HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Laser System Health Diagnostics at EU XFEL

A voucher project with DESY

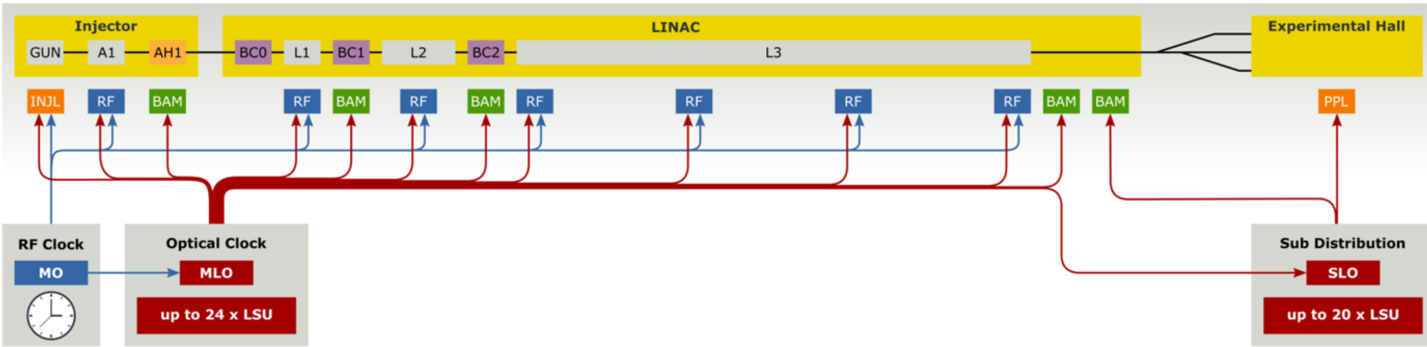
HELMHOLTZ AI

Helene Hoffmann

HZDR / 9 March 2021

Background

Background



- this is the EU XFEL
- synchronisation of all elements is crucial
- optical signals from lasers (MLO and SLO) are used for that

