Report Pervasive Computing Assignment 4

1) Data collection

Measurement tool: I downloaded the app “Sensor Logger” because it is able to track the acceleration in a 3-dimensional space and the sampling rate can be accurately chosen. Furthermore, the recorded tracks can be simply exported as csv-files. I used it for both devices.

Sampling rate for the recording: According to [2], 100 % of human movements shall lay below 20 Hz, 99 % below 15 Hz, and 98 % below 10 Hz. Beyond that, there is lot of human activity recognition research, which also tracks human body movements, which applies sampling rates in a range of 1 to 20 Hz [3]. Therefore, the sampling rate was set to **20 Hz** for both devices.

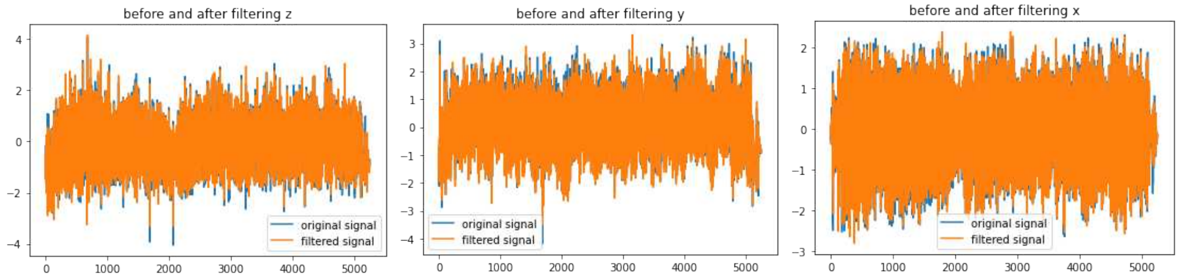
Data recording: I used two smartphones as devices. One I fixated on the wrist of the right arm where the screen pointed towards the body, the other one came into my right front trousers‘ pocket with the screen pointing in body movement direction. Both devices had were set upright. As it turned out, synchronizing the two devices can be quite struggling. I tried to hit the start buttons of the two devices simultaneously but of course this wasn‘t easily doable. The synchronziation will be done in the pre-processing step.
I successfully collected data for a time frame of 65 minutes at once. The street was part-wise iced such that I walked carefully sometimes. The right arm on which the device was fixed swung freely. To be sure that the second device registers the leg movements, I wore this time tight trousers.
To avoid data loss, I split the data collecting process into several recordings which follow chronological on each other. Because one time point represents one sample, I ended up with a **final dataset of 65825 samples after pre-processing.**

2) Pre-processing

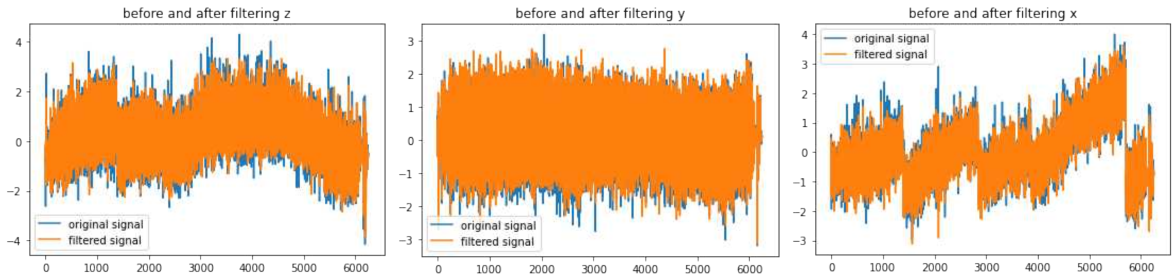
Cutting: For each signal I cut the first and last 10 seconds away to be sure that only the process of walking was recorded. It always took some time to position the devices. Moreover, it is important to synchronize the devices. The csv accelerometer files‘ first column represents the device‘s timestamps. In the lecture was mentioned that it is impossible that two devices have the exactly same current point in time. However, as I activated the recording buttons circa at the same time and we can assume that we don‘t need the point in time very precisely, I came to the conlusion to compare the first 11 digits of each timestamp of the recordings of the two devices. For instance, having the timestamp 1705060351195432700, only 17050603511 mattered for file comparison. To guarantee that the two devices record the same time interval, I had a look the first timestamps (after the 10 second cut-off) of the two simulatneously recordings and took the one which is later in time. This later in time timestamp marked the start of the two recordings.
Same line of thought accounts for the end of the accelerometer files. In this case the ending timestamp, which is earlier in time, marks the end of both device csv-files.
Through that cutting process and the usage of the same sample frequency the accelerometer files of the devices have the same amount of datapoints and are therefore suitable for the regression task.

