

RADBOUD UNIVERSITY NIJMEGEN



FACULTY OF SCIENCE

Diagnosing orthostatic hypotension with near-infrared spectroscopy: A machine learning approach

DONDERS INSTITUTE
Master thesis

Master Data Science

Supervisors:

Donders Institute:
Dr. Theodore Nikolettopoulos

Author:
Enrico Schmitz

Radboud University:
Mirthe van Diepen
Prof. dr. Tom Heskes

Second reader:
Radboud University:
Dr. Tom Claassen

August 22, 2022

Abstract

Individuals that experience an impaired recovery of blood pressure after standing up, are diagnosed with orthostatic hypotension. Individuals with orthostatic hypotension can experience dizziness and lightheadedness after standing up which may cause falls. The current method of diagnosing orthostatic hypotension is based on the orthostatic challenge. Measuring blood pressure before standing up and after one minute of standing and again after three minutes of standing. This challenge is a snapshots and might lead to wrong diagnoses. To be able to make a more precise diagnosis, continuous measurement of an individual's blood pressure for several hours is needed. This can be done with the portalite 3. This device measures near-infrared spectroscopy signals, which is a possible proxy for the blood-pressure signal. From the near-infrared spectroscopy signals, characteristics of the blood pressure drop can be estimated. In this thesis, the following research question was answered: "Can we use machine learning to diagnose orthostatic hypotension from near-infrared spectroscopy signals?". The aim of the research was therefore fourfold, namely: 1) to investigate whether orthostatic parameters can be inferred from blood pressure signals, 2) to investigate whether orthostatic parameters can be inferred from near-infrared spectroscopy signals, 3) to investigate if a direct mapping can be made from near-infrared spectroscopy signals to blood pressure signals and 4) to investigate whether the future SBP and DBP can be predicted using past near-infrared spectroscopy signals. The orthostatic parameters are defined to be the parameters used by experts to make a diagnoses on orthostatic hypotension. Different types of neural networks, a gradient boosted decision tree and linear regression were used. It was found that the developed models for orthostatic parameter inference can make correct predictions on the regular subjects but perform worse on the rarer cases. Using past near-infrared spectroscopy signals to predict future systolic and diastolic blood pressure is incapable of forecasting the blood pressure based on the near-infrared spectroscopy signals. The models and especially the gradient boosted decision tree used to estimate systolic and diastolic characteristics prove to be a viable candidate to be used for diagnosing orthostatic hypotension over an eight hour period.

List of abbreviations and acronyms

ABP = Arterial blood pressure
BP = Blood pressure
CNN = convolutional neural network
DBP = Diastolic blood pressure
Dxy = Deoxygenated haemoglobin
GRU = Gated recurrent unit
LOO-CV = leave one out cross-validation
LR = Linear regression
LSTM = Long short-term memory
MAE = Mean absolute error
MLP = Multi-layer perceptron
MSE = Mean squared error
NIRS = Near-infrared spectroscopy
OH = Orthostatic hypotension
Oxy = Oxygenated haemoglobin
RNN = Recurrent neural network
SBP = Systolic blood pressure
XGB = XGBoost

Contents

1	Introduction	3
2	Data	5
3	Methods	8
3.1	Pre-processing	8
3.1.1	Unrealistic measurements and Hampel filter	8
3.1.2	Flatliner removal	9
3.1.3	Extract SBP and DBP	10
3.1.4	Outlier detection	10
3.1.5	Resample to 1Hz	12
3.2	Standing-standing task	12
3.2.1	Input and output	12
3.2.2	Parameter models	13
3.2.3	Mapping models	14
3.3	Time-series prediction task	14
3.3.1	Architectures	15
3.3.2	Mappers	16
3.4	Optimize a model	16
3.5	Comparing models	16
4	Results	17
4.1	Pre-processing	17
4.1.1	Unrealistic measurements and Hampel filter	17
4.1.2	Flatliner removal	18
4.1.3	Extract SBP and DBP	18
4.1.4	Outlier detection	19
4.2	Standing-standing task	22
4.2.1	BP input	22
4.2.2	NIRS input	24
4.3	Time-series prediction task	28
5	Discussion	31
6	Conclusion	33
	Appendices	39
A	Standing-standing task	39
A.1	Hyperparameters for orthostatic parameter models using SBP and DBP input	39
A.2	Hyperparameters for orthostatic parameter models using Oxy and Dxy input	39
B	Time-series prediction task	40
B.1	Healthy subject model hyperparameters	40
B.2	OH subject model hyperparameters	41

1 Introduction

When standing after sitting or lying down, some people experience dizziness or lightheadedness. The dizziness is originating from a *blood pressure* (BP) drop, the BP signal measured is the *arterial BP* (ABP). In severe cases, this may lead to falls which are especially hazardous for older adults. The BP drop is caused by blood pooling in the legs as a result of gravitational forces. The blood flow back to the heart is reduced, leading to a lower output for the heart causing a decline in ABP and an immediate decrease in blood flow to the brain [42], which can cause dizziness [13]. Typically, the diagnosis of severe BP drop is made using a sit-stand or supine-stand challenge in the hospital called the orthostatic challenge. Besides the fact that this set-up is artificial, it is also a snapshot which may lead to false outcomes. For instance, the white-coat syndrome can cause abnormal BP and affect the diagnosis [49].

Individuals who suffer from this severe BP drop and have an impaired BP recovery after standing for three minutes are diagnosed with *orthostatic hypotension* (OH). Especially older adults are affected by OH and in severe cases, this can lead from dizziness to symptoms like chest pain [30]. The diagnosis of OH is made if, after three minutes of standing, the reduction in *systolic BP* (SBP) is more than 20 mmHg from the baseline SBP or if the reduction in *diastolic BP* (DBP) stays below 10mmHg from the baseline DBP [7, 30]. The baseline BP is the BP measured while sitting down or being supine. OH is most common for older adults but could also occur for younger people. The severity of the symptoms caused by OH can differ between individuals.

OH can be asymptomatic as well as symptomatic, with an acute or chronic onset. Possible common symptoms include dizziness, lightheadedness, blurred vision, weakness, fatigue, nausea, palpitations, and headache [2, 51]. Less common symptoms include syncope, dyspnea, chest pain, and neck and shoulder pain [30]. The complications caused by OH are bone fractures or concussions due to falls, but could also lead to strokes or heart diseases because of the fluctuations in BP [29, 53].

As OH can be caused by underlying health issues, it is useful to diagnose asymptomatic OH [2, 12, 35]. Underlying health issues that can cause OH include, but are not limited to, anaemia, dehydration, blood loss, heart conditions, Parkinson's disease and prolonged immobility [2, 7, 12, 30, 35, 51]. Mendez et al. [35] showed that the risk factors causing OH are different for females and males. For example, a SBP of equal to or greater than 140 mmHg was associated with an increased risk of OH only for females.

The existing method of diagnosing OH is by measuring SBP and DBP in a sitting or supine position and measuring SBP and DBP again after standing up [22]. This method is very sensitive to incorrect diagnoses as it only provides a few measurements of blood pressure drops in an artificial environment. Furthermore, a very low amount of measurements are done. During the existing method, BP is measured using a manual BP monitor in combination with a stethoscope or an automatic BP monitor. Studies are ongoing on looking into the possibility to measure blood pressure drops with a portable BP monitor in a home setting for eight hours. It is hypothesized that this would improve diagnosis as BP measurements will be available for a longer period and in the patient's natural environment. Unfortunately, the existing devices for continuous BP measurements are finger cuffs which do not perform well over a longer period [52].

Machine learning models, such as *linear regression* (LR), a decision tree and neural networks, are used more and more to solve healthcare prediction challenges. The goal of a LR model is to learn whether and, if so, which linear relationship exists between the given input and an expected output. It is a simple and fast inference model. The downside of LR is that it cannot learn complex nonlinearity's [59]. An example of a decision tree is *XGBoost* (XGB) [1, 17], which is a gradient-boosted decision tree. XGB has been used to predict hypertension outcomes based on medical data by Chang et al. [16] and is capable of learning non-linear relationships. The last type of machine learning models often used are neural network methods. The *multi-layer perceptron* (MLP) can be used for classification and regression [39]. Sequential models can be used to predict future SBP and DBP based on past *near-infrared spectroscopy* (NIRS) signals which is can be used as an alternative/proxy for the BP signals and easier to measure in a home setting. Also, a sequential model can be used, such as the *recurrent neural network* (RNN). RNN is a layer which will use the output of one time-step for the next cell, learning from what they have seen before and trying to find a pattern in our sequence. The RNN iterates over the time-steps of a sequence and keeps an internal state to use the previously fed in previous time-steps [57]. The downside of RNN is that they can have “vanishing gradients” and “exploding gradients” when using long sequences [24, 10, 44]. This RNN layer has been expanded to contain gates to reduce the problems with the gradients. The *gated recurrent unit* (GRU) is a RNN with an additional reset and update gate. Another recurrent layer is the *long short-term memory* (LSTM) [24], these are known to perform well when used for time-series [31]. The last layers we can use for time-series are *convolutional neural networks* (CNN) [32] layers, these layers learn spatiality in the data.

This thesis is part of the PrOHealth-project [5] at the Donders Centre for Neuroscience, which is an acronym for *prevention and detection of OH using wearable sensor technology*. The ultimate goal of the PrOHealth-project is to be able to detect if OH will occur when standing up, to take precautions or warn people nearby. The wearables use sensors leveraging a technology called NIRS which describes the cerebral oxygenation levels of blood. The NIRS signals are *oxygenated haemoglobin* (Oxy) and *deoxygenated haemoglobin* (Dxy). Changes in cerebral haemoglobin concentration to a large extent depend on cerebral blood flow [38]. Allowing NIRS to be used as a possible proxy for continuous BP measurements, and thus to be used to detect the BP drop after standing up.

In this thesis, the following research question was answered: “Can we use machine learning to help diagnose orthostatic hypotension from NIRS signals?”. The aim of the research was therefore fourfold, namely: 1) to investigate whether orthostatic parameters can be inferred from blood pressure signals, 2) to investigate whether orthostatic parameters can be inferred from near-infrared spectroscopy signals, 3) to investigate if a direct mapping can be made from near-infrared spectroscopy signals to blood pressure signals and 4) to investigate whether the future SBP and DBP can be predicted using past near-infrared spectroscopy signals. The last task is the ultimate goal of the PrOHealth project and for this reason we make this big step from inference to forecasting to see if the goal of PrOHealth is achievable with the used method.

2 Data

The data used in this study is acquired by the geriatrics department at the Radboud UMC hospital. The NIRS captured with the portalite 3 and ABP captured with the finapres are measured simultaneously. In this study, we have 41 subjects. 10 subjects are considered young adults (age 20 to 27) and 31 subjects are specified as older adults (age 65 to 88). Three subjects had severe OH symptoms and were thus diagnosed with OH. Four subjects had a very slow blood pressure recovery close to OH but did not report any symptoms or very light symptoms. The subjects that came very close to OH but recovered just fast enough to not be diagnosed with OH are interesting because physical resilience might lead to a quicker recovery [28]. For these subjects the current method of diagnosing OH is misleading since they could suffer from symptoms but recover just enough to not be diagnosed with OH. All of the subjects diagnosed with OH are in the older adult group. Figure 1 shows all subjects and their reported symptoms (healthy - light- severe). All subjects are of Dutch nationality, Figure 2 shows the age and gender grouping of the subjects.

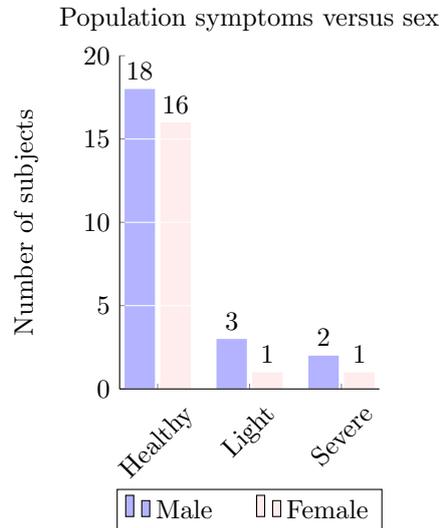


Figure 1: Subjects divided based on the severity of OH-related symptoms. Healthy subjects did not experience any symptoms, subjects in the light group experienced light symptoms and subjects in the severe group experienced severe OH symptoms.

All subjects participated in two challenges. During the first challenge the subjects needed to stand up from a sitting position and the second challenge required the subject to stand up from a supine position. Each challenge is repeated three times, so in the end, for each subject, six measurements are gathered for a transition from baseline (sitting or supine) to standing. The subject is seated or supine for five minutes and stands for three minutes.