Objective

Objective

- experiments need the synchronisation laser to work properly

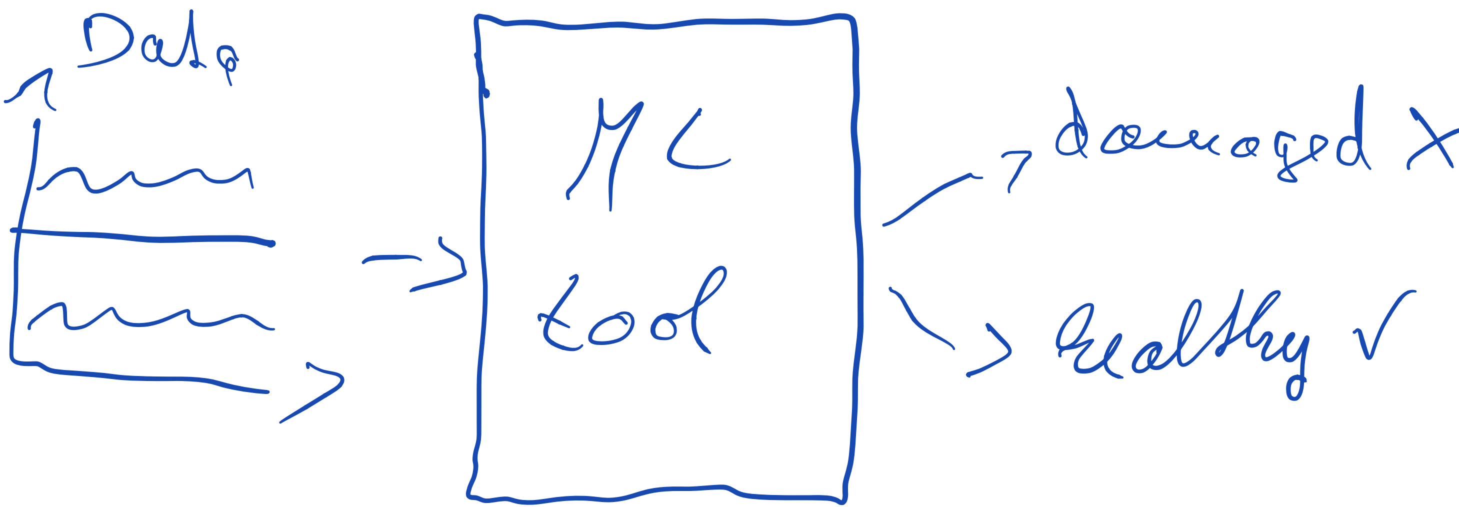
Objective

- experiments need the synchronisation laser to work properly
- damage of the synchronisation laser is not obvious

Objective

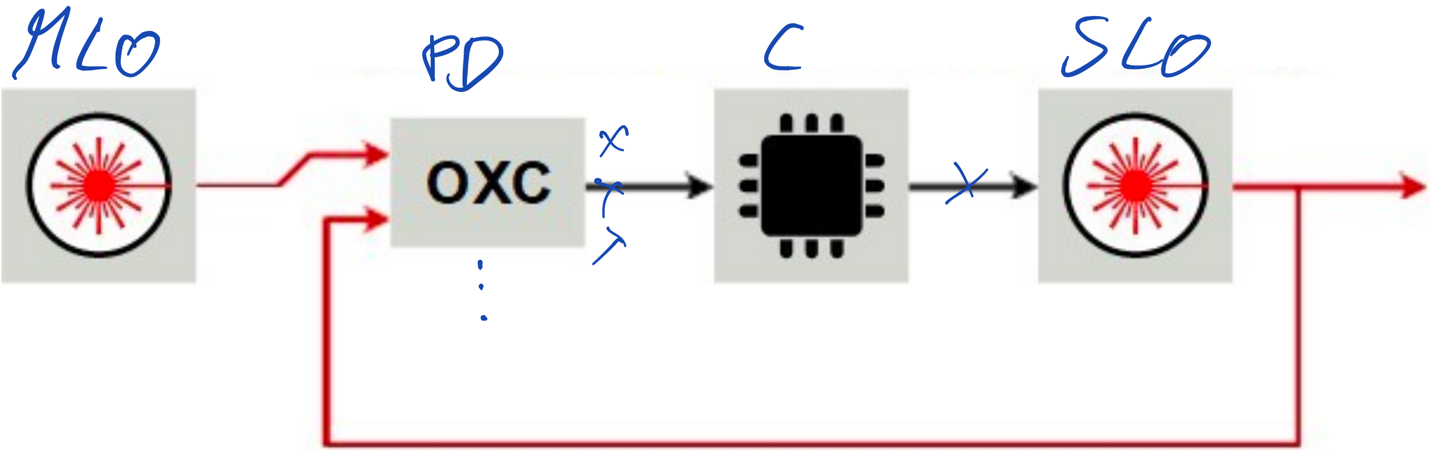
Goal:

Early and reliable damage detection for synchronisation laser



Data

Data source



Example Data - all channels

- time-series data

Example Data - all channels

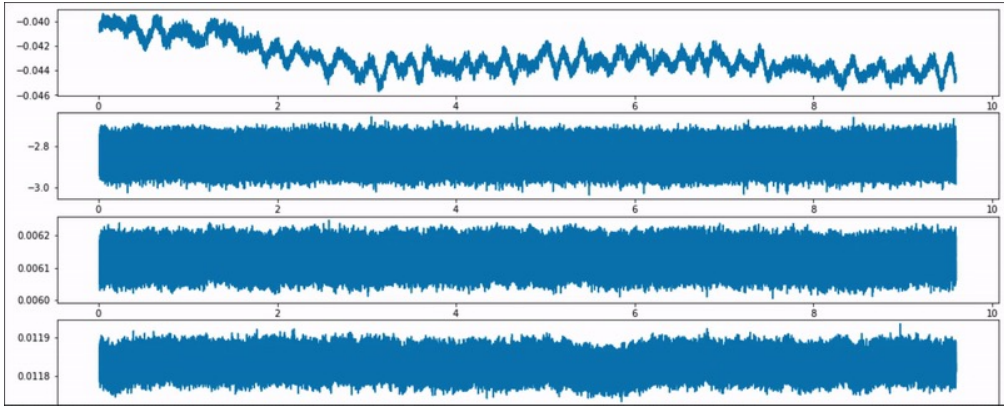
- time-series data
- time snippets of equal length, duration: 10 seconds

