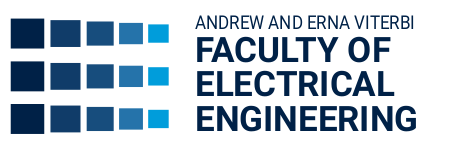
****

**Blood Pressure estimation based on PPG using neural network**

Project Submitted By: Dorin Alon & Shirili Shelef

Project Supervisors: Ron Teichner & Dr. Danny Eitan

TODO:

להוסיף את הקוד עם הנחיות

להוסיף מספרי גרפים

**Contents:**

1. Abstract:
2. Introduction:
   1. Motivation
   2. Project goal
   3. Medical background
3. Theoretical background
   1. neural networks
   2. LSTM
4. Project flow
   1. Synthetic signals
   2. Choosing the input signals to the net
   3. Focus on a single patient
   4. Data normalization
   5. Signal flow
   6. Presenting the result
5. Results:
   1. Creation and optimization of the net
   2. Final Architecture
   3. Training on a single patient and predicting BP of the same patient
   4. Training on a single patient and predicting BP of different patients
6. Conclusion:
   1. Discussion
   2. Future works
7. Appendix:
   1. results from synthetic signals
   2. results from train and test on the same patient
   3. results from train on a single patient and test on another
   4. Code and user instructions // dorin- fix code
8. References

**1. Abstract**

Blood pressure (BP) is a significant vital that is monitored for every patient in intensive care. BP could indicate patients' cardiovascular status. Nowadays, BP is measured throughout an invasive manner - catheter directly into an artery.

This form of measurement is not convenient for the patient and could lead to an infection. Photopletysmogram (PPG) is a signal measured in a non-invasive manner – a clips on the patient's finger. In this project, we will suggest a way to estimate BP using neural network (NN) type LSTM with PPG as it's only input.

**2. Introduction**

**2.1 - motivation:**

Critically ill patients are monitored continuously with vital physiologic signals. The medical team needs to be in control over the patient signals, specifically blood pressure (BP). The way to measure this vital is a catheter into an artery, this way the signal can be presented continuously on a monitor and the measurement is more accurate. Another way to measure BP is with a cuff on a patient's limb, but this measurement is taken periodically, and the result isn't accurate enough and not continuous. Therefore, patient's BP is monitored in an invasive way, which can lead to an infection and thrombosis (clot).

There is no direct formula which connects BP values to other vitals that are measured in non-invasive ways, such as electrocardiogram (ECG), respiratory impedance (RI), and PPG. Nevertheless, the physiologic connection between the signals exists, so artificial intelligence was required to learn a patient and deduce the connection. In this project we will present the neural network we used to estimate BP using PPG.

**2.2 – project goal:**

As mentioned in the motivation above, a patient's blood pressure is measured in a very invasive way. The goal of this project is to estimate a patient's blood pressure by means of deep learning methods, functioning on non-invasive continuously measured vitals, such as ECG, RI and PPG. By that, it is possible to avoid BP measuring at all, or to measure it for a short period of time. The estimation is required to be quite accurate, and predict BP elevation or descent, since it can indicate a patient's status, and alert the medical crew before an emergency.

**2.3 – medical background**

Our database is composed of 500 critically ill patients, from the children ICU department of Hospital A in Toronto.   
For every patient, the vitals PPG, ECG, RI and BP were sampled for a consistent period of 40 minutes. For every patient we got several 40 minutes sampling segments. We received raw, noisy and not normalized signals. Usually their Y axis values were meaningless, while their shape and wave pattern are the meaningful values.

BP, stands for blood pressure, is the pressure of circulating blood on the walls of blood vessels. It is usually expressed in terms of systolic pressure (maximum during heartbeat), over diastolic pressure (minimum in between two heartbeats). Blood pressure is measured in a very invasive way, by inserting a catheter to an artery. The blood pressure signals from our database are sampled at 125 Hz, and the transformation to units of mmHg as shown in equation 1.

Equation 1: converting values from the data into mmHg

PPG, stands for photoplethysmogram, is an optically obtained [plethysmogram](https://en.wikipedia.org/wiki/Plethysmograph) that can be used to detect blood volume changes in the microvascular bed of tissue. A PPG is often obtained by using a [pulse oximeter](https://en.wikipedia.org/wiki/Pulse_oximeter) which illuminates the [skin](https://en.wikipedia.org/wiki/Skin) and measures changes in light absorption. The PPG signals from our database are sampled in 125 Hz, with no consideration to values in Y axis nor to measuring units.

ECG, stands for electrocardiogram, represents the electrical activity of the heart using electrodes placed on the skin. Changes in the normal ECG pattern occur in numerous cardiac abnormalities, including cardiac rhythm disturbances. The ECG signals in our database are sampled in 500 Hz, with no consideration to Y axis values nor to measuring units. The valuable information from this signal is the wave's shape and pattern.

RI, stands for respiratory impedance, represents the analysis of pressure, flow or volume of the patient's respiration. It is measured in the same way as ECG, using the electrodes on the patient's skin to measure the chest volume. This signal is sampled in 62.5 Hz, also with no consideration to Y axis values.

Until these days, no one has ever found a strong empirical equational connection between the four signals. Yet, just by looking at the signals' shapes, we manage to witness their relation. In this project we will examine the capability of the neural network in finding a connection based on the given examples.

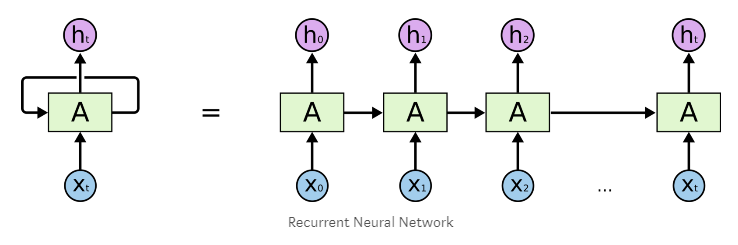
**3. theoretical background**

**3.1 neural networks:**

Neural networks are a set of algorithms, trying to imitate the human brain, and therefore are built from individual parts approximating neurons. The neural network is designed to recognize patterns by learning from various examples. It can be used for a classification problem, in which the network task will be to identify or recognize certain things from a given dataset, based on labeled examples. It can also be used for a clustering problem, in which the task will be detection of similarities. The main use of neural networks in this project is answering a regression problem. Deep learning using NN is able to establish correlations between features in a dataset, or between present and future events.

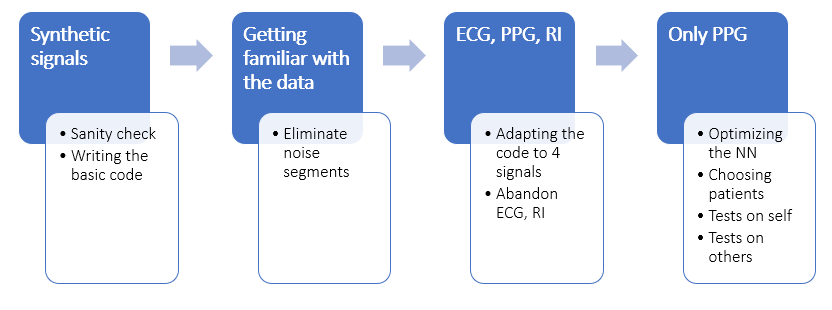
**3.2 LSTM:**

Long Short-term memory (LSTM) is an artificial Recurrent neural network (RNN) architecture that can process an entire sequence of data, and not a single point. This quality of memory of the past is necessary for learning a patient's behavior. The downside of LSTM is the need for a powerful processing unit, so we used the GPU of the lab.



*Figure 1 – recurrent neural network*

**4. Project flow**



*Figure 2 – project flow diagram*

**4.1 – synthetic signals:**

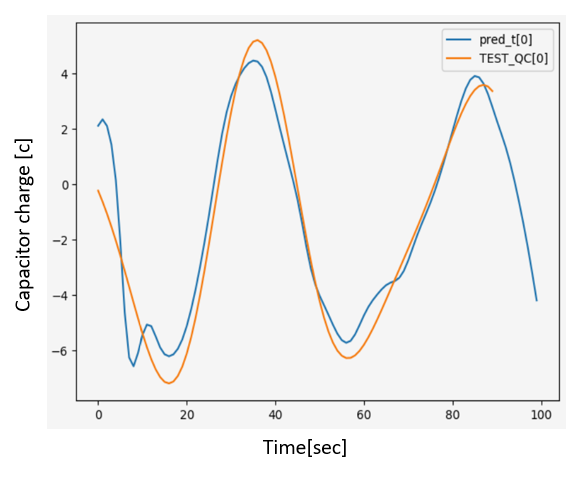
In order to ensure that our neural network is working well, we performed a sanity check. We created four types of synthetic signals that simulate an electric circuit. We created an equation that has a mathematical solution, and by that we examined whether the neural network is able to learn a proven existing connection. The four signals are source voltage, resistor voltage, inductor current and capacitor charge. We especially chose the synthetic signals so that they will be somehow equivalent to our medical signals, and this experiment will simulate the medical situation.

