Predicting Blood Pressure: A Deep Learning Outlook into Photoplethysmography Signals

Justin Wu
Department of Computer Science
Stanford University
justwu@stanford.edu

D M Raisul Ahsan
Department of Computer Science
Stanford University
dahsan@stanford.edu

Samuel Winson Tanuwidjaja
Department of Civil Engineering
Stanford University
samuelwt@stanford.edu

ABSTRACT

Photoplethysmography (PPG) is a non-invasive optical technique that has been known to be a successful indicator of Blood Pressure (BP). In this study, we aim to develop classical and deep learning models on PPG signal data in order to predict Diastolic and Systolic BP. We construct several regression and classification models (SVR, Random Forest, ANN, LSTM, etc) and implement feature reduction algorithms (Pearson Correlation, ANOVA, etc) to model BP with less features than standard models. We are able to develop models (trained on exclusively PPG data) that perform at similar or better accuracy levels than recent, state-of-the-art models. Further, we show that it is possible to obtain comparable results with significantly less features.

1. INTRODUCTION

Blood pressure (BP) readings are widely considered in the medical community as a reliable indicator of an individual's cardiovascular health [1]. BP readings, both systolic and diastolic, are traditionally measured using cuff-based devices; however, more recent advances in technology have given rise to BP predictions using signals derived by non-invasive devices such as the electrocardiogram (ECG) and the photoplethysmogram (PPG) [2, 3].

Our group aims to build BP predictive models using exclusively PPG signal data, as PPG data can be easily collected. Our group sets out to do both regression and classification tasks, respectively. The input to our models are handmade numerical features that are extracted from the PPG signal waveform, and the output will be systolic (SBP) and diastolic BP (DBP) values for regression, and hypertension level for classification. For regression, our group explored SVR, Random Forest, ANNs, and LSTMs. For classification, our group explored Softmax Regression, Random Forest and ANNs. Our group aims to understand which methods performs well, while still having a relatively simple architecture and utilizing the least number of extracted features from the PPG signals.

2. RELATED WORK

Plenty of prior work has been done in translating PPG signals to BP values and many have achieved reasonable amounts of success in doing so [4–6]. Convolutional Neural Networks (CNNs) is a deep learning method that has been repeatedly used in literature to solve this problem. Slapničar et al. (2019) utilized what is known as a spectro-temporal ResNet, which in its architecture comprises multiple CNN layers [4], whereas Schrumpf et al. (2021) employed four different neural network architectures, three of which are CNNs [6]. Their models performed sufficiently well, both of which achieving Mean Absolute Errors (MAEs) of less than 10, in general, for both SBP and DBP.

CNNs are also popular in literature for solving the classification version of this problem. Fuadah & Lim (2022) varied the number of convolutional layers used in their model, but upon training with only PPG data, their models yield low test accuracies ranging between 47% and 61% [7]. Other researchers have employed variations of CNNs in solving the classification problem, also with a varying degree of success, with accuracies ranging from as low as 76% to as high as 93% [8–10].

As we have observed, the use of CNNs does not necessarily guarantee performance, but the architecture are typically very involved and complicated. Moreover, in a lot of these papers, PPG data is not exclusively used when building models.

Random Forest, Support Vector Regression (SVR), Linear Regression are some non-deep learning techniques that have been used to predict specific SBP and DBP values [2, 11]. Random Forest performs surprisingly well in both the papers that are cited, yielding MAEs of lesser than 10 for both SBP and DBP. A simple four-layer neural network has been used as well for SBP and DBP prediction, yielding MAEs that are almost lesser than 5 for both SBP and DBP [12]. Unlike some of the previous CNN-related papers such as Slapničar et al. (2019), Kurylyak et al. (2013) and Thambiraj et al. (2020) extracted features from the PPG waveforms first and set them as input, rather than inputting the PPG waveforms themselves. We believe that this is key in the ability of the 'simpler' models to learn as well or even better than the more architecturally-complicated models.

3. DATA

3.1 Dataset

The dataset was obtained from UCI Machine Learning Repository. The dataset contains processed data from the MIMIC

II database. In the dataset, there is a total of 12000 instances of PPG, ABP, and ECG signals taken at a sampling frequency of 125 Hz. The ABP dataset provides the ground truth for Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP). The extracted SBP and DBP were used to filter out the extreme outlier examples with $SBP < 60$ or $SBP > 200$ and $DBP < 40$ or $DBP > 160$ and $SBP < DBP$. After the filtering process, a total of 9488 examples were kept. Among the 9488 examples, 6641 examples were used for training and 2847 examples were used for testing.

