ML study program for particle accelerators at IJCLAB

Viacheslav Kubytskyi

and Hayg Guler

Overview

- Accelerators related topics and ML

- Several case studies. Difficulties and progress.

- Discussion

Advanced control and optimization techniques

Accelerator related topics:
- Tuning/optimization/control
- Data analysis
- Simulation/modeling
- Prognostics/alarm handling/anomaly-breakout detection

&

ML Techniques :
Supervised Learning (Regression, Classification) ,
Unsupervised Learning (Clustering,
Dimensionality reduction),
Reinforcement Learning.

Our **goal** is investigation and demonstration of applicability and efficient use of Machine Learning (ML) techniques for advanced control and optimization of particle accelerators.

Improving accelerators:
- machine tuning and beam dynamics, beam parameters extraction, dealing with noisy data;
- diagnostics and control of high intensity laser
- virtual detectors for machine monitoring purposes

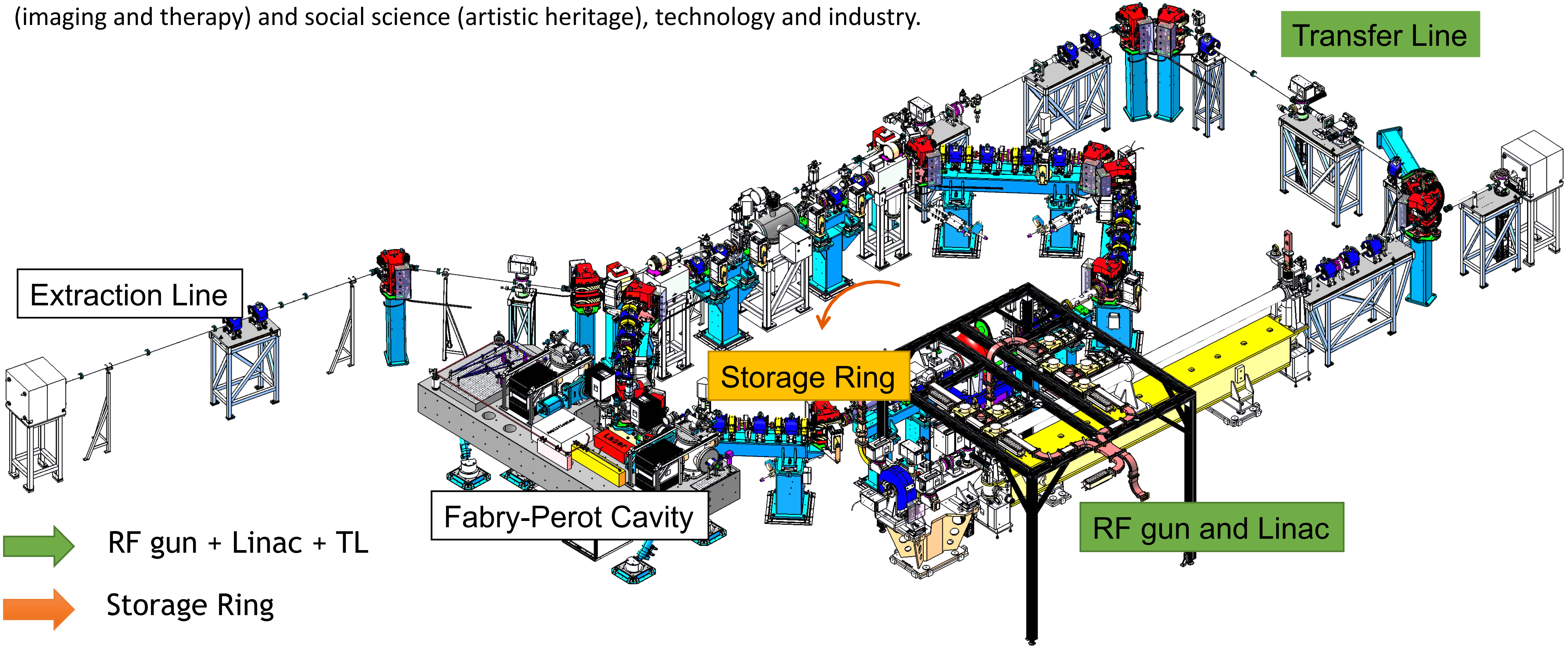
Practically: ML group is working on set of problems with further generalization. Collaboration with ML field experts.

Puppy
or
muffin?

THOMX project at IJCLAB

ThomX - Compact source of X-rays

Producing a compact source of directional X-rays, with high performance, very bright, monochromatic and with adjustable energy for application to the field of medical science (imaging and therapy) and social science (artistic heritage), technology and industry.