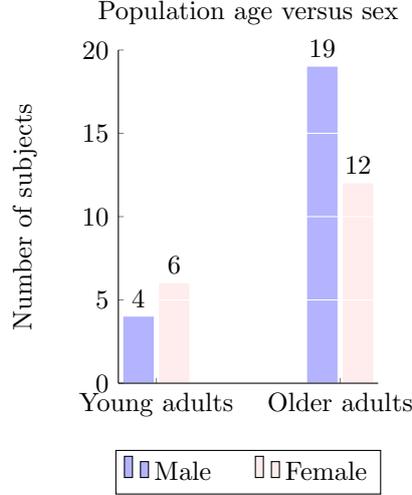


Figure 2: The number of subjects in the young adults (age 20 to 27) group and older adults (age 65 to 88) group.

The imbalance in the data set made it impossible to solve a classification problem using an OH label, since this would cause a bias in classifying a subject to be healthy if in fact that subject is not. Instead, characteristics (or parameters) of the BP recovery after standing were chosen that are used for the diagnosis of OH. We will refer to those parameters as orthostatic parameters. The recovery is very important to diagnose OH because slow recovery between 30 and 60 seconds is associated with frailty and falls [37]. The recovery amplitudes were extracted after 15, 20, 30, 40, 50, 60, 120 and 150 seconds. Besides the recovery parameters, the drop amplitude was inferred and after how many seconds of standing the drop happened. These parameters were extracted directly from the SBP or the DBP, see Figure 3. The baseline is the mean over a 30 seconds window taken 40 to 10 seconds before standing up,

$$\text{Baseline} = \text{mean}(\text{BP}_{-40:-10}),$$

and was used to calculate the

$$\text{Drop amplitude} = \text{Baseline} - \text{minimum}(\text{BP}_{0:30}),$$

and

$$\text{Recovery amplitude (at recovery time } n) = \text{Baseline} - \text{mean}(\text{BP}_{n-5:n+5}),$$

with the recovery times as specified above. The other parameters that were inferred were the mean recovery rate between 50 and 60 seconds

$$\text{Recovery rate} = \text{mean}(\text{BP}_{50:60})/\text{Baseline},$$

which is associated with cognitive decline and mortality in subjects suffering from Alzheimer's [21]. The last orthostatic parameter was the steepness of the drop (drop rate),

$$\text{Steepness} = \text{Drop amplitude}/\text{drop time}.$$

which is associated with impaired physical performance [36].

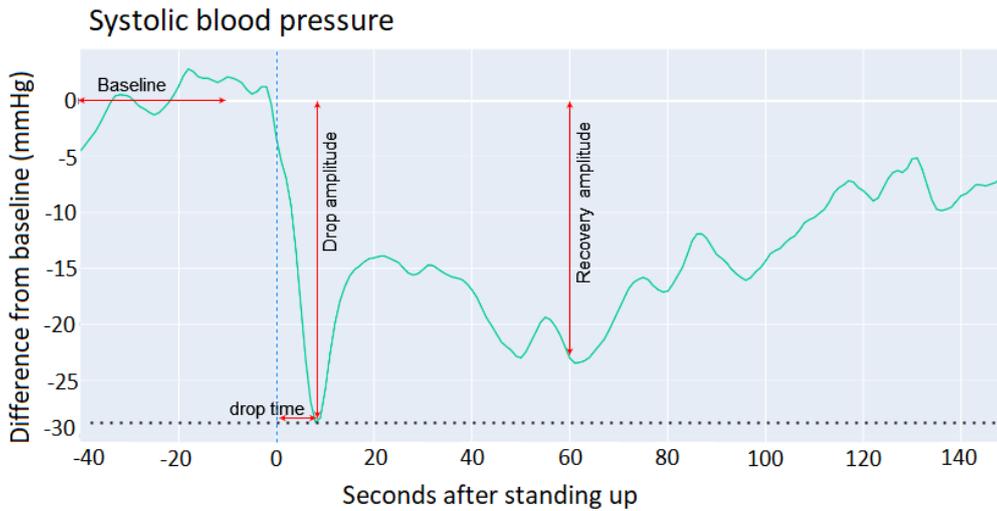


Figure 3: Some orthostatic parameters were directly extracted from the SBP (and DBP) curve. In this example the SBP of subject PHI037 is shown. Subject PHI037 was a healthy older adult female. The baseline was the average SBP (or DBP) taken over a window from 40 to 10 seconds before standing up. Where the x-axis is 0, the subject is standing up. The drop amplitude was the difference between the baseline and the minimum SBP (or DBP) within 30 seconds of standing up. The drop time was how many seconds it takes to reach the minimum. Recovery amplitude was the difference between the baseline and the mean SBP (or DBP) of a specified time window. The parameters that could not be extracted directly from the SBP (or DBP) curve were recovery rate and steepness of the drop, these were calculated using the extracted orthostatic parameters.

Due to the low amount of data available in this research, the decision was made to group all subjects during the first step of the model development, combining the two challenges containing three repeats for the 41 subjects. Leaving us with 246 individual challenges. The next step was to investigate whether the ProHealth goal can be achieved by making a prediction about the future SBP and DBP using NIRS for a single individual.

3 Methods

3.1 Pre-processing

Before model development, the data first needed to be preprocessed, which included cleaning and scaling it. Since measurements were collected with two different devices, namely the portalite 3 (to measure NIRS signals) and the Finapres (to measure ABP), it needed to be checked whether the data was at the same frequencies. The portalite 3 measures at 100Hz while the Finapres measures at 200Hz. The ABP measurements were scaled down from 200Hz to 100Hz.

The cleaning part of the pipeline is shown in Figure 4. The first step was filling any missing entries after which outliers from the ABP, Oxy and Dxy were removed using a Hampel filter. Also, unrealistic measurements were removed. Then, flatliners from the ABP were removed and the SBP and DBP were extracted from the ABP. The SBP, DBP, Oxy and Dxy were then filtered with a Butterworth filter. In the next paragraphs, the methods used are explained in more detail.

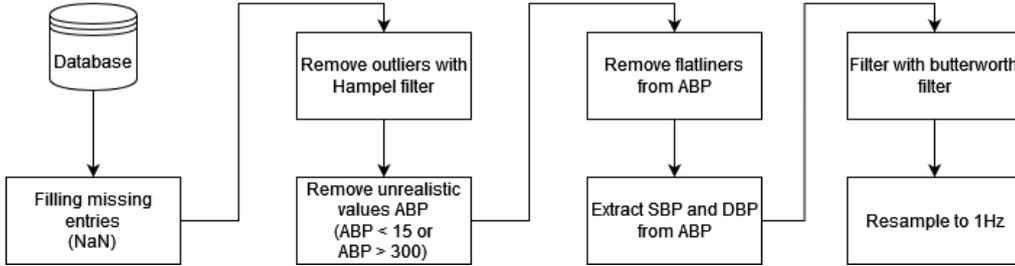
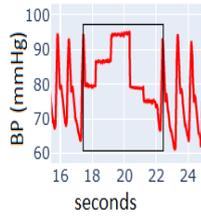


Figure 4: The pipeline for cleaning of the data. The missing entries are replaced using interpolation and followed by the Hampel filter. After that we remove unrealistic and outlier measurements and flatliners before extracting SBP and DBP. Then the NIRS, SBP and DBP were smoothed using a Butterworth filter. In the end, the data was resampled to 1Hz.

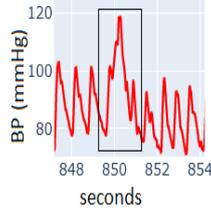
The ABP signal can contain three types of artefacts, namely flatliner, spikes and extrasystole, see Figure 5. The first was caused by active calibration during the experiment. The second by tension in the finger during measurement. The third is a biological artefact and was an extrasystole which is an extra heartbeat and can happen even to healthy subjects.

3.1.1 Unrealistic measurements and Hampel filter

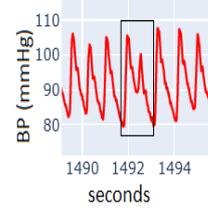
After filling in the missing entries in the data by using interpolation, the first thing we need to do is to remove ABP measurements caused by technical errors in the device. The approach used in this paper is based on Paviglianiti et al. [45]. This method can filter out the spike artefact (Figure 5b). First, the Hampel filter will remove outliers if they are far away from the mean ABP value taken over a two-second time window. The Hampel filter starts with calculating the mean absolute difference and checks if this is bigger than three times the standard deviation. If this is the case we replace the measurement with the median over the two-second window as is protocol with the Hampel filter [46].



(a) A flatliner is a technical artefact, caused by an active calibration during the experiment.



(b) Spikes are technical artefacts, caused by tension in the finger during measurement.



(c) Extrasystoles are biological artefacts, which is an extra heartbeat.

Figure 5: Possible artefacts in the arterial blood pressure signal. These example artifacts are found in PHI027, this is an older adult male showing OH characteristics. The technical artifacts 5a and 5b are important to remove from the data because these have no biological relevance. The biological artifact 5c will not be removed since this is a biological occurrence.

Then all ABP measurements above 300 mmHg and below 15 mmHg are removed since these are unrealistic measurements as stated by Paviglianiti et al. [45]. All removed unrealistic measurements are set to NaN, this will ensure that during the extraction of SBP and DBP these measurements will not be taken into account.

3.1.2 Flatliner removal

The flatliner artefact (Figure 5a) will be filtered out of the data, at 100Hz, in the following way: for each step i , denote the current BP by BP_i and the BP one step ahead by BP_{i+1} , then define:

$$\begin{aligned} \Delta BP_i &:= BP_{i+1} - BP_i, \\ \hat{BP}_i &:= \begin{cases} 1, & \text{if } \Delta BP_i > 0, \\ -1, & \text{if } \Delta BP_i < 0, \\ 0, & \text{if } \Delta BP_i == 0, \end{cases} \\ \tilde{BP}_i &:= \frac{1}{n} \left| \sum_{i=1}^n \hat{BP}_i \right|. \end{aligned}$$

With $n = 100$ at 100Hz giving us a window of one second. Since this still shows a lot of false positives, we also look into the amount of unique BP values within one second ($n = 100$). Let $v = (v_1, v_2, \dots, v_m)$ be a vector such that v_i is the number of unique BP values in an one-second time window starting at time-step i with $i = 1, 2, \dots, m$. Where m is the number of one-seconds windows fitting in the total experiment time. Then

$$\begin{aligned} \hat{v}_i &:= \frac{v_i - \min(v)}{\max(v) - \min(v)}, \\ \tilde{v}_i &:= \frac{1}{n'} \left| \sum_{[\max(0, i - \frac{1}{2}n)]}^{[\min(\frac{1}{2}n, m)]} \hat{v}_k \right|, \end{aligned}$$

Where $n' = \min(i + \frac{1}{2}n, m) - \max(0, i - \frac{1}{2}n)$. Now we combine $\tilde{\text{BP}}_i$ and \tilde{v}_i into one value for all measurements:

$$\begin{aligned} F_i &:= (\tilde{\text{BP}}_i + \tilde{v}_i)/2, \\ \text{Flatliner} &:= \begin{cases} 1, & \text{if } F_i < 0.5, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (1)$$

The removal of the flatliner artifacts from the ABP signal is done using the equations above. A signal is created where a point can be zero or one. If the point is one, it might be a measurement point in a flatliner artefact. If the measurement point is zero it probably is not in a flatliner artefact. If the equation (1) identifies more than 1.2 seconds of continuous ABP signal as flatliner, that part will be removed. SciPy [61] will mark it as a flatliner using the `find_peaks` function. This returns the timestamps at which the flatliner occurs. With the returned timestamp we can remove the flatliner from the ABP signal. The artefact is then removed up till the point we reach the DBP value again.

3.1.3 Extract SBP and DBP

Using the ABP we can extract the SBP and DBP, this is done using SciPy's `find_peaks` this function allows us to find peaks in the data. First, we set the minimum height of the peaks to be the 90th quantile over a five-second window, meaning that SciPy will never try to find a peak in the ABP signal below this 90th quantile. In the end, we interpolate the SBP and DBP back to 100Hz.

3.1.4 Outlier detection

Outlier detection is a common approach to remove anomalies and can be used to clean up a time-series signal. The outlier detection is performed on the SBP and the DBP. For this method, we tested several different types of outlier detection.

Interquartile range The interquartile range is a very efficient way to detect outliers [58]. It is based on statistics and uses the 0.25 quantile (Q_1) and the 0.75 quantile (Q_3) to detect outliers. First, the interquartile range is calculated using

$$\text{IQR} = Q_3 - Q_1$$

and then calculates a lower bound with

$$\text{lower threshold} = Q_1 - 1.5 * \text{IQR}$$

and an upper bound using

$$\text{upper threshold} = Q_3 + 1.5 * \text{IQR}.$$

If the sample is above the upper bound or below the lower bound it is classified as an outlier.

K-nearest neighbours The K-nearest neighbours algorithm [19] is a classifier which uses distance to classify data. It classifies a sample based on the surrounding points. For $k=3$ it will look for the three closest points and then classifies the unknown sample as the most occurring class. If the class occur the same amount of time it will look to the closest point in the distance. This can also be used for outlier detection [48], K-nearest neighbours will identify all points far away from the other points as an outlier. This is unsupervised K-nearest neighbours, where we compare the distance of one point to its nearest neighbours with the average distance of a point to its nearest neighbours. This will return the distance score where the interquartile range method is used to identify outliers based on the distances returned by K-nearest neighbours.

