Predicting Blood Pressure using Features of PPG Waveforms

Franklin Zhu

July 3, 2022

Abstract

Blood pressure is a key indicator of numerous medical conditions, leading it to be a component that needs to be constantly monitored. However, there are limitations to using common blood pressure measuring devices, one being that they can only be used at rest. A possible solution to this is using photoplethysmogram (PPG) and arterial blood pressure waveforms (ABP) to estimate a patient's blood pressure. Machine learning techniques, such as Deep RNN and ResNet have already been used to predict blood pressure in this manner with mean absolute errors of 8.54 and 9.43 respectively for systolic blood pressure (SBP). Therefore, I have chosen to use the Random Forest technique to analyze the same processed MIMIC-III waveform data in an effort to compare the results of unimplemented machine learning techniques in blood pressure analysis to already tested ones.

1 Introduction

With the rise in mortality rates of cardiovascular diseases [4], it is becoming more and more important to diagnose them as early as possible. One of these key conditions is hypertension, which is defined as either systolic (SBP) or diastolic blood pressure (DBP) being consistently higher than normal [1]. Using blood pressure to notice various stages of hypertension is critical for patients, as this allows them to get proper treatment to recover. Blood pressure is known to fluctuate due to numerous factors, including age, so the early identification of problems can help patients live healthy lifestyles and avoid drug treatment [2].

As noted previously, the most common measuring tool of blood pressure, the electronic sphygmomanometer, requires cuffs and forearm pressure to measure blood pressure [9]. There are many variables that can affect this reading, such as stress, posture, and operation of the machine. Due to this, some patients have a higher blood pressure reading in medical settings compared to at home. Since the diagnosis of hypertension prefers consistent readings, this cuff technique can lead to uncertainties of the state of a patient's blood pressure.

One of the best alternate solution to measuring blood pressure without cuffs right now is through the deep learning analysis of the features of PPG. PPG can be easily measured with a pulse oximeter, making it convenient to obtain and consistently measure [11]. The pulse oximeter shines light, usually infrared, through a patients fingertip, and measures the changes in light absorption to generate a PPG. The pulse oximeter can be placed on one's index finger and can be transported smoothly, allowing patients to take measurements anywhere. This allows for possible inspection of blood pressure while sleeping, or during other states that can help better diagnose cardiovascular issues.

Currently, there are many features of the PPG signal that have been shown to be correlated to blood pressure. These include crest time (CT), the velocity plethsmogram (VPG), the acceleration plethysmogram (APG), etc [7]. By processing certain features, a relationship can be formed with blood pressure and this can be used to predict blood pressure with machine learning.

2 MIMIC-III PPG Waveform Processing

This study uses waveforms from the MIMIC III waveform database matched subset [5] [8] [6], which can be found at https://physionet.org/content/mimic3wdb-matched/1.0/. Each patient can have

several signals with different types of waves, ranging from seconds to hours. This data is processed and filtered to obtain PPG and ABP signals, which is used to calculate necessary features of both waveforms. Then, the data is fed as input to train a Random Forest to use certain features of PPG signals to predict blood pressure.

2.1 Processing Raw Data

Since the MIMIC-III database is extremely large (a few terabytes in size), I first downloaded the data to an external hard drive. The MIMIC-III waveform data for each patient is separated into many .dat (data) and .hea (header) files, with the .dat files containing segments of various types of signals. An overall layout.hea file specified which .dat files contained which types of data and other useful information, such as the sampling frequency, gain , etc. The data in these .dat files is in binary and nothing meaningful could be extracted from them directly. Therefore, I ended up using wfdb2mat, a command from the WFDB software package that creates a .mat (MATLAB) and descriptor .hea file from a .hea file and several .dat files, to convert the .dat files into .mat files. To use this software, I downloaded the source code from https://archive.physionet.org/physiotools/wfdb-windows-quick-start.shtml and compiled the program with gcc. Although .mat files also cannot be opened directly, there is software available to load information from .mat files, allowing me to read the signal data. So, I implemented a script in python using Cygwin (a Linux terminal for Windows machines) to convert all the patient data into the .mat file format and then read signal information from each file.