The capacitor simulates a blood vessel, its capacity simulates the volume of blood that the vessel can contain.

Equation 2: sanity check to the NN with 4 inputs

We divided the signals to train and test parts, and used our LSTM network.

the results are shown below:



this network has the following characters: hidden layers = 30, num layers = 1, tau = 10, iterations number = 200, where tau is the time difference between the train segment and the test segment. With the tau variable, we examine the NN's ability to predict future outcomes. We can see that the NN succeeded in the current evaluation of , and also in future evaluation (the descent in the end).

*Figure 3*

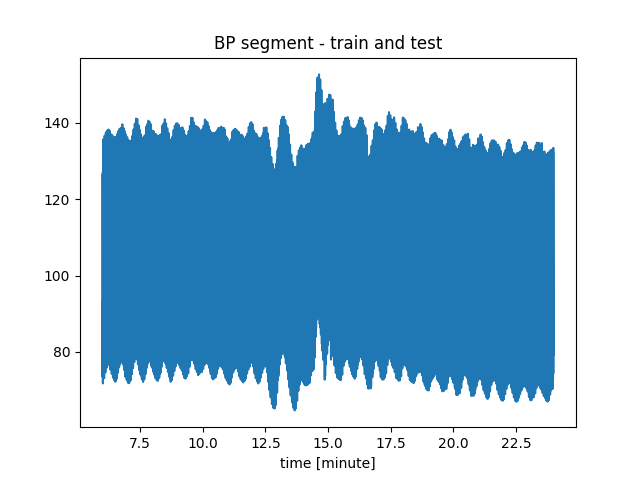
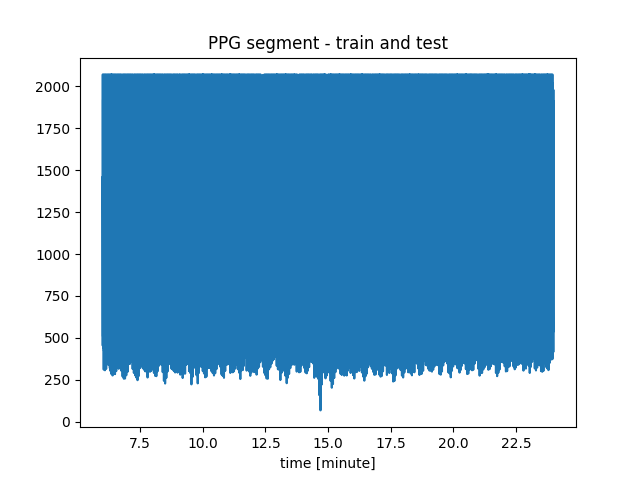
After the sanity check, we were confident with our LSTM network, and ready to test it on the medical signals.

**4.2 - Choosing the input signals to the net:**

By looking at our data, the most similar vital to BP is PPG – they are sampled in the same ratio, they seem to have similar cyclicbehavior, and medically it makes sense to research their correlation – every heart beat delivers blood saturated with oxygen, so a change in blood pressure is correlated with change in oxygen level in it (which is exactly what PPG indicates).

Moreover, we performed several trials for estimating BP based on all the signals in the database: PPG, RI, ECG. Those trials were made before the batches optimization that was done for the NN, which means 3 input to the NN, each contains a long vector of training data. As a result, the NN could not estimate BP in a good way – the estimation was bad. For comparison, the same NN with the same features that got only PPG as input, estimated well the BP of the patient.

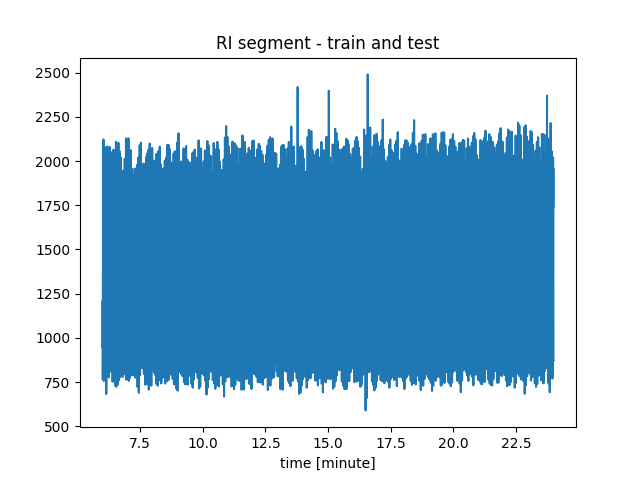
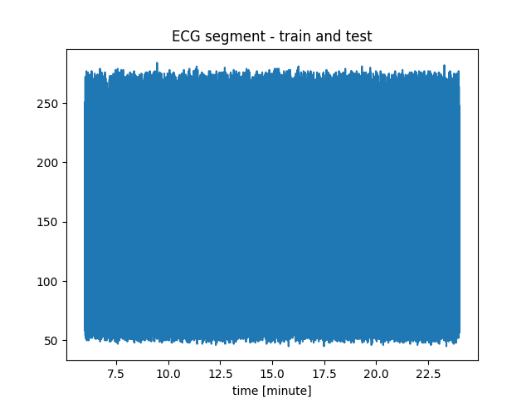
All the above led us to use the PPG vital as the only input to the NN.

We took patient 2728529-6532 (which has various BP and it's segments weren’t noisy) to perform this experiment.

*Figure 3.1*

*Figure 3.2*

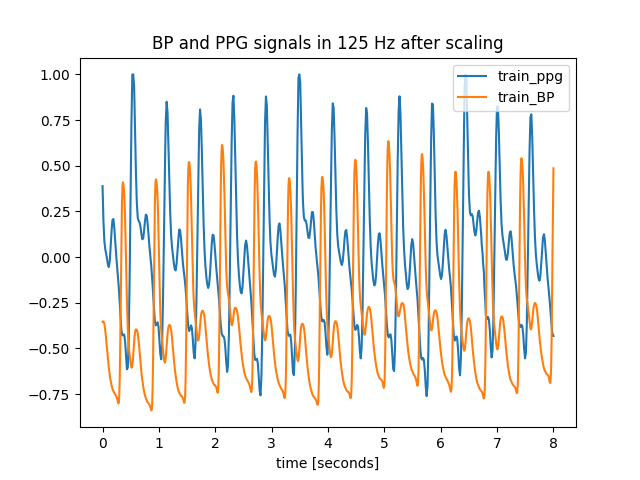
*Figure 4.2*



*Figure 4.1*

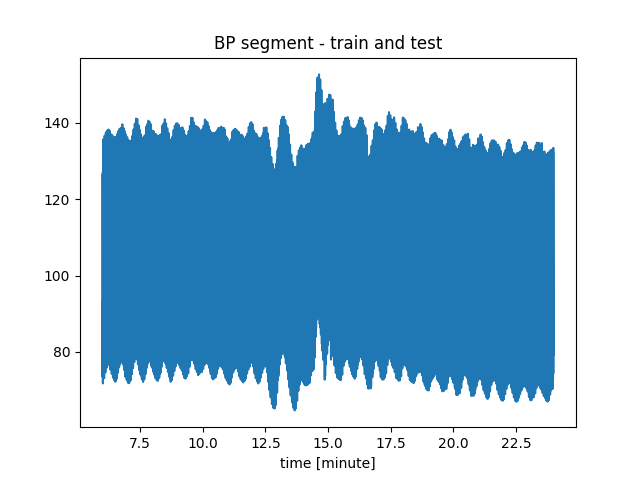
In order to use all 4 signals, we converted ECG and RI to the same sample rate as BP, PPG- 125 Hz. The scaling in the figure below is essential for NN inputs, and we will discuss about scaling and normalizing our data in the next chapter.

*Figure 5.1*



*Figure 5.2*

Division for train and test segments:

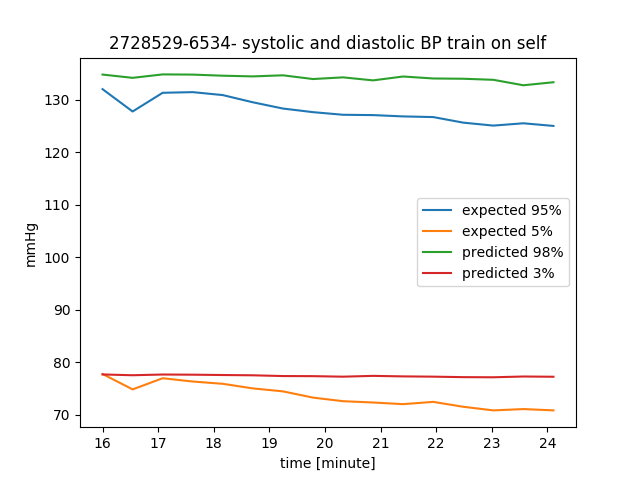


**test**

**train**

*Figure 6*

After 9 minutes of training that included all 4 signals above as inputs to the NN (ECG, RI, PPG), we performed a test over the next 9 minutes. The inputs to the test part were as in the training part: ECG, RI, PPG:



Average error – 0.049

maximal error – 0.052

*Figure 7*

At this point, comparing the estimation we got based only on PPG signal as input, we realized that ECG and RI doesn't contribute to the learning process of the NN.

