Derivation of Blood Pressure Features from Photoplethysmograph Using Signal Processing and Machine Learning Techniques

1st Meg Sharma
Biomedical Engineering
Ryerson University
Toronto, Canada

***Abstract*—This paper explores methods of preprocessing, feature extraction and machine learning as applied to the study of a blood pressure (BP) data set with corollary photoplethysmogram (PPG) data. Notch frequency removal was conducted using a notch filter and a bandpass filter was applied to the given BP and PPG signals along with zero phase filtering. Feature extraction was performed on both sets of data to produce critical points, including systolic pressure, diastolic pressure and dicrotic notch pressure values. The features extracted from the PPG signal were then used to improve the precision of the machine learning model. Labels were produced through feature extraction of the BP signal. The TPOT library was used to automate the machine learning pipeline for systolic and diastolic BP. BP features were successfully derived from the PPG signal, with a mean square error of 395.493 for systolic BP for training and 395.493 for validation. For diastolic BP, the MSE was -218.362 for training and and -218.386 for validation.**

***Index Terms*—blood pressure, photoplethysmography, machine learning, feature extraction, CLABSI, invasive lines, central lines, noninvasive monitoring, critical care**

I. Introduction

The gold standard of blood pressure (BP) monitoring for critically ill patients is the central line blood pressure waveform. This waveform can be associated with infection risk, such as central line associated-bloodstream infection (CLABSI). CLABSI occurs when bacteria or other pathogens enter a patient's central line and cause infection and are associated with increased morbidity, mortality and healthcare costs. An increase in the number of line accesses is associated with an higher probability of CLABSI [1].

A central line is an intravascular access device or catheter that terminates at or near the heart or one of the major vessels. It can be inserted either centrally or peripherally. A central line can be placed at the pulmonary artery, superior vena cava, inferior vena cava, brachiocephalic veins, internal jugular veins, subclavian veins, external iliac veins, common iliac veins or femoral veins [2]. A hollow transducer, depicted in Figure 1 situated at the central line produces a blood pressure waveform that is monitored by clinicians to assess the cardiovascular health of the patient, such as for infusion or hemodynamic monitoring [1]. The arterial pressure wave, shown in Figure 2 travels faster than actual blood flow as it represents the left ventricular contraction that is conducted through the aortic valve and blood vessels before reaching the catheter, traveling

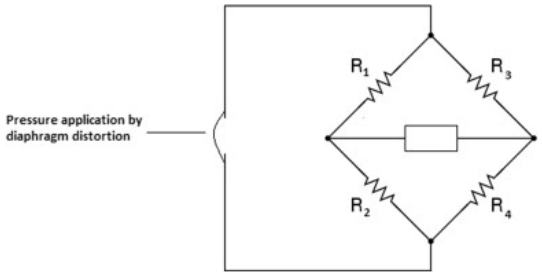

Fig. 1. Blood pressure transducer [2]

up the fluid column to reach the Wheatstone bridge transducer [3]. This waveform can be stored and retrospectively studied

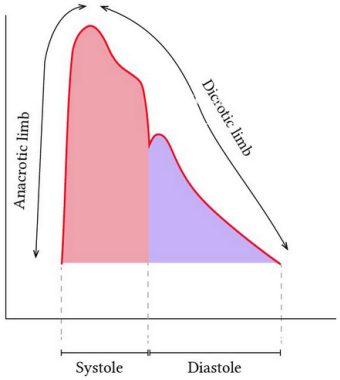

Fig. 2. BP waveform [4]

as the motivation informing this paper's signal processing approach to arterial blood pressure data. Typically, in the ICU, the pulse oximeter is placed on the index finger of the patient to measure mean blood oxygen saturation levels [5]. The photoplethysmogram (PPG) signal, can be determined from the pulse oximeter. The time variant PPG signal, originating from Aoyagi and Yoshiya, is measured by quantifying the blood volume change when the skin is illuminated. Light absorption changes can be calculated. The blood volume signal has a pulsating component called the AC component and a non-pulsating component, called the DC component. The AC component is divided by the DC component to normalize the

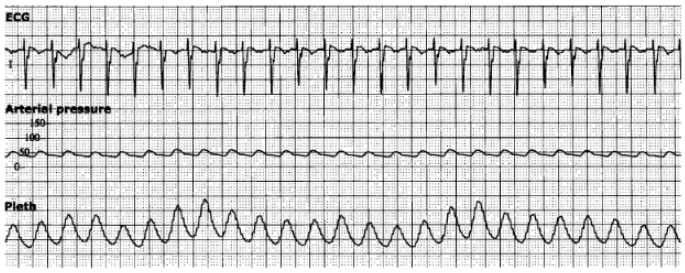

Fig. 3. Pediatric ICU patient ECG, BP and PPG data [5]

signal and then scale the waveform within a specific range [6].

