# Machine Learning-based Estimation of Cardiac Contractility from Peripheral Pressure Waveform

Machine Learning - Project 2

Marija Lazaroska, Deborah Scherrer Ma, Méline Zhao
Machine Learning TA. Semih Günel, Laboratory Supervising TA. Vasiliki Bikia
Hosting Laboratory: Laboratory of Hemodynamics and Cardiovascular Technology
*EPFL Lausanne, Switzerland*

***Abstract*—Machine learning is now a well embedded method for scientific data analysis and also various types of predictions. According to the World Health Organization, cardiovascular diseases (CVD) are the leading cause of death in the world and are held responsible for more than 30$\%$ of deaths worldwide per year [1]. Part of those deaths occur prematurely and could be avoided if high risk factors were identified early enough to provide suitable treatments. Bikia et al. recently investigated the role that machine learning could play in predicting CVD's risk factors in the population based on non-invasive pressure-based inputs, which can be easily acquired in the clinic. Various machine learning algorithms have been applied to several cardiac and aortic pressure measurements to predict aortic systolic pressure (aSBP), cardiac output (CO) as well as end-systolic elastance ($E_{es}$) and exhibited encouraging results for further studies on enhancing the predictions' accuracy [3]. In the present project, we investigate how $E_{es}$'s prediction can be improved without the use of Ejection Fraction (EF), a parameter which has been often used by prior methods to estimate $E_{es}$. To this end, we have explored the potential in accurately estimating $E_{es}$ by leveraging non-invasive features extracted from the entire peripheral (brachial) blood pressure wave. We tested various regression algorithms and compared their performance to the one of Convolutional Neural Network (CNN).**

## I. Introduction

As described in the abstract, CVDs are nowadays of great concern and thus assessing someone's risk to develop them is of great stake. Bikia et al. applied Random Forest, Support Vector (SVR), Ridge and Gradient Boosting Regression to predict aSBP, CO and $E_{es}$, using non-invasive readily available clinical measurements. They obtained good accuracies, namely a regression coefficient of no less than 92$\%$ was reached for $E_{es}$ when using SVR with EF included as input. This result is to be put in comparison with an r value of 37$\%$, which is the best under lack of EF values as feature, obtained using Ridge Regression[3].

These results highlight the high sensitivity of $E_{es}$'s prediction to EF data as input. As a matter of fact, EF is a feature which is often acquired invasively through cardiac catheterization, for instance. Imaging measurements as MRI allow to obtain EF non-invasively. However, given that EF depends on End-Diastolic Volume (EDV), accurate interpretation of EF is possible only with additional measurements of EDV. The stake of using machine learning to predict aSBP, CO and $E_{es}$ is to allow a non-invasive and clinically translatable approach for CVDs risk factors assessment. In this project, we focus on improving the $E_{es}$ predictions using only brachial blood pressure (brBP) waveforms data. Based on Bikia et al., we started by implementing regression functions to build reference values.

The initially provided dataset contains the following features:

- Brachial Systolic Blood Pressure (brSBP)
- Brachial Diastolic Blood Pressure (brDBP)
- Heart Rate (HR)
- Carotid-to-Femoral Pulse Wave Velocity (cfPWV)
- Ejection Fraction (EF)

Those feature values along with the true values of $E_{es}$ were given for n = 4,018 individuals. Beside these previously cited static features, brBP recording (waveform) over time was also made available. All data were obtained *in silico* using a validated model[3]. They were firstly used for training the regression models in the first part of our project.

The cfPWV is defined as the propagation time of the pulse wave from the carotid to the femoral arteries, and constitutes a measure of arterial stiffness. Regarding EF, its accurate evaluation can be performed through the techniques stated before, which are yet considered rather costly. To overcome these limitations, we explored the entire brBP waveform to look for relevant features which could help in effectively predicting $E_{es}$ and replace EF and cfPWV.

Moreover, knowing that hypertension, i.e. persistent raised blood pressure [2], is a predisposition for CVDs, we also split the data into Hypertensive and Normotensive subsets. Hence, to assess the relevance of splitting the data and of the pressure-related features, we applied regression algorithms and observed whether potential improvements could be brought by them. Eventually, we implemented CNN with the whole waveform data, to compare with the manually extracted features.

