Status of Data-Driven Beam Trajectory Anomaly Detection at European XFEL

3rd ICFA Beam Dynamics Mini-Workshop on Machine Learning Applications for Particle Accelerators

A. Sulc, R. Kammering, T. Wilksen

Chicago, IL, 2 Nov 2022

DESY.

European XFEL

The European XFEL:

- Consists of superconducting cavities that boost electrons.
- The electrons are then directed through specially arranged magnets (undulators).
- Then they emit extremely short and intense X-ray flashes.
- These X-ray flashes are then distributed to three beamlines (SASE).

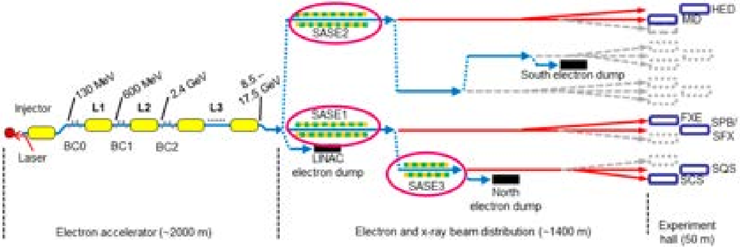
Data-Driven Predictive Maintenance on European XFEL

- Thousands of devices are involved in EuXFEL.
- Many components are operating in **extreme conditions** (radiation, heat...).
- At any moment, any component can **fail**.
- When a component fails, it can lead to undesired **downtime**.

Examples

Orbit Monitoring

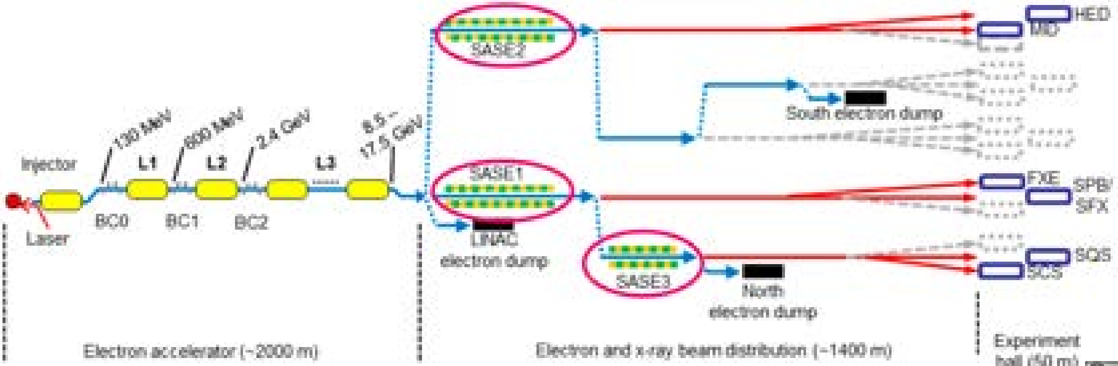
- Analyzing electron orbits in undulator SASEs.
- Various types of problems are indicated by variations in orbits.



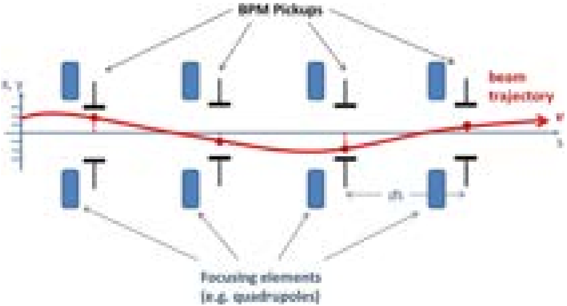
Orbit Monitoring

Assumption

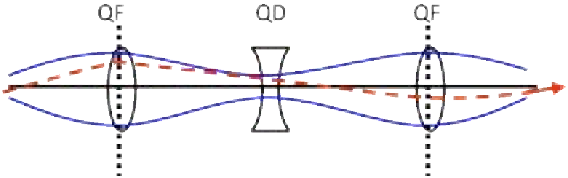
There is a systematic pattern shown in orbits given by the physical construction of EuXFEL.