3.2 Feature Extraction

We only extracted features from PPG waveforms since our goal is to achieve accuracy solely by using the PPG features. We employed feature extraction methods devised by Thambiraj et al [2]. and Kachuee et al [13]. These methods were also built on many prior studies. A total of 8 physiological features (Figure 2), and 20 temporal features were initially selected. The 20 temporal features were derived utilizing the 8 features extracted from the waveform (Figure 1). Out of the 8 physiological features, 5 were extracted directly from the waveform, and the sixth feature AI (Augmentation) was from the ratio of the inflection point and systolic peak, $AI = \frac{x}{y}$. The other 2 physiological features are the maximum (I_H) and minimum (I_L) amplitudes of the pulse signal. Table 1 provides a comprehensive list of features.

Temporal features	$ST, DT, DW10, SW10{+}DW10, \frac{SW10}{DW10}, DW25, SW25 + DW25, \frac{SW25}{DW25}, DW33, SW33 + DW33, \frac{SW33}{DW33}, DW50, SW50 + DW50, \frac{SW50}{DW50}, DW66, SW66 + DW66, \frac{SW66}{DW66}, DW75, SW75 + DW75, \frac{SW75}{DW75}$
Physiological features	$I_H, I_L, S1, S2, S3, S4, AI, LASI$

Table 1. 28 Features

3.3 Blood Pressure Classes

The SBP and DBP values were divided into 5 classes (table 2) as recommended by American Heart Association for the classification algorithms.

Classes	SBP	and/or	DBP
Low Blood Pressure	< 120	and	< 80
Normal Blood Pressure	$120 - 129$	and	$80 - 89$
Elevated Blood Pressure	$130 - 139$	or	$80 - 89$
Hypertension Stage 1	> 140	or	> 90
Hypertension Stage 2	> 180	or	> 120

Table 2. Blood Pressure Classes

4. METHODOLOGY

Our baseline model for regression and classification involved Linear and Softmax Regression, respectively. We followed this work with more sophisticated algorithms that are summarized in the following subsections. After developing models for regression and classification, we aimed to select a few of the best-performing algorithms to perform *dimensionality reduction* on the number of features; that is, we want to optimize a performance-complexity trade-off problem between the accuracy and the number of features in our respective models.

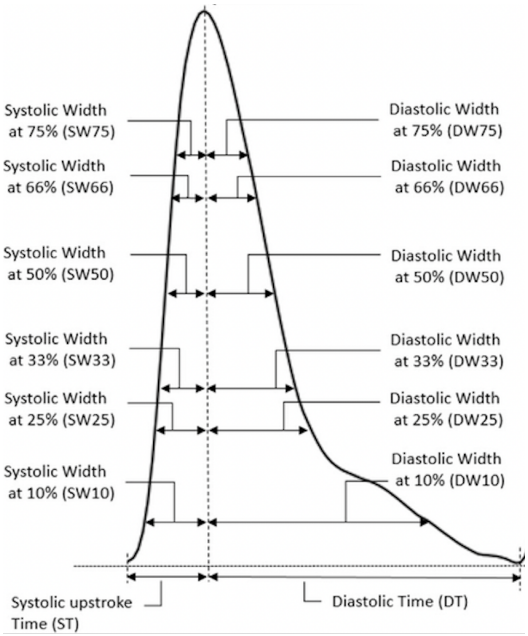

Figure 1. 8 PPG temporal features [14]

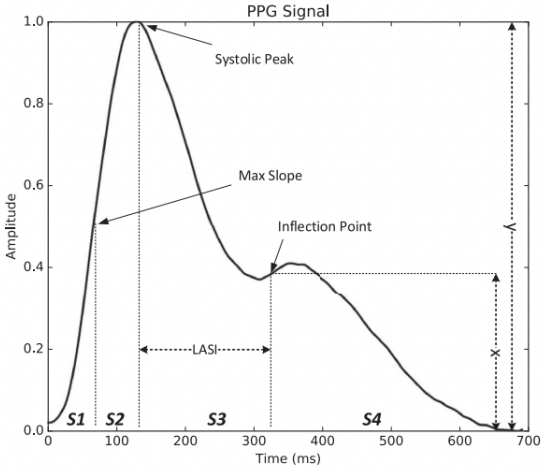

Figure 2. 5 PPG physiological features [13]

4.1 Prediction

The first class of algorithms we work with are designed to *predict* values and classes of (blood pressure) given data associated to features and their labels.