Local outlier factor The method created by Breunig et al [14] is based on K-nearest neighbours. It uses K-nearest neighbours to calculate the density of samples and compare the density of one point to a neighbouring point, where points with a lower density will have a higher chance of being an outlier. So the anomaly score depends on how isolated the object is concerning the surrounding neighbourhood, the anomaly score being the distance between the point and the group centre.

Isolation forest The last outlier detection method is isolation forest [33], this method randomly selects a feature. In our case that could be the Oxy, Dxy or the BP time-series signals. From this feature a randomly chosen split value (threshold) between the maximum and minimum of the selected feature will be chosen:

1. A random feature is selected and assigned to a binary tree.
2. For this feature, select a random threshold.
3. Measurements lower than the threshold will branch out to the left and else will go to the right.
4. Repeat steps 2 and 3 until each sample is isolated or until the maximum depth of the binary tree is reached.

The above steps are repeated to construct random binary trees. Measurements that are isolated early in the tree are likely to be outliers. The isolation forest method makes use of an ensemble creating multiple trees and using a majority vote to identify outliers, meaning that the most common outcome will be the general outcome.

Smoothing Besides the original outlier detection methods, smoothing is also considered since both the Oxy and Dxy and SBP and DBP time-series are extremely noisy which makes it unnecessarily complicated for the model to make a prediction. The method for smoothing BP is to use a five-second average [60] but this causes a delay in the data. For this reason, we decided to use a Butterworth filter [15]. The filter contains three different components, the pass band, stop band and the transition band. The passband contains the range of Hz for which the filter can let the frequency pass. The stop band specifies the amount of Hz below or above which the filter should filter out the pattern caused by the wander at the specified Hz. A cutoff value is used for which every frequency less than the cutoff value will pass with a magnitude of one and a phase of zero. For this to work a transition band is needed. This allows a transition area between the pass and stop band, this creates a smoother transition between the two. Butterworth has a flat passband response but the transition band is slow.

The Butterworth smoothing method is used for SBP and DBP with a cutoff of 0.2Hz (five seconds) and a polynomial of order two to reproduce the same results of the five seconds average without the delay. The Butterworth filter can be used for Oxy and Dxy [27] as well, the Oxy and Dxy are filtered using a filter with a lower bound 0.01Hz, an upper bound 0.5Hz and polynomial order two [20].

3.1.5 Resample to 1Hz

Now that our data is clean and every signal (Oxy, Dxy, SBP and DBP) is at 100Hz. We resample from 100Hz to 1Hz to make the predictions less computationally expensive. During this resampling, we extract the mean, standard deviation, minimum and maximum from the Oxy and Dxy signal. This resampling will calculate the statistics over a window of 100 measurement points.

3.2 Standing-standing task

In this task, we perform the estimation of orthostatic parameters given BP or NIRS signals during standing while doing the orthostatic challenge. The task is performed twofold, at first, the SBP and DBP are fed in as input data. This should be a simple task since the orthostatic parameters are extracted from the SBP and DBP. This will be a baseline to ensure we can infer orthostatic parameters from the SBP and DBP time-series. After using the SBP and DBP as input we perform the same task but with Oxy and Dxy input data, using parameter models. And finally, experiment with a model to create a direct mapping for Oxy and Dxy to SBP and DBP, this is the mapping model. In this section, we describe the prediction models used for this study, see Table 1 for an overview.

Table 1: Models use standing to standing data, these models use standing Oxy and Dxy (or SBP and DBP) data and output SBP and DBP in the same time window. With two different tasks, one to estimate orthostatic parameters of the SBP and DBP and the other to create a direct mapping from Oxy and Dxy to SBP and DBP.

Task	Model
Parameter model	Linear Regression (LR)
	XGBoost (XGB)
	MLP
	CNN
Mapping model	Time-series MLP

3.2.1 Input and output

The models will first be tested with SBP and DBP input to see if we can extract the orthostatic parameters directly from the time-series. The Oxy and Dxy signals are used as input and will contain 40 seconds of baseline (sitting or supine) data and 150 seconds of standing data. This data is then used to infer the orthostatic parameters of SBP and DBP. Here we perform leave one group out cross-validation, leaving the repeats of one subject for each unique group (older adult man, older adult female, young adult male and young adult female) out of the data set for testing and using the others for training.

In this research, we test different types of models, where one type of model will use the input Oxy and Dxy (or SBP and DBP) signals and try to estimate the orthostatic parameters of the SBP and DBP. The orthostatic parameters are as defined in section 2. Especially the recovery will be a lot slower for subjects who have OH, we would expect to see healthy subjects recover to a baseline BP quicker. This model will estimate these orthostatic parameters for both the SBP and DBP, allowing an expert to easily make a diagnosis for the subject. The other model will try and map the Oxy and Dxy directly to SBP and DBP after standing up. As stated before two different types of models are created, the parameter models are most interesting for the clinician who will need to make the OH diagnosis. The full curve mapping will be most helpful to visualize that we can map NIRS to BP and this proves that NIRS can be used as a proxy for BP for a non-subject specific model.

3.2.2 Parameter models

In this research, we investigate multiple models for standing-standing tasks, namely XGB, LR, CNN and a MLP. These models will be used to infer BP characteristics. For this task, we cannot make use of the previous mentioned RNN, GRU or LSTM layers. Since these previously mentioned models require sequential data and our output data is not sequential, these models become unusable.

The most basic model is LR using Scikit [47] basic implementation. LR is used to study the linear relationship between two variables, which will work well in the case that there is a linear relationship. Another model used is the XGB model, which is a model based on gradient boosting. Making use of decision trees results in a fast model. Each leaf of the tree contains a score, which will be used in an ensemble of multiple trees to calculate the final prediction. Boosting will add models to improve the current ensemble model, the added models will be added until it shows no improvement after adding extra models. These added models will be based on the mistakes the current model makes since the added models will try to correct the mistakes of the original model. The idea is to construct the new base learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble [41]. XGB implements gradient boosting trees and has high efficiency, flexibility, and portability. Making it in theory one of the fastest inference methods described in this paper. To be able to use XGB the input and output need to be 2D. This will be achieved by making the time-steps as features, by redefining the time-steps as

$$(\# \text{ subjects}, \# \text{ time-steps}, \# \text{ features}) \rightarrow (\# \text{ subjects}, \# \text{ time-steps} * \# \text{ features}).$$

The other available models are created with Keras. This contains a model consisting of dense layers which in its simplest form is a MLP. The other makes use of a convolutional layer which tries to learn spatial relations between the input and output data. In our case, we use a 1D convolutional layer [26]. The CNN layers use the dot product between two matrices, one being the input and another smaller matrices being the “learnable” kernel. The kernel slides across the input data and adjusts the weights in the kernel to adjust the estimation and make the model fit better to the data. In this case, we use two 1D matrices since we are using time-series data. Hence, the name 1D convolutional layer.

3.2.3 Mapping models

The last model is a time-distributed MLP layer which instead of inferring orthostatic parameters from the model will map the standing Oxy and Dxy signals to the standing SBP and DBP signals in the same time window. This model will show if it is possible to directly map Oxy and Dxy signals to SBP and DBP, using a model trained on all the different subjects. Using a time distributed part which will use the same biases and weights for each time-step. Trying to find a direct mapping for Oxy and Dxy to SBP and DBP, taking Oxy and Dxy of one timestep and translating this to SBP and DBP at the same timestep.

3.3 Time-series prediction task

Besides the models using standing input (as defined in section 3.2) to get information about the standing output this research also implements models which will try to map the past Oxy and Dxy directly to the future SBP and DBP. In this case, we make a unique model for each subject and make use of the supine orthostatic challenge data, where one repeat is used for testing and the other two for training. Using 240 seconds of the past Oxy and Dxy signals to predict 120 seconds into the future for SBP and DBP.

The models used in this section will consist of two parts, the first part uses past Oxy and Dxy to predict future Oxy and Dxy, this part will be called the *architecture*. The architecture is a two-branch model with one branch inputting the Oxy and Dxy and the other receiving the movement sensor data to adjust for the transition in the data. The second part will map the future Oxy and Dxy to the future SBP and DBP, this will be called the *mapper*. This is done to allow for recursive forecasting which would require again the future values of the input features. With recursive forecasting, we take the output of one time-step and take it as input for the next. During training we use teacher forcing, this is a method for quickly and efficiently training recurrent neural network models that use the ground truth from a prior time step as input.

In this task, we can make good use of sequential models like RNN and the with gates extended GRU and LSTM, shortly described in section 1. The GRU is similar to a RNN but has an update gate which will decide if the cell state should be updated with the new data and a reset gate which will decide if we need to use the cell state from our previous cell [18]. Instead of two gates as in the GRU, the LSTM consists of three gates, the forget gate chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten, and the input gate tries to learn new information from the input to this cell and the output gate passes the updated information from the current timestamp to the next timestamp. The LSTM originally only looks forward through the data, if we would want the LSTM to look forward and backwards through the data a *bidirectional LSTM* (BiLSTM) can be used. BiLSTM [56] consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. This increases the amount of context the model has and in essence doubles the amount of data. Both the LSTM and the BiLSTM can be stacked upon each other returning the full sequence between layers to train an LSTM based on the output of an LSTM. This increases the depth of the model and allows the model to better fit the data. An encoder-decoder setup can be created, the encoder returns its cell states and these cell states are then used as initial states in the decoder. The cell states are the forget gate, input gate and output gate. The encoder summarizes the input sequence into state vectors and these vector states are then used to estimate the wanted output.

The encoder-decoder in turn can be expanded with an attention layer first mentioned by Bahdanau et al. [9]. This mimics attention as to how humans use it, meaning that the model will focus on a part of the data and will ignore other parts. This will allow the model to focus on small but important parts of the data and highlight their part in the data. Also in this task, we can make use of the CNN layer, the CNN layer is faster than the LSTM [31], and can predict the future time-steps based on the past. All existing parts can be seen in Table 2. The architecture will be seen as a hyperparameter and is optimized by Optuna.

Table 2: The time-series prediction models, one of the three architectures is chosen to use past Oxy and Dxy to forecast future Oxy and Dxy. The chosen mapper, chosen out of the 14 available mappers, will use the future Oxy and Dxy and predict the future SBP and DBP.

Architectures	encoder-decoder LSTM		NBeats	DeepAR	
Mappers	MLP	RNN	GRU	LSTM	CNN
	BiLSTM	StackedLSTM	StackedBiLSTM	encoder-decoder LSTM	encoder-decoder with attention LSTM

3.3.1 Architectures

The three architectures used in this part are the LSTM encoder-decoder, NBeats and the probabilistic model DeepAR. The first architecture is a simple encoder-decoder LSTM. This will allow us to have a different number of past steps and future steps, the encoder will get in the past Oxy and Dxy and the decoder will translate this to future Oxy and Dxy.

A NBeats model can also be used for the past Oxy and Dxy to predict future Oxy and Dxy. NBeats stands for “neural basis expansion analysis for interpretable time-series forecasting”. This deep learning model proposed by Oreshkin et al. [43] is a model created for univariate times series point forecasting and is expanded to allow for multivariate outputs. Since we output both SBP and DBP signals, we will use the multivariate variant created by Remy [50]. This model uses trend and seasonality stacks to dissect the data and calculate a trend and seasonality in the data. The NBeats model is known to be flexible and to perform well on a wide array of time-series forecasting problems. Since night-time BP is typically lower than BP during the day [40], the seasonality stack can be used to learn this difference between nightly and daily BP.

DeepAR created by Salinas et al. [55] is a probability model which uses a Gaussian likelihood to learn a distribution from an input sample. The used implementation is based on Arrigoni [8] and will infer the mean (μ) and standard deviation (σ) from the data using deep learning. DeepAR uses an encoder-decoder LSTM network to infer μ and σ of the Gaussian from the provided training data. The mean μ during training will be returned as the expected output value. The loss is Gaussian likelihood and is based on μ and σ returned by the model. This allows us to create an expected distribution for a sample and return the probability of the predicted value. The DeepAR model generates probabilistic forecasts, this allows the model to learn complex patterns such as seasonality and uncertainty growth over time from the data [55]. Probabilistic forecasts quantify the uncertainty in a prediction, resulting in a better-interpreted result where the uncertainty is visualised [23]. With the transition performed in the orthostatic challenge it is useful to know the uncertainty of the model, since we would expect that the uncertainty is bigger during a transition. This allows us to get an idea how the BP signals are behaving even when the model is uncertain and gives a mediocre prediction.

The LSTM encoder-decoder and DeepAR use a CNN for the movement branch, since the movement sensor will only have peaks when transitioning, not while standing. We use a CNN with a large filter to learn the spatial information from which we can extract if we are standing. After the CNN we will perform AveragePooling1D. And then another CNN layer with AveragePooling1D. The output of the AveragePooling is flattened and fed into a dense layer. The NBeats architecture has its way to deal with exogenous variables [43].

