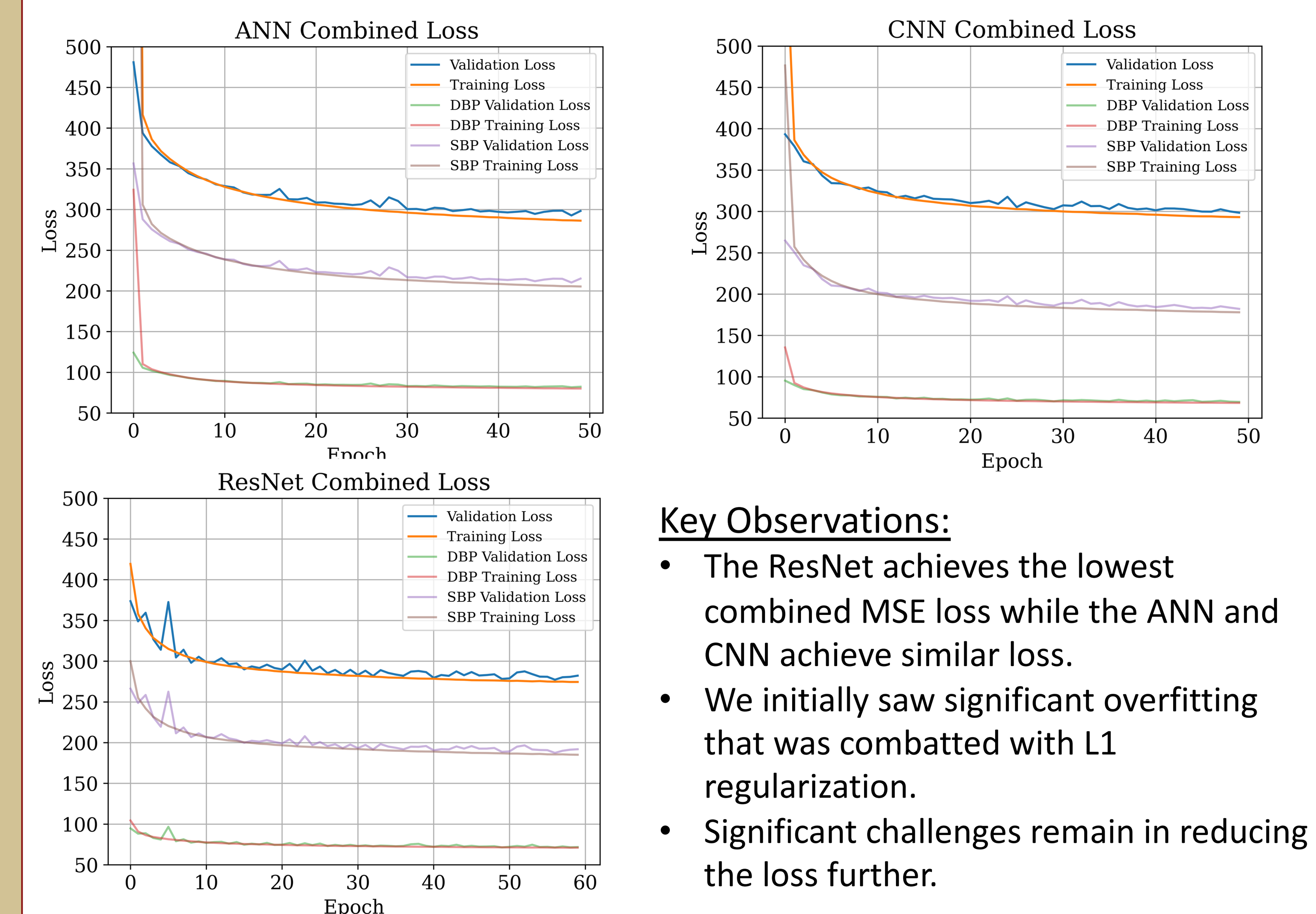
| Model | SBP Results | | | | | | DBP Results | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Featurized | | | Raw Signal | | | Featurized | | | Raw Signal | | |
| | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ |
| **Random Forest** | 13.957 | 17.940 | 0.156 | 12.178 | 16.054 | 0.327 | 7.816 | 10.097 | 0.214 | 7.282 | 9.551 | 0.300 |
| **SVR Regression** | 15.361 | 19.534 | 0 | 11.802 | 15.977 | 0.333 | 9.019 | 11.381 | 0.001 | 7.019 | 9.470 | 0.312 |
| **Adaboost Regression** | 14.532 | 18.436 | 0.109 | 13.898 | 17.461 | 0.204 | 8.399 | 10.500 | 0.150 | 8.402 | 10.508 | 0.153 |
| **Ridge Regression** | 15.052 | 19.310 | 0.022 | 15.077 | 19.245 | 0.033 | 8.624 | 10.949 | 0.076 | 8.875 | 11.277 | 0.024 |