Example Data - all channels

- time-series data
- time snippets of equal length, duration: 10 seconds
- very high rate: number of data points per sample: 3 249 984

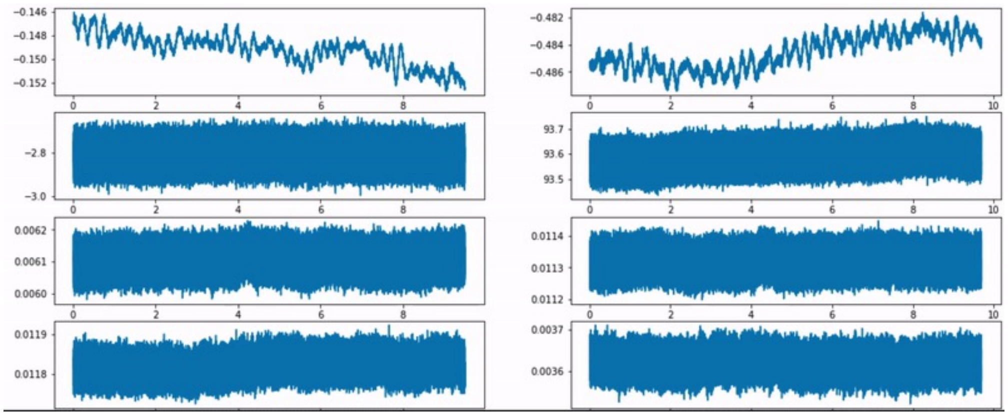
Example Data - all channels

- time-series data
- time snippets of equal length, duration: 10 seconds
- very high rate: number of data points per sample: 3 249 984



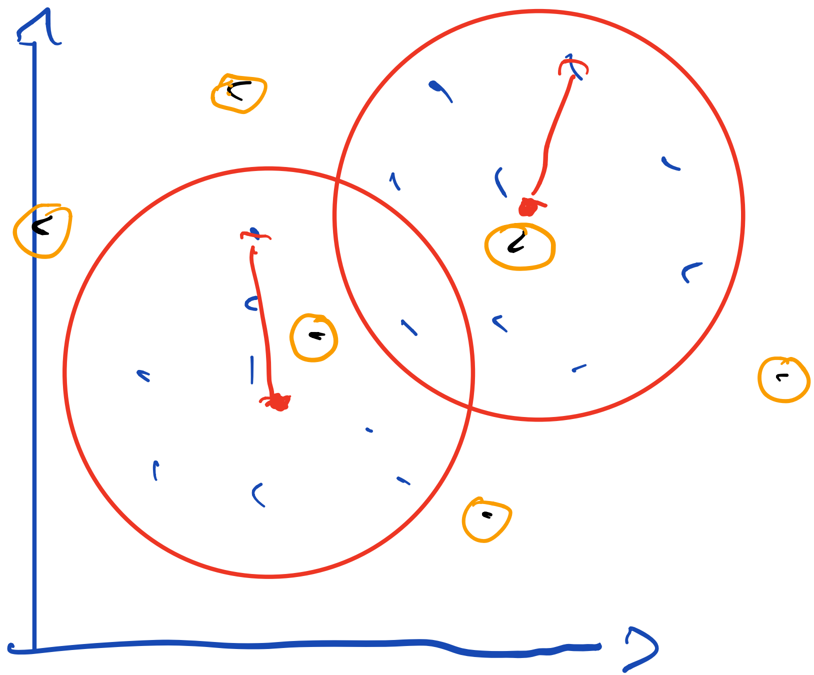
Example Data: healthy vs. damaged

- healthy data: for training + testing (58 samples), damaged data: for testing only (22 samples)
- one example comparison:



Algorithm

k-means



Data transformation

Feature extraction

Data transformation

Feature extraction

- requirement: low dimensional representation of time-courses

Data transformation

Feature extraction

- requirement: low dimensional representation of time-courses
- here: used `tsfresh`-package

Data transformation

Feature extraction

- requirement: low dimensional representation of time-courses
- here: used `tsfresh`-package
    - simple features: mean, min, max, ...