RF gun + Linac + TL

Storage Ring

Fabry-Perot Cavity / X-line

Waiting for ASN permission to start

Typical difficulties during commissioning

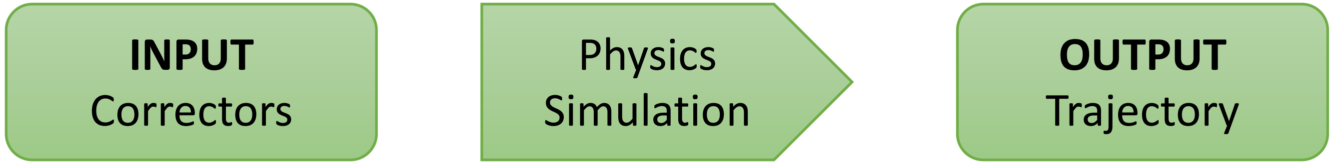
- **Linac and transfer line:** find day to day repeatable beam conditions is challenging task (temperature conditions, laser jitter, …). Reproducible beam dynamics. Injection matching with Ring.

- **Ring:** Need to stabilize machine, find stable orbit.
  First turns – need to match several parameters.
  There are empirical ways. Machine expert can do (experience).
  We search for automatization/ guiding. Study case.

RING: Study case. Some details.

Single particle Trajectory (several turns):

1. Control parameter: Corrector magnets. 12 independent variables in transverse horizontal/vertical planes.
2. Physics simulation. Accelerator Toolbox (MATLAB).
3. Measured: trajectory/n-turns/orbit represented by 120 variables (12 BMPs x 10 Turns).



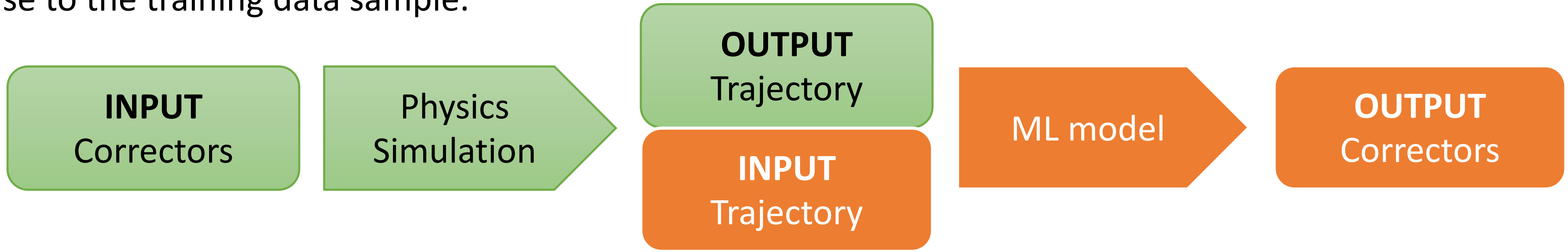
Supervised learning:

1. Prepare vectors of corrector and calculate corresponding trajectories.
   MATLAB Engine API for Python.
2. Train model to predict correctors based on the trajectory input.
   XGBRegressor + MultiOutputRegressor. Jupyter notebook at server 48 threads ~ 1 hour.

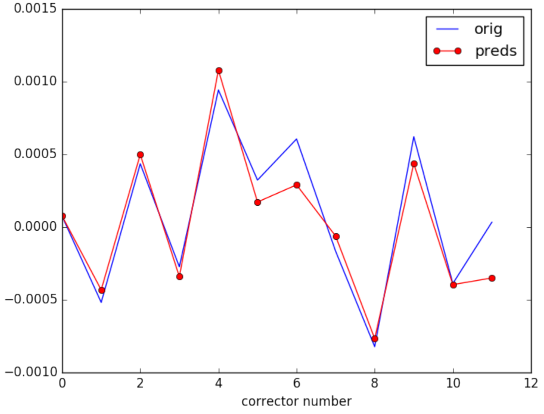
In real life: noise in measurements, BPM readings at low statistics not so good, etc.
Once orbit is found: large statistics allows to use well developed methods in accelerator physics (SVD, optimization, etc.).

Tests, validation:

1) Correctors->Measure Trajectory(simulation) ->Insert to ML model(ML)-> predicted Correctors
2) How different initial and predicted correctors? How different trajectories?
3) Add noise to the training data sample.