| Model | Train Results | | | | | | Test Results | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SBP | | | DBP | | | SBP | | | DBP | | |
| | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ |
| **ANN** | 10.908 | 14.332 | 0.440 | 6.922 | 8.948 | 0.365 | 11.124 | 14.668 | 0.411 | 7.029 | 9.07 | 0.348 |
| **CNN** | 10.040 | 13.341 | 0.514 | 6.310 | 8.270 | 0.457 | 10.125 | 13.519 | 0.506 | 6.396 | 8.40 | 0.457 |
| **ResNet** | 10.286 | 13.604 | 0.495 | 6.453 | 8.429 | 0.437 | 10.287 | 13.724 | 0.489 | 6.499 | 8.525 | 0.450 |

Key Observations:

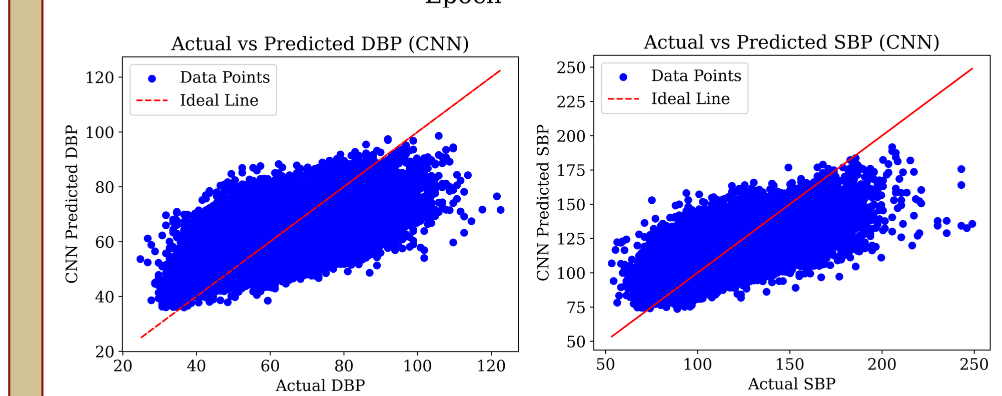
- The **raw signal representation performed best** across all models since it contained the most information.
- The linear auto-encoder performed poorly likely since it introduced too much distortion (see Embeddings). The autoencoder embeddings produced a SBP MAE of 15.789 so we did not include these results here.
- The CNN performs the best, though all neural nets are able to bring errors down significantly.
- Our **best MAE of 10.125** with the CNN is similar to those published in literature (9.43 with CNN in [2], 11.53 with RF in [4]). With more resources, we can process more data and improve our neural net performance.

## Discussion




Key Observations:

- The ResNet achieves the lowest combined MSE loss while the ANN and CNN achieve similar loss.
- We initially saw significant overfitting that was combatted with L1 regularization.
- Significant challenges remain in reducing the loss further.




- Actual vs. predicted curves indicate that there is some structure to the incorrect BP prediction with higher BPs being less accurate – more model optimization is needed.

## Next Steps

Our unique approach of using the raw PPG signal while applying minimal processing is shown to yield MAE's on par with literature on PPG-only BP prediction. This indicates that our methodology and implementation are somewhat effective. However, our models' best MAE of 10.1 mm of Hg is not useful for medical diagnosis. With more time and resources, we may explore the following:

- LSTM models for patient-specific BP prediction
- Creating multimodal models incorporating other biosignals that are easy to measure and potentially linked to BP
- Analyzing the convolutional layer output of our CNNs to gain insight into the learned features

## References


[1] Seungman Yang, Jangjay Sohn, Saram Lee, Joonnyong Lee, and Hee Chan Kim. Estimation and validation of arterial blood pressure using photoplethysmogram morphology features in conjunction with pulse arrival time in large open databases. *IEEE Journal of Biomedical and Health Informatics*, 25(4):1018–1030, 2021

[2] Nejc Mlakar Slapniˇcar, Gašper and Mitja Luštrek. Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network. *Sensors*, 9(15), 2019

[3] HC. Lee, Y. Park, and S.B. Yoon. Vitaldb, a high-fidelity multi-parameter vital signs database in surgical patients. *Nature Scientific Data*, 9(279), 2022

[4] Umapathy Mangalanathan V. Jeya Maria Jose M. Anand Geerthy Thambiraj, Uma Gandhi. Investigation of the effect of womerseley number, ecg and ppg features for cuff-less blood pressure estimation using machine learning. Biomedical Signal Processing and Control, 60, 2020


Stanford University