Data transformation

Feature extraction

- requirement: low dimensional representation of time-courses
- here: used `tsfresh`-package
  - simple features: mean, min, max, ...
  - more sophisticated features: fft-coefficients, entropy, absolute energy, ...

Data transformation

Feature extraction

- requirement: low dimensional representation of time-courses
- here: used `tsfresh`-package
  - simple features: mean, min, max, ...
  - more sophisticated features: fft-coefficients, entropy, absolute energy, ...

Dimensionality reduction

- reduce dimensionality of data using: principle component analysis (PCA)
- combines features into defined number of principle components

Results

Using the package

```
import anomaly_detection_package
from anomaly_detection_package.anomaly_detection import
    AnomalyDetectionDesy
data = AnomalyDetectionDesy()
data.load_data_desy()
data.train_test_set_creation(test_size = 0.2)
data.fit(PCA_n_components = 2)
data.get_most_important_features(n_features=10,component=0)
```

Using the package

```python
import anomaly_detection_package
from anomaly_detection_package.anomaly_detection import
    AnomalyDetectionDesy
data = AnomalyDetectionDesy()
data.load_data_desy()
data.train_test_set_creation(test_size = 0.2)
data.fit(PCA_n_components = 2)
data.get_most_important_features(n_features=10,component=0)
```

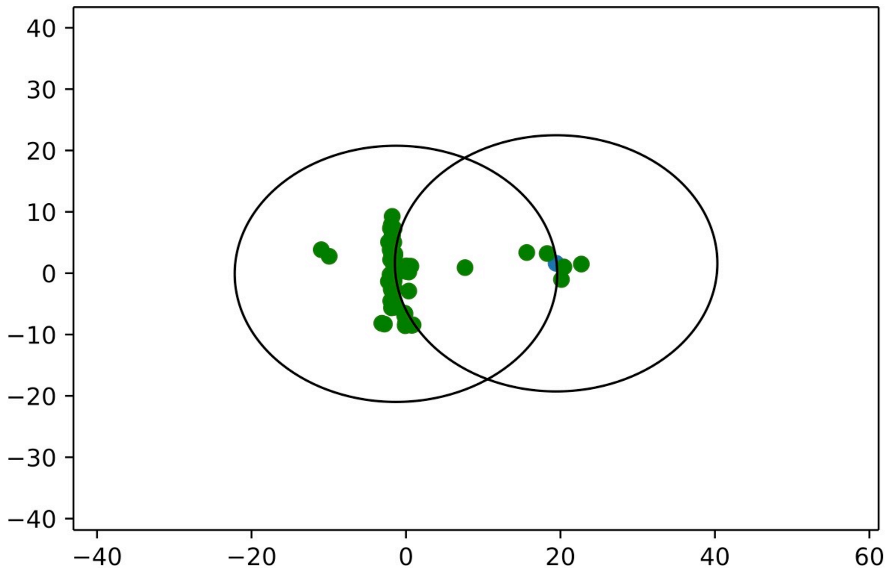
```
feature             channel             PCA component_
-------             -------             --------------
maximum             RF_HARM_MAG.SPEC    0.162954
maximum             OXC_IN_N.SPEC       0.162324
maximum             RF_FUND_MAG.SPEC    0.162307
standard_deviation  OXC_IN_P.SPEC       0.157876
standard_deviation  RF_FUND_MAG.SPEC    0.157875
standard_deviation  OXC_IN.SPEC         0.157875
standard_deviation  OXC_IN_N.SPEC       0.157873
standard_deviation  RF_HARM_MAG.SPEC    0.157866
standard_deviation  RF_FUND_PHASE.SPEC  0.157857
mean                RF_HARM_MAG.SPEC    0.150212
```