Orbit Monitoring - FODO Lattice



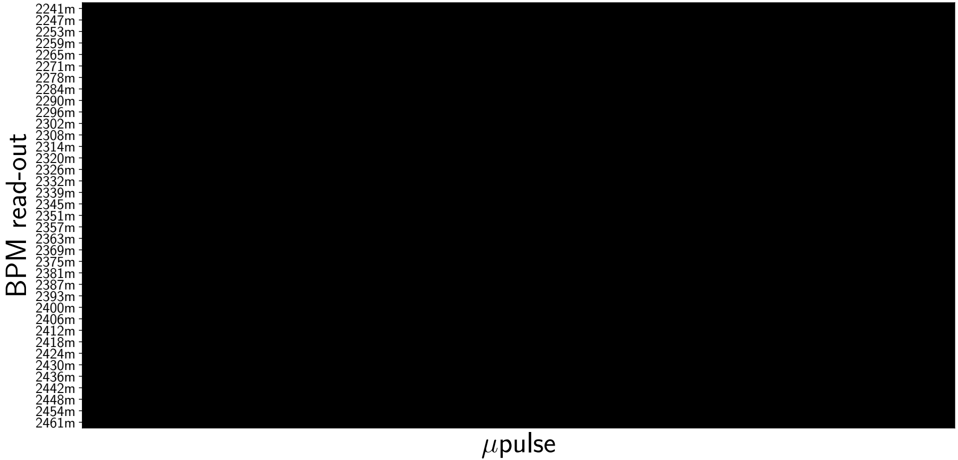
Source: [Wendt(2011)]



Source: [Holzer(2006)]

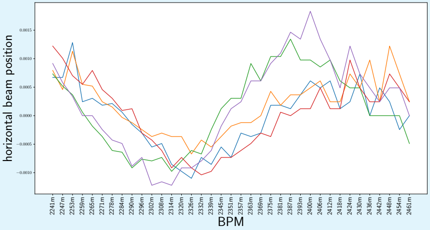
Orbit Monitoring

Orbit Monitoring



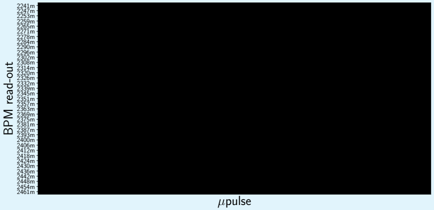
Data-Driven Orbit Monitoring

Model-Based Orbit Monitoring



- Credits to **R. Kammering**.
- Fit a sine and measure residual.
- **Pros** It is **fast** and **intuitive**.
- **Cons** No intra-bunching instabilities.

Model-Free Orbit Monitoring

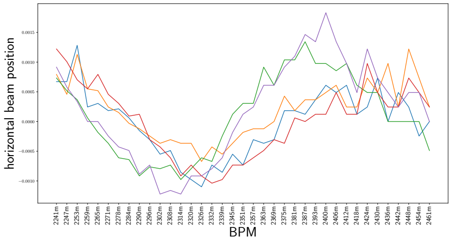


- Series of BPM read-outs.
- Train a model with the anomaly loss.
- **Pros** Takes into consideration intra-bunching instabilities.

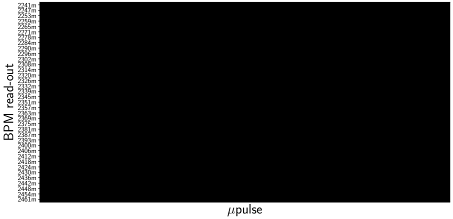
Data

- **No labels**, just logbook records.
- We rarely have an **interpretation** how a failure is propagated to orbit (if so).
- We have 84351 minutes which spans from **Feb 25 2022** to **May 19 2022**.
- Each minute record has ≈600 macro-pulses (1 minute, 10 Hz) ordered sequentially.
- Only **first bunch** in the bunch train is considered!

BPM/position (each line is one bunch)



μ-pulse vs BPM (stack of bunches)



Model-Free Data-Driven Orbit Monitoring

$$\underbrace{\underset{\text{input}=2|\text{BPM}|}{\mathbf{h}_1 = \text{A Layer}(\mathbf{x})} \longrightarrow \cdots \longrightarrow \phi = \text{Linear}(\mathbf{h})}_{f_\theta(\mathbf{x})} \longrightarrow \underset{\text{input}=16,\ \text{output}=1}{s(\phi) := \|\phi - \mathbf{c}\|_2^2}$$

- Units that are scoring a series of BPM read-outs.
- Each input is stacked in horizontal and vertical positions from all beamlines.
- We **do not** have any **faulty labels**.
- **Unsupervised one-class anomaly** loss [Ruff(2018)]:

$$L(\theta) = \| f_\theta(\mathbf{x}) - \mathbf{c} \|_2$$

- Trained features f_θ can be used for further analysis, e.g. TSNE embedding.

