Machine Learning Techniques for Accelerators

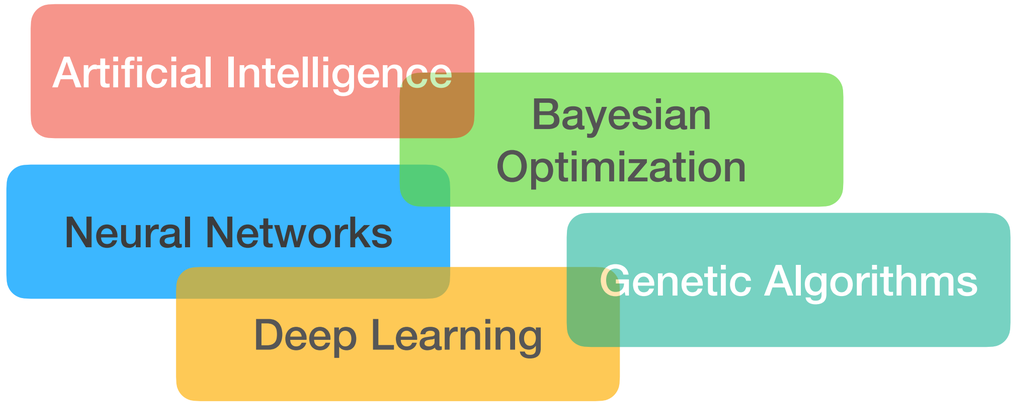
Elena Fol

CERN, BE-ABP-LAF

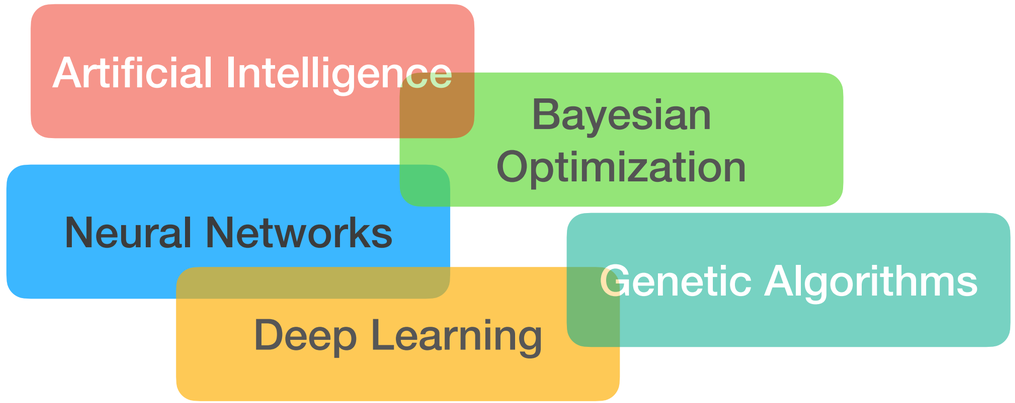
Many thanks to D. Schulte, C. Rogers, G. Franchetti, R. Tomas and OMC Team

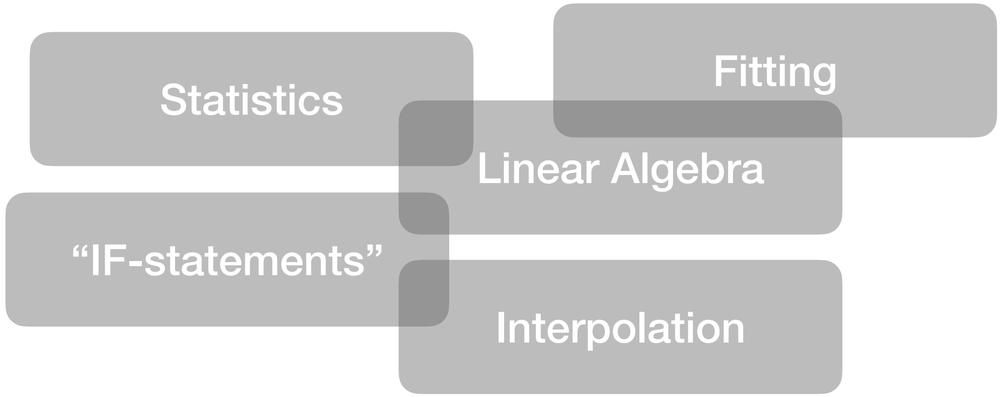
3rd Workshop on Low Emittance Lattice Design - LEL 2022
26th - 29th of June 2022

What does “Machine Learning” mean?