Training results

```
data.plot_clusters_2D()
```

Training results

```
data.plot_clusters_2D()
```

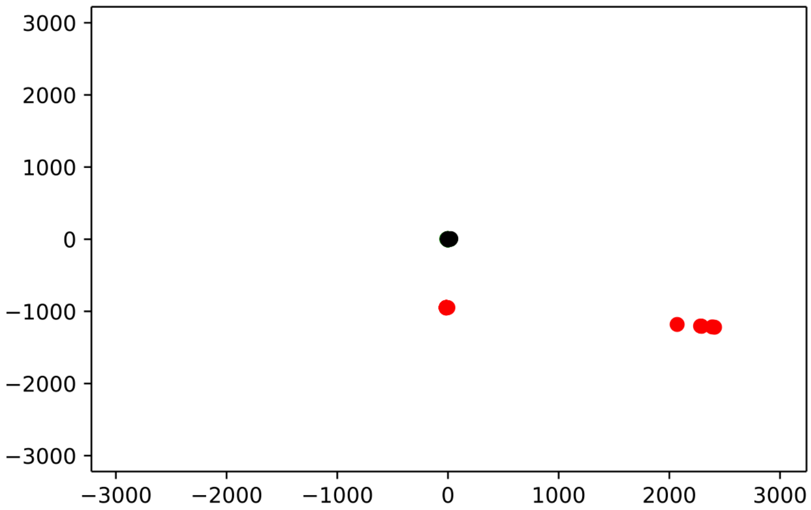


Test results

```
data.test()
data.plot_clusters_2D()
```

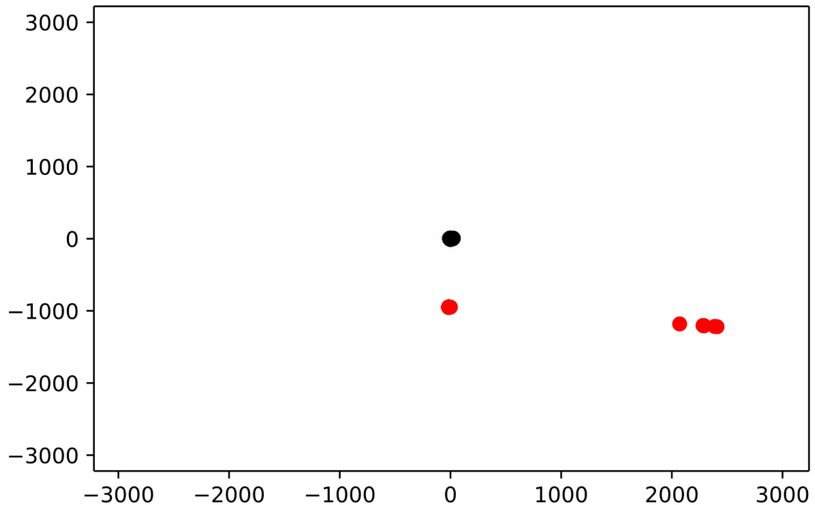
Test results

```
data.test()
data.plot_clusters_2D()
```



Test results

```
data.test()
data.plot_clusters_2D()
```



Accuracy:

Overall and reproducible accuracy:
1.0

Caution:

- healthy and damaged data is from laser of same type, but not from same laser

Caution:

- healthy and damaged data is from laser of same type, but not from same laser
- not every damage is the same

Caution:

- healthy and damaged data is from laser of same type, but not from same laser
- not every damage is the same
- for different laser types the results might look different

Caution:

- healthy and damaged data is from laser of same type, but not from same laser
- not every damage is the same
- for different laser types the results might look different

$\rightarrow$ Transferability can not be guaranteed

Conclusion

The used algorithm is suitable for the task!

Conclusion

The used algorithm is suitable for the task!

Goal:

Early and reliable damage detection for synchronisation laser

Conclusion

The used algorithm is suitable for the task!

Goal:

Early and reliable damage detection for synchronisation laser