3.3.2 Mappers

The mapper will try to convert the future Oxy and Dxy signals to SBP and DBP. Since LSTM layers can be used in several different ways, six different architectures are created. A simple LSTM layer, a BiLSTM layer, stacked LSTM, stacked BiLSTM, encoder-decoder LSTM, encoder-decoder LSTM with attention and as a competitor for the LSTM, we use a MLP, RNN, GRU and CNN. All the models used as mappers can be seen in Table 2.

3.4 Optimize a model

Model hyperparameters are optimized using a combination of MLflow [3] and Optuna [4], MLflow is used for saving and visualizing the model and Optuna is the package responsible for optimization of the model. Optuna selects parameters and performs multiple trials with different parameter sets. The parameter set is selected using tree-structured parzen estimator [11], this is an iterative process and uses the history of evaluated hyperparameters. Based on the history it creates a probabilistic model and this model suggests the next set of parameters [11]. Optuna is also able to prune trials that most likely will not lead to an improvement. This pruning is based on successive halving, this process evaluates all completed parameter sets and removes the worst half of the parameters [25]. All models created with Keras use early-stopping. The training is stopped if the validation loss is not decreasing for a set number of epochs, this was done to avoid over-training to reduce over-fitting. In the appendices A.1 and A.2 we can see the hyperparameters used in this research for the standing-standing task (section 3.2). The hyperparameters used for the time-series prediction task (section 3.3) are also in the appendices B.1 and B.2. Optuna returns an optimized model and this model can then be used to fit the data. After fitting the model it will save the used weights in MLflow and this allows us to reload the model and not have to train a ‘new’ model every time we want to use a model.

3.5 Comparing models

MLflow has an user interface which can be used to compare loss values of the models trained and see which model has the lowest loss and what parameters are used for this model. This was used to pick the best model and can be used to see the best model for each different task. To compare models we will be using the *mean absolute error* (MAE) and the *mean squared error* (MSE), which are widely used as metric and loss functions to evaluate the output of a regression model [54, 6, 34]. The reported scores will be the average of *leave one out cross-validation* (LOO-CV), which means that cross-validation is performed where we separate all repeats from one subject to be the test set.

4 Results

In this section, the various results of the methods described in section 3 are given and explained. Starting with the methods described in section 3.1.

4.1 Pre-processing

As specified in section 3.1 there are a few steps to remove anomaly data and extract the data in a format which will be used for the models (described in sections 3.2 and section 3.3). The first step is to remove unrealistic or very high or low measurements.

4.1.1 Unrealistic measurements and Hampel filter

In Figure 6, we can see the results of using the Hampel filter and removing the unrealistic measurements. As seen in Figure 4, we first perform the Hampel filter to remove most of the anomaly peaks (Figure 6), but this does not remove them all. For this reason, we also remove all measurements still above 300mmHg and below 15mmHg, this way we end up with a cleaner signal.

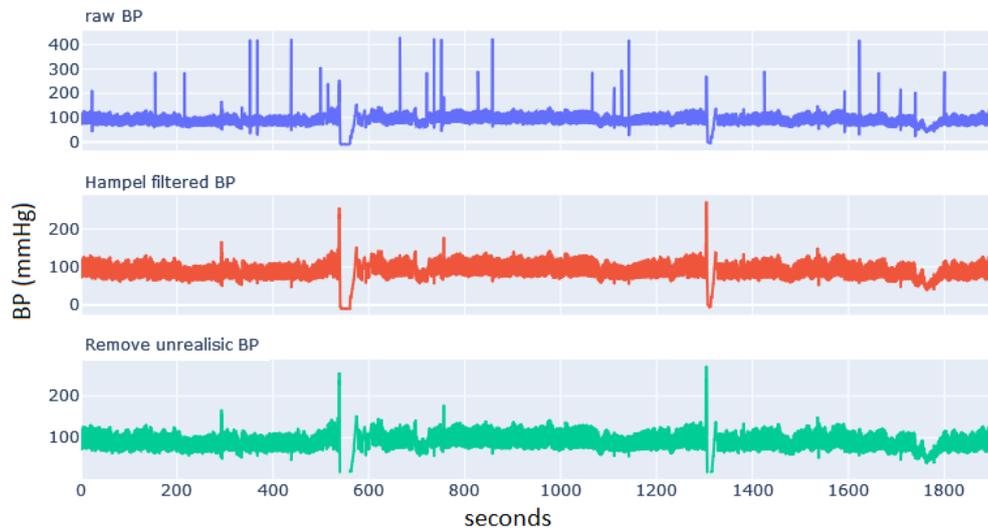


Figure 6: Removing unrealistic measurements from the ABP signal (blue line) of subject PHI037, this is a healthy older adult female. The first step is removing anomalies with the Hampel filter, the graph with the red line shows the removal of some anomaly peaks but still includes ABP measurements below zero. In the final step the unrealistic ($15 \text{ mmHg} < \text{ABP}$ or $\text{ABP} > 300 \text{ mmHg}$) measurements are removed.

4.1.2 Flatliner removal

After removing all the unexpected high peaks the flatliners need to be removed. Using the flatliner detection method defined in section 3.1.2 we can identify flatliners. In Figure 7, the result of the method shows that the method is capable of removing the flatliner from the data. The BP is removed in such a way that all incomplete BP waves are removed, an example can be seen in Figure 7. In Figure 7a, we can see that the flatliner starts in the middle of a BP curve, meaning that if we would only remove the flatliner it would leave half of the BP curve. Figure 7b shows an ABP signal after cleaning up any curves that contain flatliners. This leads to the models (described in sections 3.2 and section 3.3) receiving cleaner input data and not being biased by anomalies.

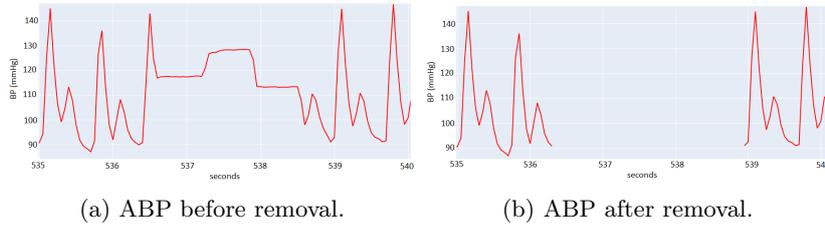


Figure 7: A sample of an ABP signal on 100Hz in the data set with a flatliner taken from a healthy young adult subject (PHI001). Here, we can see the ABP signal before (7a) and after (7b) the removal of the flatliner.

4.1.3 Extract SBP and DBP

Since we are mainly interested in the SBP and DBP we first need to extract the SBP and DBP from the ABP. In Figure 8, the result of the extraction can be seen and this shows that the SBP is the maxima of the ABP and the DBP is the minima of the ABP signal.

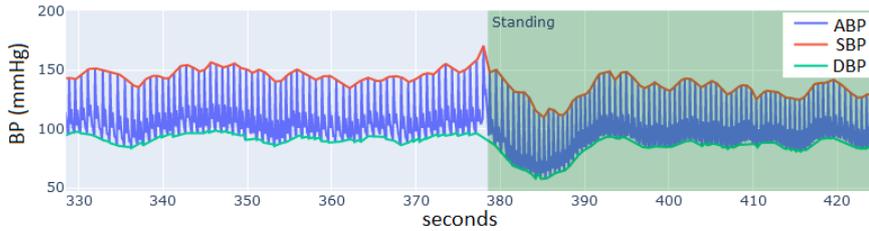


Figure 8: Extracting SBP (red) and DBP (green) from ABP (blue) for a healthy young adult subject (PHI001). Extracting SBP is done by taking the maxima of the ABP. To extract the DBP the minima of the ABP are retrieved. In this figure we see a transition from sitting (white background) to standing (green background).

4.1.4 Outlier detection

In Figure 9, we can see the results of the four outlier detectors: K-nearest neighbours (Figure 9a), interquartile range (Figure 9b), isolation forest (Figure 9c) and local outlier factor (Figure 9d). Red points are the detected outliers, and we can see that the majority of points in the drop of blood pressure are specified as outliers. These are not the outliers we are looking for, and we conclude that the selected outlier detectors are not well-suited for this application. For this reason the outlier detection is excluded from the data cleaning pipeline (Figure 4). Instead we can use the Hampel filter to remove unrealistic high peaks in the data that are significantly different from the other measurements.

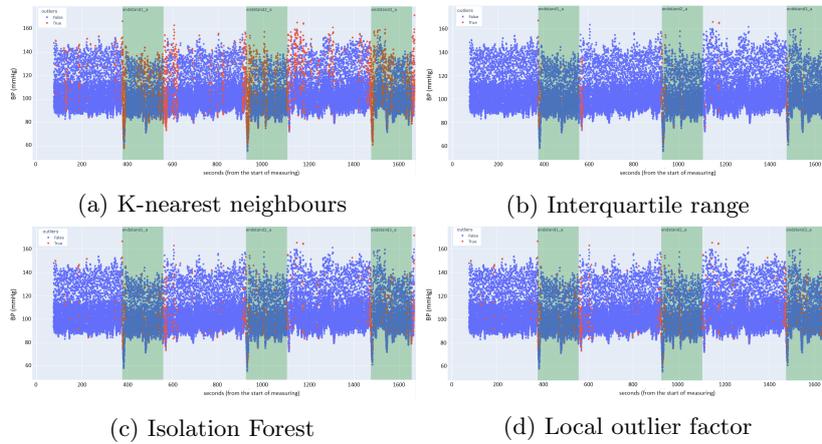


Figure 9: Outlier detection methods applied to the ABP on a healthy young adult subject (PHI001), identifying non-stationary measurements as outliers (red). Most BP measurements during standing up (before green square) and sitting down (after green square) are marked as outliers, while most others are marked as normal (blue).

Another possible way to filter outliers is to use smoothing. The smoothing method chosen was the Butterworth smoothing [15], this is a low-pass filter. The difference between the raw signal and one smoothed with Butterworth can be seen in Figures 10 and 12. Smoothing for NIRS and BP are separately done. The Oxy and Dxy are filtered (Figure 10) using a filter with lower bound 0.01Hz, upper bound 0.5Hz and polynomial order two [20].

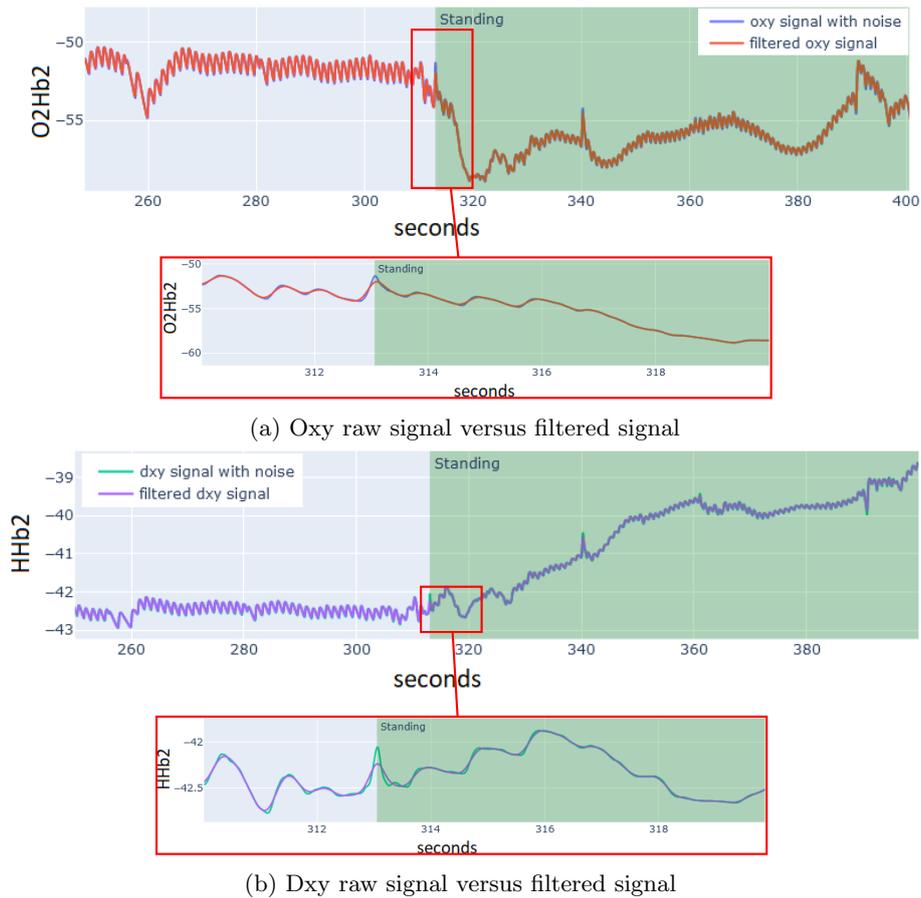


Figure 10: The NIRS signals are filtered with a Butterworth filter with lower bound 0.01Hz, upper bound 0.5Hz and polynomial order two. This removes sudden peaks and makes the data have less variability. This example is for subject PHI038, an older adult male showing OH characteristics.