4.1.1 Baseline

Our base-line Linear regression is a prediction method that works by optimizing a linear hypothesis function $h_\theta(x) = \theta^T x$ under the cost function $J(\theta) = \|X\theta - y\|^2$ via gradient descent, where X is the design matrix and y contains the associated labels. We also implement Ridge Regres-

sion, which performs a similar task, but is intended to analyze data that may suffer from highly correlated features. Thus, it performs L_2 regularization, yielding a cost function of: $J(\theta) = \|X\theta - y\|^2 + \lambda\|\theta\|^2$.

For our baseline classification, we implement Softmax Regression, which is a generalization of logistic regression, defined by $softmax(z)_i = \frac{\exp z_i}{\sum_{j=1}^{k} \exp z_j}$. Softmax returns a vector of probabilities, indicating the probability that some input would belong to each class.

4.1.2 Support Vector Regression

SVR is often seen as the generalization of SVMs, going from an output from a finite set to a continuous-valued output [15]. The objective in SVR is to minimize the L_2 norm of the coefficient vector - not the squared error. Here, the aim is to find a function that minimizes the prediction error. The error term is instead handled in the constraints, where we set the absolute error less than or equal to a specified margin, called the maximum error, ε (epsilon). We can tune epsilon to gain the desired accuracy of our model. Our new objective function and constraints are as follows: We'll use the radial basis function (RBF) kernel, defined as:

$$K(x, z) = \exp\left(-\frac{\|x - z\|_2^2}{2\sigma^2}\right)$$

4.1.3 Random Forest Regression

Random forest regression is a type of ensemble regression algorithm that works by training multiple decision trees on a dataset and then combining the predictions of each tree to make a final prediction. The random forest algorithm works by training each decision tree on a random subset of the data and using a random subset of the features to make predictions. This randomness helps to reduce overfitting and improve the accuracy of the model. The final prediction is made by taking the average of the predictions of all the decision trees in the forest. Random forest regression is an effective method for dealing with nonlinear and complex relationships in the data.

4.1.4 Neural Networks

Neural Networks are able to detect highly-nonlinear patterns within complex data by adjusting the strength of the connections between *neurons* in the network via backpropagation. In this study, we implement two different general architectures. For regression, we'll use a standard feed-forward Neural Network and use the Mean Squared-Error loss function. For classification, we tune a similar model, but we append a softmax function at the end. We'll use the Cross-Entropy as our loss function:

$$CE(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

4.1.5 Long Short-term Memory Networks

LSTMs are a special kind of Recurrent Neural Network that are capable of learning long-term dependencies within the data. At each time step, an LSTM unit works by predicting the next value h_t in a sequence by remembering information from past inputs, intermediates, and predictions. They are generally governed by the following principals (consult Figure 3).

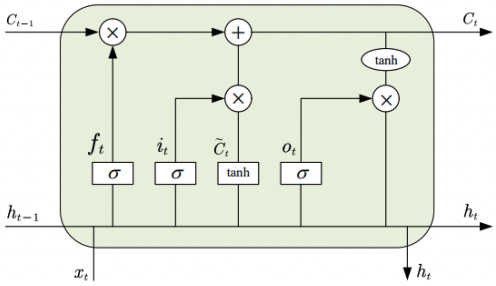

Figure 3. Vizualization of a single LSTM cell [16]

$$f_t = \sigma(W_f C_t + U_f h_{t-1} + b_f), \quad i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad C_t = f_t \odot C_{t-1} + i_t \odot C_t$$

$$\tilde{C}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c), \quad h_t = o_t \odot \sigma_h(C_t)$$

where σ is the sigmoid function, σ_h is the $\tanh$ function, and W, U, b are weight matrices and bias vectors that are learned during training. At each time slot, we calculate the input gate i, forget gate f, and output o. After these intermediate values are calculated, we update the current unit C_t, and output the prediction h_t [16]. While the previous regression models rely on features that we extracted from the PPG data, Long Short-term Memory Networks (LSTMs) rely on sequential data. Instead of extracting input features, we will be feeding in the actual wave signal as time-series data.