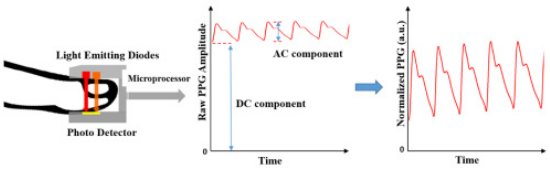

Fig. 4. PPG instrumentation, with AC and DC components shown [6]

There are variations in individual PPG signals [7]. The clinical PPG waveform is shown in comparison to BP and ECG waveform from a pediatric ICU patient in Figure 3 [5].

The measurement of BP from a PPG signal has shown a great deal of promise, with previous work spanning the use of the Fast Fourier Transform to calculate normalized BP waveforms fro the PPG as well as linear and neural network system identification techniques to correlate BP and PPG. This approach relied on auto-regression to extract waveform features and provided the justification for this paper's machine learning approach as it confirmed the link between BP and PPG waveforms. Other techniques have relied on multiple signal inputs, namely integrating the ECG and PPG waveform data to produce a reliably accurate derivation of BP [6].

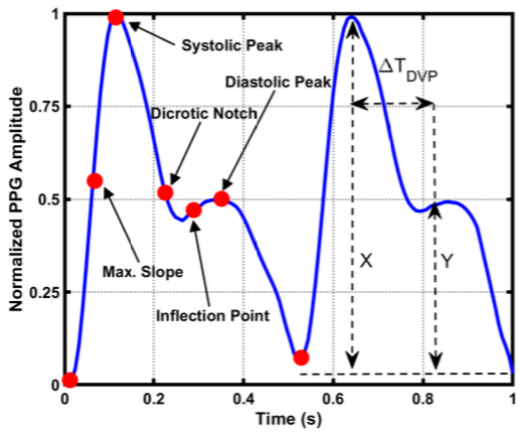

Fig. 5. PPG key features [7]

Furthermore, photoplethysmography (PPG) data can also be used to assess the patient's cardiovascular state [REF]. The PPG waveform shows the arterial oxygenation level plotted against time. Key features are shown in Figure 5, including the maxmimum slope point, dicrotic notch, inflection point and diastolic peak. The dicrotic notch demarcates the end of systole and the beginning of diastole in the central arteries pressure waveform, being more prominent in the BP waveform than the PPG waveform [7].

When the blood pressure propagates towards the fingertips from the left ventricle, the systolic peak results. The diastolic peak, another key feature, is produced when the reflected blood pressure from small blood vessels of the lower body propagates towards the aorta and fingertips [7].

II. Methods

The data of length 32.061 million samples were used out of a Kaggle data set of 1000 individuals. A sampling frequency of 125 Hz was used to produce all samples. A blood pressure signal obtained from invasive arterial blood pressure (in mm Hg), photoplethysmograph from fingertip and electrocardiograph from channel II [8].

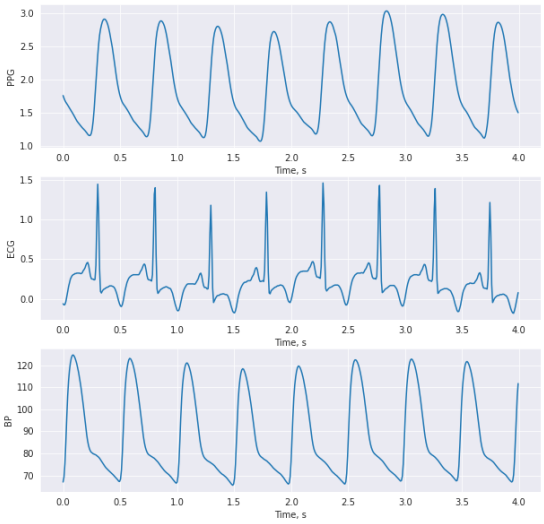

Fig. 6. BP, PPG and ECG signals from Physionet data [8]

Key features of the PPG pulse are shown in Figure III. The technique for feature extraction is shown whereby the waveform is divided in ascending and descending components, the descending component is isolated. First and second derivatives are computed and plotted to determine features.