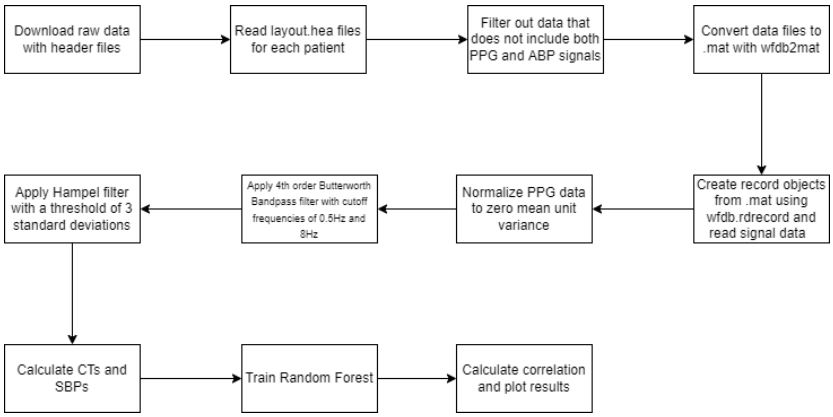

Figure 1: Flowchart of the experiment

Before I began visually inspecting the samples and deciding which filters to apply, I removed patient files that did not contain both PPG and ABP waveform data. To do this, I used information specified in the .hea files generated from wfdb2mat. As demonstrated by Figure 2, each line in the .hea file specifies which file contains which waveform, and properties of each waveform. Since I combined all the signals into one .mat file, the file names were all the same. I ignored the rest of the information which were in the middle columns, as this specified specific properties that I did not need to process the data. In the right-most column, the type of signal was specified. In the example shown in Figure 2, the number II means a certain type of electrocardiogram (ECG) signal, which is not necessary data. However, the .hea also notes that the .mat file contains PLETH and ABP signals, which are the PPG and ABP data. Since both of these waves are necessary to train the random forest, I used a python script that read through each .hea file and looked for those that contained both PLETH and ABP to filter out patients that did not have both ABP and PPG recordings.

Once I finished this, I plotted random segments of around 10,000 points for random patients to check for common problems that the signals might have. To get this data from the .mat files, I first attempted to use the loadmat function from scipy.io, but after directly extracting the data into numpy arrays and plotting them, I noticed that the plots were extremely irregular. This problem occurs because loadmat only retrieves the raw data points, which do not take into account the baseline, frequency, gain, etc. specified in the .hea file. Instead of writing a new method to read the .hea file and scale the data,

I decided to use the wfdb.rdrecord method from the WFDB software package to increase processing efficiency. The rdrecord method creates a record object with data that has already been adjusted to the specifications in the header file, so by reading information from each record, I was able to plot waveforms to visually examine them.

```
p087119-2156-01-04-17-25m 3 125 38797500 17:25:55 04/01/2156
p087119-2156-01-04-17-25m.mat 16+192 127(-64)/mV 8 0 -32768 -25071 0 II
p087119-2156-01-04-17-25m.mat 16+192 255(-128)/NU 8 0 -32768 -7335 0 PLETH
p087119-2156-01-04-17-25m.mat 16+192 4.825(-410)/mmHg 10 0 -32768 31134 0 ABP
# Location: tsicu
#Creator: wfdb2mat
#Source: record p087119-2156-01-04-17-25  Start: [17:25:55.000 04/01/2156]
```

Figure 2: Example of a .hea file generated from wfdb2mat

After examining the data, I noticed many sections with high frequency noise and some outliers. Therefore, I decided to filter each waveform to reduce noise and check for sensor errors. I tried to use the same methods as described by a previous study [10]. First, I normalized the PPG signal to zero mean unit variance. Then, I applied a 4th-order Butterworth band-pass filter to smooth out data below 0.5Hz and above 8Hz, as these areas can be attributed to noise. Afterwards, I applied a Hampel filter to reduce outliers. Although I attempted to use the same data processing as the previously stated paper, I did not obtain similar results. This leads me to beleive that the differing results could be attributed to possibly the parameters used for the filters and the way they were implemented.