Models

Linear - The Simplest

Bunches **scored independently**, i. e. no correlations of adjacent bunches are taken into consideration.

Recurrent - The Classics for Sequences

Has an **”internal state”** and takes into consideration other bunches and their order

Convolutional - Translation Invariant

Does not have an internal state, but is **translation invariant**. It still considers the order.

Transformers - Permutation Invariant

Takes co-occurrences of some bunches into consideration (if the positional encoding is not used) - i. e. **permutation invariant**.

Linear Units

$$\underbrace{\underset{\text{input}=|\text{BPM}|,\ \text{output}=|\text{BPM}|}{\mathbf{h}_1 = \text{Linear}(\mathbf{x})} \longrightarrow \underset{\text{input}=|\text{BPM}|,\ \text{output}=|\text{BPM}|}{\mathbf{h}_1 = \text{Linear}(\mathbf{x})} \longrightarrow \underset{\text{input}=|\text{BPM}|,\ \text{output}=16}{\phi = \text{Linear}(\mathbf{h}_2)}}_{f_\theta(\mathbf{x})} \longrightarrow \underset{\text{input}=16,\ \text{output}=1}{s(\phi) := \|\phi - \mathbf{c}\|_2^2}$$

- The **simplest**.
- Individual bunches are **independent**.
- Surprisingly **powerful** already.
- Very **hard to interpret** (in general).

Recurrent Neural Networks - Old Classics

$$\underbrace{\underset{\text{input=|BPM|, output=|BPM|}}{\mathbf{h}_1 = \mathrm{LSTM}(\mathbf{x})} \longrightarrow \underset{\text{input=|BPM|, output=|BPM|}}{\mathbf{h}_1 = \mathrm{LSTM}(\mathbf{x})} \longrightarrow \underset{\text{input=|BPM|, output=16}}{\phi = \mathrm{Linear}(\mathbf{h}_2)}}_{f_\theta(\mathbf{x})} \longrightarrow \underset{\text{input=16, output=1}}{s(\phi) := \|\phi - \mathbf{c}\|_2^2}$$

- A **classic way to handle sequences**, literally a few lines of code to make a model.
- **Slow** (very inefficient on GPUs).
- Results are **similar** to the purely **linear** model.

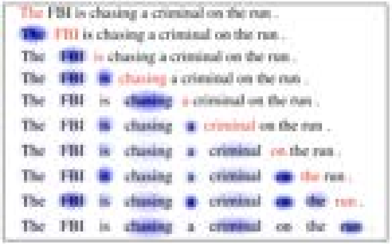
Transformers

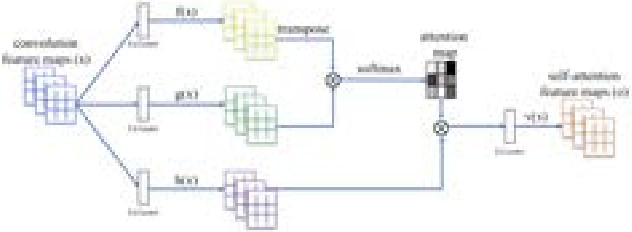
$\mathbf{h}_1 = \text{Transformer}(\mathbf{x})$ → $\phi = \text{Linear}(\mathbf{h}_1)$ → $s(\phi) := \|\phi - \mathbf{c}\|_2^2$

input=|BPM|, output=128 | input=128, output=16 | input=64, output=1

$$\underbrace{\mathbf{h}_1 = \text{Transformer}(\mathbf{x}) \rightarrow \phi = \text{Linear}(\mathbf{h}_1)}_{f_\theta(\mathbf{x})}$$

- Transformers have self-attention mechanism, puts other inputs into a **”context”**
- **Permutation invariant**, scores all bunches in correlation with other bunches (if no positional encoding is used).