A. Preprocessing

The preprocessing of the signals imported from the data set is important as it allows for the removal of any potential noise inherent in the signal. This preprocessing was conducted in such a way that it could be applied to any of the three signals (ECG, PPG, BP) present in the dataset that was used in the algorithm. There were three important techniques used in order to perform the necessary preprocessing on the signals.

The first preprocessing technique used to limit the potential noise of the signal is a notch filter. Notch filters have a wide

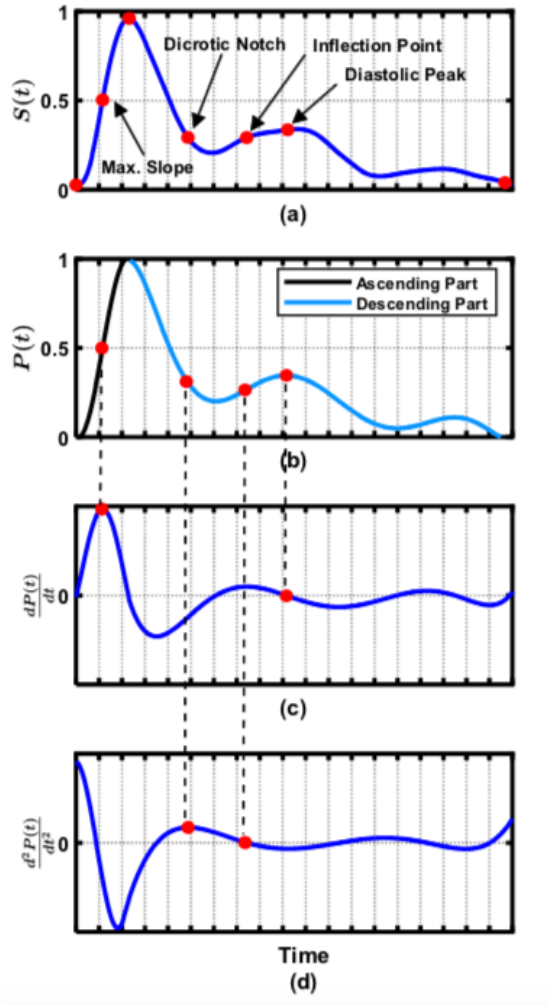

Fig. 7. (a) Key points of PPG pulse (b) Division of PPG into ascending and descending components (c) First derivative waveform used to determine maximum slope point and diastolic peak (d) Second derivative waveform used to determine dicrotic notch [7]

variety of uses in signal processing, particularly when there is a specific frequency component that needs to be removed from a signal [9]. One of the important notch filter applications to the preprocessing of the signals used in this project is the notch filters ability to reject 60 Hz AC interference in biomedical signals [9]. This 60 Hz AC interference is seen in a variety of signals due to the frequency of the power lines in North America and should be removed from the signals in the dataset before feature extraction. In order to achieve this an IIR notch filter was implemented in Python with a cutoff frequency of 60 Hz.

The second method used in order to preprocess the signals before feature extraction was the creation of a bandpass butterworth filter within Python. This filter is chiefly important as it allows for a specific range of frequencies to be selected out of the signal and any frequencies outside of the given range would be suppressed. This is particularly important as a bandpass filter like this can potentially limit the effect of any motion artifacts in the data set as well as prevent high-frequency noise in the signal [10].The important design decision in the creation of this filter was the bandpass frequency to be used in the filter. It is important that this frequency encapsulates the approximate frequency range that would be suitable for measurement of the human heart beating, as that is what is essentially being measured in the three signals available in the data set. Typically, PPG signals are often found within the frequency range of 0.5 Hz to 4 Hz [10]. For the particular band pass filter designed as part of this project, the frequency range decided upon was a range of 0.5 Hz to 8 Hz.