We tried several trainings with PPG, which led us to optimize the NN and understand its limitations. The main limitation was choosing a patient from the database. We chose a patient's signals that stood in several criteria:

* + BP segments must be diverse enough (include increasing and decreasing of BP). The segments should include wide range of values, because the NN can't estimate values that it has never seen.
  + Both BP and PPG segments should be properly measured and without noise. Our database is raw, and the signals come from critically-ill patients, so the data can contain exceptions and spurious signals that can confuse our NN. The noise can originate from movement of the patient, treatment from the medical staff or measurement noise from the monitor or the recording unit.
  + We tried to choose a patient that was recorded long enough (every segment is 40 minutes approximately), so we could test the trained NN for different periods of time after the training period – for example 2 hours after the train, 8 hours, or even two days after training.

**4.3 -** **Focus on a single patient:**

At this point of the project, we realized that a personalized NN is required in order to estimate BP of a patient. This goes along with self-adapted medicine, an approach that customises the treatment to the unique patient. Moreover, training over multiple patients would probably deliver the mean result, which wouldn’t be accurate for a patient that doesn’t have the same features of the training patients. In addition, our data base is composed from mostly children in Toronto, which is a homogeneous population. Multi patients training on this data set wouldn’t be suitable to a diverse population.

The shape of PPG signal can change due to the way of the measuring tool of PPG changes – the way the clips is placed on a patient's finger, which figure it is placed on, and the medical condition of the patient. All of these can affect the shape of PPG and affect the NN estimation, and the PPG shape can be different over several patients.

Therefore, we decided to focus on a self-training method, and learn a single patient.

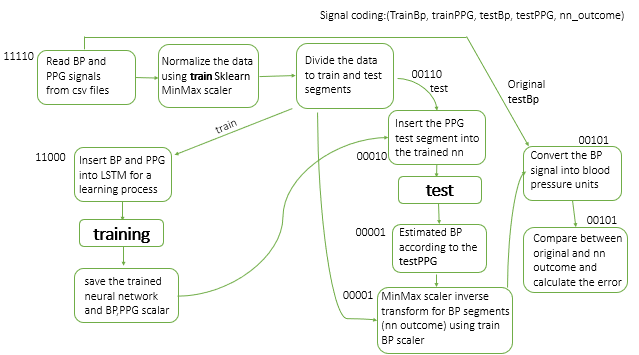
**4.4 -** **Data normalization:**

The raw data of BP and PPG segments is delivered in values of ~2500. The NN requires inputs in range of [-1,1], due to its activation function – hyperbolic tangent. The scaler for center and normalization is calculated using python package – sklearn. The scaler is calculated on the training segment, and is saved in order to use it in the test segment normalization (to simulate the reality of the ICU). This could be a problem because between the training segment and the test segment, BP could escalate and then the scaler is different – the values of the test segment could be higher/lower than the training segment.

**4.5 – signal flow:**

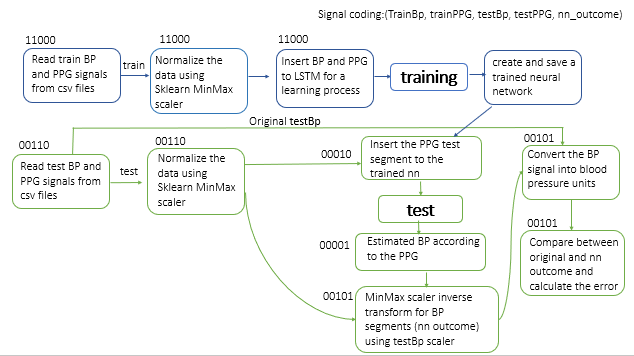
In the following diagrams we will show the signal flow from the database to the BP estimation:

**Train and test on the same patient:**

****

*Figure 8 – Train and test on the same patient*

**Train on one patient and test on another:**



*Figure 9 – Train on one patient and test on another*

**4.6 -** **Presenting the result:**

The output graph we will inspect to evaluate our NN contains a graph for systolic value (higher line) and diastolic value (lower line). We used an average over percentage value for every 30 seconds. The 95% of the BP signal should represent systolic value and 5% of the BP signal should represent the diastolic value. The NN output and the expected BP (the expected signal is drawn from the database) is on the same graph so we can compare them.

**5. Results**

**5.1 - Creation and optimization of the net:**

As we began our journey of creating and optimizing our LSTM, we made a research about its features. We used an open source code of LSTM network creation (the link is given in the appendix). We chose Adam to be our network's optimizer. Adam is an adaptive learning rate method, based on stochastic gradient descent.

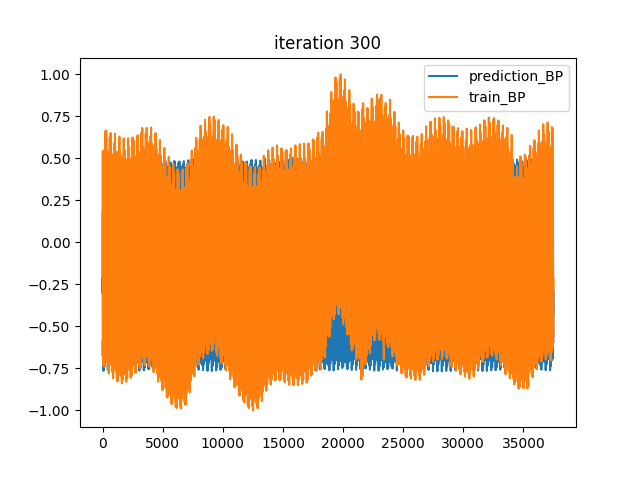
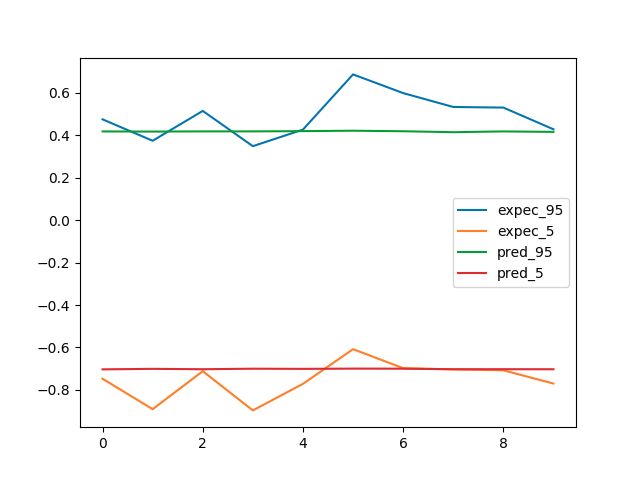
Maybe we don’t need to specify all the nn characters?

At first, we tried a straightforward approach – 1 big vector of 10 minute as training set. We performed many trials for the network features – we changed the number of layers, number of hidden layers, amount of iterations, and step size. Bellow we will present an example of the network's outcomes along with changing one of its attributes- the number of LSTM layers.

Network's features:

* Hidden layers number: 30
* Step size: 1e-3
* Number of iterations: 300

In the next graphs we will present the last training iteration over 5 minutes, and a test over the following 9 minutes.

One LSTM layer:

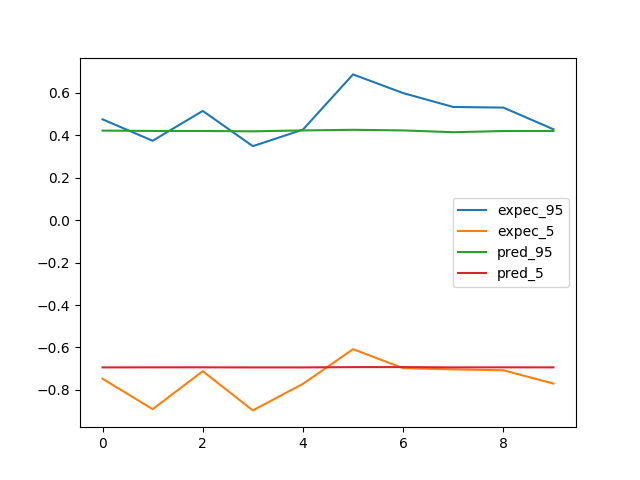
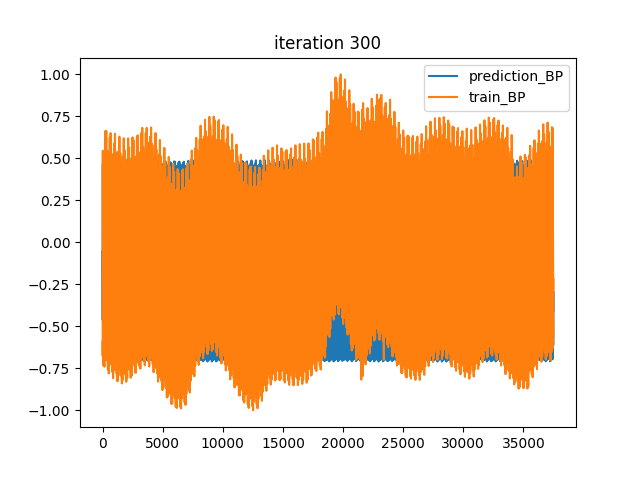
Average error- 0.0815