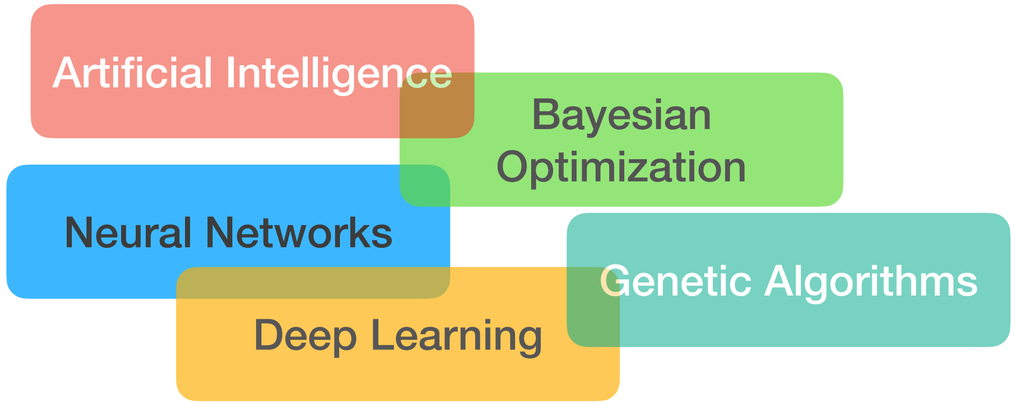


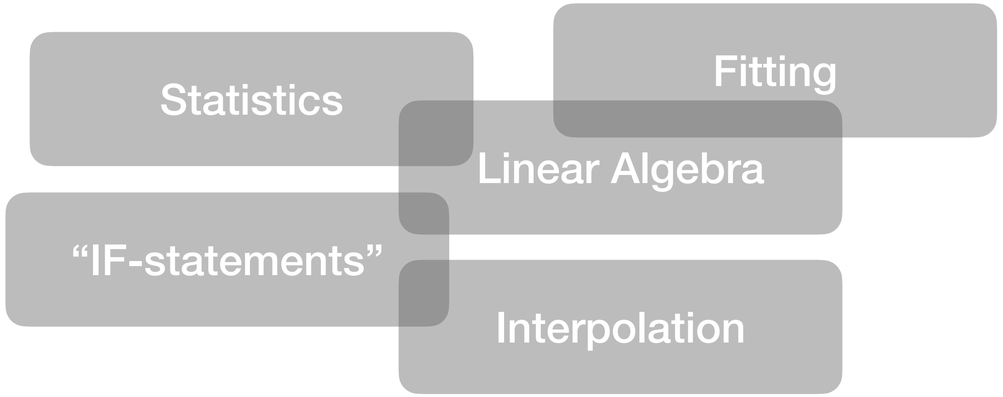
What does “Machine Learning” mean?





What does “Machine Learning” mean?





- **Strong AI:** Should be able to perform any task, indistinguishable from human —> does not exist (yet?)

- **Narrow AI:** Execute specific tasks —> outperforms humans in e.g. financial forecasting, medical diagnostics

- **Machine Learning** = a toolkit consisting of many different concepts and mathematical methods
  - ✓ Building solution for **specific tasks**, **without providing explicit instructions**

Learning from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task
  - ➢ learn from examples instead

Learning from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task

➢ learn from examples instead

Learning from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task

➢ learn from examples instead

→ **Cat?**

Relevant ML concepts and definitions

Supervised Learning

- **Input/output pairs** available
- Learn a mapping function, **generalizing for all provided data**
- Predict from **unseen data**

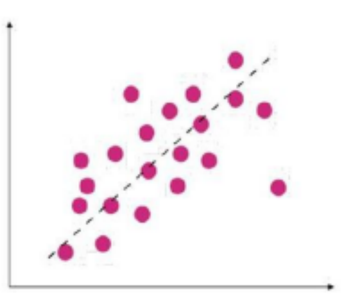
Unsupervised Learning

- **Only input** data is given
- Discover structures and patterns

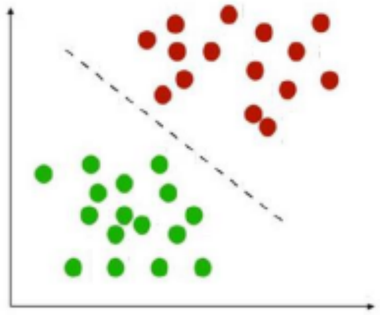
Reinforcement Learning

- No training data
- Interact with an environment
- Trying to learn optimal sequences of decisions

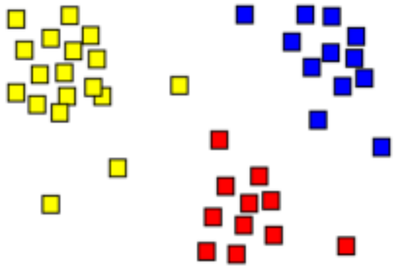
Regression



Classification



Clustering



Learning from data: Supervised Learning

example 1
example 2
example 3
.
.
.

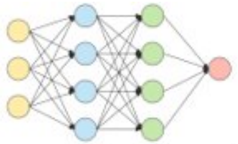
Training input data

Training output data

Learning from data: Supervised Learning

example 1
example 2
example 3
.
.
.