The final preprocessing step for the signals in this project was the use of zero-phase filtering when applying the bandpass and notch filters. This was achieved through the use of the filtfilt function in the SciPy library for Python. The way this function operates is by applying the filter in question to the signal once forwards and a second time backwards, effectively creating a zero-phase filter [11]. Zero-phase filtering is important as it can implement the filter while causing minimal distortion to the waveform of the original signal as well as not causing any delay to the original signal [11].

Fig. 8. Preprocessing Pipeline

A block diagram, shown in Figure 8 demonstrates the processing steps taken to isolate PPG-derived BP signals from the dataset, which were then compared to the arterial BP features.

B. Feature Extraction

Several features were extracted from both the BP and PPG signals to create labels and features, respectively. The signal was isolated into ascending and descending components, as shown in Figure 9.

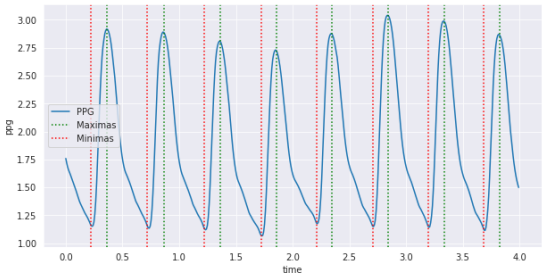

Fig. 9. PPG signal with labeled max and min points

For an interval length of 4s of PPG signal, the descending portion of the BP or PPG signal is then normalized to amplitude of 1, with consideration of the systolic peak to the end. The first derivative is computed to find the maximum slope point using polyfit(). Since the diastolic peak is not easily detectable at this point, the second derivative is computed and

a polynomial is fitted once again. Through experimentation, the diastolic peak was isolated as the last local minimum in the descending interval selected. Furthermore, the dicrotic notch was isolated from the PPG signal as it is defined as the point where the second derivative of the PPG is a local maxmimum. The dicrotic notch occurs before the diastolic peak as can be seen in Figure 10 [7].

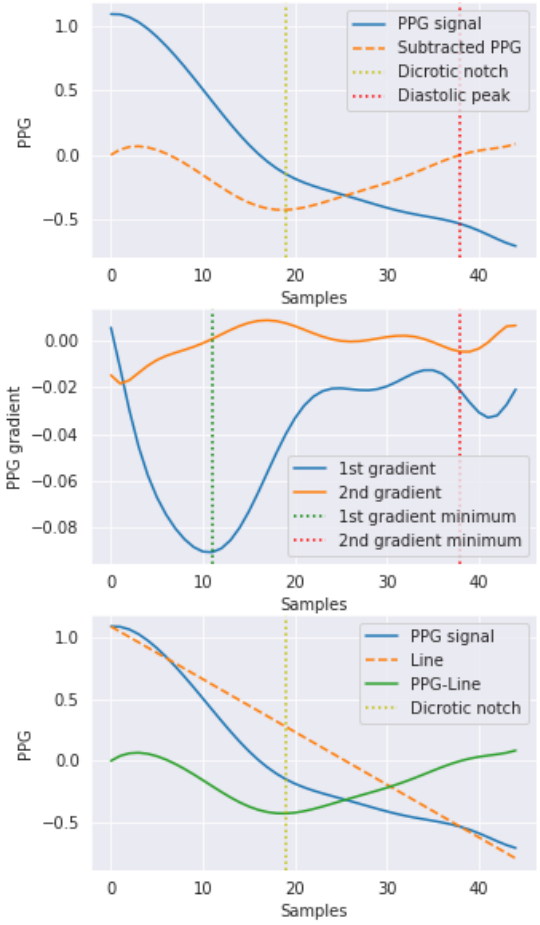

Fig. 10. 4s interval sample data

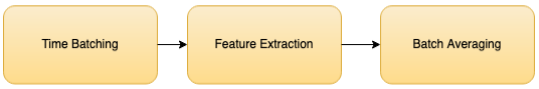

Fig. 11. Feature Extraction Pipeline

C. Feature Correlation

In the heat map diagram, Figure 12, the correlation values of various features are in the range of -1 to 1. The correlation of PPG with itself and BP with itself is 1. There is a high correlation between PPG features, as indicated by the darker green. There is also a high correlation between BP features, similarly indicated in the figure. There is no correlation between PPG and BP.