Maximal error- 0.3

*Figure 10.2*

*Figure 10.1*

Two LSTM layers:

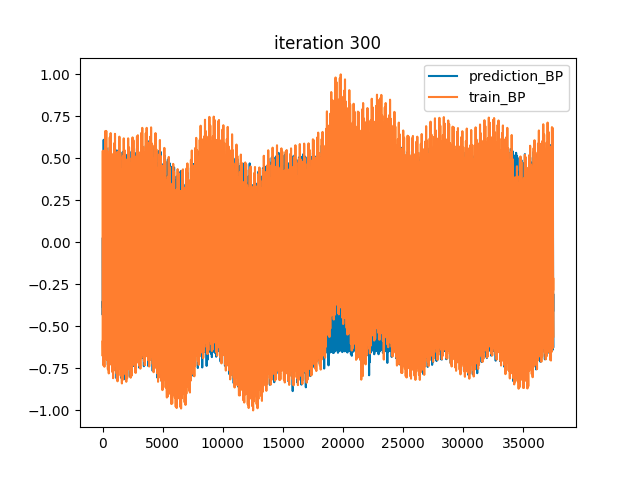
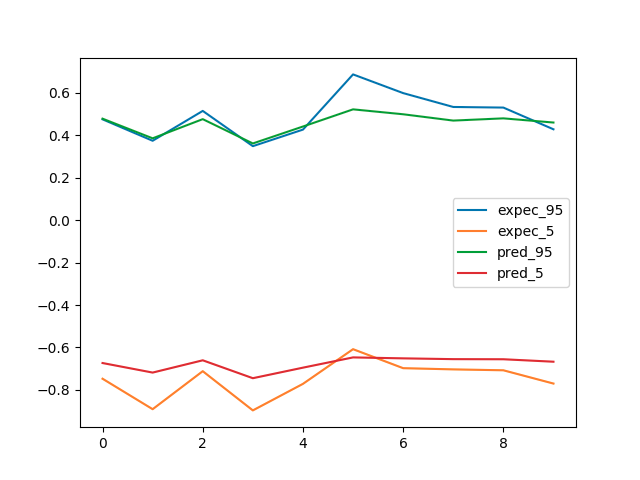


Average error- 0.086

Maximal error- 0.25

*Figure 11.1*

*Figure 11.2*

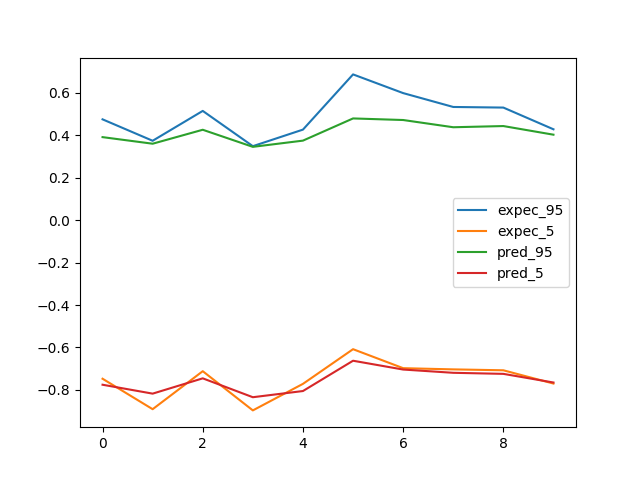
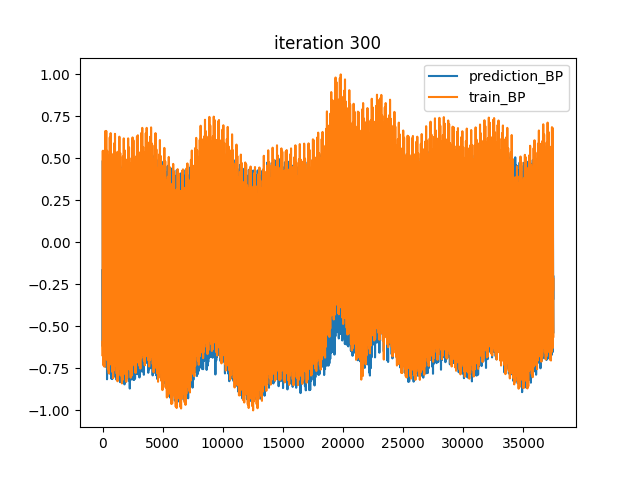
Three LSTM layers:

Average error- 0.0575

Maximal error- 0.2

*Figure 12.2*

*Figure 12.1*

Four LSTM layers:

Average error- 0.0402

Maximal error- 0.22

*Figure 13.1*

*Figure 13.2*

We can see that as we add LSTM layers, the network's results are getting better- the maximal error and the average error are decsending.

Eventually, we were able to achieve good results. Yet, the whole process of training took a lot of time – more than 1 hour (over GPU) for 10 minute of training set.

In order to take advantage of the parallel qualities of GPU and shorten the train process - we tried a batch approach.

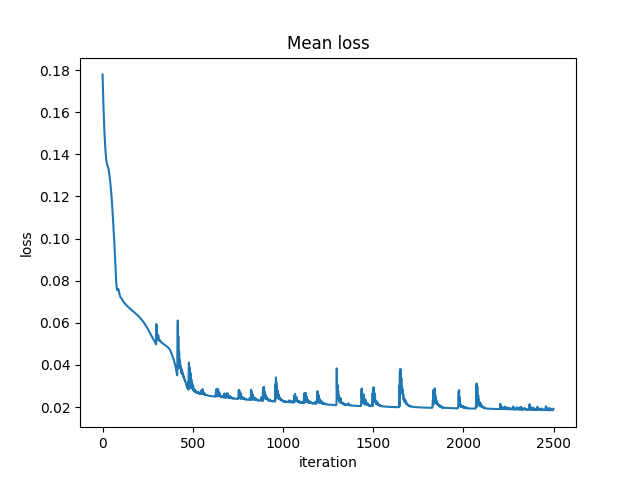
**5.2 - Final Architecture:**

After several trials, we decided to perform the train procedure with samples, which are equivalent to approximately 9 minutes (sample ratio is 125 Hz). We divided this training set to batches of 16 seconds, which means 32 batches. This way we reached improvement by 32 time than without batches (without batches means the whole training set in one vector).

On one hand, using GPU can reach 2500 train iterations in 10 minutes. On the other hand, we utilized the memory of the LSTM for only 16 seconds. The results were good even though we lost the advantage of the long-term memory. This could indicate that a long signal doesn’t contribute to the NN in order to learn a patient, and that 16 seconds of the past is enough in order to estimate a patient's BP in the future.

The final parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **Layers** | **Hidden size** | **Step size** | **Iterations** |
| 2 | 12 | 1e-3 | 2500-3500 |



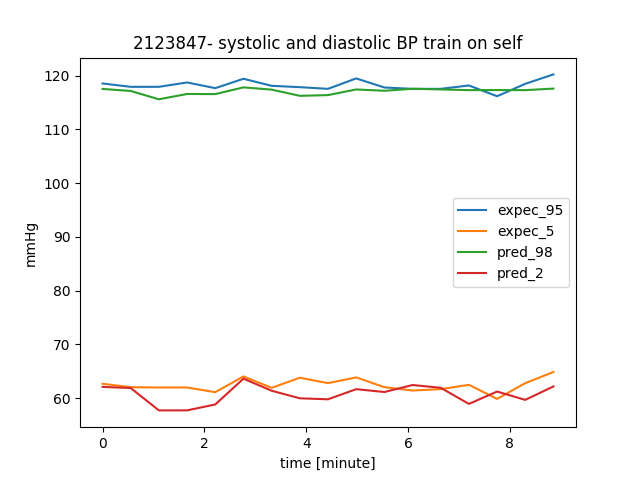
*Figure 14*

**5.3 - Training on a single patient and predicting BP of the same patient:**

We succeeded in learning a patient's behavior and infer about future and past times (of the certain patient). We took a train set that was various enough, we made sure that the BP segment contained increasing and decreasing values in BP. The variation is important in the train segment because the NN can't produce values that it hasn’t seen in the train part.

With 10 minute of train segment, the NN succeed in estimating future BP based on PPG segment.