4.2 Feature Selection

Here, we outline the techniques we used for identifying and removing unnecessary or redundant features in our dataset. Techniques such as univariate chi-squared, lasso, etc., which did not perform as well, will be excluded from the outline.

4.2.1 Pearson's Method

Pearson's correlation coefficient is a statistical measure that is used to determine the strength and direction of the linear relationship between two variables. By calculating the correlation

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

between pairs X, Y of features, Pearson's method can help identify which features are strongly correlated with each other, and which features are not. The idea is that features that are strongly correlated with each other can be removed from the dataset without significantly affecting the overall accuracy of the model, because the information contained in those features is already present in other features in the dataset.

4.2.2 One-way ANOVA

One-way ANOVA (short for "analysis of variance") tests for statistically significant differences in the means between two or more independent groups [17]. In our context, the groups can be considered as the individual features. A null hypothesis (H0) would be that there are no significant differences in the means between all considered features. If the null hypothesis (H0) cannot be rejected, then this might

imply that the pair of features may not be significantly different from each other in terms of means and could potentially be removed due to redundancy.
The one-way ANOVA test can be characterized by the following F statistic:

$$\mathrm{F} = \frac{\sum_{i=1}^{K} n_i \left(\bar{Y}_i - \bar{Y}\right)^2 / (K-1)}{\sum_{ij=1}^{n} \left(Y_{ij} - \bar{Y}_i\right)^2 / (N-K)}$$

where $\bar{Y}_i$ is the mean of group i, n_i is the number of datapoints in group i, $\bar{Y}$ is the overall mean, K is the number of groups, Y_{ij} is the j^{th} datapoint in group i, and N is the total number of datapoints in the whole population.The larger the F value, the larger the inter-group variance relative to the intra-group variance, and once it crosses a certain threshold, the null hypothesis (H0) can be rejected.

4.2.3 Random Forest Model Selection

For our Random Forest model, we employed a feature selection technique provided from scikit-learn known as SelectFromModel. SelectFromModel assigns an importance score to each of the features via a specific attribute, and then eliminates features that have an importance score lower than the threshold. For random forest, the specific attribute is the mean decrease in impurity (MDI).

The framework by which MDI is used is described extensively in literature [18–20]. The following formulation to calculate the MDI is adapted from [18].

$$MDI = \frac{1}{ntree} \sum_{t=1}^{ntree} (MP_{tj} - M_{tj})$$

where $ntree$ is the number of trees in the forest, M_{tj} is the error of tree *t* before permuting the values of predictor variable X_j, and MP_{tj} is the error of tree *t* after permuting values of predictor variable X_j.
If the predictor is not associated with the results, a permutation of the predictor will not change the error by much, meaning that the importance score above will be close to zero [18].

5. EXPERIMENTS & RESULTS

5.1 Regression

We run Linear, Ridge, Support Vector, and Random Forest Regression (100 trees) on the 28-feature input data. We employ several LSTM networks with the best performing one having an encoder-decoder architecture, where each LSTM layer contains 512 units. The idea is for the first layer to extract features from the Time-Series data, which will be fed into the second layer, which makes predictions through a dense, time-distributed sequnce prediction layer. Note that here, we input the raw PPG pulse-wave as time-series data, rather than inputting extracted features. We also fine-tune a standard 4-layer Neural Network regressor model with hidden layers containing $80, 40, 20$ neurons, respectively. We use a learning rate of 0.001 under 1000 epochs. We evaluate them primarily on the Mean Absolute Error, which is a computationally efficient evaluation metric that provides clear intuition and performance on the mode, and is the metric used by the Association for the Advancement of Medical Instrumentation (AAMI) for validating BP monitoring devices. The results can be summarized in Table 3 (note that all values are measured in mmHG).