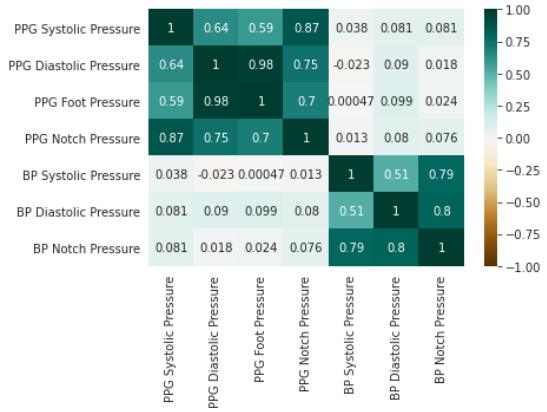

Fig. 12. Heat maps

D. Machine Learning

The TPOT machine learning library, as seen in Figure 13 was used to train the most accurate model using automated machine learning [12]. TPOT uses genetic algorithms, which

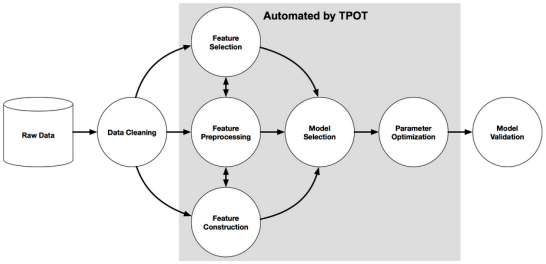

Fig. 13. Automated search of ML model using Python's TPOT library [12]

is inspired by Darwinian nature selection, to optimize and generate solutions [12]. Genetic algorithms have 3 components:

- Selection: at every generation, each solution is evaluated
- Crossover: the most fit solution is selected and crossover occurs to create a new population
- Mutation: the children from the new population are mutated randomly and the process is repeated once more to obtain the best solution

To train the systolic estimator, for instance, the following source code was used:

```
tpot_systolic = TPOTRegressor(generations=10,
    verbosity=2,random_state=random_state,
    population_size=20, cv=3,
    early_stop=4, warm_start=True)
tpot_systolic.fit(train[selected_features],
    train['BP Systolic Pressure'])
tpot_systolic.export('tpot_exported_
    pipeline_systolic.py')
```

10 generations were used, which indicates the number of iterations to run during the pipeline optimization process, with a 60-40 training-validation split. Population size indicates the number of individuals that are retained in programming the

new population at every generation. cv denotes the cross-validation strategy to evaluate pipelines; Random state indicates the seed of the pseudo-random number generator used (the same results are generated for that seed). Early stop denotes stopping before 10 generations is reached if the pipeline is optimized earlier with no improvement in the optimization process to speed up the model selection. For 10 generations of population size 20, TPOT evaluated 200 pipeline configurations, which is akin to 200 hyperparameter combinations of a ML algorithm. The best pipeline was determined and the output was written to a Python file. Upon completion, the test error was computed for validation purposes, using the following syntax:

```
-tpot_systolic.score(valid[selected_features],
    valid['BP Systolic Pressure'])
```

Since the score syntax computes the negative mean square error (MSE), a negative sign was added to the output to produce a positive MSE.

III. Results

6 features were extracted from preprocessing and feature extraction pipeline implementation, as seen in Figure 14.

	PPG Systolic Pressure	PPG Diastolic Pressure	PPG Foot Pressure	PPG Notch Pressure	BP Systolic Pressure	BP Diastolic Pressure	BP Notch Pressure
count	64121.000000	64121.000000	64121.000000	64121.000000	64115.000000	64115.000000	64115.000000
mean	2.241751	1.137596	1.087083	1.518052	119.948096	76.379230	92.048810
std	0.504900	0.225979	0.212501	0.339785	21.937135	16.032734	18.315878
min	0.401760	0.176651	0.151096	0.287390	59.763475	50.553648	56.005374
25%	1.975888	1.068589	1.036046	1.368449	104.757881	64.935731	77.749464
50%	2.208993	1.130661	1.080482	1.496858	116.793466	72.642037	87.968232
75%	2.606549	1.233138	1.167155	1.700880	133.437567	83.556405	105.195287
max	3.569892	2.311632	2.309677	2.804497	197.569324	190.863791	191.519644