Normally BP is smoothed by taking a rolling window of five seconds and averaging it [60]. The downside is that this will lead to some shift in the data, for this reason, the Butterworth filter was used. In Figure 11, we can see the delay caused by taking the average of the five-second rolling window. The Butterworth filter is very similar but without the delay.

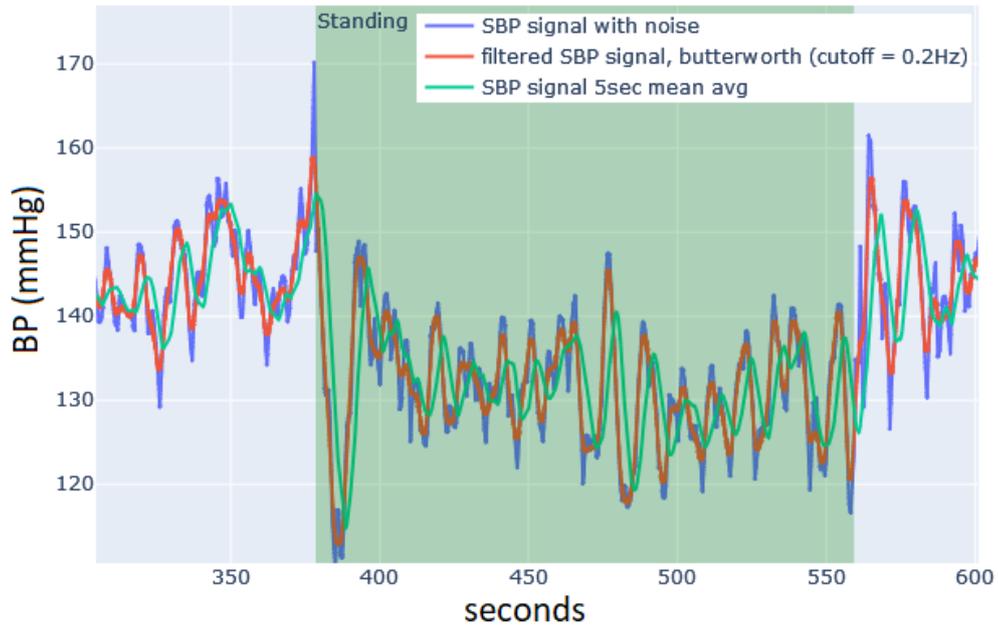
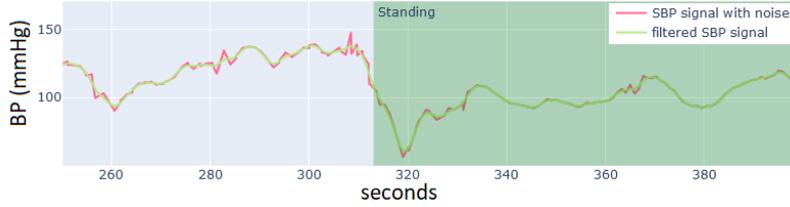
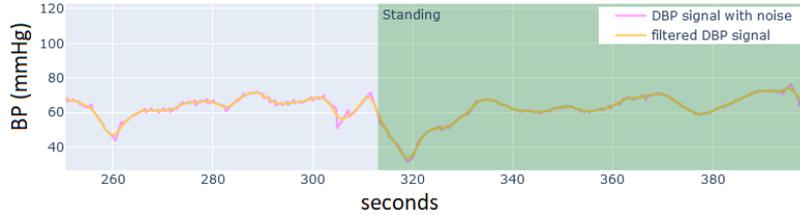


Figure 11: Comparing rolling average versus Butterworth smoothing, in SBP at 100Hz for subject a healthy young adult male subject (PHI001). The golden standard, filtering BP by taking the average over a 5-second window shows a delay. Filtering with a Butterworth filter shows a similar result without the delay, the Butterworth filter used an upper bound of 100Hz, a lower bound 0.2Hz and the polynomial order of two.

The SBP and DBP are filtered with 0.2Hz which is one measurement for every five seconds. In Figure 12, we can see how the filtered SBP and DBP changed after the smoothing is applied.



(a) SBP raw signal versus filtered signal



(b) DBP raw signal versus filtered signal

Figure 12: The SBP and DBP is filtered with a Butterworth filter with lower bound 0.2Hz, upper bound 100Hz and polynomial order two. The data in this figure is from subject PHI038, an older adult male showing OH characteristics. The smoothing removes irregular peaks to make the data less erratic.

4.2 Standing-standing task

For the standing to standing models, we used 40 seconds of baseline (sitting or supine) with 150 seconds of standing data as input data. The input data is used to infer the orthostatic parameters for the SBP and DBP in the case of the parameter models. With the mapping model, we use the input data to map the Oxy and Dxy directly to SBP and DBP, trying to map one time-step of Oxy and Dxy to SBP and DBP of the same time-step. The first step was to see if the orthostatic parameters can be inferred directly from the SBP and DBP.

4.2.1 BP input

The models (LR, XGB, MLP, CNN) described in section 3.2 are trained using SBP and DBP signals in baseline and standing as input, and they predict the orthostatic parameters.

Table 3: Results of leave one out cross-validation (LOO-CV), showing linear regression (LR) achieving the lowest loss. Input are the SBP and DBP signals containing the standing part of the experiment.

Name	average LOO-CV MAE	average LOO-CV MSE
XGBoost	8.02	$1.44 \cdot 10^2$
CNN	$1.04 \cdot 10$	$2.16 \cdot 10^2$
MLP	$1.62 \cdot 10$	$5.70 \cdot 10^2$
Linear regression	4.30	$4.62 \cdot 10$

In Table 3, we can see that the LR model has the lowest average CV MAE and MSE in comparison to the other models for this task. In Figure 13, we can see a few examples of the predicted orthostatic parameters against the true orthostatic parameters. This shows that the predicted orthostatic parameters are close to the actual orthostatic

parameters, but some are easier to extract than others. Especially the recovery values (Figures 13e, 13f, 13d) are easier for the model to learn while the drop time (Figure 13b) and steepness (Figure 13c) are significantly harder. The estimated orthostatic parameters can be used to reconstruct the SBP and DBP curves and an example of this can be seen in Figure 14, the SBP and DBP curves are reconstructed for an older adult subject (PHI014) with OH characteristics.

In this section we have shown that using the SBP and DBP we can estimate the specified orthostatic parameters with the parameters estimation models in Table 2. With the underlying thought that the Oxy and Dxy can be used as a proxy for the SBP and DBP, the next step is to use NIRS input.

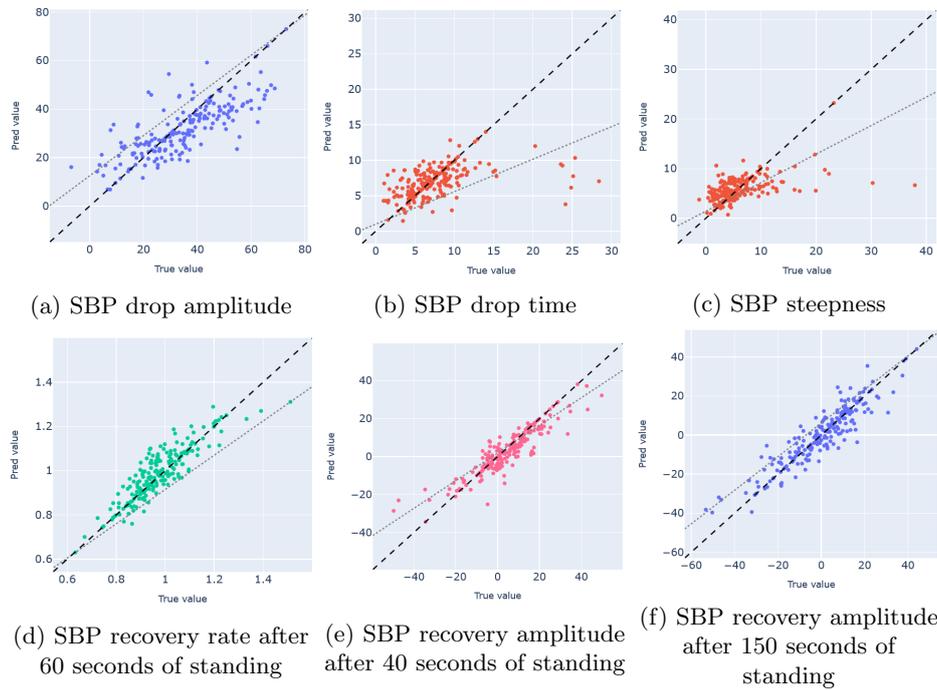


Figure 13: The comparison of some of the true orthostatic parameters versus predicted orthostatic parameters using SBP and DBP input. The parameters are predicted by the best performing model (linear regression) for the standing-standing task with BP as input. The prediction is on the y-axis and the true orthostatic parameters are on the x-axis. In the ideal situation, all points would lay on the dashed black line, the grey dotted line is the trend-line.

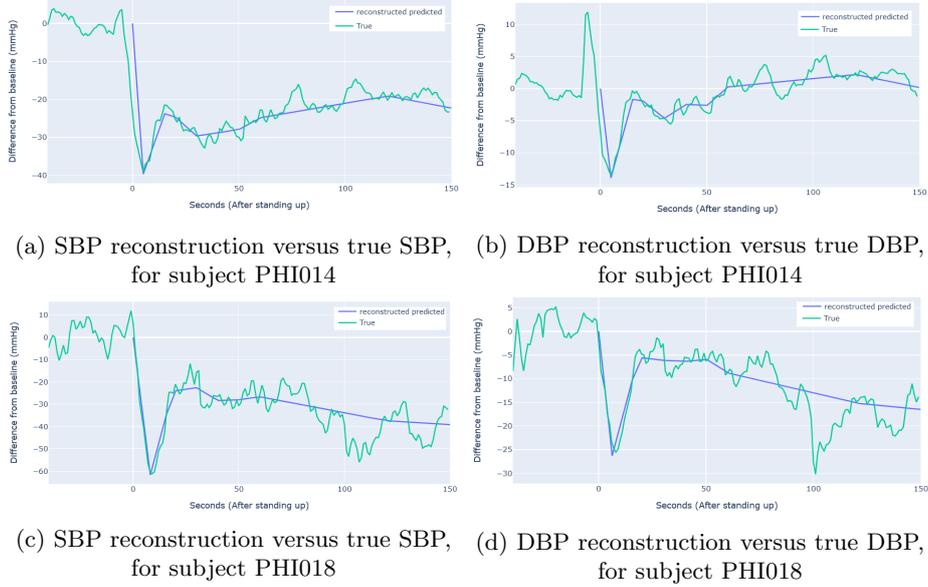


Figure 14: Reconstruction of the SBP and DBP based on the estimated orthostatic parameters. Prediction from the best performing (linear regression) model with BP input. The predicted parameters are used to estimate the true SBP and DBP (green line) and can be used to draw a the predicted SBP and DBP (blue line).

4.2.2 NIRS input

Now the model is fed the Oxy and Dxy signals and with these signals, we estimate the orthostatic parameters for the SBP and DBP. Using the same models as for the standing-standing task with BP as input, we can see the results in Table 3. The higher losses reflect that the task is harder, but in this case, the XGB model has the lowest average LOO-CV MAE and MSE. The comparison in Figure 15 reflects the higher losses in comparison to Figure 13 and it shows that the model is not capable of predicting rare orthostatic parameter values. In Figure 16a and Figure 16c, we can see that the results differ significantly for each subject. Performing well on one and worse on another.

Table 4: Results of leave one out cross-validation (LOO-CV) for orthostatic parameter estimation. The input is the Oxy and Dxy signal containing the standing part of the experiment. XGBoost has the lowest average MAE and MSE loss for the estimation of orthostatic parameters, while the time-series MLP is incapable of mapping Oxy and Dxy to SBP and DBP.