Filtering: The low-pass filter enables the removal of noise and gravity acceleration [1]. I had a closer look at the low-pass **Butterworth filter**. As mentioned in the sampling rate choice, 99 % of human movenments shall be covered by 20 Hz [2]. Therefore, the filter threshold was set to that value. Besides that, [4] outlines that strong phase distortions might happen when an unsuitable filter order and frequency are selected. [3] also mentions that the filtering for each of the 3 dimensions shall happen separately.

I experimented with the filter setting. The order of the filter should be as high as possible, else I can observe strong shifts in the signal. Therefore, I ended up with an **order of 9 and a frequency of 20**. The following plots give an outlook of the filtered z-score normalized signal for one time recording.



Device A/wrist example snippet



Device B/right trousers‘ front pocket example snippet

It can be observed that the Butterworth filter doesn’t lead to a drastic distortion in time domain. Furthermore, clear differences in the signals between the two devices can be observed for all recorded time snippets. Device B tends to build a wave like structure whereas device A doesn’t do so.

Sample forumulation: To use the pre-processed data in WEKA, it was stored as arff-file. One sample/line was defined as (z-axis device A, y-axis device A, x-axis device A, z-axis device B, y-axis device B, x-axis device B) or when putting in the used feature names (Z1, Y1, X1, Z2, Y2, X2). In WEKA it is possible to remove all features which aren‘t of interest currently.

Train-test-split: For this task a train-test-split is needed. In the exercise slides was the request to apply an 80-20-split. This was done by loading the created arff-file. For the linear regression and multi-layer perceptron the time order doesn‘t matter. Therefore, for these two methods the dataset samples were shuffled to assure diversity in the train and test set. In contrast, gaussion process regression makes use of auto-correlation and thus it is important to keep the natural signal sequence. The dataset was for this case only split. Even further, WEKA had troubles when working with a huge dataset by using gaussian process. It simply stopped the process without feedback or raised the error message that the dataset would be too big. Therefore, I only used 1/10 of the whole dataset for it. The other two methods worked with the full dataset. The splitted dataset is then stored as train and test arff-files.

3) Regression

MAE = Mean Absolute Error; it is the reported measure
MAE scores in thick letters indicate that the current hyper-parameter modification is kept for further experiments.