Fig. 14. Dataframe's features with statistical analysis

The following pipelines were generated by the TPOT regressor for the systolic BP, as shown in Figure 15 and for diastolic BP, as shown in Figure 16. The best pipeline for

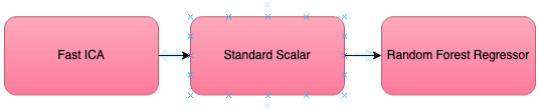

Fig. 15. Systolic BP pipeline

the determination of systolic BP was a combination of the FastICA, Standard Scaler and Random Forest Regressor. FastICA is a fast algorithm for Independent Component Analysis taken from the sklearn Python library that takes the parameter tolerance of 0.9, which describes the tolerance on update at each iteration. The Standard Scaler is used for preprocessing that implements the Transformer API to compute the mean and standard deviation on the training data that can be reapplied on the testing set. Lastly, the Random Forest Regressor was also used in the model, with the following parameters selected:

```
RandomForestRegressor(bootstrap=True,
    max_features=0.45, min_samples_leaf=11,
    min_samples_split=9, n_estimators=100)
```

A random forest is a meta estimator that fits multiple decision trees and uses averaging to produce the best overall accuracy. 100 estimators denotes 100 trees in the forest used, with a minimum number of samples at a leaf node to be 11, a minimum number of sampels to be split as 9, and a maximum 0.45 features to be considered when looking for the best fit [13].

Fig. 16. Diastolic BP pipeline

The best pipeline for determination of diastolic BP was a combination of K Neighbors Regressor, Max Absolute Scaler and PCA. K Neighbors Regressor was chosen to have the following parameters:

```
KNeighborsRegressor(n_neighbors=50,
    p=2, weights="distance")
```

50 neighbors were selected with a power parameter for the Minkowski metric set to 2, which denotes the use of Euclidean distance as the distance metric used for the tree (the default metric is minkowski). Lastly, the weight type was distance. A Max Absolute Scalar was also used in training, which scales each features by its maximum absolute value (to a max value of 1.0) and does not take any parameters. Lastly, PCA was used, with parameters:

```
PCA(iterated_power=6, svd_solver="randomized")
```

where PCA denotes Principal Component Analysis is used for strongly correlaed variables, whereby linear dimension reduction takes placed using Singular Value Decomposition of the data to project it to a lower dimensional space to reduce the data. The SVD solver was selected as "randomized" since the input data was large with fewer than 0.8 of components needed to be extracted. and an iterated power of 6 for the power method computed by SVD solver [13].

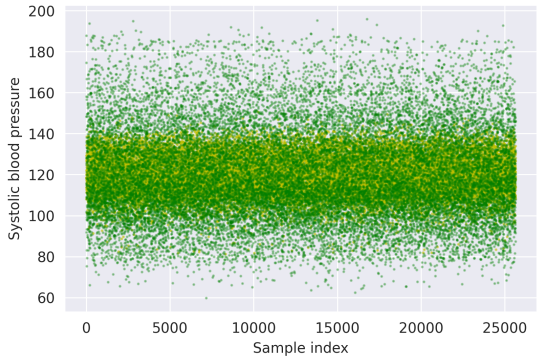

Fig. 17. Systolic pressure signal produced through use of machine learning. Yellow - predicted values, green - true values.

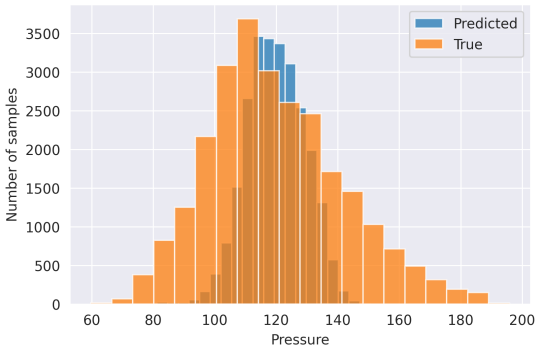

Fig. 18. Systolic pressure signal produced through use of machine learning. Blue - predicted values, orange - true values.