## II. Regression Models

### A. Data Pre-processing

Instead of using the whole set of data, containing all five aforementioned features, for training the models, as done in the paper[3], we split it into 80$\%$ of training set and 20$\%$ of testing set. In order to proceed to regression using different algorithms, we standardized the used train and test features using MinMaxScaling[9], according to the data pre-processing step performed by Bikia et al.

### B. Methods

After scaling all our input data, we applied Random Forest[6], Support Vector[10], Ridge Regression[7] and Gradient Boosting Regression[5]. To assess the quality of the regression algorithm, for each iteration of the 10-fold cross-validation performed on the training set, we proceeded to a 10-fold internal cross-validation using the Grid Search method[8] to find the best-fit hyperparameters. The accuracy of the predictions was eventually assessed by computing the correlation coefficient (r), the coefficient of determination ($R^2$), the root-mean-square error (RMSE) and the mean absolute error (MAE) on the testing set.

### C. Results

The best accuracies for each of the method along with the corresponding tuned hyperparameters can be observed in **Table I** and **Table II**, respectively. Surprisingly, the highest accuracy was obtained using Gradient Boosting Regression as opposed to SVR for Bikia et al. team's results. This discrepancy could be explained by the train-test split of our data, which is assumed to affect the choice of the best-fit hyperparameters of the models. As a matter of fact, Gradient Boosting provided a r score, between predicted $E_{es}$ and their true values, of 95% as compared to 90% for SVR.

TABLE I: Results of the 4 regression models used for predicting the $E_{es}$. RMSE and MAE are given in the same units as the output $E_{es}$, [mmHg/mL]

| Model | r score | $R^2$ score | RMSE | MAE |
|---|---|---|---|---|
| Random Forest | 0.94 | 0.89 | 0.027 | 0.020 |
| SVR | 0.90 | 0.80 | 0.037 | 0.029 |
| Ridge Regression | 0.89 | 0.80 | 0.037 | 0.029 |
| Gradient Boosting | 0.95 | 0.91 | 0.025 | 0.020 |

TABLE II: List of hyperparamters' values for each regression method, used to obtain the results of **Table I**

| Models | Hyperparameters | Values |
|---|---|---|
| Random Forest | max_depth | 20 |
| | n_estimators | 1000 |
| SVR | C | 10 |
| | gamma | 1 |
| Ridge Regression | alpha | 0.1 |
| Gradient Boosting | learning_rate | 0.1 |
| | n_estimators | 1000 |

## III. Exploitation of Brachial Blood Pressure Waveform

### A. Features Extraction & Pre-processing

As previously emphasized, the stake of this project is to find non-invasive features which can substitute EF values in predicting $E_{es}$ values. Thus, we define the Control feature group as having the same features from the regression models in the first part, but without EF. We also evaluated the possibility of not using the feature cfPWV, given that it requires additional measurements to be obtained.

We first explored the brBP waveforms to seek for relevant features to predict $E_{es}$. Some features used before remain, namely brSBP, brDBP and HR. From the waveform, we also added the features brachial Pulse Preassure (brPP = brSBP-brDBP) and Mean Arterial Pressure (MAP). These features formed our Group 1 dataset.

As awaited, the dynamical behavior of the blood pressure is a consequence of the mechanical properties of the blood vessels, which can partly be underlined by its value of $E_{es}$. Considering this dependency, we opted for computing the values of the derivatives of the waveform. More specifically, we extracted the maximum, minimum values of the derivatives and also the time point at which they did occur. These, along with Group 1 features, formed our Group 2 dataset. Hence looking upon the derivative-related features, the following ones were extracted for further regression:

- Maximum of the derivative (max (dP/dt))
- Minimum of the derivative (min (dP/dt))
- Derivative's maximum timepoint
- Derivative's minimum timepoint

In this extend, we also investigated integral-related features and namely the total area under the blood pressure waveform curve. The addition of this feature to the Group 2 formed our Group 3 dataset.

- Total area

A representation of the features can be observed in **Figure 1**

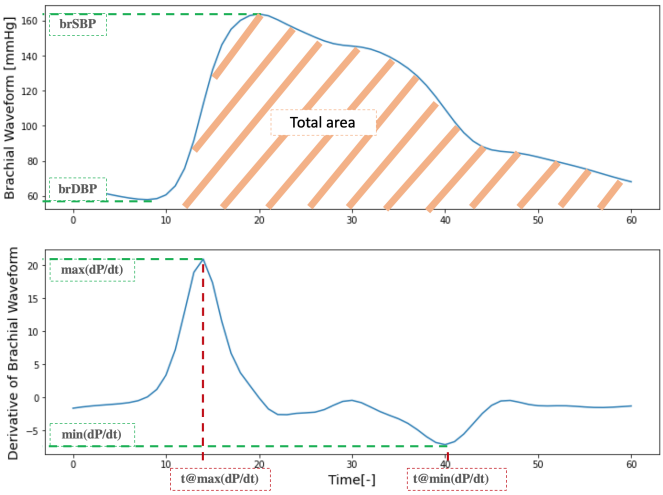


Fig. 1: Example of brachial waveform (top) and its derivative (bottom), along with graphical representations of the used features.

Beside adding these new features, we also seek for a mean of splitting the whole dataset. It seems thus natural to us to split the individuals according to their extremum pressure values[4]. With regard to that, we divided the 4,018 data into:

- Hypertensive Group (if brSBP $>$ 135 mmHg and brDBP $>$ 85 mmHg)
- Normotensive Group

Finally, these new features are split into training and testing set and normalized using MinMaxScaling.

### B. Method

Following the results of the Regression Models part of the project, the choice was made to apply Gradient Boosting Regression to these new sets of features and to observe the performances. Regression was applied to the whole dataset, as well as to the 2 separated groups, Hypertensive and Normotensive, for each of the four different features groups. Similar to the first part of the project, the data were split into training and testing set, in a 80%-20% ratio. Eventually, we prevented overfitting behavior of the regressor by performing 10-fold cross-validation on the training set. The presented results were obtained from the testing set.

### C. Results

The results are summarized in **Table III**. Between the Control and Group 1, we do not observe a notable difference of r and $R^2$ values, which shows that the cfPWV feature can indeed be replaced with no major losses. As hypothesized, the derivative features provided much improvement in the accuracies, where we see a change in the r score, going from around 30% to around 90%. The Group 3, which adds the total area feature, shows a slight improvement of r and $R^2$ scores compared to Group 2.

It is worth noticing that these results are subjected to variance. Thus, a simple variance analysis of the r values was made for each of the 3 splits, as shown in **Figure 2**.

TABLE III: Gradient Boosting Regression results for split data groups *All*, *Hypertensive* and *Normotensive* with different input features *Control*, *Group 1*, *Group 2* and *Group 3*, as described in III-A. RMSE and MAE's unit are given in [mmHg/mL].

| Group | Split | r | $R^2$ | RMSE | MAE |
|---|---|---|---|---|---|
| Control | All | 0.3929 | 0.1438 | 0.0788 | 0.0610 |
| | Hypertensive | 0.1807 | 0.0249 | 0.0681 | 0.0529 |
| | Normotensive | 0.3702 | 0.1276 | 0.0860 | 0.0659 |
| Group 1 | All | 0.3742 | 0.1355 | 0.0804 | 0.0626 |
| | Hypertensive | 0.1704 | 0.0081 | 0.0675 | 0.0531 |
| | Normotensive | 0.3504 | 0.1199 | 0.0811 | 0.0636 |
| Group 2 | All | 0.9033 | 0.7980 | 0.0410 | 0.0295 |
| | Hypertensive | 0.9233 | 0.8399 | 0.02710 | 0.0193 |
| | Normotensive | 0.8917 | 0.7940 | 0.0377 | 0.0281 |
| Group 3 | All | 0.9110 | 0.8190 | 0.0382 | 0.0267 |
| | Hypertensive | 0.9238 | 0.8479 | 0.0247 | 0.0177 |
| | Normotensive | 0.8978 | 0.8011 | 0.0402 | 0.0294 |

Interestingly, we observe a very poor accuracy for the Hypertensive split using Control and Group 1 features compared to the Normotensive split. Nevertheless, after adding extra features, the Hypertensive split shows the best accuracies for Groups 2 and 3. This confirms the well-established physiological fact that the waveform's dynamic of hypertensive people is different from normotensive people.

The hyperparameters were found manually among the following values: learning_rate = {0.001, 0.05, 0.01, 0.1} and n_estimators = {1000, 1500, 1750, 2000}. The best results were obtained using the values from **Table IV**.

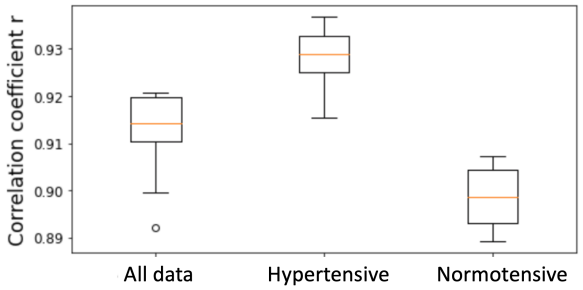


Fig. 2: r coefficient boxplot for Group 3 and $n = 10$ runs.

TABLE IV: Hyperparameter's values used to obtain the results of Table III

| Features Group | learning_rate | n_estimators |
|---|---|---|
| Control | 0.001 | 1750 |
| Group 1 | 0.001 | 1750 |
| Group 2 | 0.05 | 1750 |
| Group 3 | 0.05 | 2000 |

## IV. Implementation of Neural Network

### A. Methods

In order to investigate potential improvement in the method, we implemented a 1D CNN. The reason we chose CNN was because of its ability to learn and automatically extract features from raw input data, the brBP waveforms in our case.

Our CNN consists of 4 convolutional layers and one Max-Pooling layer. We also use a Flatten layer followed by a Linear layer. After the first 3 convolutional layers, we used an activation ReLU layer (**Table V**). We also tried using a Dropout layer to avoid overfitting the data. However, the model was performing better without it and by observing the loss there was no overfitting, therefore this layer was omitted.

TABLE V: CNN model architecture

| |
|---|
| 1D Conv Layer |
| ReLU Layer |
| 1D Conv Layer |
| ReLU Layer |
| 1D Conv Layer |
| ReLU Layer |
| 1D Conv Layer |
| Max Pooling Layer |
| Flatten Layer |
| Linear Layer |

Firstly, we trained our 1D CNN using input data with only one channel, meaning we used only the raw time series from the brBP waveforms as features.

From the aforementioned analysis on the waveforms we concluded that the derivatives of the waveforms are good predictors for the $E_{es}$. Thus, we decided to train a CNN with a second channel which includes the derivatives. As input data, we used both the raw time series and the derivatives.

In the waveform analysis, the total area between the two minimum points under the waveform also showed good results

in predicting $E_{es}$, hence we tried using it as well as an input data. Therefore, in addition to the 2 channels input data, we added the area as a third channel. The area had a (samples_size, 1) dimension which didn't match with the dimension (samples_size, 80) of the 2 channels. So we proceeded with creating a (samples_size, 80) dimension for the third channel by repeating the area value 80 times.

The CNN model was trained using 300 epochs and after each epoch we observed the correlation coefficient r, the $R^2$ score and the MSE loss on the testing set. Moreover, we used the Adam optimizer. For selecting the best learning rate as well as the batch size used in each epoch, we performed hyperparameter tuning. The selected hyperparameters are presented in **Table VI**.

TABLE VI: Best hyperparameter's values, obtained using Adam optimizer, for 1D CNN with different numbers of channels

| | **batch_size** | **learning_rate** |
|---|---|---|
| ConvNN with 1 channel | 32 | 0.001 |
| ConvNN with 2 channels | 32 | 0.001 |
| ConvNN with 3 channels | 32 | 0.001 |

### *B. Results*

The first CNN model we trained using one channel input performed well, giving good results, similar as those obtained in the research paper when using EF features.

The second CNN model using the 2 channels input performed better than the first one and the regression models used in the paper by Bikia et al. (**Figure 3**). This confirmed that the derivatives of the waveforms are good predictors.

However, the third model using the 3 channels input showed worse results than ones of the CNN model using 2 channels. This suggests that including the total area as inputs is not a good estimator for the NN (**Table VII**).

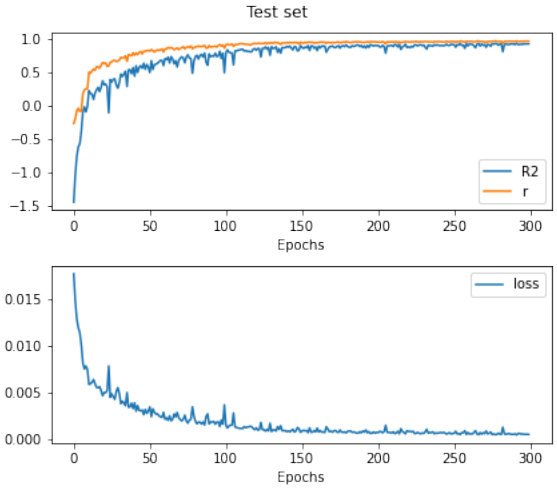


Fig. 3: Epochs vs R2 score and r correlation coefficient *(top)* and loss *(bottom)* on testing set for CNN model using 2 channels

TABLE VII: Results using the CNN

| | **r score** | **$R^2$ score** | **MSE** |
|---|---|---|---|
| ConvNN with 1 channel | 0.79 | 0.90 | 0.001 |
| ConvNN with 2 channels | 0.96 | 0.92 | 0.00049 |
| ConvNN with 3 channels | 0.88 | 0.75 | 0.001 |

## V. Discussion and Outlook

In this project, we achieved improvements in the $E_{es}$ predictions without the use of the EF and cfPWV features. Indeed, as the results section of this report suggests, it is possible to reach reasonably high accuracies and even improve them by only exploiting features contained in the brBP waveform. This observation is of great interest, given that the brBP wave can be easily acquired in the clinic by simple non-invasive means of measurement.

Beside the waveform features we've retrieved, other features could have been interesting to extract as predictors. Unfortunately, some extra features which could be used for regression analysis were not possible to be obtained from the dataset. Namely, the dicrotic notch, which is a mark of the end of the systole and beginning of the diastole, was not visible in all the waveforms. Hence, all features which depended on the dicrotic notch, such as the time at the dicrotic notch, the systolic and the diastolic areas, were not withdrawn from the dataset. In addition to the used features, the following ones were considered: (i) upstroke systolic area, which is the area from the beginning of systole until brSBP, (ii) inverse of the upstroke systolic area, (iii) upstroke systolic area / brPP ratio, (iv) mean of the derivative. These features showed no significant results' improvements, and thus were not used for the final predictions. Interestingly, the CNN with 2 channels showed better results than Gradient Boosting, suggesting that the brBP waveform and its derivative values could contain other important features not used for the regression.

Considering the heterogeneity in a random sample of the population, we also investigated the difference in the predictions for individuals with two cardiovascular conditions, namely, hypertensive and normotensive cases. This data split highlighted the fact that a same feature can have different weight and significance in predicting $E_{es}$ for the corresponding subgroups. This aspect was not our project's main point, but we do think that by addressing an increased concern to this background diversity while assessing risk factors, it is possible to enhance the predictions' accuracy.

For time concern, we did not fully explore hyperparameters tuning possibilities. However, one should consider that hyperparameters' optimization can be leveraged to increase predictions' accuracies.

Finally, as exploratory analysis of the models, the results obtained are promising. Eventually, these models should be applied in values taken *in vivo* to validate the predictions. Also higher-sampling-frequency acquired data are susceptible to provide additional information and thus enable extraction of extra features, which could of help for regression analysis.

## REFERENCES


[1] URL: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1.

[2] URL: https://www.who.int/westernpacific/health-topics/hypertension.

[3] Vasiliki Bikia et al. "Noninvasive estimation of aortic hemodynamics and cardiac contractility using machine learning". In: *Scientific Reports* 10.11 (2020), p. 15015. ISSN: 2045-2322. DOI: 10.1038/s41598-020-72147-8.

[4] Niels Gobin et al. "Mesure ambulatoire de la pression artérielle sur 24 heures". In: *Forum Médical Suisse – Swiss Medical Forum* 12.3132 (2012). ISSN: 1661-6146, 1661-6138. DOI: 10.4414/fms.2012.01093. URL: https://doi.emh.ch/fms.2012.01093.

[5] *sklearn.ensemble.GradientBoostingRegressor¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html#sklearn.ensemble.GradientBoostingRegressor.

[6] *sklearn.ensemble.RandomForestRegressor¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

[7] *sklearn.linear_model.Ridge¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html.

[8] *sklearn.model_selection.GridSearchCV¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[9] *sklearn.preprocessing.MinMaxScaler¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[10] *sklearn.svm.SVR¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html.