For example, we trained on a patient over 9 minutes: (the trained NN and the scalers were saved)

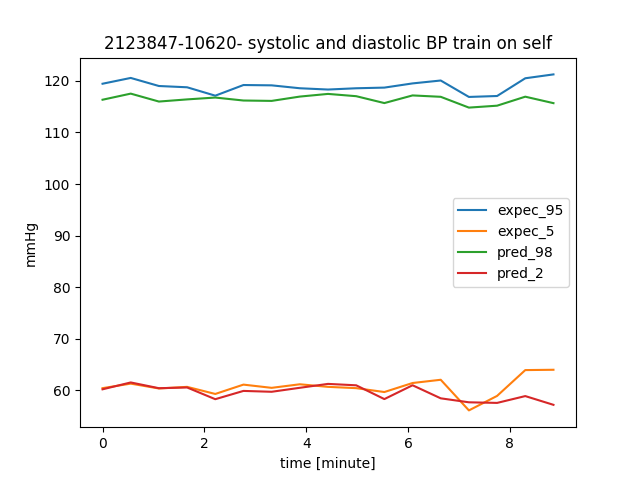
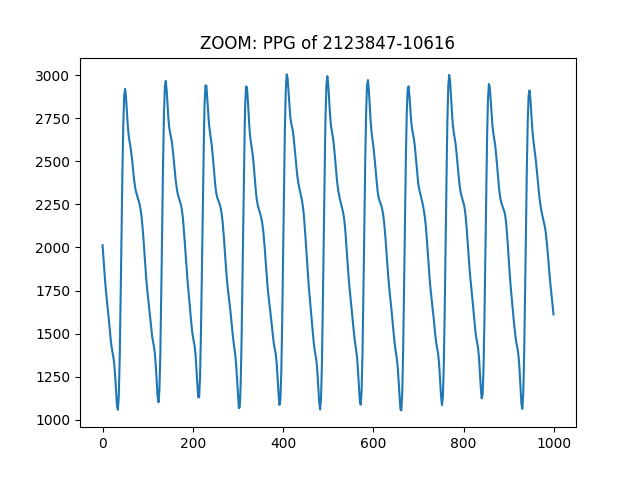


maximal error – 0.049

Average error – 0.027

*Figure 15*

Test after 2.5 hours:



maximal error – 0.049

Average error – 0.029

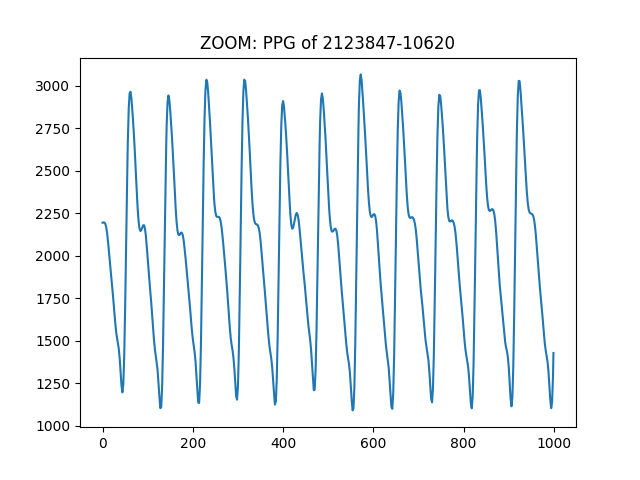
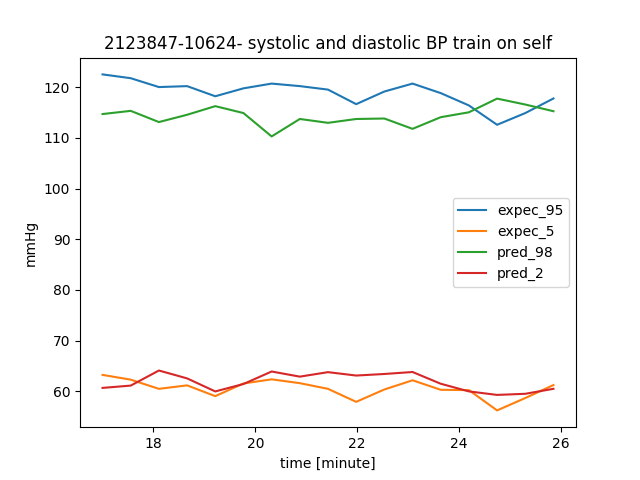
*Figure 16.1*

*Figure 16.2*

Test after 5.5 hours:

Average error – 0.047

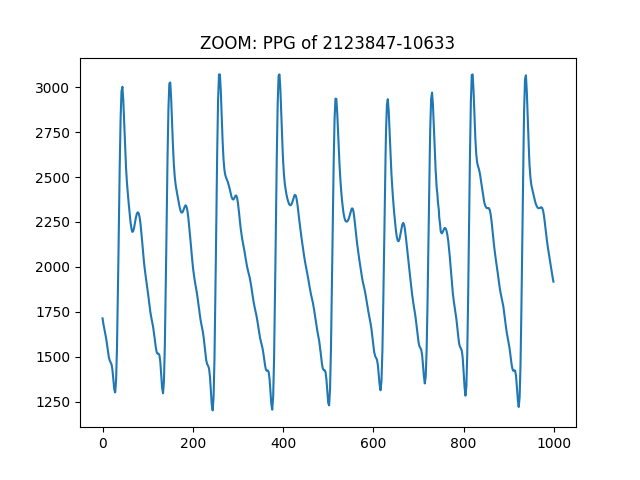
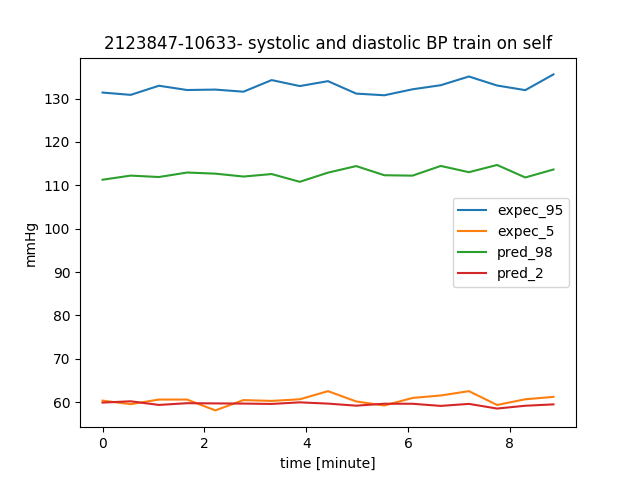
maximal error – 0.081



*Figure 17.1*

*Figure 17.2*

Test after 11 hours:



Average error – 0.148

maximal error – 0.162

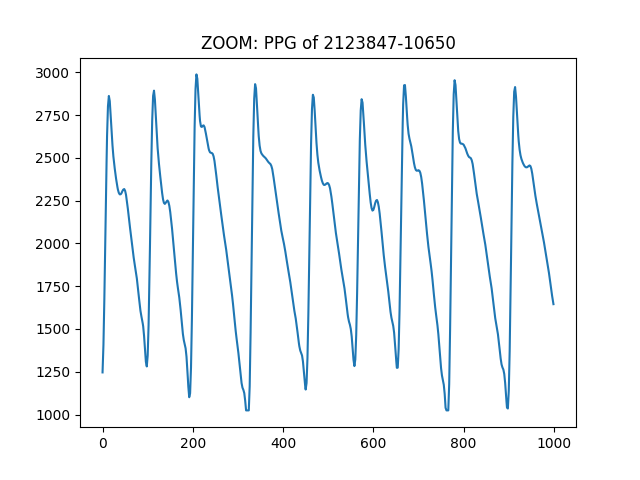
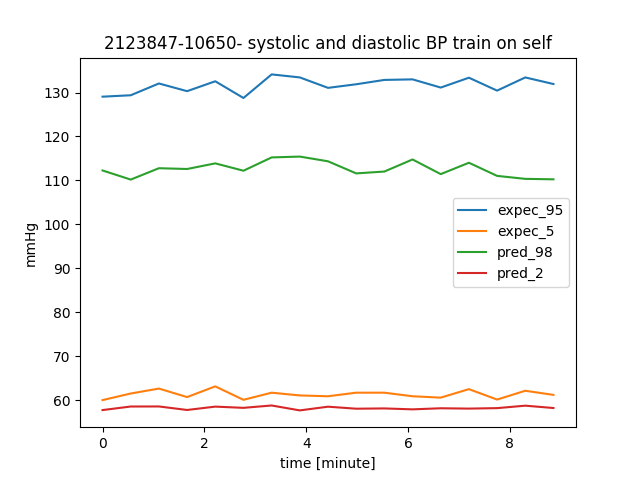
*Figure 18.2*

*Figure 18.1*

Test after 22 hours:

*Figure 19.1*

*Figure 19.2*



maximal error – 0.164

Average error – 0.147

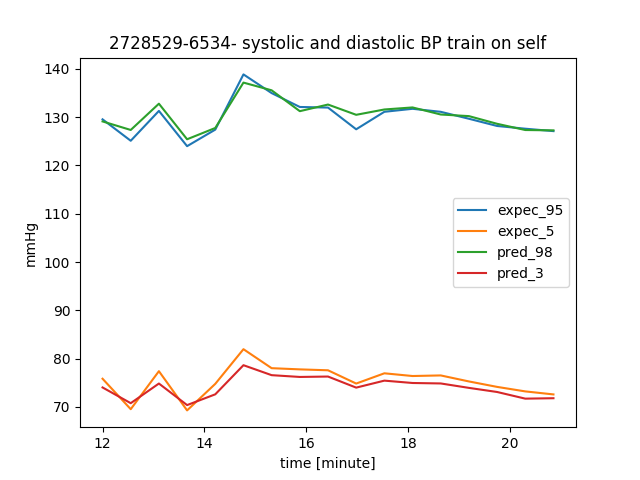
We can see that sometimes the NN estimates BP values close to reality in the next few hours after the train, and sometimes the estimation isn't that good. More results are detailed in the appendix.