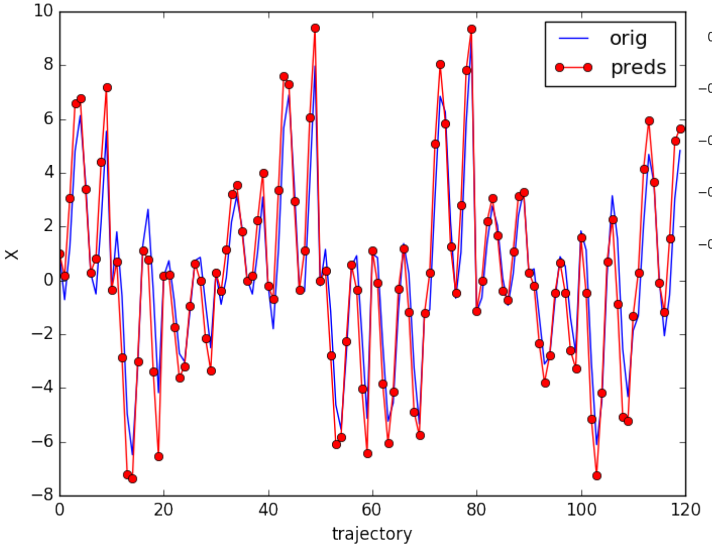


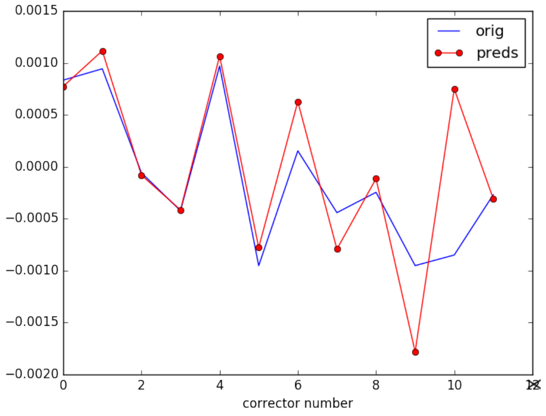
measured
predicted



Correctors

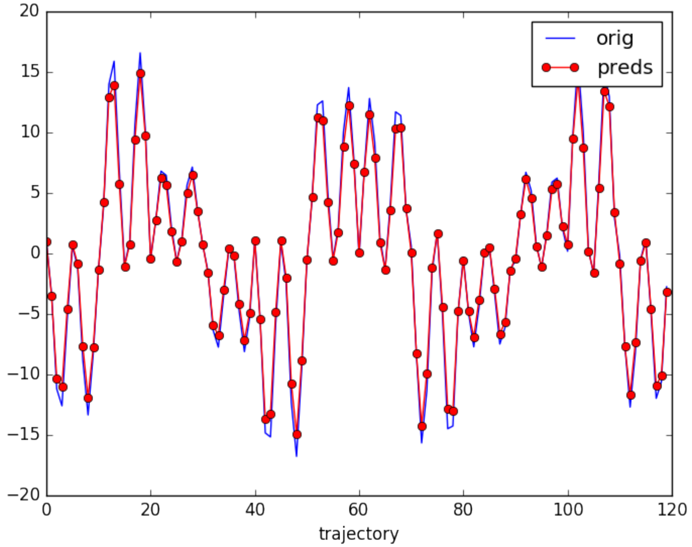
Trajectory





Correctors

Trajectory



LINAC: Study case. Image analysis for beam tuning.

Beam shape measurement by YAG screens is destructive.

Is it possible to retrieve beam parameters non-destructively but using only some accelerator parameters?

Fast tuning? Way to ideal beam ?

Use and analyze images using supervised learning, associate accelerator parameters with images

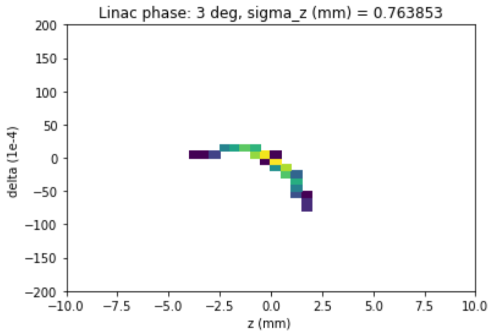
Some details:

Beam parameters->Simulation-> Images->Training-> Beam Parameters

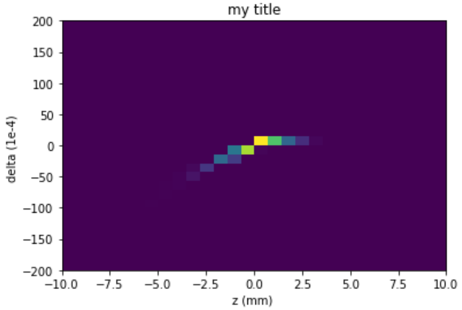
CNN : Keras (Tensorflow) model with simulated particle beam = f(RF_phase). Parameters->