Algorithm	MAE for DBP		MAE for SBP	
	Train	Test	Train	Test
Linear Regression	15.97	16.08	16.09	16.01
Ridge Regression	17.07	16.98	16.63	16.18
Support Vector Reg.	17.42	17.19	17.42	17.19
LSTM (2 Layer)	7.383	7.543	7.100	7.293
RF (100 Trees)	1.018	1.602	1.344	2.082
NN (4 Layer)	2.565	3.139	1.826	4.767

Table 3. Regression Results over 28-feature input data

Random Forest and Neural Networks clearly outperform the other models and in fact (as we'll see later) most other models in the literature. Now, we'll perform *feature reduction*: on the Random Forest Regressor, we'll implement SelectFromModel and in our Neural Network, we'll use Pearson's method to reduce the number of features. We measure the accuracy of our models with respect to the number of features.

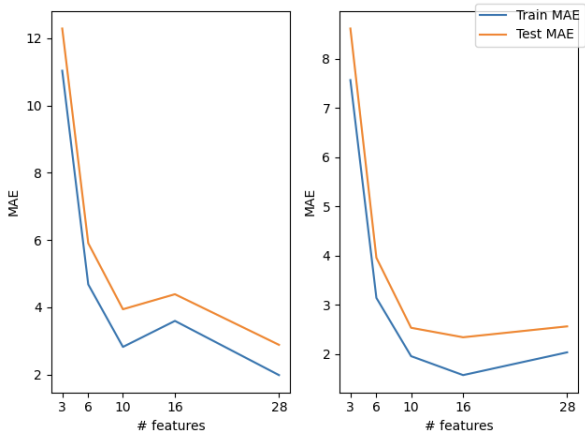

Figure 4. NN Feature Reduction on SBP & DBP

In Random Forest, we obtain strong accuracy at 6 features, while in our Neural Network, we obtain strong accuracy at 10 features. This may be due to the fact that employing feature reduction allows our model to generalize on unseen data since it eliminates insignificant features that may clutter the test data.

5.2 Classification

For 5-class classification, we obtain a baseline accuracy of 58% using Softmax Regression. We fine-tune a 4-layer Neural Network (with the same parameters as our NN regressor) and Random Forest Classification Algorithm with 100 trees. Both of these models perform significantly better than our baseline, with our NN reaching 96% and our RF achieving 99% accuracy.
Once again, we'll perform feature reduction to see if we can obtain strong accuracy with less features. On our Neural Network, we'll use One-Way ANOVA and in our Random Forest, we'll implement RF Model Selection (Table 4) Notice again that with only 3 features, our Neural Network still performs well, while our Random Forest is able to consistently perform strongly. This can most likely be attributed to the strong depths that each Decision Tree in

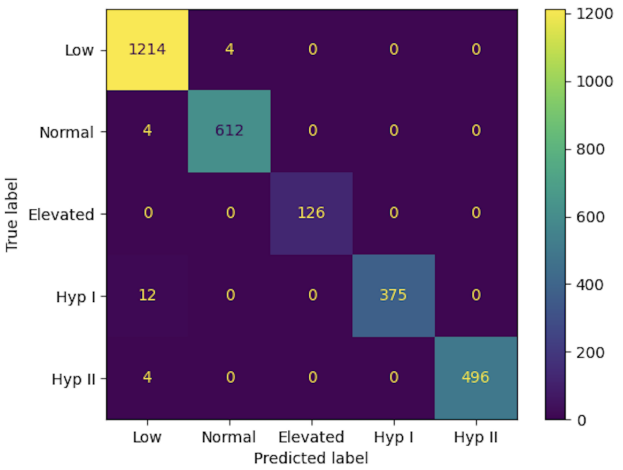

Figure 5. Confusion Matrix for RF (28 features)

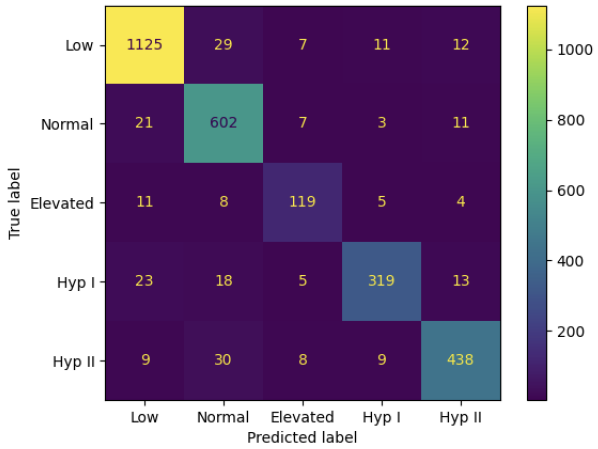

Figure 6. Confusion Matrix for NN (28 features)