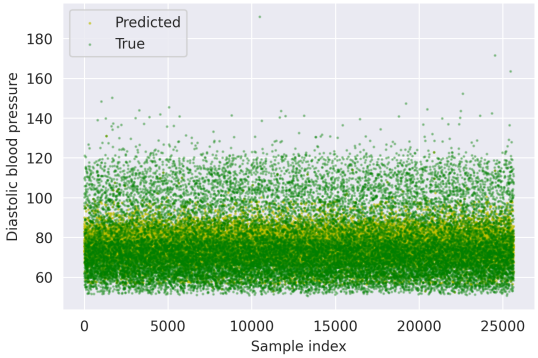

Fig. 19. Diastolic pressure signal produced through use of machine learning. Yellow - predicted values, green - true values.

Using the mean standard error, systolic peak BP and diastolic peak BP values were compared between the true BP and the PPG-derived BP, demonstrating good validation of the training sets for both systolic and diastolic pressures.

Feature	Value
Systolic Training MSE	395.493
Systolic Validation MSE	395.493
Diastolic Training MSE	-218.362
Diastolic Validation MSE	-218.386

IV. Discussion

The biomedical signal analysis of BP and PPG requires several steps, including preprocessing and preparing the data, selecting appropriate features, feature extraction and engineering, model selection and validation and tuning of hyperparameters in order to effectively produce an accurate machine learning model for feature prediction of BP from PPG. As was demonstrated in heat map, BP and PPG signals are not correlated, thus the feature extraction pipeline must precisely map the PPG's diastolic peak to the foot (signal minimum) of the BP signal.

To achieve this objective, preprocessing was performed on the signal using a notch filter, Butterworth filter and zero-phase filtering. Morphological features were used in this paper

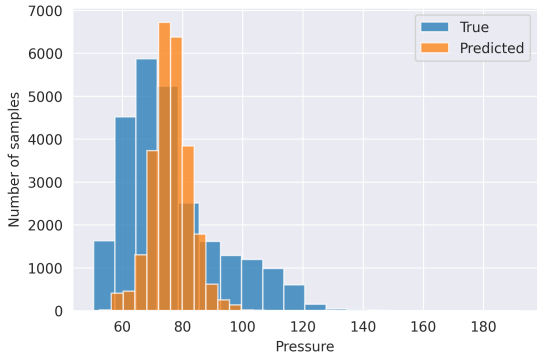

Fig. 20. Diastolic pressure signal produced through use of machine learning. Blue - predicted values, orange - true values.

to derive BP features from a PPG waveform signal. Through the feature extraction of BP, systolic and diastolic pressure were extracted. This feature extraction process was repeated for PPG. However, the PPG waveform does not have as a prominent a peak for diastolic pressure, and thus the second derivative was computed to extract this feature from the data set. The auto-machine learning TPOT library was used to train the derivation of BP from PPG features using genetic programming.

Out of the 64121 total number of samples, 6 values were dropped. 64,115 samples were used with a 60-40 train-validate split: 38,469 samples for training and 25,646 samples for validation. BP features were successfully derived from the PPG signal, with a mean square error of 395.493 for systolic BP for training and 395.493 for validation. For diastolic BP, the MSE was -218.362 for training and and -218.386 for validation.

In examination of systolic and diastolic plots, green points had a greater range of distribution, while the yellow points are more centrally distributed. The pressure range of the predicted BP was found to be narrower than the true BP. Thus, it is apparent that the model is optimized for a Gaussian distribution and feature extraction does not indicate the variation in pressure seen in the BP. A shortcoming of the approach is that ML algorithm does not successfully predict BP values far from the median systolic. This is further evidenced by the left skew of the distribution with a right tail that is not accounted for similarly, for diastolic, the prediction is skewed to the right.

There were many limitations in the approach utilized in this study. Due to limited information about the dataset, variability in the patient BP and PPG were not available, which does not allow inferences to be made in the context of the ICU patient. Moreover, only 3 features were extracted from the BP and PPG, producing labels and features, respectively, that did not provide an accurate enough training set upon which the model could learn. More features can be extracted in future work.