Linear Regression

initial default tests: The outputs are sorted from condition 1 (first WEKA output) to condition 12 (last WEKA output). Each row is read from left to right.

```
Time taken to build model: 0.37 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.2 seconds

=== Summary ===

Correlation coefficient                  0.0113
Mean absolute error                      0.7818
Root mean squared error                  0.9884
Relative absolute error                 99.9923 %
Root relative squared error             99.9936 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.14 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.08 seconds

=== Summary ===

Correlation coefficient                  0.0413
Mean absolute error                      0.781
Root mean squared error                  0.9876
Relative absolute error                 99.8975 %
Root relative squared error             99.9156 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.02 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.07 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                      0.8166
Root mean squared error                  0.9859
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            13165
```

```
Time taken to build model: 0.06 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0.0051
Mean absolute error                      0.8166
Root mean squared error                  0.9859
Relative absolute error                100.0006 %
Root relative squared error             99.9991 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.03 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                      0.7863
Root mean squared error                  0.9876
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            13165
```

```
Time taken to build model: 0.05 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds

=== Summary ===

Correlation coefficient                  0.0074
Mean absolute error                      0.7863
Root mean squared error                  0.9877
Relative absolute error                 99.9957 %
Root relative squared error            100.0075 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.37 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.2 seconds

=== Summary ===

Correlation coefficient                  0.0113
Mean absolute error                      0.7818
Root mean squared error                  0.9884
Relative absolute error                 99.9923 %
Root relative squared error             99.9936 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.1 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.13 seconds

=== Summary ===

Correlation coefficient                  0.0133
Mean absolute error                      0.8202
Root mean squared error                  0.9952
Relative absolute error                 99.9907 %
Root relative squared error             99.9921 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.05 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds

=== Summary ===

Correlation coefficient                 -0
Mean absolute error                      0.801
Root mean squared error                  0.9979
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            13165
```

```
Time taken to build model: 0.08 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                  0.0103
Mean absolute error                      0.801
Root mean squared error                  0.9979
Relative absolute error                100.0103 %
Root relative squared error            100.0095 %
Total Number of Instances            13165
```

```
Time taken to build model: 0.05 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0
Mean absolute error                      0.7916
Root mean squared error                  0.9856
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            13165
```

```
Time taken to build model: 0.09 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                  0.0395
Mean absolute error                      0.7911
Root mean squared error                  0.9849
Relative absolute error                 99.9401 %
Root relative squared error             99.9339 %
Total Number of Instances            13165
```

Considering the almost not exsting correlations, the signals of the two devices doesn‘t seem to be linearly related. Besides that, it doesn‘t make sense to have a closer look at both conditions using the two same variables but with switched input and output. For instance Y1 = w*Y2 + b can be rewritten as Y2 = (Y1 – b)/w. Therefore, it seems to be redundant to always consider both versions. Even further, the hyper-parameter modification has no impact on the MAE. Most likely that‘s because linear regression is a very simple algorithm for which nothing needs to be changed.

best performing configurations:
The configurations 1, 2, 5, 6, 7, 11, and 12 have MAE scores of lower than 0.8 but always higher than 0.78. They almost not differ from the worse configurations.

worst performing configurations:
The worst performing configurations are the negative of the best performing ones, namely **configurations 3, 4, 8, 9, and 10. Their MAE lays between 0.801 and 0.8202**. They are only slightly worse than the best performing configurations. It turns out that linear regression isn’t able to handle this type of task. Also computing linear regression with a signal correctly ordered in time doesn’t lead to a better performing MAE. There doesn’t seem to be a linear connection between the signals of devices A and B.

Gaussian Process Regression

initial default tests: The outputs are sorted from condition 1 (first WEKA output) to condition 12 (last WEKA output). Each row is read from left to right.

```
Time taken to build model: 337.12 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.39 seconds

=== Summary ===

Correlation coefficient                  0.0051
Mean absolute error                      0.8313
Root mean squared error                  1.034
Relative absolute error                100.2965 %
Root relative squared error            100.3228 %
Total Number of Instances             2194
```

```
Time taken to build model: 406.35 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.33 seconds

=== Summary ===

Correlation coefficient                 -0.0087
Mean absolute error                      0.6464
Root mean squared error                  0.7994
Relative absolute error                100.6238 %
Root relative squared error            100.5098 %
Total Number of Instances             1317
```

```
Time taken to build model: 308.82 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.2 seconds

=== Summary ===

Correlation coefficient                 -0.0003
Mean absolute error                      0.8411
Root mean squared error                  0.9825
Relative absolute error                100.09   %
Root relative squared error            100.0152 %
Total Number of Instances             1317
```

```
Time taken to build model: 265.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.16 seconds

=== Summary ===

Correlation coefficient                 -0.0174
Mean absolute error                      0.8405
Root mean squared error                  0.9828
Relative absolute error                100.0113 %
Root relative squared error            100.054  %
Total Number of Instances             1317
```

```
Time taken to build model: 298.78 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.2 seconds

=== Summary ===

Correlation coefficient                  0.0005
Mean absolute error                      0.8021
Root mean squared error                  1.0686
Relative absolute error                100.1726 %
Root relative squared error            100.1813 %
Total Number of Instances             1317
```

```
Time taken to build model: 415 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.38 seconds

=== Summary ===

Correlation coefficient                  0.0285
Mean absolute error                      0.7994
Root mean squared error                  1.0646
Relative absolute error                 99.83   %
Root relative squared error             99.8058 %
Total Number of Instances             1317
```

```
Time taken to build model: 365.52 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.23 seconds

=== Summary ===

Correlation coefficient                 -0.0094
Mean absolute error                      0.8672
Root mean squared error                  1.0291
Relative absolute error                100.0388 %
Root relative squared error            100.0224 %
Total Number of Instances             1317
```

```
Time taken to build model: 321.63 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.22 seconds

=== Summary ===

Correlation coefficient                  0.0325
Mean absolute error                      0.8655
Root mean squared error                  1.0278
Relative absolute error                 99.8415 %
Root relative squared error             99.8949 %
Total Number of Instances             1317
```

```
Time taken to build model: 299.03 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.21 seconds

=== Summary ===

Correlation coefficient                 -0.0003
Mean absolute error                      0.7306
Root mean squared error                  0.9417
Relative absolute error                 99.9856 %
Root relative squared error             99.9498 %
Total Number of Instances             1317
```

```
Time taken to build model: 287.31 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.22 seconds

=== Summary ===

Correlation coefficient                 -0.0062
Mean absolute error                      0.7309
Root mean squared error                  0.9413
Relative absolute error                100.0288 %
Root relative squared error             99.9138 %
Total Number of Instances             1317
```

```
Time taken to build model: 294.84 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.21 seconds

=== Summary ===

Correlation coefficient                  0.0005
Mean absolute error                      0.6831
Root mean squared error                  0.855
Relative absolute error                 99.9307 %
Root relative squared error            100.0097 %
Total Number of Instances             1317
```

```
Time taken to build model: 521.31 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.44 seconds

=== Summary ===

Correlation coefficient                 -0.0152
Mean absolute error                      0.6874
Root mean squared error                  0.8611
Relative absolute error                100.554  %
Root relative squared error            100.7254 %
Total Number of Instances             1317
```

There is no correlation which would be worth to mention, all of them are very close to 0. Looking at the MAE some configurations seem to do a little bit better than others. These are **configuration 2, 11, and 12**. Each of them has a MAE **of smaller than 0.7**. However, considering that the values are z-score normalized where the majority lays between -3 and 3, this is still a huge error score, measuring the mean distance between the prediction and ground truth. For further hyper-parameter modifications the configurations with a MAE lower than 0.7 are considered.

configuration 2: MAE = 0.6464 | configuration 11: MAE = 0.6831 | configuration 12: MAE = 0.6874

kernel:
Puk → 0.667 **0.683** 0.6892
RBFKernel → **0.646** 0.6831 **0.6856**

noise:
0.5 → 0.6502 0.6837 0.6874
1.5 → 0.644 0.6826 0.6864
2.0 → 0.6433 **0.6825** 0.6841
3.0 → 0.6428 0.6825 0.6837
4.0 → 0.6426 0.6826 **0.6836**
5.0 → **0.6425** 0.6827 0.6836

debug = True → 0.6425 0.6825 0.6836

Final MAE: **0.6425 0.6825 0.6836**

best performing configurations:
There is a slight decrease of MAE in all three investigated configurations observable. However, this improvement only takes place with the beginning of the third decimal place. The total gain of this examination seems to be limited. **Configuration 2** still outperforms the other ones. Here all axes of device A are taken to predict the x-axis of device B.

worst performing configurations:
The worst performing configurations reach a MAE score of at least 0.85. Namely, these are **configurations 7, MAE of 0.8672, and 8, MAE of 0.8655**. Those are worse than for linear regression. This shows that a simple model doesn't always underperform more complex models.
It is striking that configuration 7 and 8 both try to predict the x-axis of device A.

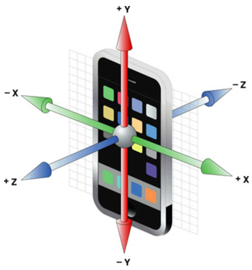


Figure 1: axes in the three-dimensional space of a smartphone.
Retrieved from: https://images.saymedia-content.com/.image/t_share/MTczOTg4MTA1Mzc0NzM3NzE1/10-best-accelerometer-apps.jpg

Looking at the figure 1 above and knowing about the fact that device A was fixed at the wrist, it becomes clear that the arm swung along the x-axis. However, device B at the upper leg didn't swung in the same tact as the arm. It was more like that the leg moved almost twice times slower than the arm. Therefore, analyzing the signal curve device B (X2, Y2, Z2) could give only little information about the x-axis of device A.

Multilayer Perceptron Regression (MLP)

initial default tests: The outputs are sorted from condition 1 (first WEKA output) to condition 12 (last WEKA output). Each row is read from left to right.

```
Time taken to build model: 3.66 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.07 seconds

=== Summary ===

Correlation coefficient                  0.0079
Mean absolute error                      0.7848
Root mean squared error                  0.9891
Relative absolute error                100.3773 %
Root relative squared error            100.0634 %
Total Number of Instances             13165
```

```
Time taken to build model: 7.09 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0.0457
Mean absolute error                      0.7801
Root mean squared error                  0.9875
Relative absolute error                 99.779  %
Root relative squared error             99.8989 %
Total Number of Instances             13165
```

```
Time taken to build model: 3.8 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                 -0.0015
Mean absolute error                      1.0376
Root mean squared error                  1.2042
Relative absolute error                127.0679 %
Root relative squared error            122.136  %
Total Number of Instances             13165
```

```
Time taken to build model: 6.88 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0.0132
Mean absolute error                      1.0397
Root mean squared error                  1.2066
Relative absolute error                127.3196 %
Root relative squared error            122.3803 %
Total Number of Instances             13165
```

```
Time taken to build model: 3.74 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds

=== Summary ===

Correlation coefficient                  0.0057
Mean absolute error                      0.8207
Root mean squared error                  1.0128
Relative absolute error                104.3681 %
Root relative squared error            102.5487 %
Total Number of Instances             13165
```

```
Time taken to build model: 6.84 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0.0161
Mean absolute error                      0.8164
Root mean squared error                  1.0089
Relative absolute error                103.8266 %
Root relative squared error            102.155  %
Total Number of Instances             13165
```

```
Time taken to build model: 3.5 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                  0.0139
Mean absolute error                      0.8905
Root mean squared error                  1.1242
Relative absolute error                108.5577 %
Root relative squared error            112.9492 %
Total Number of Instances             13165
```

```
Time taken to build model: 6.92 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                  0.011
Mean absolute error                      0.8917
Root mean squared error                  1.1258
Relative absolute error                108.6956 %
Root relative squared error            113.1116 %
Total Number of Instances             13165
```

```
Time taken to build model: 3.85 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                 -0.0053
Mean absolute error                      1.2522
Root mean squared error                  1.4862
Relative absolute error                156.3339 %
Root relative squared error            148.9437 %
Total Number of Instances             13165
```

```
Time taken to build model: 6.92 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.03 seconds

=== Summary ===

Correlation coefficient                  0.0115
Mean absolute error                      1.2486
Root mean squared error                  1.4827
Relative absolute error                155.8938 %
Root relative squared error            148.5917 %
Total Number of Instances             13165
```

```
Time taken to build model: 3.83 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.04 seconds

=== Summary ===

Correlation coefficient                  0.0052
Mean absolute error                      0.9172
Root mean squared error                  1.1074
Relative absolute error                115.8729 %
Root relative squared error            112.3649 %
Total Number of Instances             13165
```

```
Time taken to build model: 6.79 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.06 seconds

=== Summary ===

Correlation coefficient                  0.0359
Mean absolute error                      0.916
Root mean squared error                  1.106
Relative absolute error                115.7271 %
Root relative squared error            112.2161 %
Total Number of Instances             13165
```

The correlations almost don‘t exist as they not even reach a value of 0.1. Interestingly, the MAE is even higher than for the linear regression. This is obviously the case for **configurations 3, 4, 9, 10, 11, 12 which reach all at least a MAE of 0.9.** Because the signal values are z-score normalized, most of the values range from -3 to 3. Therefore, a MAE of 0.9 is relatively high as it describes the mean distance between prediction and ground truth. For these the y-axis of both devices or the z-axis of the device A should be predicted. The MLP seems to struggle with this even harder. For further hyper-parameter modifications the configurations with a MAE lower than 0.8 are considered.

configuration 1: MAE = 0.7848 | configuration 2: MAE = 0.7801

learningRate:
0.1 → 0.798 0.8059
0.2 → 0.7986 0.7915
0.4 → 0.7831 **0.7774**
0.5 → **0.7784** 0.7781
0.6 → 0.7793 0.7781

momentum:
0.0 → 0.7809 0.7825
0.1 → 0.7794 0.7789
0.3 → **0.7781** 0.7795
0.4 → 0.7781 0.7865
0.5 → 0.7782 0.7968

hiddenLayers:
1 → 0.7781 0.7777
3 → 0.7781 0.7774
6 → 0.7781 0.7774
12 → 0.7781 **0.7773**

trainingTime:
10 → 0.7781 0.7776
100 → 0.7781 0.7773
600 → 0.7781 0.7774

decay = True → 0.7806 0.7812

Final MAE: **0.7781** **0.7773**

best performing configurations:
After hyper-parameter modification **configuration 1**’s MAE improves by 0.0067 and **configuration 2**’s MAE enhances by 0.0028. Also here it becomes clear that this is only a minimal improvement. The strength of MLP is it to learn by combining features. However, this task doesn’t really seem to be suited for it as configuration 2 with its three feature inputs is minimally better than configuration 1 which only has X1 as variable input.

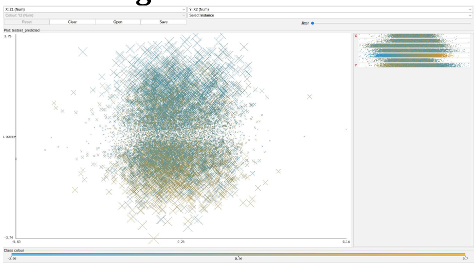
worst performing configurations:
Configurations 3, 4, 9, and 10 all reached a **MAE of higher than 1.0**. Therefore, these are the worst performing method-configuration combinations of the whole report. The z-score normalized values have a standard deviation of 1. These configurations have a mean distances between prediction and ground truth of even higher than one standard deviation. They are also the worst performing configurations of linear regression. They all have in common that they try to predict the y-axis. Figure 1 tells that the y-axis represents the distance to the ground for the set upright devices. During walking the height change to the ground could be relatively small. The acceleration along the y-axis could be minimal. Therefore it can be difficult to predict the small changes. That could

become even harder as the arm and leg don't move synchronized like it was explained in the gaussian process section.

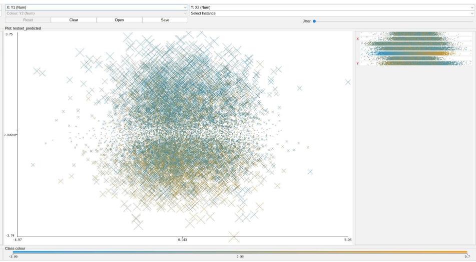
Visualizations

In this section the three different models are compared for configuration 2.

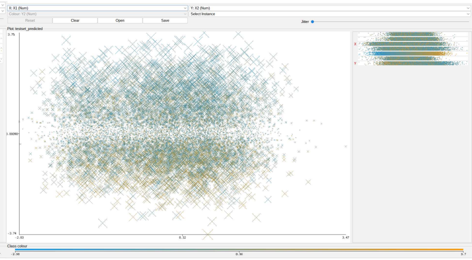
Linear Regression



x: Z1 | y: X2

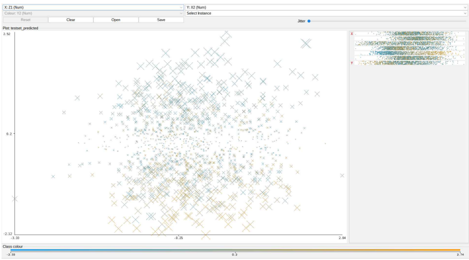


x: Y1 | y: X2

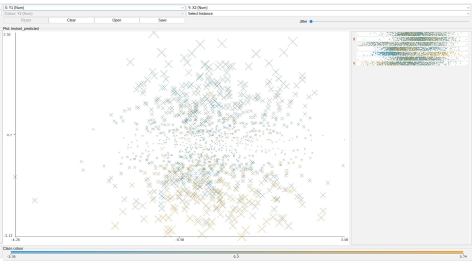


x: X1 | y: X2

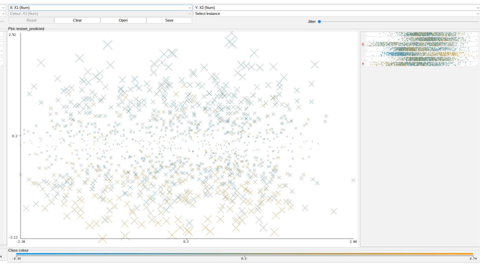
Gaussian Process Regression



x: Z1 | y: X2

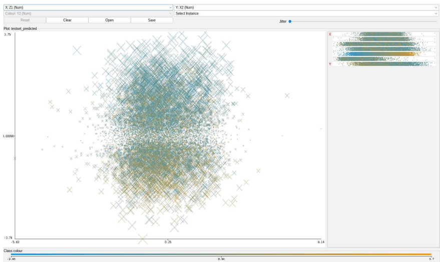


x: Y1 | y: X2

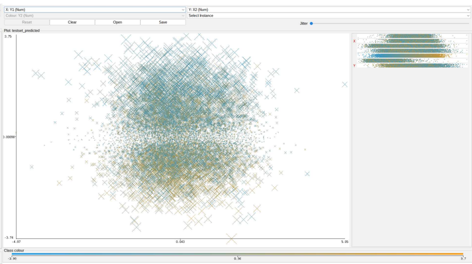


x: X1 | y: X2

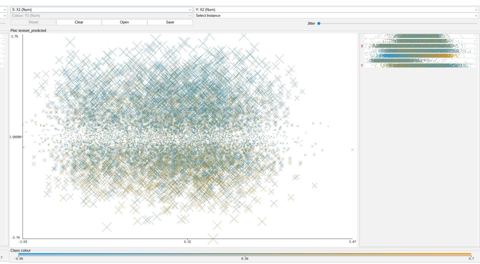
MLP



x: Z1 | y: X2



x: Y1 | y: X2



x: X1 | y: X2

All the plots show that one signal axis of device A is not enough to split the x-axis of device B into several intervals and therefore successful prediction on point cannot work.

Summary

The overall **best performing** configuration is achieved by the **gaussian process regression** method with the usage of **configuration 2 with a MAE of 0.6425**. The **worst** case happens for **MLP configuration 9 with a MAE of 1.2522**. Both MAE score aren't impressive in their performance. However, they differentiate by 0.6097 which is a massive gap for z-score normalized values. It follows that the configuration and method choice has a huge impact on the prediction performance.

References

[1] Khan, A., Siddiqi, M., & Lee, S.-W. (2013). Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors*, *13*(10), 13099–13122. https://doi.org/10.3390/s131013099

[2] Khusainov, R., Azzi, D., Achumba, I., & Bersch, S. (2013). Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations. *Sensors*, *13*(10), 12852–12902. https://doi.org/10.3390/s131012852

[3] Sousa Lima, W., Souto, E., El-Khatib, K., Jalali, R., & Gama, J. (2019). Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors*, *19*(14), 3213. https://doi.org/10.3390/s19143213

[4] Wang, W., Guo, Y., Huang, B., Zhao, G., Liu, B., & Wang, L. (2011). Analysis of filtering methods for 3D acceleration signals in Body Sensor Network. *International Symposium on Bioelectronics and Bioinformations 2011*. https://doi.org/10.1109/isbb.2011.6107697