Features	Random Forest		Neural Network	
	Train	Test	Train	Test
16	1.0000	0.9944	0.9637	0.9631
10	1.0000	0.9888	0.9809	0.9325
6	1.0000	0.9845	0.9899	0.9006
3	1.0000	0.9888	0.9727	0.9391

Table 4. Multi-class Accuracy after Feature Reduction

the Random Forest can hit. Now that we've seen the performance of both regression and classification models over a various number of features, we can pinpoint which specific features are most correlated with blood pressure. We see that among the algorithms, S_1, S_2, S_3, and S_4 are all commonalities, indicating that these features likely contribute highly to Blood Pressure.

5.3 Comparison & Discussion

Our results demonstrate a drastic improvement in the accuracy of PPG-based blood pressure estimation, making it a useful tool for self-monitoring. While it may not be feasible to use photoplethysmography (PPG) to estimate blood pressure without any calibration for everyone, this technique can be used as a quick and easy screening tool for hypertension in certain populations.

Neural Network Regression (SBP)	DT, $DW10$, $\frac{SW10}{DW10}$, $DW25$, $\frac{SW66}{DW66}$, I_H, I_L, $S1$, $S2$, $S3$
Neural Network Regression (DBP)	ST, $DW10$, $SW10 + DW10$, $\frac{SW10}{DW10}$, I_H, I_L, $S1$, $S2$, $S3$, $LASI$
Neural Network Classification	$DW10$, $\frac{SW10}{DW10}$, $DW25$, I_H, I_L, $S1$, $S2$, $S3$, $S4$, $LASI$
Random Forest (same features selected in all cases)	ST, DT, I_H, I_L, $S1$, $S2$, $S3$, $S4$, AI, $LASI$

Table 5. Top 10 features for different algorithms

Author	Task	Acc.	MAE (DBP)	MAE (SBP)	Method
Thambiraj et al.	Reg.	N/A	9.51	13.53	RF
Slapničar et al.	Reg.	N/A	6.88	9.43	ResNet
Tazarv et al.	Reg.	N/A	3.70	2.02	NN
Sun et al.	Class.	93%	N/A	N/A	2D-CNN
Fuadah et al.	Class.	95%	N/A	N/A	CNN
Ours	Both	96%	3.139	4.767	NN
Ours	Both	99%	1.602	2.082	RF

Table 6. Comparison With Other Studies

A note of caution with the models developed is that the MIMIC and UCI dataset used in building the models come from a more homogeneous pool of patients, that is ICU patients. This may hinder future replicability of our results if the dataset used is derived from patient backgrounds that are much more heterogeneous (younger people, etc.).

Another thing of note that we can visually observe from the confusion matrices for both RF and NN is that prediction mistakes, though rare, yield "extreme"-valued predictions. For example, in Figure 5, 4 patients with low BP is predicted to be classified as Hypertension II. This compels future users of these models to caution against trusting the models' predictive results definitively.

6. CONCLUSION & FUTURE WORK

Our results show that our selected features are able to predict systolic and diastolic blood pressure more accurately than most existing models in the literature (see Table 6). Our Random Forest Regressor performs best for BP prediction, while both Random Forest and Feed-Forward Neural Networks perform strongly for BP Classification. We also showed that we can recreate similar high accuracy by selecting significantly fewer features. Our results provide the first analysis of specific PPG features that contribute to BP (that we know of).

There are several improvements and methods we aim to incorporate in the future. Deep Multitasking Network Regression are Neural Networks that predict more than one value at a time. They work by optimizing a loss function that combines the loss for each prediction class. Implementing Multitask may yield more accurate results if the features in our data are highly correlated with each other (which they seem to be). We may also consider implementing a CNN-BiLSTM network on our time series data; the goal here is for the CNN to extract temporal features from the data, that the Bidirectional LSTM can use to make predictions. This method may also be useful for *continuous* blood pressure monitoring, that is, predicting the pulse wave for Arterial Blood Pressure.

7. CONTRIBUTIONS

Justin worked on Classical ML Models (LR, RR, SVR), Random Forest, LSTM Models, dataset selection, and the write-up.

Ahsan worked on data processing, feature extraction, Neural Network Models, feature selection algorithms, result processing, helper code, write-up.