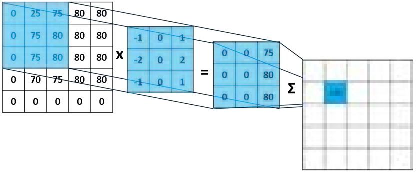


Source: https://lilianweng.github.io/posts/2018-06-24-attention/

Convolutional Neural Networks

$$\underbrace{\underset{\text{input}=|\text{BPM}|}{\mathbf{h}_1 = \text{Conv2d}(\mathbf{x})} \longrightarrow \mathbf{h}_2 = \text{Conv2d}(\mathbf{h}_1) \longrightarrow \underset{\text{output}=16}{\phi = \text{Linear}(\mathbf{h}_2)}}_{f_\theta(\mathbf{x})} \longrightarrow \underset{\text{input}=16,\ \text{output}=1}{s(\phi) := \|\phi - \mathbf{c}\|_2^2}$$

- **Translation** invariant.
- Unlike RNNs very **fast** on GPUs!
- The dynamic number of bunches was eliminated via calculating maximum over features along bunch dimensions.
- Therefore a single number for an arbitrarily long sequence.



Source: https://mlnotebook.github.io/post/CNN1/

CNN + Transformers

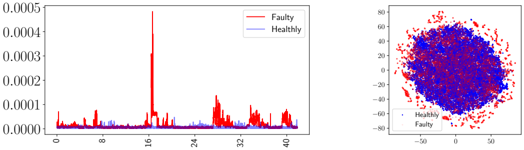
$$\underbrace{\underset{\text{input}=|\text{BPM}|}{\mathbf{h}_1 = \text{Conv2d}(\mathbf{x})} \longrightarrow \mathbf{h}_2 = \text{Transformer}(\mathbf{h}_1) \longrightarrow \underset{\text{output}=16}{\phi = \text{Linear}(\mathbf{h}_2)}}_{f_\theta(\mathbf{x})} \longrightarrow \underset{\text{input}=16,\ \text{output}=1}{s(\phi) := \|\phi - \mathbf{c}\|_2^2}$$

- **Translation and permutation** invariant.
- Still fast on the GPU.
- Fixes the problem with the dynamic number of bunches, each bunch is like a token after being processed by CNN.

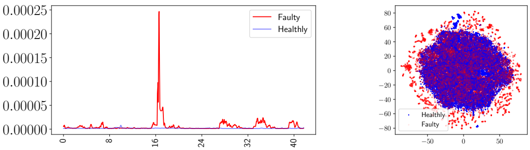
Example 1

The undulator server crashed after an unusual selection of colors for individual cells