A derived BP signal was successfully produced from the PPG features through the use of machine learning techniques. Thus, cuffless BP monitoring can be implemented using a PPG device that is both noninvasive and allows for continuous

patient monitoring without the use of an invasive line. While invasive lines such as the central venous line and arterial line (used as the BP waveform in this data) remain the gold standard for cardiovascular monitoring of cardiac critical care unit patients, the results of this paper present the possibility of accurate, lower cost and less invasive means of detection beyond a patient's ICU stay. A PPG sensor can acquire a waveform through the low-cost implementation of a LED light from their smartphone.

V. Conclusion

This paper focused on the extraction of PPG signals to derive key features of invasive arterial BP. This work can be implemented in portable smartphone that noninvasively captures PPG signals to then derive a BP signal, and other physiological feature data, from the subject. The low difference in MSE between traininng and validation sets implemented in this study demonstrate the potential for high accuracy BP data that can be acquired at a low-cost, continuously and noninvasively from the patient beyond their ICU stay.

Future work can be conducted to explore critical care unit artifacts, such as line access interruptions in the BP waveform, to assess infection risk due to CLABSI in both pediatric and adult ICU settings. The signal processing and model training on ICU-specific patient data is the next step in validating the techniques used in this paper in order to assess different morphological features of medicated patients or those with abnormal conditions that may cause alterations in the model's ability to detect BP from PPG. Furthermore, additional features can be extracted and TPOT generations can be increased to test more hyperparameters and improve the quality of the derived BP features.

Through the development of a machine learning-based model for artifact detection, real-time patient health risk assessments can be developed to better inform clinical staff and provide timely, precision health-based care to the patient.

References

[1] M. L. Ling, A. Apisarnthanarak, N. Jaggi, G. Harrington, K. Morikane, L. T. A. Thu, P. Ching, V. Villanueva, Z. Zong, J. S. Jeong, and C.-M. Lee, "APSIC guide for prevention of Central Line Associated Bloodstream Infections (CLABSI)," *Antimicrobial Resistance & Infection Control*, vol. 5, p. 16, May 2016.

[2] E. O'Brien, B. Waeber, G. Parati, J. Staessen, and M. G. Myers, "Blood pressure measuring devices: recommendations of the European Society of Hypertension," *BMJ : British Medical Journal*, vol. 322, pp. 531–536, Mar. 2001.

[3] I. Moxham, "Physics of Invasive Blood Pressure Monitoring," *Southern African Journal of Anaesthesia and Analgesia*, vol. 9, pp. 33–38, Feb. 2003.

[4] S. A. Esper and M. R. Pinsky, "Arterial waveform analysis," *Best Practice & Research Clinical Anaesthesiology*, vol. 28, pp. 363–380, Dec. 2014.

[5] B. Frey, K. Waldvogel, and C. Balmer, "Clinical applications of photoplethysmography in paediatric intensive care," *Intensive Care Medicine*, vol. 34, pp. 578–582, Mar. 2008.

[6] X. Xing and M. Sun, "Optical blood pressure estimation with photoplethysmography and FFT-based neural networks," *Biomedical Optics Express*, vol. 7, pp. 3007–3020, Aug. 2016. Publisher: Optical Society of America.

[7] N. Hasanzadeh, M. M. Ahmadi, and H. Mohammadzade, "Blood Pressure Estimation Using Photoplethysmogram Signal and Its Morphological Features," *IEEE Sensors Journal*, vol. 20, pp. 4300–4310, Apr. 2020.

[8] "Cuff-Less Blood Pressure Estimation."

[9] J. Piskorowski, "Suppressing harmonic powerline interference using multiple-notch filtering methods with improved transient behavior," *Measurement*, vol. 45, no. 6, p. 1350–1361, 2012.

[10] J. Moraes, M. Rocha, G. Vasconcelos, J. V. Filho, V. D. Albuquerque, and A. Alexandria, "Advances in photopletysmography signal analysis for biomedical applications," *Sensors*, vol. 18, no. 6, p. 1894, 2018.

[11] A. D. Cheveigne and I. Nelken, "Filters: When, why, and how (not) to use them," *Neuron*, vol. 102, no. 2, p. 280–293, 2019.

[12] "TPOT in Python," Sept. 2018.

[13] "scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation."