Report Pervasive Computing Assignment 3

1) Screwing Process and Recognition

1. Data recording

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, recorded tracks can be simply exported as csv files.

Sampling rate for the recording: According to [4], 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 [5]. Therefore, the sampling rate was set to **20 Hz**.

Data recording: I used a screwdriver with cross slot to work with 4 screws of sharpeners which are shown in the picture below. As the sharpeners are made of steel, there is the advantage that the material isn’t strongly modified over time. Thus, less noise should happen due to abrasion.
Each sharpener was fixed with the non-dominant, right hand while the left hand screwed. I didn’t rest my left elbow on the table because that could restrict the moving space of my wrist. On the left wrist, I fixed my smartphone for measurements.
For all conditions of all tasks the smartphone had the same position. The microphone pointed towards my left hand and the backside of the phone touched my arm such that I could operate on the device. The smartphone was fixed with help of 4 hair ties. So it was flexibly movable by any of the four tie regions.
For each class I collected 60 samples over 2 days. For instance, that means that I screwed a screw into a sharpener with a certain grip 60 times. Each screw with the related whole were used 15 times for each condition. **Each sample is stored as a csv-file in the belonging class folder within the folder “dataset”.**

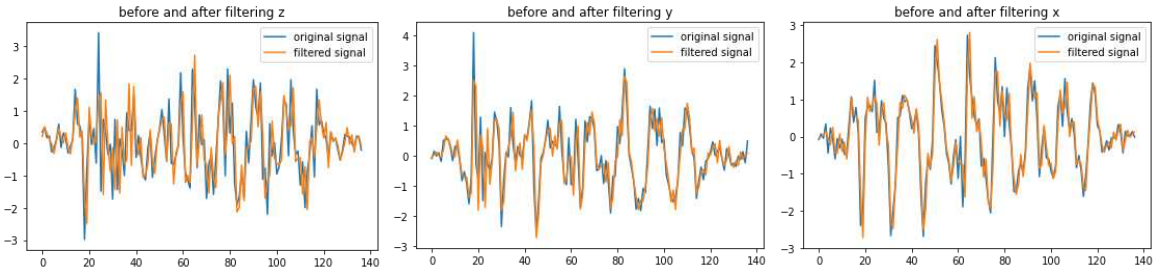
2. Pre-processing, segmentation, feature extraction, classification

Pre-processing and segmentation

Cutting: For each signal I cut the first and last 0.4 seconds away to be sure that only the process screwing is recorded.

Filtering: The low-pass filter enables removal of noise and gravity acceleration [3]. 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 [4]. Therefore, the filter threshold was set to that value.

Besides that, [6] outlines that strong phase distortions might happen when an unsuitable filter order and frequency are selected. [5] 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, Grip 1 unscrewing.



In general, there are slight shifts noticable. However, changing the filter order and frequency would lead to worse results.

Windowing: In the lecture was mentioned that 1.5 second windows are already large enough to enable human motion recognition. Each of my sample files only contain one action without null-class (there was no sequence of doing nothing). Furthermore, the repetive motion ends after at the latest 1.5 seconds. Therefore, it should be fine to set the hop size such that no overlap is given. As [1] and [2] set their window and hop sizes considering a power of 2 structure, I set the **window and hop size to 32 samples each**. As I have 20 samples per second, this gives a bit more than 1.5 second intervals.
The windows are not considered as individual samples but only as part of samples. It follows, that the distribution of samples keep unchanged. **For each condition 60 samples** exist. It is worth mentioning that the STFT matrix for each sample has exact 2 columns/windows, even so the samples aren't of same duration in time domain. This is very handy for feature extraction as each sample has the same amount of features then.

Feature extraction

For feature extraction I considered the papers [1] and [2]. Both of them place accelerometers on wrists and also consider fine motor hand movements, like teeth brushing, laundary folding, and working at the computer [1] or photo taking and rope jumping [2]. So, there is no screwing involved but actions which aren't to far away in their idea. Both papers work with the **frequency domain** in which the **DC, energy, and information entropy** are determined. Also the **normalized cross product** can be calculated by comparing each value of STFT of one dimension (x/y/z) with the corresponding value of the STFT of the other dimension [2]. I stored those features for each window but also took the overall window mean as additional features.
[1] gave my the idea to compute the **correlation between the single dimensions in time domain**. The paper considered correlation to align different measurement devices with each other. However, for me it was also interesting to see how the different axes are involved with each other.
In total, I ended up with 39 features.

Classification

class legend:

- 'Grip1_screw': 1
- 'Grip1_unscrew': 2

All reported scores are weighted average F-measures!

kNN

default: 0.667

number of neighbors:
3 → 0.741
5 → 0.758
8 → 0.725
9 → 0.733
11 → 0.750
19 → 0.783 (keep for further hyper-parameter experiments with kNN)
21 → 0.775
25, 31 → 0.741

distanceWeighting: no impact on F-measure

nearestNeighbourSearchAlgorithm:
ChebyshevDistance → 0.700
FilteredDistance → 0.599
ManhattanDistance → 0.807 (keep)
MinkowskiDistance → 0.783

distanceWeighting:
1/distance → 0.799
1-distance → 0.787

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          97               80.8333 %
Incorrectly Classified Instances        23               19.1667 %
Kappa statistic                          0.6167
Mean absolute error                      0.3905
Root mean squared error                  0.4133
Relative absolute error                 78.0915 %
Root relative squared error             82.6615 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,733    0,117    0,863      0,733    0,793      0,624    0,870     0,867     1
                 0,883    0,267    0,768      0,883    0,822      0,624    0,870     0,827     2
Weighted Avg.    0,808    0,192    0,815      0,808    0,807      0,624    0,870     0,847

=== Confusion Matrix ===

  a  b   <-- classified as
 44 16 |  a = 1
  7 53 |  b = 2
```

Later under “Features of interest” it will become clear that there already exist some features or feature combinations that separate sub-groups of classes well. kNN is a model which separates the classes in high-dimensional space considering the features and their combinations. When even in low-dimensional space a rough separation is visible then it is probable to observe even better separation in high-dimensional space. This could be an explanation why kNN performs well (F-measure of 0.807). Moreover, for the model it seems to be easier to detect class 2/unscrew samples correctly (7 false positives) than to classify class 1/screw samples correctly (16 false negatives).

Naive Bayes

default: 0.758

useKernelEstimator = True → 0.775 (keep)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          93               77.5    %
Incorrectly Classified Instances        27               22.5    %
Kappa statistic                          0.55
Mean absolute error                      0.2593
Root mean squared error                  0.4303
Relative absolute error                 51.8599 %
Root relative squared error             86.0661 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,800    0,250    0,762      0,800    0,780      0,551    0,829     0,830     1
               0,750    0,200    0,789      0,750    0,769      0,551    0,829     0,808     2
Weighted Avg.  0,775    0,225    0,776      0,775    0,775      0,551    0,829     0,819

=== Confusion Matrix ===

  a  b   <-- classified as
 48 12 |  a = 1
 15 45 |  b = 2
```

Naive Bayes reaches a F-measure score of 0.775. Looking at the confusion matrix it becomes clear that class 1/screw samples are as good detectable as class 2/unscrew samples because the diagonal true prediction are roughly the same and the classes are of equal size.

Decision tree (J48)

default: 0.616

confidenceFactor, binarySplits, useLaplace, numFolds, unpruned, subtreeRaising: no impact on F-measure

useMDLCorrection = False → 0.656 (keep)

minNumObj:

1 → 0.657
3 → 0.667 (keep)
4 → 0.633
5 → 0.658
6 → 0.633
10 → 0.650
20 → 0.568

```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          80               66.6667 %
Incorrectly Classified Instances        40               33.3333 %
Kappa statistic                          0.3333
Mean absolute error                      0.3434
Root mean squared error                  0.542
Relative absolute error                 68.677  %
Root relative squared error            108.399  %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,667    0,333    0,667      0,667    0,667      0,333    0,688     0,641     1
               0,667    0,333    0,667      0,667    0,667      0,333    0,688     0,663     2
Weighted Avg.  0,667    0,333    0,667      0,667    0,667      0,333    0,688     0,652

=== Confusion Matrix ===

  a  b   <-- classified as
 40 20 |  a = 1
 20 40 |  b = 2
```

Assignment 2 has a similar topic but instead of activity classification it was about measurement unit location detection. Moreover, I worked with time domain features there. Back then, J48 performed the best. In this assignment it is far away from that title (F-measure of 0.667). Considering the high score of the kNN, it seems like that the frequency domain features aren't well suited for the tree structure. It seems to be the case that the feature combination applied at once is important.

Multilayer Preceptron

default: 0.750

learningRate:
0.01 → **0.808 (keep)**
0.05, 0.1 → 0.742
0.2 → 0.717

momentum:
0.0, 0.01, 0.1 → 0.799
0.3 → 0.791

batchSize: no impact on F-measure

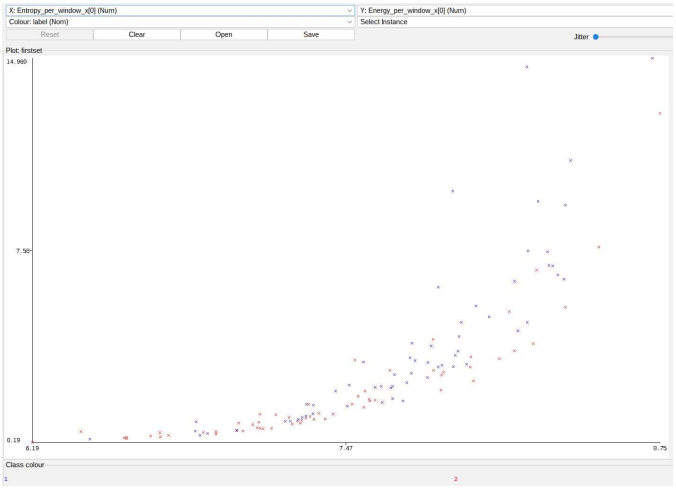
trainingTime:
400, 450, 550 → 0.799
600 → 0.783
1000 → 0.800

hiddenLayers:
1 → **0.816 (keep)**
2 → 0.791
3, 6, 10, 20 → 0.808
25 → 0.791
30 → 0.799

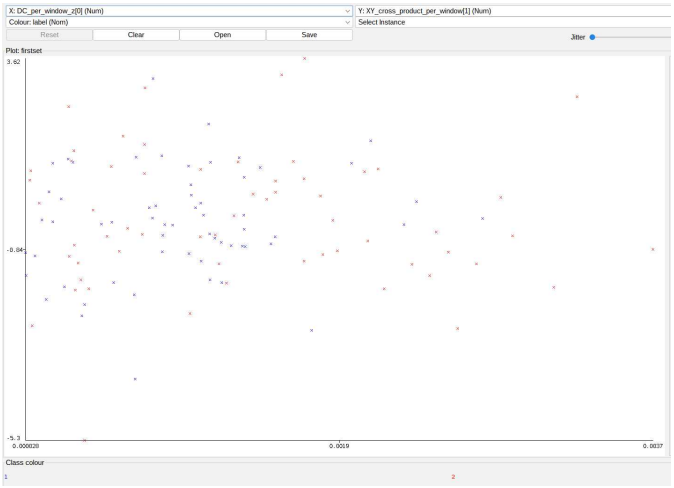
```
Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          98               81.6667 %
Incorrectly Classified Instances        22               18.3333 %
Kappa statistic                          0.6333
Mean absolute error                      0.2889
Root mean squared error                  0.3727
Relative absolute error                 57.7781 %
Root relative squared error             74.5344 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,750    0,117    0,865      0,750    0,804      0,639    0,871     0,902     1
                 0,883    0,250    0,779      0,883    0,828      0,639    0,871     0,819     2
Weighted Avg.    0,817    0,183    0,822      0,817    0,816      0,639    0,871     0,861

=== Confusion Matrix ===

  a  b   <-- classified as
 45 15 |  a = 1
  7 53 |  b = 2
```

Considering the confusion matrix off-diagonal values, the MLP seems to have the same problem as kNN, a high false negative rate. An explanation could be that I had a variation in movement style while screwing. An even bigger dataset collected over several months with breaks in-between could be helpful to adequately represent also the varying movements of the same action.

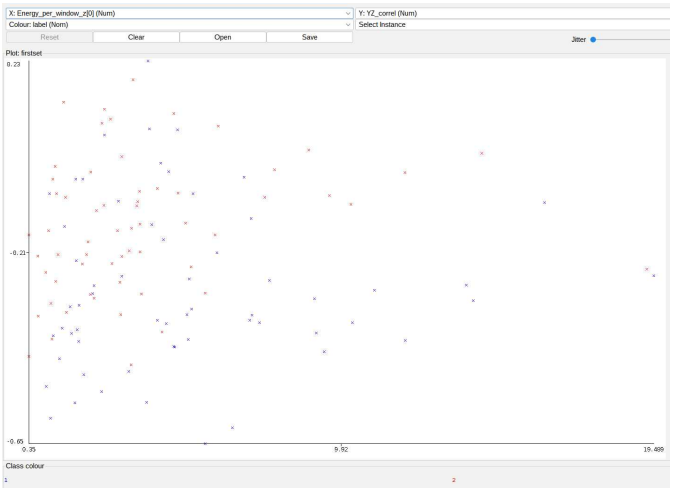
Features of interest



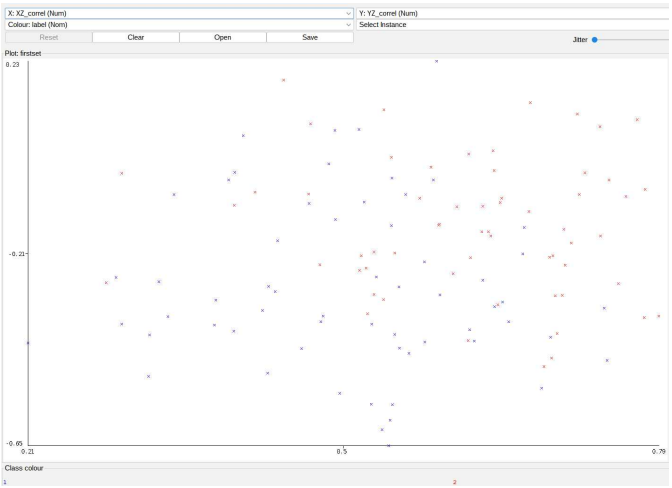
X: entropy x-signal first window
Y: energy x-signal first window



X: DC z-signal first window
Y: xy cross product second window



X: energy z-signal first window
Y: yz correlation



X: xz correlation
Y: yz correlation

The first plot shows that class 2/unscrew samples tend to have smaller energy values for the first window of the x-signal STFT and also for the entropy for the first window of the x-signal STFT.
The second illustration gives an idea that class 1/screw samples have lower and class 2/unscrew samples score higher in the DC value in the first window of the z-signal STFT.
The third graphic tells us that class 1/screw samples tend to build up a cluster in the area of medium energy strength for the first window of the z-sginal STFT and a medium negative correlation between the y- and z-signal. Observing the same energy interval area but with a small positive y- and z-signal correlation suggests class2/unscrew samples.
Also the last plot shows two clusters. The class 1/screw cluster is positioned at a small correlation between x- and z-signal and the a relatively strong negative correlation between y- and z-signal. For the class 2/unscrew cluster the opposite is the case. A high correlation between the x- and z-signal and a moderate correlation between the y- and z-signal marks the area of class 2/unscrew.

2) Screwing Grip Recognition

Data, recording, pre-processing, segmentation, and feature extraction are identical with 1).

Classification

class legend:

- Grip1 screw: 1
- Grip2 screw: 2

All reported scores are weighted average F-measures!

kNN

default: 0.736

number of neighbors:

3 → 0.812 (keep for further hyper-parameter experiments with kNN)
5 → 0.755
8 → 0.745
19 → 0.778
31 → 0.749

weighting: no impact on F-measure

nearestNeighbourSearchAlgorithm:
ChebyshevDistance → 0.741
FilteredDistance → 0.666
ManhattanDistance → 0.754
MinkowskiDistance → 0.812

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          98               81.6667 %
Incorrectly Classified Instances        22               18.3333 %
Kappa statistic                          0.6333
Mean absolute error                      0.2515
Root mean squared error                  0.3963
Relative absolute error                 50.3067 %
Root relative squared error             79.2504 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,967    0,333    0,744      0,967    0,841      0,664    0,864     0,797     1
               0,667    0,033    0,952      0,667    0,784      0,664    0,864     0,847     2
Weighted Avg.  0,817    0,183    0,848      0,817    0,812      0,664    0,864     0,822

=== Confusion Matrix ===

  a  b   <-- classified as
 58  2 |  a = 1
 20 40 |  b = 2
```

With a F-measure of 0.812, kNN reaches a high score. The confusion matrix tells that class 2/grip2 is often classified as class 1/grip 1 (20 false positives) but class 1/grip1 is almost always classified correctly (2 false negatives).

Naive Bayes

default: 0.800

useKernelEstimator = True → 0.729

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          96               80      %
Incorrectly Classified Instances        24               20      %
Kappa statistic                          0.6
Mean absolute error                      0.2057
Root mean squared error                  0.41
Relative absolute error                 41.1432 %
Root relative squared error             81.9998 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,783    0,183    0,810      0,783    0,797      0,600    0,860     0,877     1
               0,817    0,217    0,790      0,817    0,803      0,600    0,860     0,812     2
Weighted Avg.  0,800    0,200    0,800      0,800    0,800      0,600    0,860     0,845

=== Confusion Matrix ===

  a  b   <-- classified as
 47 13 |  a = 1
 11 49 |  b = 2
```

With a F-measure of 0.800, the Naive Bayes scores higher than in task 1. It seems like the probabilistic conditions are able to catch the feature dependencies.

Decision tree (J48)

default: 0.758

confidenceFactor, binarySplits, useLaplace, numFolds, subtreeRaising: no impact on F-measure

unpruned = True → 0.750

useMDLCorrection = False → 0.783 (keep)

minNumObj:
1 → 0.791
3 → 0.792
4 → 0.783
5 → 0.799 (keep)
6 → 0.766
8 → 0.792
22 → 0.716

```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          96               80      %
Incorrectly Classified Instances        24               20      %
Kappa statistic                          0.6
Mean absolute error                      0.2414
Root mean squared error                  0.4143
Relative absolute error                 48.2737 %
Root relative squared error             82.8576 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,867    0,267    0,765      0,867    0,813      0,605    0,822     0,754     1
               0,733    0,133    0,846      0,733    0,786      0,605    0,822     0,811     2
Weighted Avg.  0,800    0,200    0,805      0,800    0,799      0,605    0,822     0,783

=== Confusion Matrix ===

  a  b   <-- classified as
 52  8 |  a = 1
 16 44 |  b = 2
```

Compared to task 1 the J48 reaches a much higher score here (F-measure of 0.799). This tells us that the used features are better suited to distinguish between activities (different curve structures) than doing the same activity in contrary directions (more similar curves with opposite signs on one axis).

Multilayer Preceptron

default: 0.883

learningRate:
0.01 → 0.883
0.05 → 0.867
0.1 → 0.858
0.2 → 0.892 (keep)
0.25 → 0.892
0.4 → 0.883

momentum:
0.0, 0.01, 0.1 → 0.883
0.3, 0.4 → 0.892
batchSize: no impact on F-measure

trainingTime:
400, 600 → 0.892
1000 → 0.883

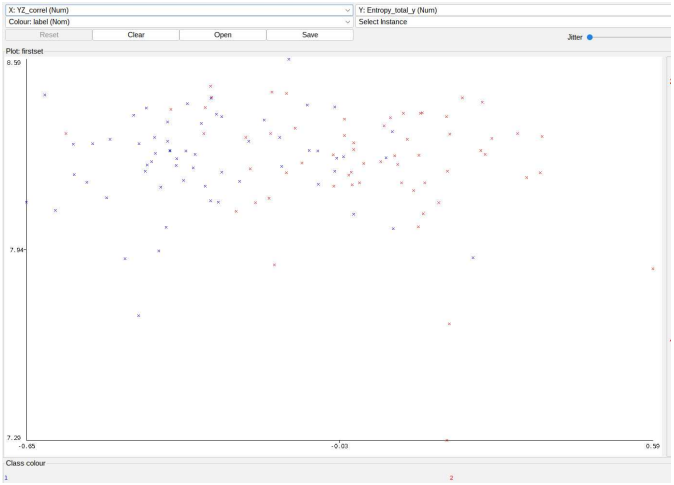
hiddenLayers:
1, 3, 6 → 0.883
10 → 0.842
30, 40, 50 → 0.892
70 → 0.883

decay = True → 0.883

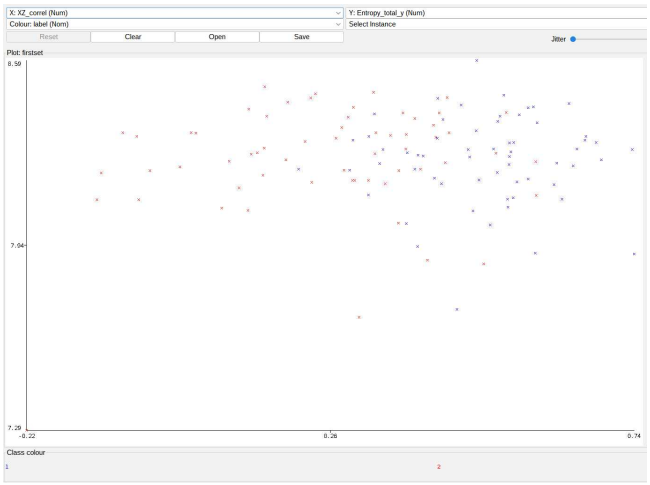
```
Time taken to build model: 1.32 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         107               89.1667 %
Incorrectly Classified Instances        13               10.8333 %
Kappa statistic                          0.7833
Mean absolute error                      0.1409
Root mean squared error                  0.3286
Relative absolute error                 28.173  %
Root relative squared error             65.7242 %
Total Number of Instances              120

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,900    0,117    0,885      0,900    0,893      0,783    0,923     0,923     1
               0,883    0,100    0,898      0,883    0,891      0,783    0,923     0,919     2
Weighted Avg.  0,892    0,108    0,892      0,892    0,892      0,783    0,923     0,921

=== Confusion Matrix ===

  a  b   <-- classified as
 54  6 |  a = 1
  7 53 |  b = 2
```

The MLP scores the best in the whole assignment and outperforms even the MLP of task 1. The confusion matrix shows that there are only a few misclassifications.

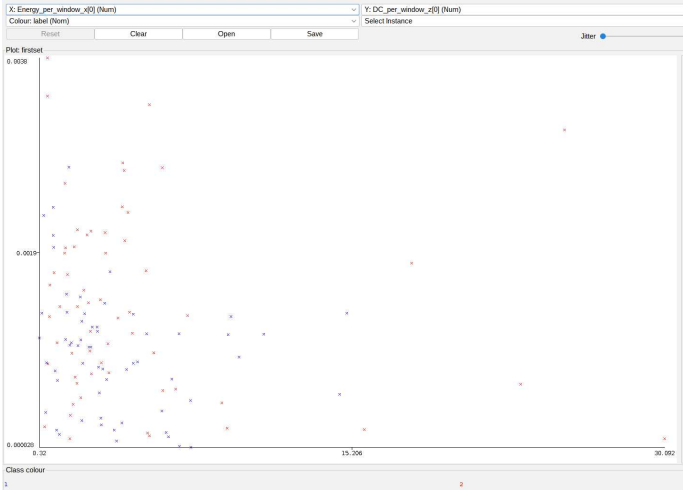
Features of interest



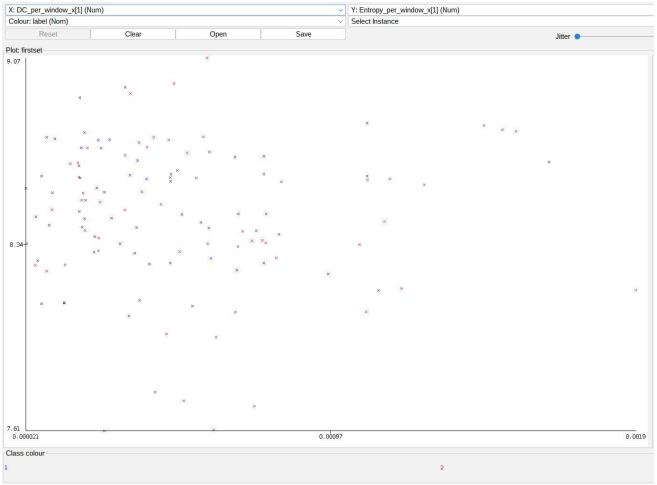
X: xy correlation
Y: Entropy y-signal over all windows



X: xz correlation
Y: Entropy y-signal over all windows



X: energy x-signal of first window
Y: STFT DC z-signal first window



X: STFT DC x-signal second window
Y: Entropy x-signal second window

The first figure shows that a big sub-sample of class 2/grip 2 can be separated by reaching a positive correlation between y- and z-sginal. The second one illustrates that a correlation between x- and z-signal below 0.26 often represents the second class grip2. The third plot tells that for high values, no matter if the energy of the x-signal of the frist window or the frequency DC of the z-signal of the first window, it is likely to meet a class 2/grip2 sample. The same can be observed with a high frequency DC value of the x-signal for the second window in the last graphic. It follows that the features DC in frequency domain and correlations in time domain may be valuable for prediction.

References

[1] Bao, L., & Intille, S. S. (2004). Activity recognition from user-annotated acceleration data. *Lecture Notes in Computer Science*, 1–17. https://doi.org/10.1007/978-3-540-24646-6_1

[2] Hong, Y.-J., Kim, I.-J., Ahn, S. C., & Kim, H.-G. (2010). Mobile Health Monitoring System based on activity recognition using accelerometer. *Simulation Modelling Practice and Theory*, *18*(4), 446–455. https://doi.org/10.1016/j.simpat.2009.09.002

[3] 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

[4] 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

[5] 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

[6] 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