Experiment	Name	average LOO-CV MAE	average LOO-CV MSE
Parameter estimation	XGBoost	8.93	$1.79 \cdot 10^2$
	MLP	$1.07 \cdot 10$	$2.35 \cdot 10^2$
	CNN	$1.02 \cdot 10$	$2.31 \cdot 10^2$
	Linear regression	$1.07 \cdot 10$	$2.79 \cdot 10^2$
full curve mapping	Time-series MLP	$4.37 \cdot 10^{10}$	$2.04 \cdot 10^{22}$

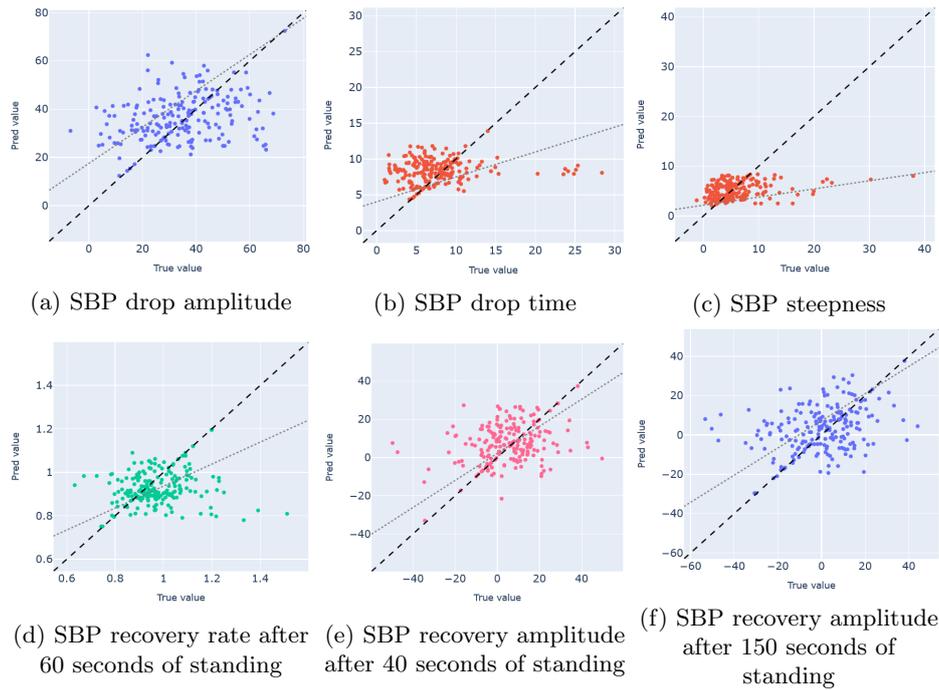


Figure 15: Comparison of the lowest loss model (XGBoost) orthostatic parameter prediction made using Oxy and Dxy input signal versus the true orthostatic parameters. The orthostatic parameter prediction is on the y-axis and the true orthostatic parameters are on the x-axis. We can see that the orthostatic parameter predictions are grouped together and that it is unable to predict the rarely occurring cases. In the ideal situation all point would lay on the dashed black line, the grey dotted line is the trend-line.

To make the results easily visible and interpretable the SBP and DBP curves are reconstructed using the estimate orthostatic parameters (Figure 16) and compared to the true SBP and DBP signals, again for the same subjects as used in Figure 14.

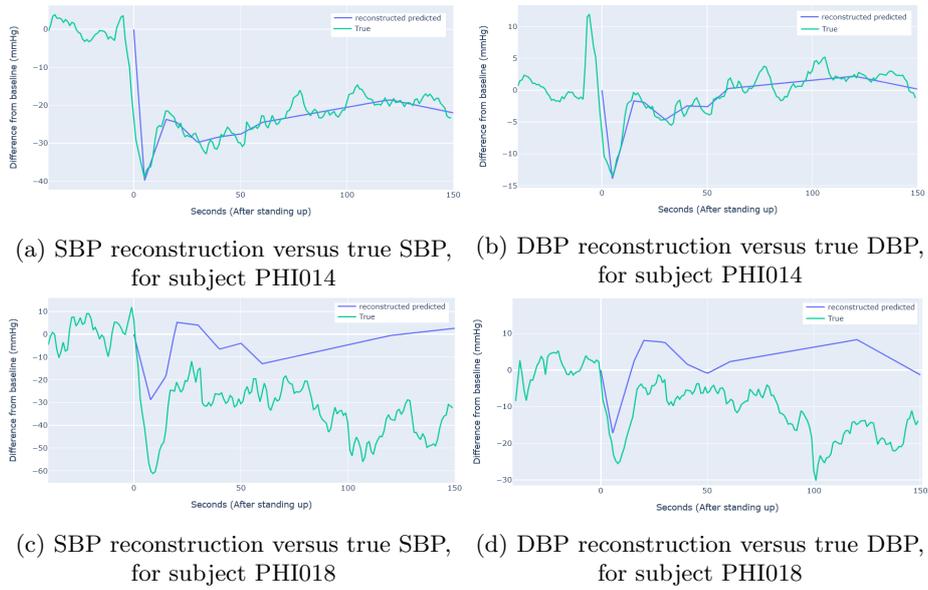
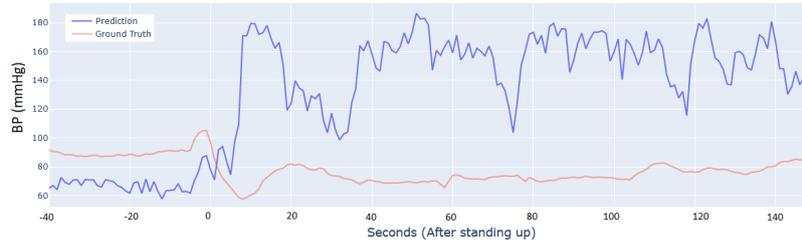
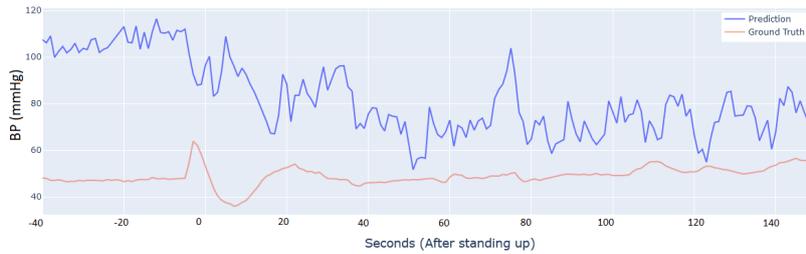


Figure 16: Reconstruction of the SBP and DBP based on the estimated orthostatic parameters. Prediction from the best (XGBoost) model with NIRS input. The predicted parameters are used to estimate the true SBP and DBP (green line) and can be used to draw the predicted SBP and DBP (blue line). Using NIRS the reconstruction of the SBP and DBP signal using the orthostatic parameters works well for one subject while performing worse on others. PHI014 is an older adult male showing OH characteristics and PHI018 is a healthy older adult female.

The last task using standing-standing data is to see if we can map Oxy and Dxy to SBP and DBP in the same time window, this model would learn how to combine the Oxy and Dxy measurement time-step to output the SBP and DBP for the same time-step. The full curve mapping was an experiment to see how well a general mapping could be made to go from Oxy and Dxy to SBP and DBP. As the results in Table 4 show, this was unsuccessful and reports a very high loss in comparison to the parameter estimation models. An example of this mapping can be seen in Figure 17, the SBP curve (Figure 17a) is not showing the drop at all. Due to the bad performance of mapping when making a non-subject-specific model, the decision was made to make use of subject-specific models with the time-series prediction task.



(a) SBP



(b) DBP

Figure 17: Mapping Oxy and Dxy to SBP and DBP with a time-series MLP for subject PHI014, an older adult male showing OH characteristics. Showing that the model is not capable of mapping Oxy and Dxy directly to SBP and DBP. We can observe that the predicted BP curve does not follow the same behaviour as the true BP curve and it even has completely different values.

4.3 Time-series prediction task

The last experiment performed in this research is to see if it is possible to reach the ultimate goal of the ProHealth project: to be able to detect if OH will occur when standing up, to take precautions or warn people nearby. In this section, we will look into the models specified in Table 2, which are used to predict future SBP and DBP based on past Oxy and Dxy time-series. We start with a model trained for a healthy young adult male subject (PHI001), the results in Table 5 show that the best performing model is the encoder-decoder LSTM architecture to convert past Oxy and Dxy to future Oxy and Dxy with an encoder-decoder with attention LSTM mapper is to translate the future Oxy and Dxy to SBP and DBP. The other models all resulted in higher loss values especially in the average LOO-CV MSE. For this reason, the focus will be on the encoder-decoder LSTM architecture with an encoder-decoder with attention LSTM mapper.

Table 5: Results of the time-series prediction for a healthy subject (PHI001), with the encoder-decoder architecture with an encoder-decoder attention mapper giving the lowest loss. Using 240 seconds of past Oxy and Dxy to forecast 120 seconds of future SBP and DBP. Results of leave one out cross-validation (LOO-CV).

Mappers	Architecture	avg LOO-CV MAE	avg LOO-CV MSE
MLP	encoder-decoder LSTM	3.95	$4.52 \cdot 10$
RNN	NBeats	5.65	$6.10 \cdot 10$
GRU	encoder-decoder LSTM	3.69	$3.96 \cdot 10$
LSTM	encoder-decoder LSTM	3.97	$3.67 \cdot 10$
CNN	encoder-decoder LSTM	3.96	$2.35 \cdot 10$
BiLSTM	NBeats	5.74	$6.88 \cdot 10$
stackedLSTM	encoder-decoder LSTM	5.37	$5.57 \cdot 10$
stackedBiLSTM	NBeats	6.37	$8.23 \cdot 10$
encoder-decoder LSTM	encoder-decoder LSTM	3.87	$4.28 \cdot 10$
encoder-decoder LSTM with attention	encoder-decoder LSTM	3.30	$3.30 \cdot 10$

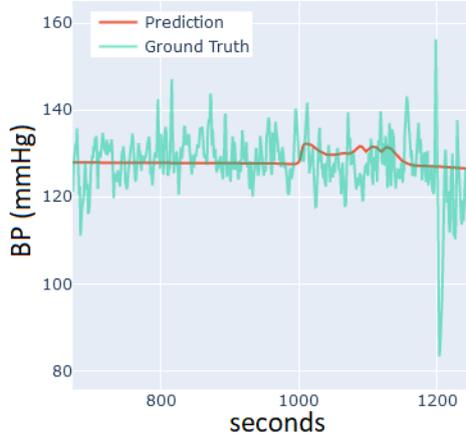


Figure 18: Results of the SBP signal time-series prediction for a healthy young adult male subject (PHI001). The SBP signal during one repetition of the challenge is forecasted over time using the Oxy and Dxy signals. Showing the results of the best model (encoder-decoder LSTM architecture with encoder-decoder with attention LSTM mapper). Using 240 seconds of past Oxy and Dxy to forecast 120 seconds of future SBP and DBP.

In Figure 18, an example of the prediction for a healthy subject can be seen. Since ProHealth is mainly focused on OH subjects we also used a subject who was diagnosed with OH during the project and see how the models (as described in section 3.3) perform on the OH subject. The results overall models (Table 6) show that in this case the LSTM encoder-decoder architecture again performs best, but this time with the CNN mapper. A prediction made by this model can be seen in Figure 19.

Table 6: Results of the SBP signal time-series prediction for an older adult male subject with OH characteristics (PHI014), with the encoder-decoder architecture with a CNN mapper giving the lowest loss. Using 240 seconds of past Oxy and Dxy to forecast 120 seconds of future SBP and DBP. Results of leave one out cross-validation (LOO-CV).

Mappers	Architecture	avg LOO-CV MAE	avg LOO-CV MSE
MLP	encoder-decoder LSTM	1.93	$1.99 \cdot 10$
RNN	encoder-decoder LSTM	1.87	$1.98 \cdot 10$
GRU	encoder-decoder LSTM	1.90	$1.85 \cdot 10$
LSTM	encoder-decoder LSTM	1.96	$2.26 \cdot 10$
CNN	encoder-decoder LSTM	1.75	$1.51 \cdot 10$
BiLSTM	encoder-decoder LSTM	2.01	$2.20 \cdot 10$
stackedLSTM	encoder-decoder LSTM	1.94	$2.15 \cdot 10$
stackedBiLSTM	encoder-decoder LSTM	1.99	$2.05 \cdot 10$
encoder-decoder LSTM	encoder-decoder LSTM	1.90	$1.95 \cdot 10$
encoder-decoder LSTM with attention	encoder-decoder LSTM	1.91	$2.03 \cdot 10$

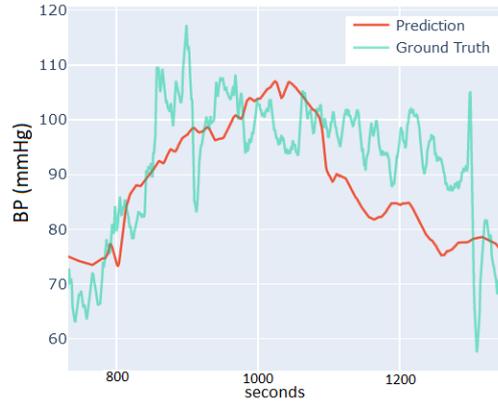


Figure 19: Results of the SBP time-series prediction for an older adult male subject with OH characteristics (PHI014). The SBP signal during one repetition of the challenge is forecasted using the Oxy and Dxy signals. Showing the results of the best model (encoder-decoder LSTM architecture with CNN mapper). Using 240 seconds of past Oxy and Dxy to forecast 120 seconds of future SBP and DBP.

Since a model is optimized for each subject separately we are comparing the best model created for one subject on the other subject. The best model layout of the healthy subject which is an encoder-decoder LSTM architecture with encoder-decoder with attention LSTM mapper is used on the OH subject. And the best model layout for the OH subject which is an encoder-decoder LSTM architecture with CNN mapper is used on our healthy subject. Figure 20 shows that if we use the same hyperparameters and layers as the best-optimized model from the OH subject (PHI014) for our healthy subject (PHI001) the prediction improves significantly. This is because Optuna test a random set of hyperparameters and the model could be sub-optimal. On the other hand, if we use the model optimized for our healthy subject (PHI001) for our OH subject (PHI014), we see a prediction which is a close estimation but further away from the true signal than when the CNN mapper is used. In section 5 we will go into the reason behind this issue.

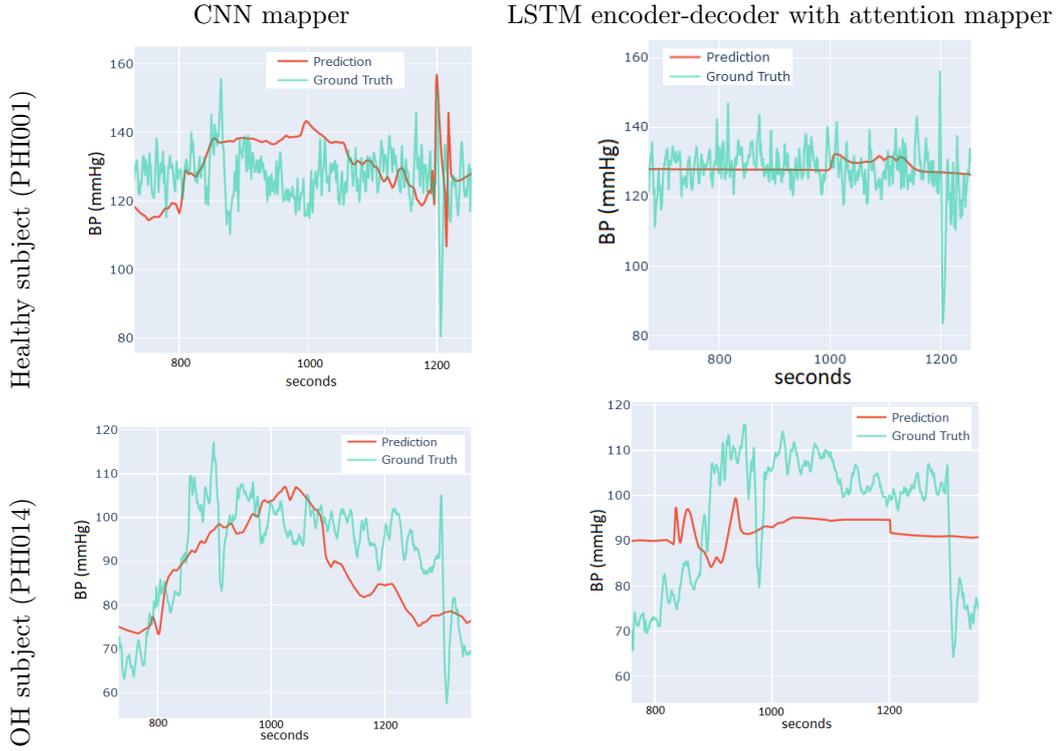


Figure 20: Results for SBP time-series prediction, comparing the results of the optimized models. The optimized CNN mapper, which is the best on an older adult subject with OH characteristics (PHI014), is performing better on the healthy subject (PHI001) than the optimized encoder-decoder with attention mapper. Using 240 seconds of past Oxy and Dxy to forecast 120 seconds of future SBP and DBP.

5 Discussion

In this section we will go over all the results presented in section 4 and other discussion points. The first thing to note is the non-homogeneous data, where we have only three subjects who would be diagnosed with OH, and the subjects are not evenly divided between male (56%) and female (44%). For this reason, during orthostatic parameter estimation we do not estimate directly for OH, but instead, predict the characteristics of the drop after standing up. The model used for the time-series prediction contains very little data (three supine-stand transitions), but was chosen to be subject-specific to not introduce a bias caused by the non-homogeneous data.

During preprocessing we first perform the unrealistic measurements removal. The performed method defined in section 3.1.1 removes a lot of unrealistic peaks, which are most likely caused by measurement errors. After removing anomaly ABP measurements with the Hampel filter and unrealistic measurements filter, there are still some remaining peaks around the removed unrealistic measurements (Figure 6), but this was not the case in all subjects. Since the peaks only remained in a few subjects the data is not cut off further to not remove too many data points. The anomalies happened only in the sitting or supine part of the data, because the clinician who provided the data already removed the subjects where the standing part of the data was affected.

The ABP signal still contains flatliners caused by the calibration, which might cause issues when extracting SBP and DBP signals. Performing the flatliner removal method defined in section 3.1.2 we can remove all of the flatliners, but this might still lead to some false positive hits, where non-flatliners are removed from the data. To limit this, there was a limit set that a maximum of 20 plateaus that can be removed. Both of these anomalies are also addressed in [45], where they remove all experiments containing these anomalies. If this was performed in this research as well, we would remove more data from the already sparse data set. For this reason, an implementation was made to handle the anomalies instead of removing them. The outlier detection methods described in section 3.1.4 show that the time-series-based outlier detection methods are not well suited when using a time-series with a sudden change, in our case caused by standing up. Most of the points which do not follow the BP curve in rest are marked as outliers which happens when standing up. Since the ABP is not relevant when diagnosing OH, the SBP and DBP are extracted from the ABP instead. The method used for extracting SBP and DBP is capable of marking all maxima and minima in the ABP signal with great precision and can handle missing data by ignoring gaps and interpolating between the found points.

The extracting of the orthostatic parameters is done by using the markers provided in the data set. These markers are time points when the subject is standing up and sitting down again. These could be delayed or ahead of time since they are measured by hand. Without a way to check for the offset, there is no method to ensure the correctness of a marker without taking brute assumptions. For this reason, the markers are assumed to be right regardless and extract the data based on the markers. In Figure 13a and Figure 15a, a measurement is shown where the drop is negative, meaning that the SBP only goes up and most likely is already recovering. This could be due to the marker being incorrect and the drop happening before the marked time. Since we already suffer from sparse data, these measurements are not removed.

The models used for parameter estimation in the standing-standing task (section 3.2.2) show promising results. The models used to extract orthostatic parameters from the SBP and DBP showed very good results in practice, but this would not be useful and is only included to see if it is possible to extract the orthostatic parameters from the SBP and DBP signals. These models perform well, Figure 13 shows that the model is capable of estimating the orthostatic parameters and could even estimate the rarer values within the range of the orthostatic parameters. The models which use the Oxy and Dxy signals, are clinically relevant and could be used to make an improved diagnosis. These models show to perform well on the more frequently occurring ranges (Figure 15) and worse on the less occurring ranges, but before we can make a conclusion, we need to experiment with bigger balanced data sets to be able to predict the now less common ranges. As we can see in Figure 14 and Figure 16 the estimated orthostatic parameters can be used to reconstruct the SBP and DBP curves. The reconstructed SBP and DBP

signals are for some subjects very close to the true SBP and DBP signals (Figure 16a and Figure 16b). For other subjects, this model predicts inaccurate orthostatic parameters, resulting in an inaccurate SBP and DBP signal that can lead to wrong conclusions, see Figure 16c and Figure 16d. A more detailed research is required but at first glance the difference seems to originate from the difference in Oxy and Dxy range, where the pattern of Oxy and Dxy is less common for the subject with the incorrect prediction. These reconstructed signals together with the other estimated orthostatic parameters can be used to make a diagnosis if the subject is suffering from OH. However, the model should be improved by training the model with more subjects to make it better at predicting the now incorrect cases.

In Figure 17, we showed that our performance of the time-series prediction task to go from Oxy and Dxy signals to the SBP and DBP curve (section 3.2.3) is not high. We conclude that this task is very difficult and it might be impossible to predict. This may be because there is no general mapping to go from Oxy and Dxy to SBP and DBP and the signals are very noisy, and with this, we could see what happens if a subject-specific mapper is trained.

On the other hand, we can see that the time-series prediction task only approaches the true SBP and DBP curve without being able to precisely capture the more detailed changes in the SBP (and DBP) signal. In Figure 20, we can see that the model trained on the OH subject (PHI014) is performing better on the healthy subject (PHI001) than the optimized model for the healthy subject (PHI001). The optimization is done by Optuna. This can be compared with a grid search and it is possible that the best hyperparameter combination is not tested. Due to time limitations, we limited the number of optimization trials to 50, but this could be increased to result in better models. If one would use a subject-specific model one could optimize the available models and always use the same optimized models instead of having a separate optimization for each subject. The models could share the same hyperparameters and by training it for each subject.

Before being able to use the orthostatic parameter estimation models in practice we would need to train them on more data. Since the model currently seems to be unable to learn a pattern for all subjects. The results showed that we can use it for some subjects and that it is capable of making a good estimation of the SBP and DBP curves, see Figure 16. To solve this problem training the models on more data would seem the obvious next step.

6 Conclusion

The use of Oxy and Dxy signals to diagnose OH was researched in this paper. We use the Oxy and Dxy signals to infer information from the SBP and DBP signals. This would allow possible OH subjects to use NIRS devices at home, during their day-to-day activities and make a diagnosis based on a period of eight hours instead of a snapshot taken at the hospital. The results show that a model can be trained to estimate orthostatic parameters which are important for diagnosing OH. The answer to the research question “Can we use machine learning to help diagnose orthostatic hypotension from NIRS signals?”. The orthostatic parameter estimation from Oxy and Dxy is feasible but is not very accurate for all subjects, this can be seen in Figure 15 and Figure 16. These results would need to be discussed with a clinician to determine the biological difference between subjects, which helps to explain the difference in performance. If this is indeed caused by a biological difference, a balanced data set with more subjects

should be able to improve the result. Another point which requires some improvement is model selection. The models hyperparameters are selected randomly by Optuna, with a limited number of combinations tested. Due to time limitations the limit of number of combinations was set to 50, but with more tested combinations or manually testing hyperparameters could result in a better hyperparameter combination. Which could in turn improve the results of the models. The time-series mapping model to directly map Oxy and Dxy to SBP and DBP in the standing-standing task is performing poor and getting very high loss values (Table 4). This shows that it is impossible to make a general mapping from Oxy and Dxy to SBP and DBP, see Figure 17. Fortunately, the parameter estimation models allow clinicians to improve their diagnosis if the subject has OH and hope to save older adults from falling injuries, since this diagnosis can now be based on a longer period. The models used for the time-series prediction task are unable to predict more than the general pattern (Figure 20) the SBP and DBP will follow and this needs more research to see if the models can be improved further. For further research, it would be interesting to see if it is possible to have a subject unique model that can map Oxy and Dxy directly to SBP and DBP, instead of a general model for all subjects. If improving the time-series predicting models does not seem to work, the possibility to predict the future orthostatic parameters of the SBP and DBP curves can be explored. If the Oxy and Dxy prove to be inadequate to predict the SBP and DBP the use of photoplethysmogram and electrocardiogram signals could be considered as explored in Paviglianiti et al. [45].

Acknowledgement

I would like to thank Mirthe van Diepen and Theodore Nikolettopoulos for their guidance and support during the thesis and helping me when it was needed. Thanks to Tom Heskens and Tom Claassen for taking their time to give me feedback and grade the thesis.

References

- [1] A gentle introduction to XGBoost for applied machine learning. <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>.
- [2] Low blood pressure (orthostatic hypotension): Causes, symptoms treatments. <https://my.clevelandclinic.org/health/diseases/9385-low-blood-pressure-orthostatic-hypotension>.
- [3] Mlflow - a platform for the machine learning lifecycle — mlflow. <https://mlflow.org>.
- [4] Optuna - a hyperparameter optimization framework. <https://optuna.org>.
- [5] PROHEALTH project - measuring cerebral blood flow. <https://www.orikami.ai/prohealth-project/>.
- [6] Ayaz Ahmad, Furqan Farooq, Pawel Niewiadomski, Krzysztof Ostrowski, Arslan Akbar, Fahid Aslam, and Rayed Alyousef. Prediction of compressive strength of fly ash based concrete using individual and ensemble algorithm. *Materials 2021, Vol. 14, Page 794*, 14:794, 2 2021.

- [7] Amy C. Arnold and Satish R. Raj. Orthostatic hypotension – a practical approach to investigation and management. *The Canadian journal of cardiology*, 33(12):1725, dec 2017.
- [8] Alberto Arrigoni. Deepar. <https://github.com/arrigoni/alberto86/deepar>, 2021.
- [9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2015.
- [10] Y Bengio, Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, J J Urgan Schmidhuber, and In S C Kremer. Gradient flow in recurrent nets: The difficulty of learning long-term dependencies. 2003.
- [11] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. 2011.
- [12] I. Biaggioni and H. Kaufmann. Orthostatic hypotension. *Encyclopedia of the Neurological Sciences*, pages 698–700, jul 2021.
- [13] Johan Bresseleers, Ilse Van Diest, Steven De Peuter, Peter Verhamme, and Omer Van Den Bergh. Feeling lightheaded: The role of cerebral blood flow. *Psychosomatic Medicine*, 72:672–680, 2010.
- [14] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, 2000.
- [15] Stephen Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.
- [16] Wenbing Chang, Yinglai Liu, Yiyong Xiao, Xinglong Yuan, Xingxing Xu, Siyue Zhang, and Shenghan Zhou. A machine-learning-based prediction method for hypertension outcomes based on medical data. *Diagnostics*, 2019.
- [17] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-August-2016:785–794, aug 2016.
- [18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1724–1734, 6 2014.
- [19] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.
- [20] Xu Cui, Signe Bray, and Allan L. Reiss. Functional near infrared spectroscopy (nirs) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics. *NeuroImage*, 49:3039, 2 2010.
- [21] Rianne A.A. de Heus, Daan L.K. de Jong, Anne Rijpma, Brian A. Lawlor, Marcel G.M. Olde Rikkert, and Jurgen A.H.R. Claassen. Orthostatic blood pressure recovery is associated with the rate of cognitive decline and mortality in clinical alzheimer’s disease. *The journals of gerontology. Series A, Biological sciences and medical sciences*, 75:2169–2176, 11 2020.

- [22] Centers for Disease Control and Prevention. Measuring orthostatic blood pressure. <https://www.cdc.gov/steady/pdf/STEADI-Assessment-MeasuringBP-508.pdf>, 2017.
- [23] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.
- [24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997.
- [25] Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. 2015.
- [26] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 4 2021.
- [27] Franziska Klein and Cornelia Kranczioch. Signal processing in fnirs: A case for the removal of systemic activity for single trial data. *Frontiers in Human Neuroscience*, 13:331, 9 2019.
- [28] M Klop. Physical resilience indicators derived from haemodynamic signals in older adults : relation to cognitive status, cognitive decline and surgical outcome, 9 2020.
- [29] William J. Kostis, Davit Sargsyan, Choukri Mekkaoui, Abel E. Moreyra, Javier Cabrera, Nora M. Cosgrove, Jeanine E. Sedjro, John B. Kostis, William C. Cushman, John S. Pantazopoulos, Sara L. Pressel, and Barry R. Davis. Association of orthostatic hypertension with mortality in the systolic hypertension in the elderly program. *Journal of Human Hypertension*, 33(10):735, oct 2019.
- [30] J. Brian Lanier, Matthew B. Mote, and Emily C. Clay. Evaluation and management of orthostatic hypotension. *American Family Physician*, 84(5):527–536, sep 2011.
- [31] Pedro Lara-Benítez, Manuel Carranza-García, and José C. Riquelme. An experimental review on deep learning architectures for time series forecasting. *International Journal of Neural Systems*, 31, 3 2021.
- [32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 12 1989.
- [33] Fei Tony Liu, Kai Ming Ting, and Zhi Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6:39, 3 2012.
- [34] Erick Martínez-Ríos, Luis Montesinos, Mariel Alfaro-Ponce, and Leandro Pecchia. A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data. *Biomedical Signal Processing and Control*, 68:102813, 7 2021.
- [35] Andrea S. Méndez, Jesús D. Melgarejo, Luis J. Mena, Carlos A. Chávez, Alicex C. González, José Boggia, Joseph D. Terwilliger, Joseph H. Lee, and Gladys E. Maestre. Risk factors for orthostatic hypotension: Differences between elderly men and women. *American Journal of Hypertension*, 31(7):797, jun 2018.
- [36] Arjen Mol, Esmee M. Reijnierse, Marijke C. Trappenburg, Richard J.A. Van Wezel, Andrea B. Maier, and Carel G.M. Meskers. Rapid systolic blood pressure changes after standing up associate with impaired physical performance in geriatric outpatients. *Journal of the American Heart Association*, 7, 11 2018.

- [37] Arjen Mol, Lois R.N. Slangen, Richard J.A. Van Wezel, Andrea B. Maier, and Carel G.M. Meskers. Orthostatic blood pressure recovery associates with physical performance, frailty and number of falls in geriatric outpatients. *Journal of hypertension*, 39:101–106, 1 2021.
- [38] Filippo Molinari, William Liboni, Gianfranco Grippi, and Emanuela Negri. Relationship between oxygen supply and cerebral blood flow assessed by transcranial doppler and near - infrared spectroscopy in healthy subjects during breath - holding. *Journal of NeuroEngineering and Rehabilitation*, 3:1–13, 7 2006.
- [39] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2:183–197, 7 1991.
- [40] M. D. Musameh, C. P. Nelson, J. Gracey, M. Tobin, M. Tomaszewski, and N. J. Samani. Determinants of day–night difference in blood pressure, a comparison with determinants of daytime and night-time blood pressure. *Journal of Human Hypertension*, 31:43, 1 2017.
- [41] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 0(DEC):21, 2013.
- [42] Mette S. Olufsen, Johnny T. Ottesen, Hien T. Tran, Laura M. Ellwein, Lewis A. Lipsitz, and Vera Novak. Blood pressure and blood flow variation during postural change from sitting to standing: Model development and validation. *Journal of applied physiology (Bethesda, Md. : 1985)*, 99:1523, 10 2005.
- [43] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio Mila. N-beats: Neural basis expansion analysis for interpretable time series forecasting. 5 2019.
- [44] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013*, pages 2347–2355, 11 2012.
- [45] Annunziata Paviglianiti, Vincenzo Randazzo, Stefano Villata, Giansalvo Cirrione, and Eros Pasero. A comparison of deep learning techniques for arterial blood pressure prediction. *Cognitive computation*, 2021.
- [46] Ronald K. Pearson, Yrjö Neuvo, Jaakko Astola, and Moncef Gabbouj. Generalized hamper filters. *Eurasip Journal on Advances in Signal Processing*, 2016, 12 2016.
- [47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [48] Yang Peng and Huang Biao. Knn based outlier detection algorithm in large dataset. *2008 International Workshop on Education Technology and Training and 2008 International Workshop on Geoscience and Remote Sensing, ETT and GRS 2008*, 1:611–613, 2008.
- [49] Mariana R. Pioli, Alessandra M.V. Ritter, Ana Paula de Faria, and Rodrigo Modolo. White coat syndrome and its variations: differences and clinical impact. *Integrated Blood Pressure Control*, 11:73, 2018.
- [50] Philippe Remy. N-beats. <https://github.com/philipperemy/n-beats>, 2020.

- [51] Fabrizio Ricci, Raffaele De Caterina, and Artur Fedorowski. Orthostatic hypotension: Epidemiology, prognosis, and treatment. *Journal of the American College of Cardiology*, 66(7):848–860, aug 2015.
- [52] Heather L. Ristuccia, Paul Grossman, Lana L. Watkins, and Bernard Lown. Incremental bias in finapres estimation of baseline blood pressure levels over time. *Hypertension (Dallas, Tex. : 1979)*, 29:1039–1043, 1997.
- [53] Kathryn M Rose, Herman A Tyroler, Christopher J Nardo, Donna K Arnett, Kathleen C Light, Wayne Rosamond, A Richey Sharrett, and Moyses Szklo. Orthostatic hypotension and the incidence of coronary heart disease: The atherosclerosis risk in communities study. Technical report, 2000.
- [54] S. Saigal and D. Mehrotra. Performance comparison of time-series data using predictive data mining techniques. *Advances in Information Mining*, 4:57–66, 2012.
- [55] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, apr 2017.
- [56] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, 1997.
- [57] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404, 8 2018.
- [58] L Sunitha, M Balraju, J Sasikiran, and E Venkat Ramana. Automatic outlier identification in data mining using iqr in real-time data. *International Journal of Advanced Research in Computer and Communication Engineering*, 3, 2014.
- [59] Jack V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49:1225–1231, 11 1996.
- [60] Nathalie Van Der Velde, Anton H. Van Den Meiracker, Bruno H.Ch Stricker, and Tischa J.M. Van Der Cammen. Measuring orthostatic hypotension with the finometer device: Is a blood pressure drop of one heartbeat clinically relevant? *Blood pressure monitoring*, 12:167–171, 6 2007.
- [61] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

Appendices

A Standing-standing task

A.1 Hyperparameters for orthostatic parameter models using SBP and DBP input

Table 7: Hyperparameters used for the models trained with SBP and DBP to estimate orthostatic parameters.

Name	Hyperparameters
CNN	dropout=0.8, filters=172, kernel_size=7, pool_size=4, strides=5, number of dense layers after the CNN = 6, units = [1520, 1221, 922, 623, 324, 26]
MLP	dropout=0.2, number of layers = 3, units = [1520, 773, 26]
XGB	colsample_bylevel = 1, colsample_bytree=0.6, gamma=0.01, grow_policy=lossguide, max_bin=96, sampling_method=gradient_based, subsample=0.8

A.2 Hyperparameters for orthostatic parameter models using Oxy and Dxy input

Table 8: Hyperparameters used for the models trained with Oxy and Dxy to estimate orthostatic parameters.

Name	Hyperparameters
CNN	dropout=0.2, filters=201, kernel_size=8, pool_size=6, strides=3, number of dense layers after the CNN = 0
MLP	dropout=0.2, number of layers = 6, units = [1520, 1221, 922, 623, 324, 26]
XGB	colsample_bylevel = 0.1, colsample_bytree=0.4, gamma=0.01, grow_policy=lossguide, max_bin=512, sampling_method=gradient_based, subsample=1

B Time-series prediction task

B.1 Healthy subject model hyperparameters

Table 9: Hyperparameters used for the models trained with past Oxy and Dxy to predict future SBP and DBP for a healthy subject (PHI001).

Mapper	Mapper hyperparameters	Architecture	Architecture hyperparameters	Movement branch parameters
MLP	-	encoder-decoder LSTM	LSTM units=56	filters=112, kernel_size=60
BiLSTM	units=192, dropout=0.07	NBeats	stack_types=('generic', 'trend', 'seasonality'), nb_blocks_per_stack=1, thetas_dim=(1, 6, 6), hidden_layer_units=96	
CNN	filters=64, kernel_size=5	encoder-decoder LSTM	LSTM units=32	filters=88, kernel_size=80
encoder-decoder with attention LSTM	units=160, dropout=0.75	encoder-decoder LSTM	LSTM units=48	filters=72, kernel_size=30
encoder-decoder LSTM	units=32, dropout=0.01	encoder-decoder LSTM	LSTM units=40	filters=88, kernel_size=60
GRU	units=128, dropout=0.1	encoder-decoder LSTM	LSTM units=64	filters=96, kernel_size=80
LSTM	units=32, dropout=0.5	encoder-decoder LSTM	LSTM units=24	filters=120, kernel_size=30
RNN	units=160, dropout=0.8	NBeats	stack_types=('generic', 'trend', 'seasonality'), nb_blocks_per_stack=1, thetas_dim=(2, 2, 10), hidden_layer_units=32	
Stacked BiLSTM	units=32, dropout=0.46, stacks=2	NBeats	stack_types=('generic', 'trend', 'seasonality'), nb_blocks_per_stack=2, thetas_dim=(2, 4, 7), hidden_layer_units=192	
Stacked LSTM	units=32, dropout=0.02, stacks=2	encoder-decoder LSTM	LSTM units=8	filters=88, kernel_size=50

B.2 OH subject model hyperparameters

Table 10: Hyperparameters used for the models trained with past Oxy and Dxy to predict future SBP and DBP for a healthy subject (PHI001).

Mapper	Mapper hyperparameters	Architecture	Architecture hyperparameters	Movement branch parameters
MLP	-	encoder-decoder LSTM	LSTM units=48	filters=96, kernel_size=40
BiLSTM	units=192, dropout=0.14	encoder-decoder LSTM	LSTM units=24	filters=88, kernel_size=80
CNN	filters=192, kernel_size=6	encoder-decoder LSTM	LSTM units=64	filters=112, kernel_size=60
encoder-decoder with attention LSTM	units=256, dropout=0.21	encoder-decoder LSTM	LSTM units=56	filters=96, kernel_size=60
encoder-decoder LSTM	units=256, dropout=0.74	encoder-decoder LSTM	LSTM units=24	filters=112, kernel_size=50
GRU	units=96, dropout=0.45	encoder-decoder LSTM	LSTM units=32	filters=72, kernel_size=20
LSTM	units=192, dropout=0.42	encoder-decoder LSTM	LSTM units=40	filters=80, kernel_size=40
RNN	units=192, dropout=0.01	encoder-decoder LSTM	LSTM units=64	filters=120, kernel_size=50
Stacked BiLSTM	units=256, dropout=0.09, stacks=3	encoder-decoder LSTM	LSTM units=32	filters=112, kernel_size=80
Stacked LSTM	units=192, dropout=0.3, stacks=1	encoder-decoder LSTM	LSTM units=64	filters=112, kernel_size=20