2.2 Butterworth Bandpass Filter

The Butterworth Bandpass is a filter used to process signals and produce a frequency response that is as flat as possible. It is also known as a "brick wall" filter and the higher the order of the filter is, the closer it gets to the "brick wall" response, as shown in Figure 3 [3]. Overall, this allows the filter to remove ripples in waveform data, smoothing out high-frequency noise in PPG waveform files. To implement the Butterworth filter, I used the pre-existing butter function from scipy.sig, but after applying it to the data, it still seemed like high-frequency noise was not reduced by visual inspection. The filter did not produce a large change in the PPG waveforms, and I am not sure why it was not as effective as expected.

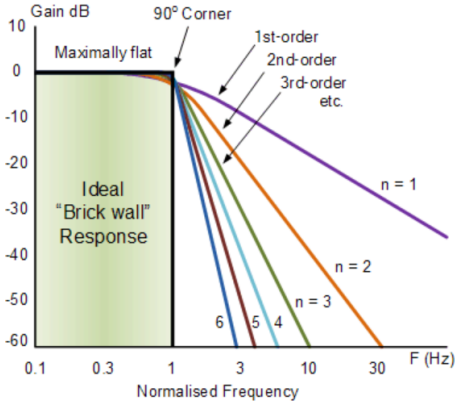

Figure 3: Different orders of Butterworth filters

2.3 Hampel Filter

The Hampel filter is a filter that replaces outliers with reasonable values by using a sliding window with a rolling median to identify outliers. For each point in the signal, a median and standard deviation

are calculated with points in the same window. If the point is more than a threshold number of standard deviations away from the median, the Hampel filter replaces this value with either a calculated reasonable value, or just the median of the window. This should have helped generally reduce the outliers and erroneous measurements that are found in a lot of the data, but it did not help fix the more important problems in the signals. The main problem of the PPG data were segments that just had low peaks. These peaks lasted for long segments of some part of the signal, and since the Hampel filter only looks at a window of data to remove outliers, it did not work well to change these sections.

Even though the noise is reduced, there are still problems in some of the waveforms. A previous study [10] found that some ABP waveforms had flat lines and flat peaks. After I examined the data, I did notice many flat lines, as not every signal lasts the full length of the recording, but I did not notice any flat peaks. The other main problem that I found was continuous low peaks for ABP and PPG waveforms, which are essentially the same as just having a flat line.

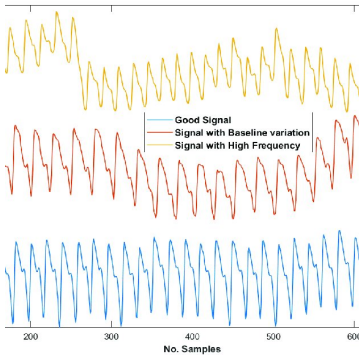

Figure 4: Different types of noise in PPG waveforms

One problem found in many signals was that flat lines appear for long periods of time in the some of the ABP samples. This could be caused by sensor failure, or the removal of the sensor, so these areas become useless and should be ignored.

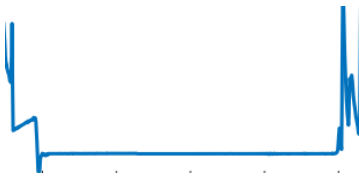

Figure 5: Flat lines in ABP waveforms

Another problem is that some ABP and PPG waveforms have a lot of small peaks that continue for long periods of time. These small peaks are almost like flat lines, and I believe this could be a result of sensor detachment or just problems in the sensor sensing something even though it is not attached. Due to this, these small peaks also cannot be used and should be ignored.

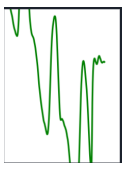

Figure 6: Small peaks in ABP waveforms

2.4 Solution

Since the filters applied did not remove all undesirable parts of the PPG signal, I was not able to apply them to every file and obtain useable information. Consequently, the only way for me to find which segments of data had normal peaks was by visual inspection. I ended up randomly examining around 10,000 points per patient for several patients without filtering the waveforms before selecting 8 patients that had relatively good PPG and ABP signals. I then used the PPG and ABP data for these patients to attempt to find a correlation between PPG data and the SBP.