Samuel worked on literature review, dataset selection, evaluation of feature selection algorithms, Softmax Regression modelling, and the write-up.

All experiments and code can be found at:
https://github.com/ahsan1578/BP-Prediction-From-PPG

8. REFERENCES

[1] D. S. Sheps, J. C. Ernst, F. W. Briese, and R. J. Myerburg, "Exercise-induced increase in diastolic pressure: indicator of severe coronary artery disease," *The American journal of cardiology*, vol. 43, no. 4, pp. 708–712, 1979.

[2] G. Thambiraj, U. Gandhi, U. Mangalanathan, V. J. M. Jose, and M. Anand, "Investigation on the effect of womersley number, ecg and ppg features for cuff less blood pressure estimation using machine learning," *Biomedical Signal Processing and Control*, vol. 60, p. 101942, 2020.

[3] A. Sahoo, P. Manimegalai, and K. Thanushkodi, "Wavelet based pulse rate and blood pressure estimation system from ecg and ppg signals," in *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*. IEEE, 2011, pp. 285–289.

[4] G. Slapničar, N. Mlakar, and M. Luštrek, "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, vol. 19, no. 15, p. 3420, 2019.

[5] A. Tazarv and M. Levorato, "A deep learning approach to predict blood pressure from ppg signals," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 5658–5662.

[6] F. Schrumpf, P. Frenzel, C. Aust, G. Osterhoff, and M. Fuchs, "Assessment of non-invasive blood pressure prediction from ppg and rppg signals using deep learning," *Sensors*, vol. 21, no. 18, p. 6022, 2021.

[7] Y. N. Fuadah and K. M. Lim, "Classification of blood pressure levels based on photoplethysmogram and electrocardiogram signals with a concatenated convolutional neural network," *Diagnostics*, vol. 12, no. 11, p. 2886, 2022.

[8] J. Wu, H. Liang, C. Ding, X. Huang, J. Huang, and Q. Peng, "Improving the accuracy in classification of blood pressure from photoplethysmography using continuous wavelet transform and deep learning," *International journal of hypertension*, vol. 2021, 2021.

[9] X. Sun, L. Zhou, S. Chang, and Z. Liu, "Using cnn and hht to predict blood pressure level based on photoplethysmography and its derivatives," *Biosensors*, vol. 11, no. 4, p. 120, 2021.

[10] C.-T. Yen, S.-N. Chang, and C.-H. Liao, "Deep learning algorithm evaluation of hypertension classification in less photoplethysmography signals conditions," *Measurement and Control*, vol. 54, no. 3-4, pp. 439–445, 2021.

[11] H. Wang, "Random forest based blood pressure prediction model from ecg and ppg signal," in *2022 12th International Conference on Bioscience, Biochemistry and Bioinformatics*, 2022, pp. 1–6.

[12] Y. Kurylyak, F. Lamonaca, and D. Grimaldi, "A neural network-based method for continuous blood pressure estimation from a ppg signal," in *2013 IEEE International instrumentation and measurement technology conference (I2MTC)*. IEEE, 2013, pp. 280–283.

[13] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless blood pressure estimation algorithms for continuous health-care monitoring," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 4, pp. 859–869, 2017.

[14] C. El-Hajj and P. Kyriacou, "A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure," *Biomedical Signal Processing and Control*, vol. 58, p. 101870, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809420300264

[15] M. Awad and R. Khanna, "Support vector regression," in *Efficient learning machines*. Springer, 2015, pp. 67–80.

[16] H. Mou and J. Yu, "Cnn-lstm prediction method for blood pressure based on pulse wave," *Electronics*, vol. 10, no. 14, p. 1664, 2021.

[17] T. K. Kim, "Understanding one-way anova using conceptual figures," *Korean journal of anesthesiology*, vol. 70, no. 1, pp. 22–26, 2017.

[18] S. Janitza, G. Tutz, and A.-L. Boulesteix, "Random forest for ordinal responses: prediction and variable selection," *Computational Statistics & Data Analysis*, vol. 96, pp. 57–73, 2016.

[19] H. Han, X. Guo, and H. Yu, "Variable selection using mean decrease accuracy and mean decrease gini based on random forest," in *2016 7th ieee international conference on software engineering and service science (icsess)*. IEEE, 2016, pp. 219–224.

[20] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," *Advances in neural information processing systems*, vol. 26, 2013.