We assume that the faults in the estimation are due to changes in PPG segment. The patient's physiology can change sometimes due to medications or medical condition. The change in PPG can also arise from the way the clip is suited on the patient's hand (figures in the appendix).

With the help of graphs ## which present the network's estimation after 11 and 22 hours, we can witness an interesting situation. It is easy to see that the network succeeded to estimate ascent or descent in BP, yet with an offset of approximately 10-20 mmHg. We assume that this offset is a result of the normalization method we used. In the training set, the BP values reached maximum of 120 mmHg. The test set was normalized using the scaler calculated from the train set, so as we reverse- normalized the output BP from [-1,1] it could only reach 120 mmHg, although its' real values reached higher.

We discovered that the NN can estimate BP in the past – which means that we managed to learn a patient's physiology.

Train on 2728529-6534:

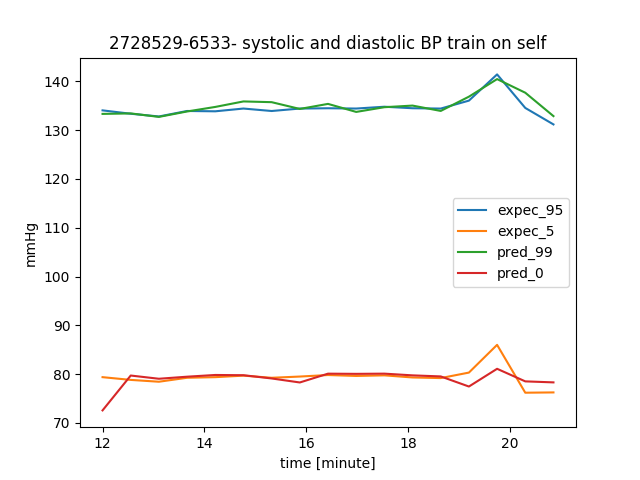


Average error – 0.015

maximal error – 0.021

*Figure 20*

Test on a segment 27 minutes earlier (before the train):



maximal error – 0.021

Average error – 0.012

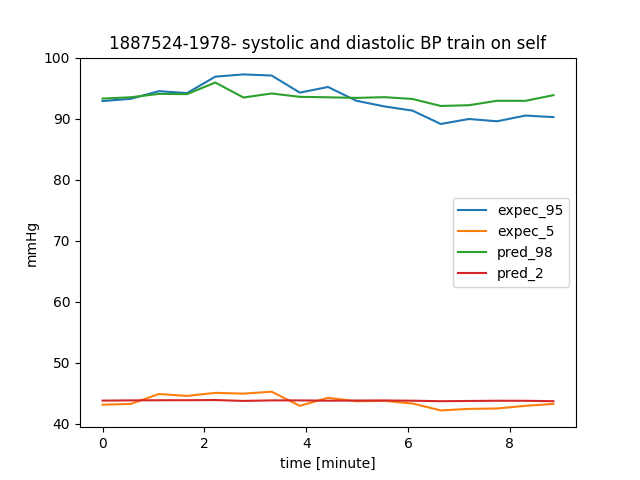
*Figure 21*

In chapter 7.2 (appendix) we can see several examples of the NN we created, that trained on 9 minutes segment on a single patient. For some patients, the estimation for the next (and sometimes previous) hours is good, and sometimes has some mismatches. We assume that these mismatches occur due to changes in PPG shape. In chapter 7.2 we can see that sometimes these changes cause bad estimation, and sometimes even though the PPG looks similar – the estimation is bad.

**5.4 - Training on a single patient and predicting BP of different patients:**

We examined a NN that was trained on a certain patient (10 minute of a various segment) – perform an estimation for another patient based on the other patient's PPG. The data normalization has been done with the test segment.

We trained a NN for 9 minutes on 1887524-1978:



Average error – 0.028

maximal error – 0.041

*Figure 22*

We performed a 9 minutes test on 2043726-9499:



maximal error – 0.06

Average error – 0.053

*Figure 23*

We can see that the estimation isn’t good – the NN couldn’t estimate the diastolic at all. The systolic estimation has a certain delay in the estimation, and the values are not reaching the expected minimum and maximum.

We assume that bad estimation of the diastolic is because the NN was exposed in the train set to values in range 40-50, and here in the test patient there are values in range of 45-55. This outcome is reasonable, because the NN can't deliver values that it wasn’t exposed to during training. Moreover, we can see that the BP of the test patient is more dynamic in comparison to the train patient. In chapter 7.3 (appendix) we can see several examples of bad estimation even when the range of BP value of the test segment consists of the range of BP value from the train segment.

**6. Conclusion**

**6.1 discussion:**

We created a neural network that can learn approximately 10 minutes of PPG and BP coordinated signals of a single patient, and able to estimate BP signal based only on PPG signals in the next period of time. We couldn’t achieve an unambiguous conclusion concerning the successfulness and accuracy of the BP evaluation, as it was changing along with the distance of time from the training set.

We've learned that in order to create successful prediction of BP, the train set should be as diverse as possible, reaching high and low values, containing elevation or descent of BP.

Upon all the above, and the research we made, we developed an algorithm for measuring a patient's blood pressure, in a non-invasive way.

**The algorithm:**

As the patient arrives to the hospital, the medical team will take his BP using a catheter to an artery. That phase must be performed for saving a training segment that includes the patient signals shape and pattern, and for our NN scalers to be calibrated.

The invasive BP measuring will continue until we get a diverse BP signal.

Through the whole process, the patient's PPG measurements will be processed alongside with the BP, as an input to the neural network. After approximately 10 minutes of training, the NN has finished it's learning process. After the catheter is out, the PPG signals will continue to flow as an input to the network, and the output is the estimated blood pressure, that can be presented on the monitor. In order for the output to be more accurate, every hour (more or less- depending on the patient's state), the patient's blood pressure will be taken using a sphygmomanometer (cuff), that produces an average value for systolic and diastolic BP at that time. These values will be delivered to the network as parameters for calibrating the scalers.   
Every 12 to 24 hours the catheter will be reconnected to the patient (in order to take blood samples). We will use that for retraining the NN and achieve better results.

With the help of the algorithm, we will achieve minimal exposure of the artery, and yet an accurate blood pressure measurement.

The algorithm we created has some flaws. First, we need to take into consideration that the PPG signal's structure and dimensions can be affected from some factors. The way the patient wears the clips, the finger he wears it on, or his physiological state can alter the signal's shape. As a result, the estimated BP will be affected as well, since it is directly calculated from the PPG.

Second, the goal is to minimize the exposed artery time, by that we get a shorter continuous BP sampling. In a shorter signal, we might get less diversity of the BP signal. The radical values will probably be lower on the systolic and higher on the diastolic scale, and the NN won't be aware of its ability to reach extreme values. Also, in a shorter signal, there is low probability of ascent or descent in the BP values, so our NN won't know how the PPG behaves in times like that, and won't be able to predict it.

Last, the PPG and BP signals normalization is performed according to scalers that were calculated based on the PPG and BP training segments. It is very likely that the dimensions are different between the train and test segments, as the patient's medical and physiological state changes as well. This normalization method is not ideal, and can impair the network's results.

**6.2 Future works:**

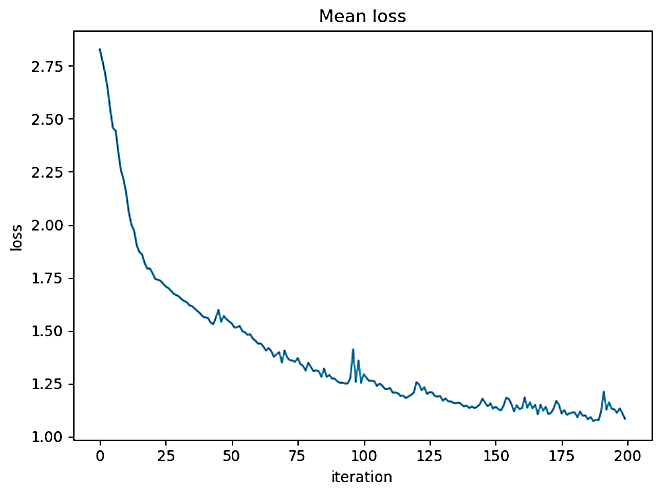
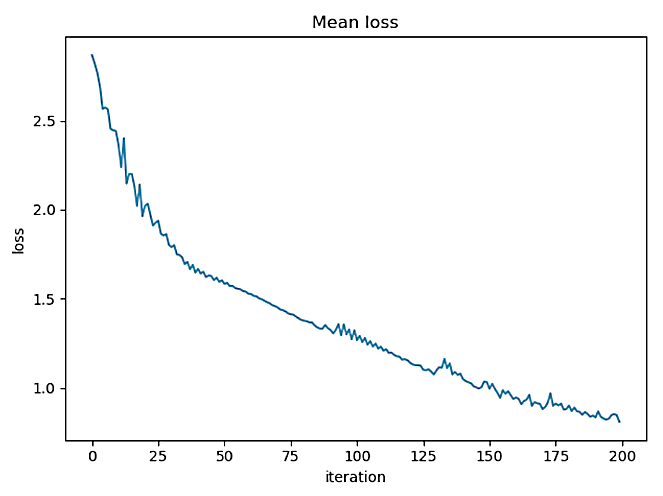
First, we noticed a connection between the PPG wave structure and blood pressure changes. We think that investigating this connection specifically in rising and falling of BP can lead to better results.

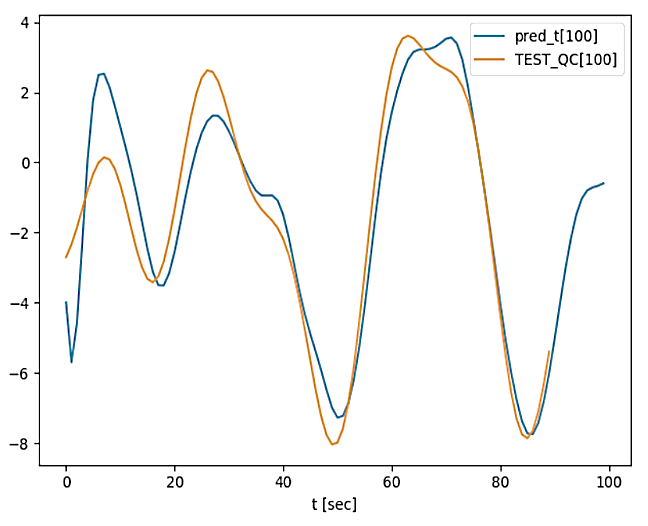
Second, we believe that the biggest flaw in our algorithm is coming from the lack of diversity in the train set. For future project, we recommend trying to perform data augmentation to the input signals, so we can achieve various BP and PPG values, but still conserve its medical correctness. In that way, we might be able to predict radical BP values, and minimize the mean error.

Third, it is interesting to use various patients as a joint input to the network, and see how another patient's BP estimation is affected. In addition, to save runtime of network training, we can use a trained network of another patient. This way, the NN weights will already be calibrated and normalized.

**7. Appendix**

**7.1 – results from synthetic signals**





Q[c]

Test 10 seconds later



Q[c]

Test 20 seconds later

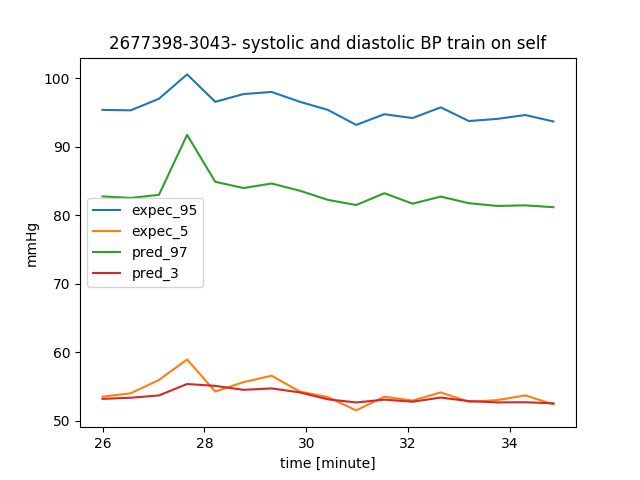
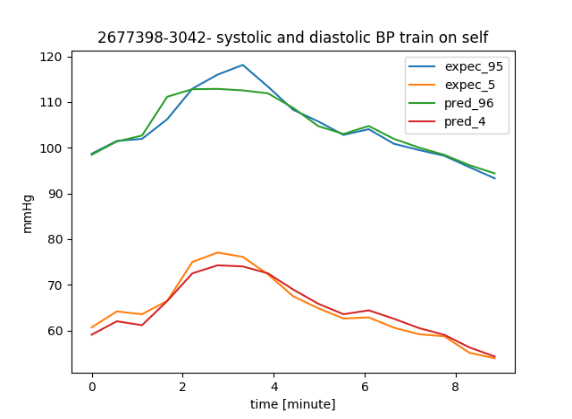
**7.2 – results from train and test on the same patient**

**Patient 2677398:**

**Test**

**65 min later**

**Train**



maximal error – 0.042

Average error – 0.016

Average error – 0.112

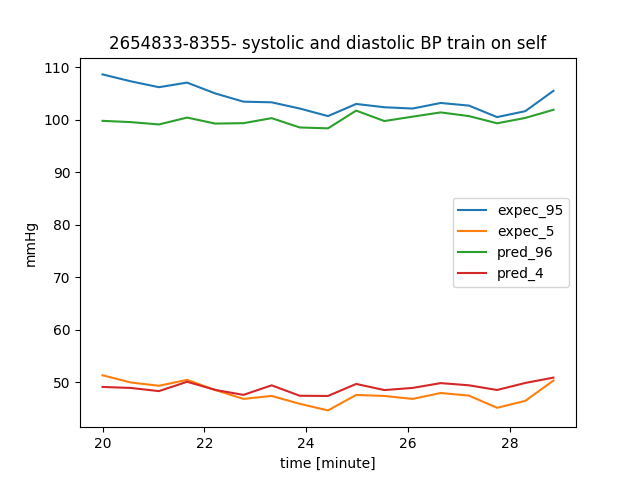
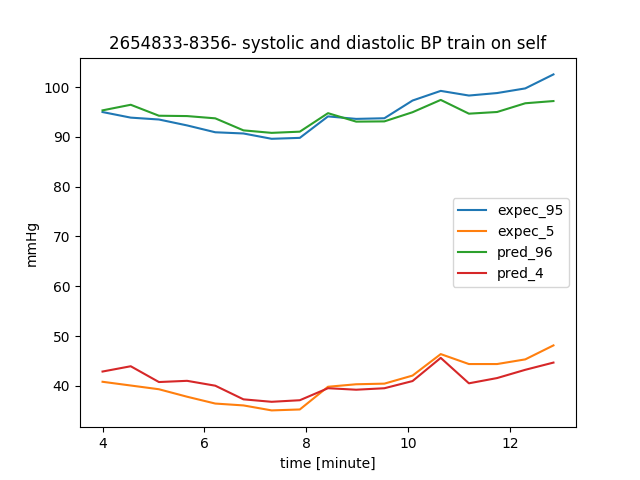
maximal error – 0.13

**Patient 2654833: estimation for the past:**

**Test**

**24 min earlier**

**Train**



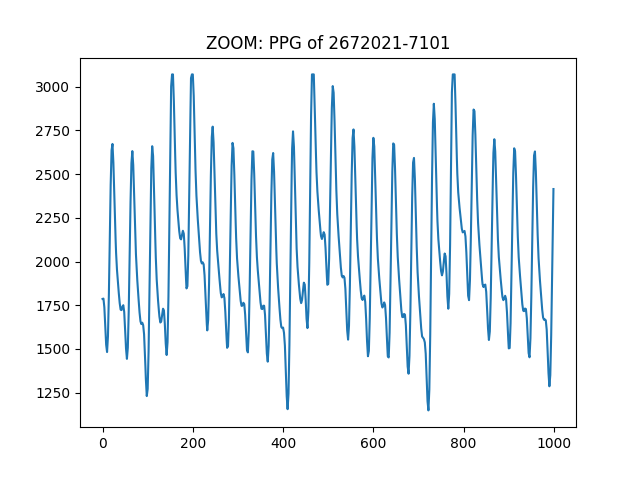
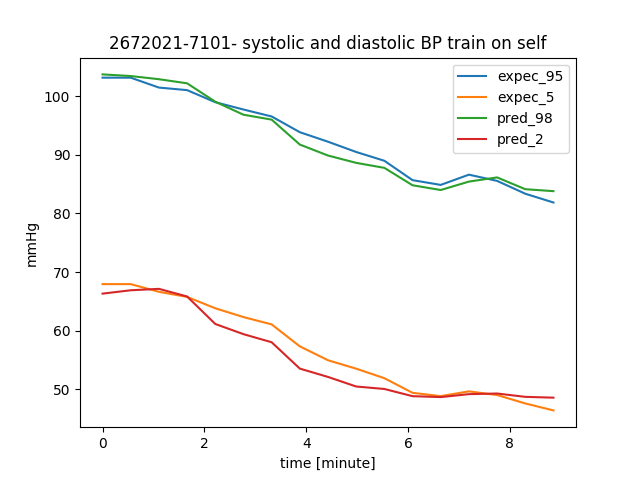
Average error – 0.035

maximal error – 0.038

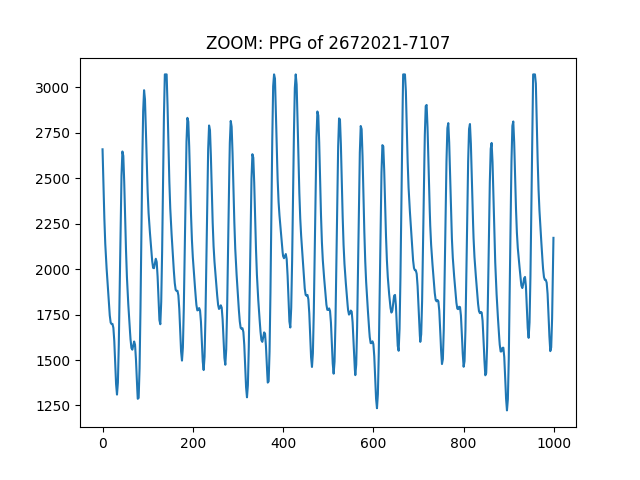
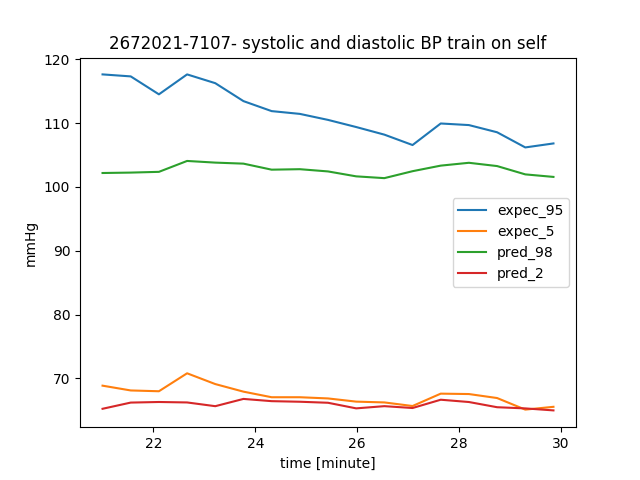
Average error – 0.046

maximal error – 0.082

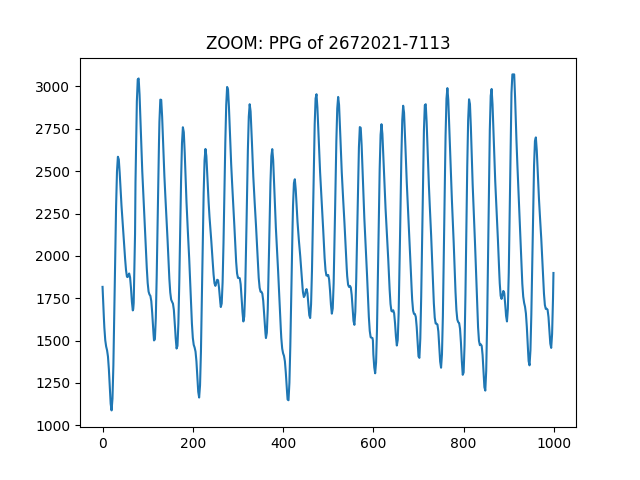
**Patient 2672021:**



**Test 4 hours later**



**Test 8 hours later**



**Train**

Average error – 0.025

maximal error – 0.037

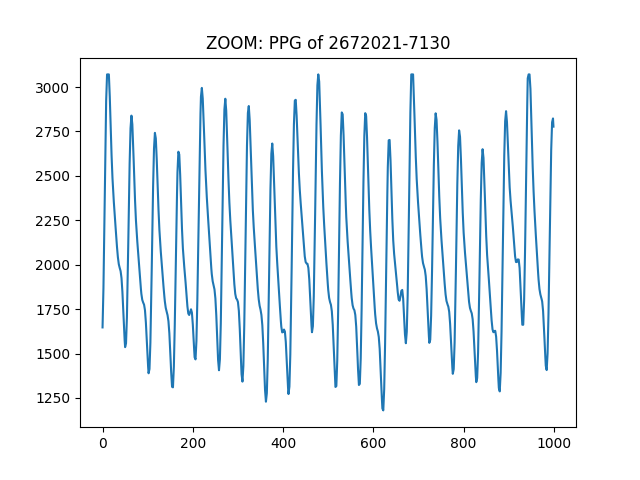
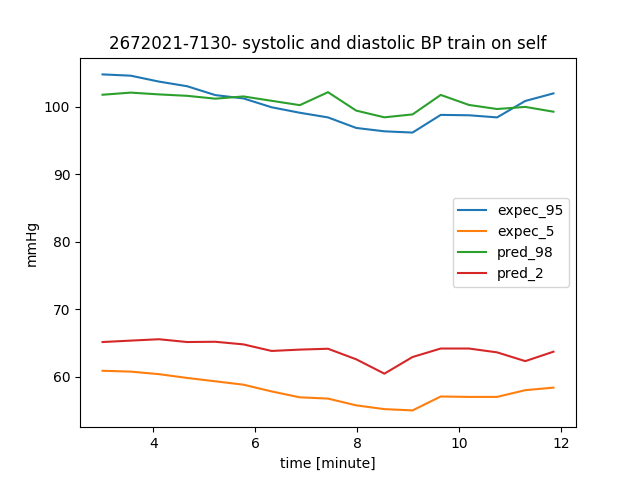
Average error – 0.084

maximal error – 0.127

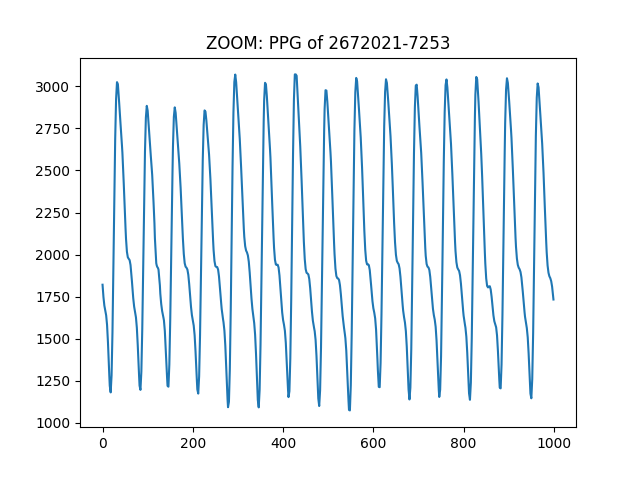
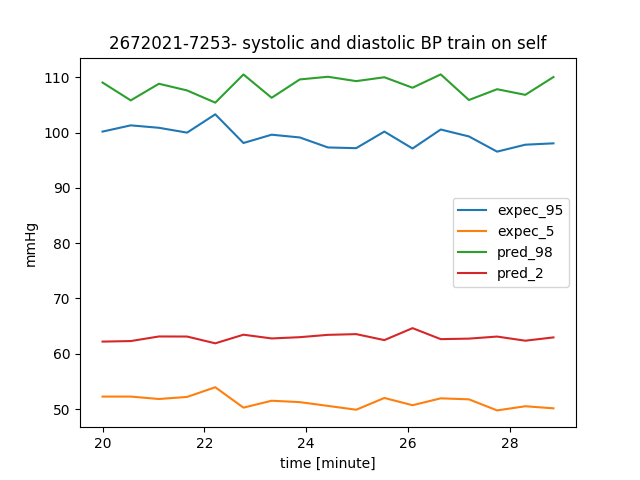
Average error – 0.083

maximal error – 0.104

**Test 19 hours later**



**Test 101 hours later**



Average error – 0.073

maximal error – 0.074

Average error – 0.115

maximal error – 0.137

**7.3 – results from train on a single patient and test on another patient:**

|  |  |
| --- | --- |
| **Train on 654833-8356**  Average error – 0.029  maximal error – 0.049 | **Test on 2043726-9499**  Average error – 0.108  maximal error – 0.123 |
| **Train on 1887524-1978**  Average error – 0.029  maximal error – 0.051 | **Test on 2043726-9499**  Average error – 0.048  maximal error – 0.074 |
| **Train on 2677398-3042**    maximal error – 0.0508  Average error – 0.023 | **Test on 2728529-6534**    maximal error – 0.178  Average error – 0.133 |

In the chart above, we can see that a NN that trains its weights according a single patient, can't estimate BP of another patient, based on the other patient PPG.

**7.4 - Code and user instructions:**

**8. References**

1. SD Goodfellow, A Goodwin, R Greer, PC Laussen, M Mazwi, D Eytan, Non-invasive Blood Pressure Estimation Using Physiological Signals Acquired by the Bedside Monitor: Insights from a Pediatric Critical Care Setting , 2019

2. Open source [code](https://nipunbatra.github.io/blog/2018/denoising.html) that was used to estimate a signal out of it's noised sample. We based our code on this example.