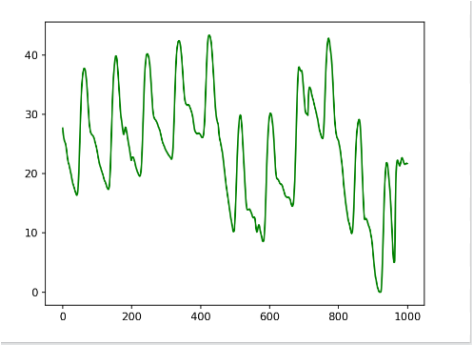

Figure 7: PPG data with too much noise even after filtering

3 Features of PPG Waveforms

PPG waveforms contain various features that are correlated to many different measurements useful for different purposes. The various key features are shown below in Figure 8. After examining a previous study about the top ten features correlated with systolic blood pressure [7], I decided to take a look at the five most correlated features and chose one that seemed the most intuitive.

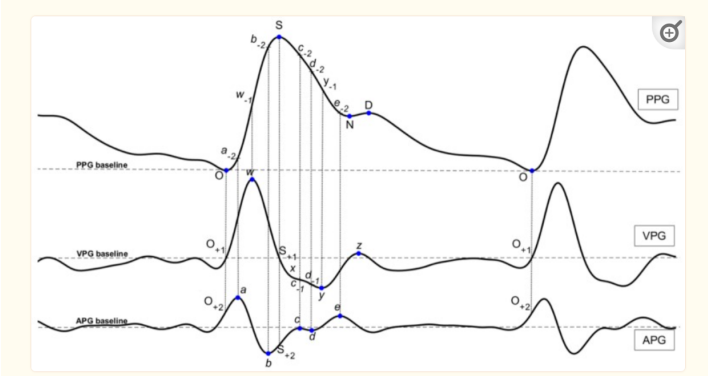

Figure 8: Characteristics in PPG and its derivatives, taken from [7]

3.1 Example Features

Some of the most correlated features to blood pressure as discovered by a previous study [7], are (b - c - d) / a, CT, b_{-2}/S, etc (letters correspond to the graph shown in Figure 8). The study also calculated the correlation coefficient of each feature, with the coefficients being 0.6903, 0.6164, and -0.6353 respectively. However, even though CT was not the most correlated component to SBP, it seemed like the most intuitive and convenient method, so I decided to first use CT to predict SBP.

3.2 Calculating Crest Time

Crest time (CT) is defined as the time from the foot of the PPG waveform to its systolic peak. I calculated CT by first finding the peaks of the PPG waveform. To do this, I tracked when the signal was increasing and decreasing and marked the points that the signal changed from increasing to decreasing. Since some of the PPG waveforms could have smaller peaks near the actual largest peak, I used the frequency of the samples (specified by MIMIC-III database as 125Hz) to check for peaks that were too close to each other. Then, I would replace these peaks with the largest peak, removing extra lower peaks that may have been detected. After finding the peaks of the PPG waveform, I found each foot by looking for the minimum value between two PPG peaks. Once each peak and its corresponding foot was found, I simply calculated the crest time by subtracting the location of the peak from the location of the foot.

Besides calculating the crest time, the SBP must also be calculated for each crest time. As shown by Figure 9, the ABP waveform peaks in alternating intervals with the PPG waveform peaks. Therefore, to find the SBP corresponding to each CT, I looked for the maximum value of the ABP signal within 2 PPG peaks. Once this was complete, I was able to pass in the CTs and corresponding SBPs to the Random Forest algorithm to train decision trees.

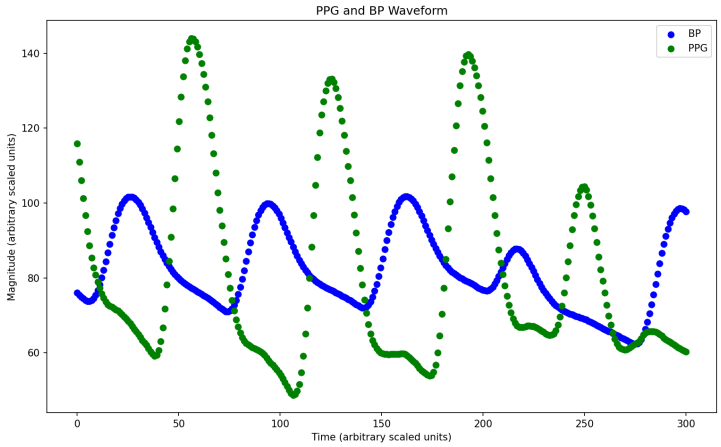

Figure 9: Plot of alternating peaks of ABP and PPG

4 Code

The code I developed used to process and feed the data into a random forest is available at https://github.com/FrZhu22/Blood-Pressure-Prediction-with-Random-Forest

4.1 Random Forest

The algorithm I chose to evaluate PPG CTs with is random forest, a technique that produces many random trees with random leafs and combines the predictions of all the trees to produce a prediction. Starting with a decision tree, a random forest picks random features to split data at every node. This process is created to produce many differing trees, which are combined into a forest. With a large number of relatively uncorrelated trees, the random forest will outperform any individual tree. Using this "crowd intelligence," a random forest is able to better predict results.

4.2 Python Code

The python code is separated into many functions, with the five main ones displayer, findCT, peakPlot, findpeaks, and estimation. First, displayer is a method used for manual inspection of data. It takes in

the name of a patient record and an interval to display with the location of ABP and PPG waveforms as array indices, and displays the PPG signal along with the ABP signal in the interval. This is useful for manually checking the quality of data and possible errors that could occur. Next, findCT calculates the all the CTs in a PPG signal along with its corresponding SBPs given a PPG record and ABP record. The findCT method also calls findpeaks, which uses the same logic as previously described to locate the actual peaks of a signal. To make sure this method works correctly, peakPlot takes in a record and an array with the indices of the peaks in the record, and plots the signal while indicating where the calculated peaks are located. This makes it simple to debug problems and allows for visual inspection in case of errors. Lastly, the estimation method applies the Random Forest algorithm to the calculated CTs and SBPs. To implement the Random Forest in python, I used the pre-existing RandomForestRegressor from sklearn, which creates a Random Forest with a specified amount of decision trees and nodes. However, I did not have enough time to attempt to optimize the Random Forest predictions by altering these features, so I used the default number of 100 decision trees for my Random Forest fitting.

Unfortunately, since I only used visually inspected samples from a few patients, the correlation coefficient that I obtained between CT and SBP was 0.3525, which was not as strong as found by a previous paper [7] that found a correlation coefficient of 0.6164. This can be seen in Figure 10, which exemplifies the found relationship between CT and SBP. I believe this difference in correlation can be attributed to the methods used of filtering data. Since the mentioned paper applied different types of filters to PPG data and the filters I applied did not completely eliminate bad data, this could result in a lower correlation. Also, because I only used a small sample of random patients with normal waveforms, it could be possible that there would be a higher correlation factor with more patient data. Due to this lack of correlation, the RandomForestRegressor had trouble fitting and predicting SBPs with CTs, and did not produce meaningful results to compare to previous methods used to predict SBP.

Another possible way to improve the predictions done by the Random Forest would be to test other features of PPG signals that are more strongly correlated with SBP. As there are various important features, some of them may be better for predicting SBP with the MIMIC-III waveform data.

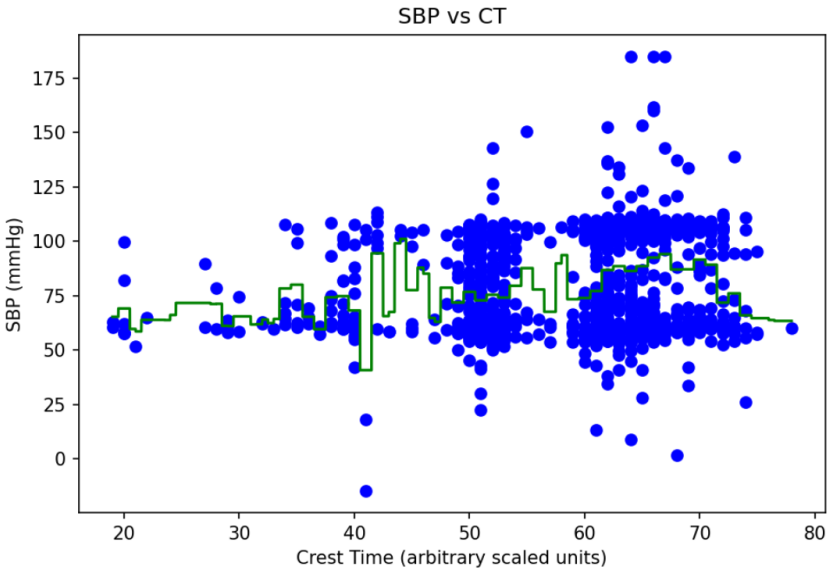

Figure 10: Inaccurate random forest fitting as a result of lack of data

5 Conclusion

Even though so much data is available online, the biggest challenge to implementing any machine learning algorithm to predict blood pressure is processing the data. Almost every waveform has prob-

lems with its peaks, or weird values during certain windows. Especially due to how long the waves are, it is also difficult to find out which problems a dataset might have. Deciding which filters to use along with how they should be configured also poses a challenge, as both over processing and under processing the signals can lead to outliers when trying to establish a correlation between CT and SBP. Also, as filtering techniques are not perfect to remove all possible errors, there is still a lot of room to improve the accuracy of machine learning algorithms to predict SBP.

Another aspect of the RandomForestRegressor that I noticed with the data that I plotted was that it tended to have overfitting in many places. This is caused by many factors, such as the number of decision trees and the way the regressor chooses how to separate the data at every tree node. With optimization and testing of different features, the Random Forest can definitely be improved to predict SBP with a much higher accuracy. The overfitting seen with the data I plotted could also be partly attributed to the low correlation factor of my points and the Random Forest would probably perform better with points that are more closely related.

Overall, machine learning techniques used to predict BP with PPG signals have a lot of potential. With enough patient data and time, these algorithms can be significantly improved and possibly replace commonly used BP measuring devices. In the case of Random Forest, there is still a lot that can be done to improve the results that it provides. In conclusion, although my Random Forest did not perform well, the most significant problem was processing the data, and not the algorithm itself. Therefore, I believe it can produce more meaningful results given more accurate data.

References

[1] Chobanian AV;Bakris GL;Black HR;Cushman WC;Green LA;Izzo JL;Jones DW;Materson BJ;Oparil S;Wright JT;Roccella EJ; ; ; *Seventh report of the Joint National Committee on Prevention, detection, evaluation, and treatment of high blood pressure.* URL: https://pubmed.ncbi.nlm.nih.gov/14656957/.

[2] Adam M Brickman et al. *Long-term blood pressure fluctuation and cerebrovascular disease in an elderly cohort.* May 2010. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2917204/.

[3] *Butterworth filter design and low Pass Butterworth filters.* July 2018. URL: https://www.electronics-tutorials.ws/filter/filter_8.html#:~:text=Ideal%20Frequency%20Response%20for%20a,ideal%20%E2%80%9Cbrick%20wall%E2%80%9D%20response..

[4] *Cardiovascular disease burden, deaths are rising around the world.* Dec. 2020. URL: https://www.acc.org/about-acc/press-releases/2020/12/09/18/30/cvd-burden-and-deaths-rising-around-the-world.

[5] A L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". en. In: *Circulation* 101.23 (June 2000), E215–20.

[6] Alistair E W Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3.1 (May 2016), p. 160035.

[7] Yongbo Liang et al. *Hypertension assessment using photoplethysmography: A risk stratification approach.* Dec. 2018. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6352119/#B23-jcm-08-00012.

[8] Benjamin Moody et al. *MIMIC-III Waveform Database Matched Subset.* 2020.

[9] Gbenga Ogedegbe and Thomas Pickering. *Principles and techniques of blood pressure measurement.* Nov. 2010. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3639494/.

[10] Gašper Slapničar, Nejc Mlakar, and Mitja Luštrek. "Blood Pressure Estimation from Photoplethysmogram Using a Spectro-Temporal Deep Neural Network". In: *Sensors* 19.15 (2019). ISSN: 1424-8220. DOI: 10.3390/s19153420. URL: https://www.mdpi.com/1424-8220/19/15/3420.

[11] Toshiyo Tamura. *Current progress of photoplethysmography and SPO2 for Health Monitoring.* Feb. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6431353/.