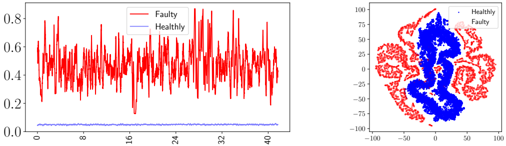
Linear



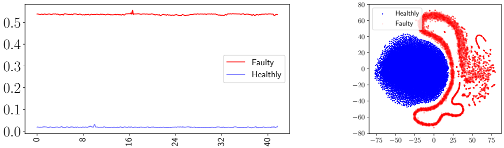
RNN



Transformer



CNN+Transformers



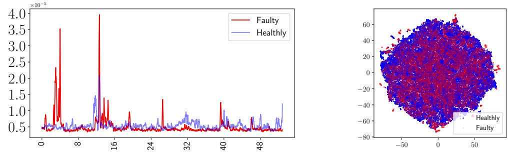
CNN

$s_{faulty} = \mathbf{0.0639}$ $s_{healthly} = 0.003$

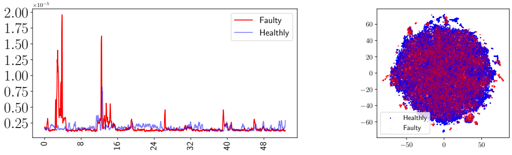
Example 2

An issue with a cold quadrupole in A6

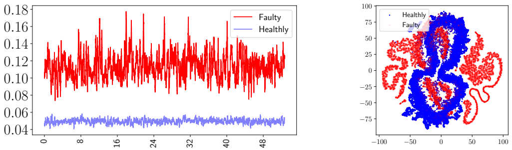
Linear



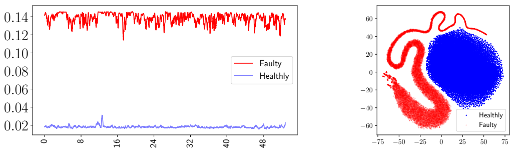
RNN



Transformer



CNN+Transformers



CNN

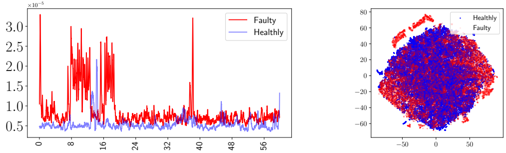
$s_{faulty} = \mathbf{0.0086}$

$s_{healthly} = 0.003$

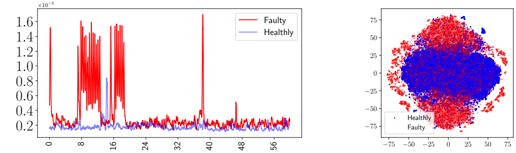
Example 3

A CPU failure on one of the control boards

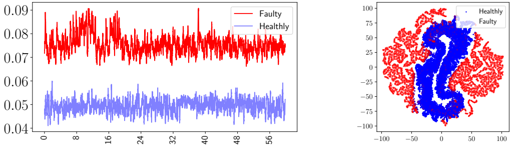
Linear



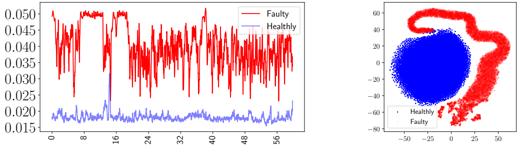
RNN



Transformer



CNN+Transformers



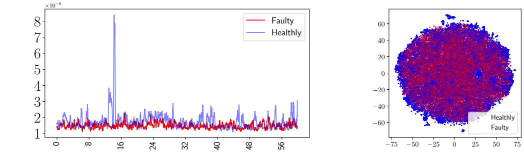
CNN

$s_{faulty} =$ **0.005** $s_{healthly} = 0.003$

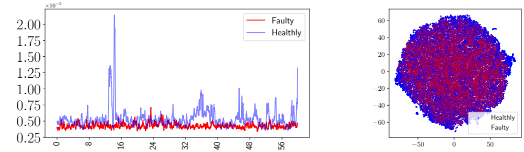
Example 4

Safety magnet at SA1/SA3

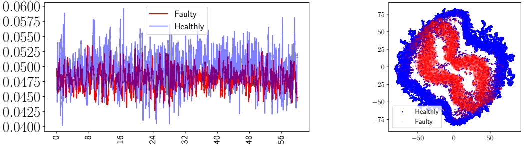
Linear



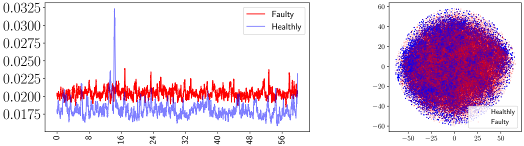
RNN



Transformer



CNN+Transformers



CNN

$s_{faulty} = 0.0003$ $s_{healthly} = \mathbf{0.003}$

Conclusion and Outlook

- We used **classical** ML models including well-established convolution neural nets (image processing) and transformers (language models).
- **Lack of interpretation** makes it **extremely challenging** (labelling, interpreting).
- Linear and RNN have lower recall but higher precision, unlike CNN and transformers.
- Especially transformers, CNN and their combinations show **how a context is important**.
- In the future we would like to try generative models, e. g. normalizing flows, (RNN)-VAE or GANs.

Thank you for your attention!

This is the joint work of R. Kammering and T. Wilksen!

https://github.com/sulcantonin/ICFA-Beam-2022

And I shouldn't forget to mention that we are very eager to collaborate!

Contact

DESY. Deutsches Elektronen-Synchrotron

www.desy.de

A. Sulc, R. Kammering, T. Wilksen
0000-0001-7767-778X
MCS DESY
antonin.sulc@desy.de

Bibliography I

- BJ Holzer.
  Lattice design in high-energy particle accelerators.
  2006.
- L Ruff.
  Deep one-class classification.
  In International conference on machine learning, pages 4393–4402. PMLR, 2018.
- Manfred Wendt.
  Overview of recent trends and developments for bpm systems.
  Technical report, Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States), 2011.

MLops Repository

We started to collect ideas about what we have talked about towards approaches and unification of MLops operations.

https://github.com/owle-ml/owle-ml

1. Fork the repo.
2. Create a pull request.
3. We (Tia Miceli, me, event. someone else) would approve it (for transparency)