Supervised learning using 60 k-Images for training

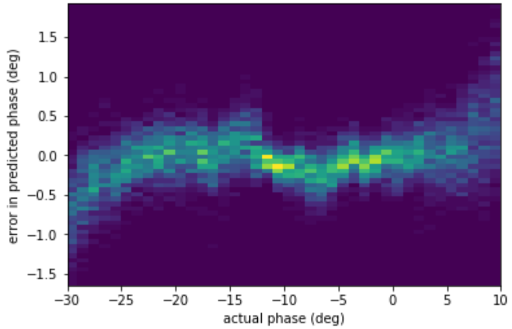
Beam longitudinal phase space



Transform as image for the training



Good accuracy on test data

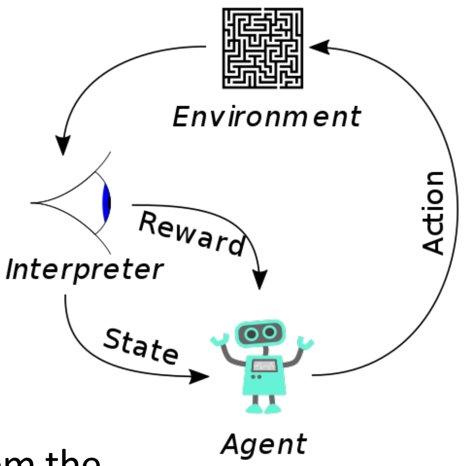


LINAC: Use of Reinforcement learning
Beam tuning, accelerator design

Motivations : Why using RL for accelerators ? Idea to mimic an operator that tunes an accelerator by playing with the set of parameters and gets rewards when he gets good beam conditions.

Methodology :

- Agent (= Virtual operator) learns from simulation and gets reward for "good" actions
- Used simulation should be fast and realistic inside OpenAI-GYM environment
- Find proper RL method (Q-leaning, Deep Q-learning etc)



For accelerator (existing at LAB) tuning purpose:

- Our goals :
  - prove the power of RL to tune an accelerator from the simulations.
  - Compare to other methods (SL for example)
  - understand the transfer learning
- Gym environment created for the Linac based beam transport code

Known problems :

- RL is less efficient than supervised learning
  - Single agent (multiple agent convergence is difficult)
  - Discrete actions vs continuous
  - continuous (needed) has weaker convergence
- Speed of the realistic simulator is an important point
- Use of supervised learning for example
- GAN could also be considered
- For us, we could increase beam current to increase the collective effects to help the learning

Subject is quite used in the Robots comunity
Forseen collaboration will be starting with
ENSTA-U2IS (robots)
Common Phd subject is discussed

Discussion

- Accelerator physics could strongly benefit from machine learning tools
  Outcomes: very precise control of parameters and machine/laser stability.
  Better understanding of space of parameters and their importance, etc.

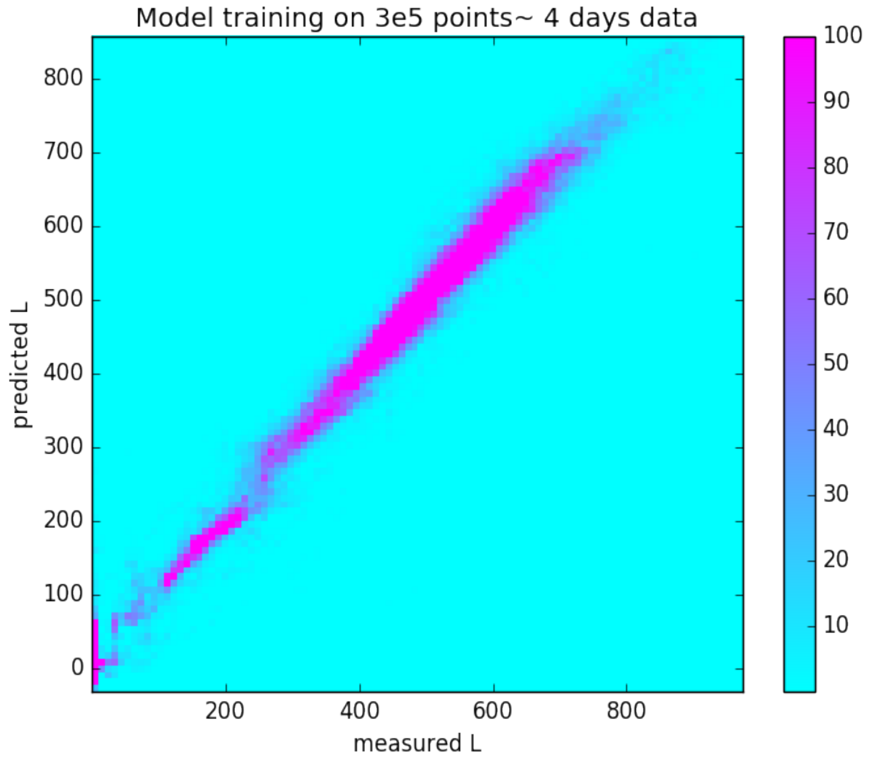
- Solve set of simple problems with ML, select “interesting” ML tools to improve accelerators.
  Qualify ML concepts for possible generalization and application for particle accelerators.
  Share with community.

BACKUP

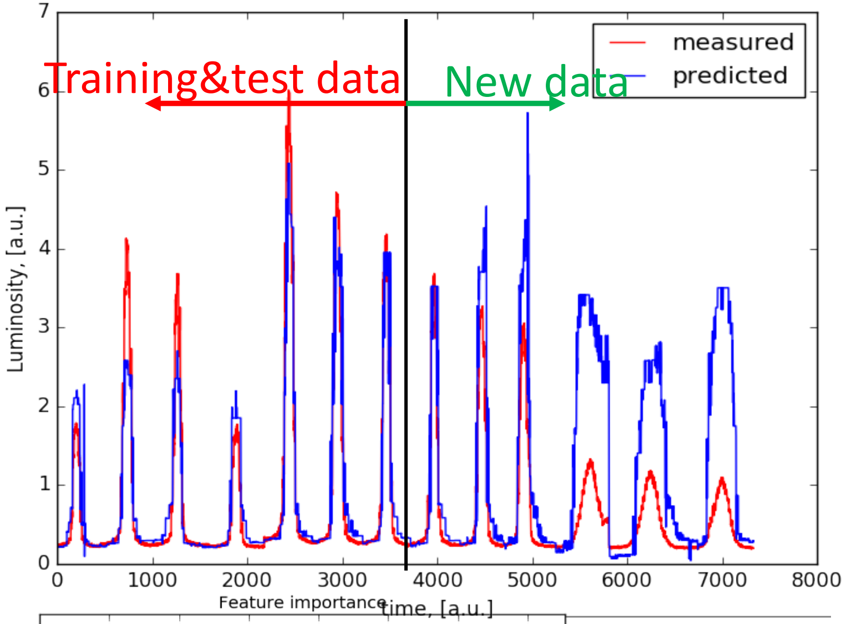
Can we apply the similar methodology other problems?

Example 1: training a virtual luminosity monitor

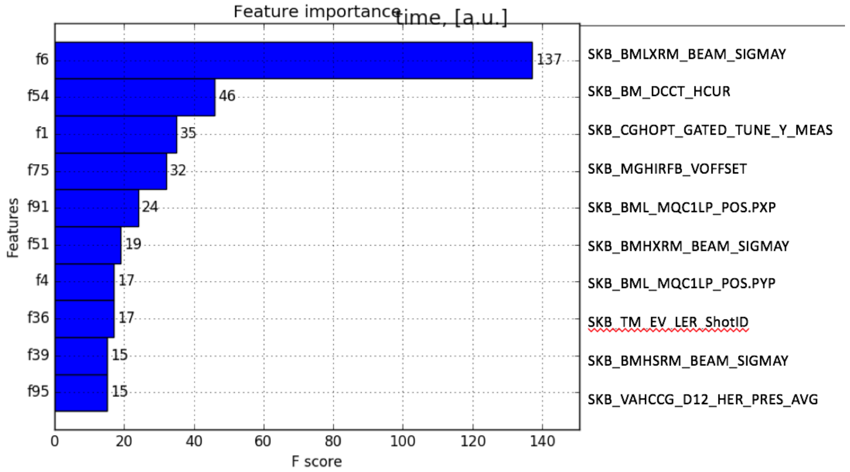
- LumiBelle2 database consists **of 165 EPICS PV variables** from LumiBelle2/SuperKEKB at **1Hz.**
- Good quality data and the PV variables are synchronous, very important for ML studies.
- XGBoost for ML model.



Example 2: training over luminosity scans



- In total 30 scans (100* 2e5 datapoints): use first 24 for training and last 6 for prediction.
- The agreement is not perfect but number of scans available is relatively small.



- Feature importance reveals strongest dependencies, however could be sensitive to the data sample chosen