Training input data



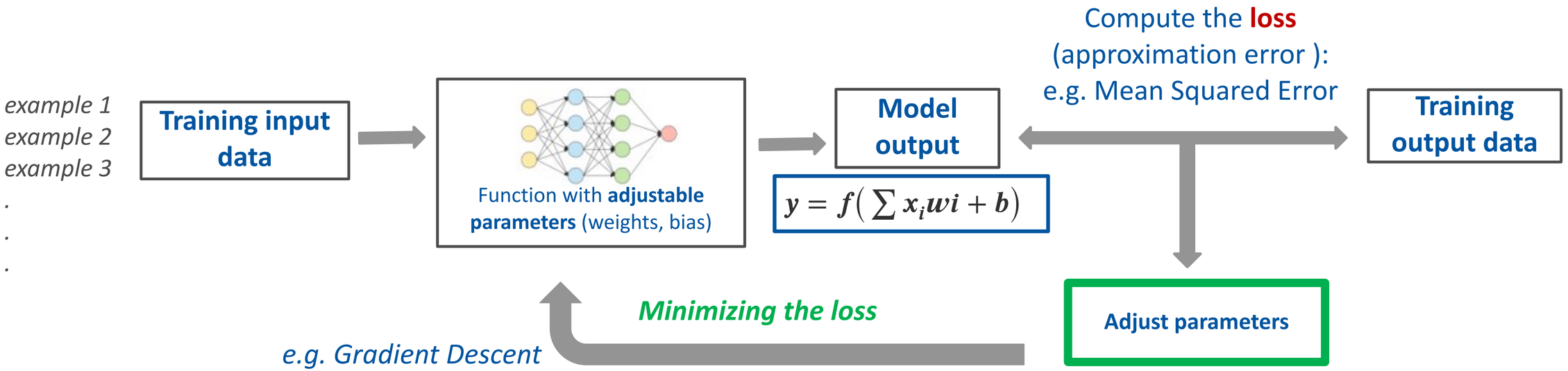
Function with **adjustable parameters** (weights, bias)

Model output

$$y = f\left(\sum x_i wi + b\right)$$

Training output data

Learning from data: Supervised Learning



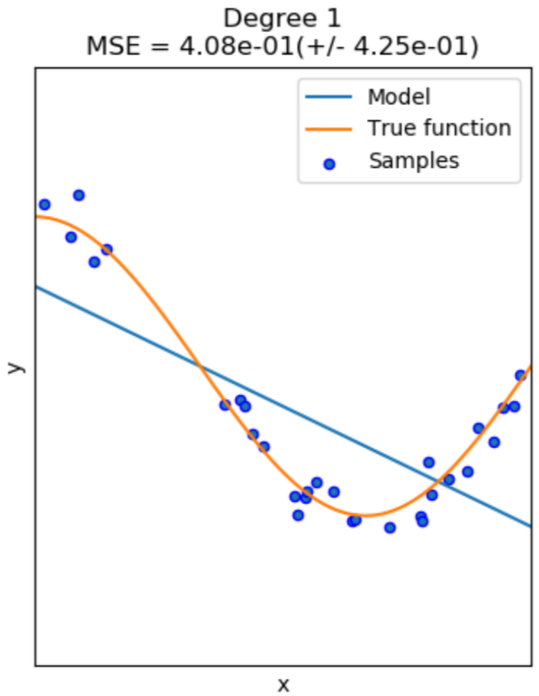
➡ **Learning from data automatically**

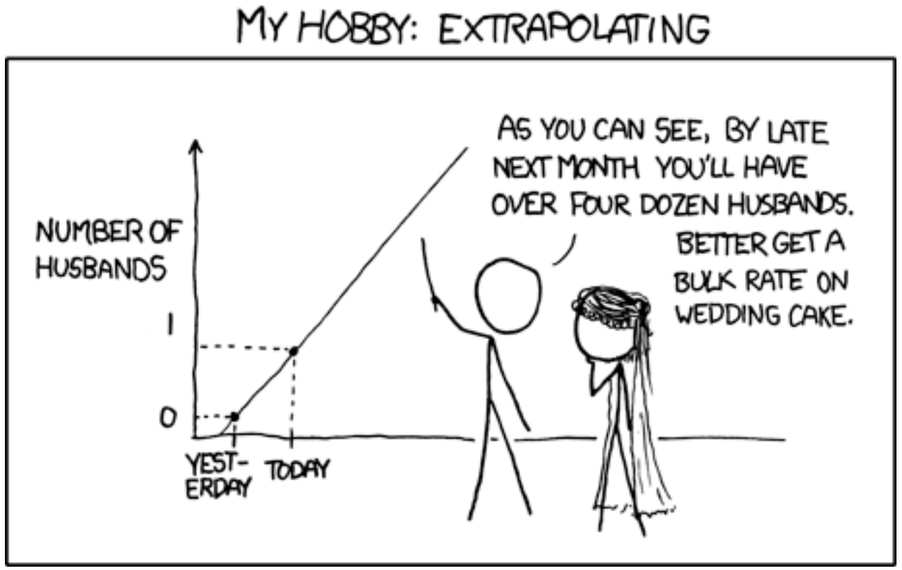
➡ **Explaining relationship between input and output variables in all training samples.**

➡ **Generalisation: the capability of explaining new cases**

- How to prove generalisation?

Training and generalization

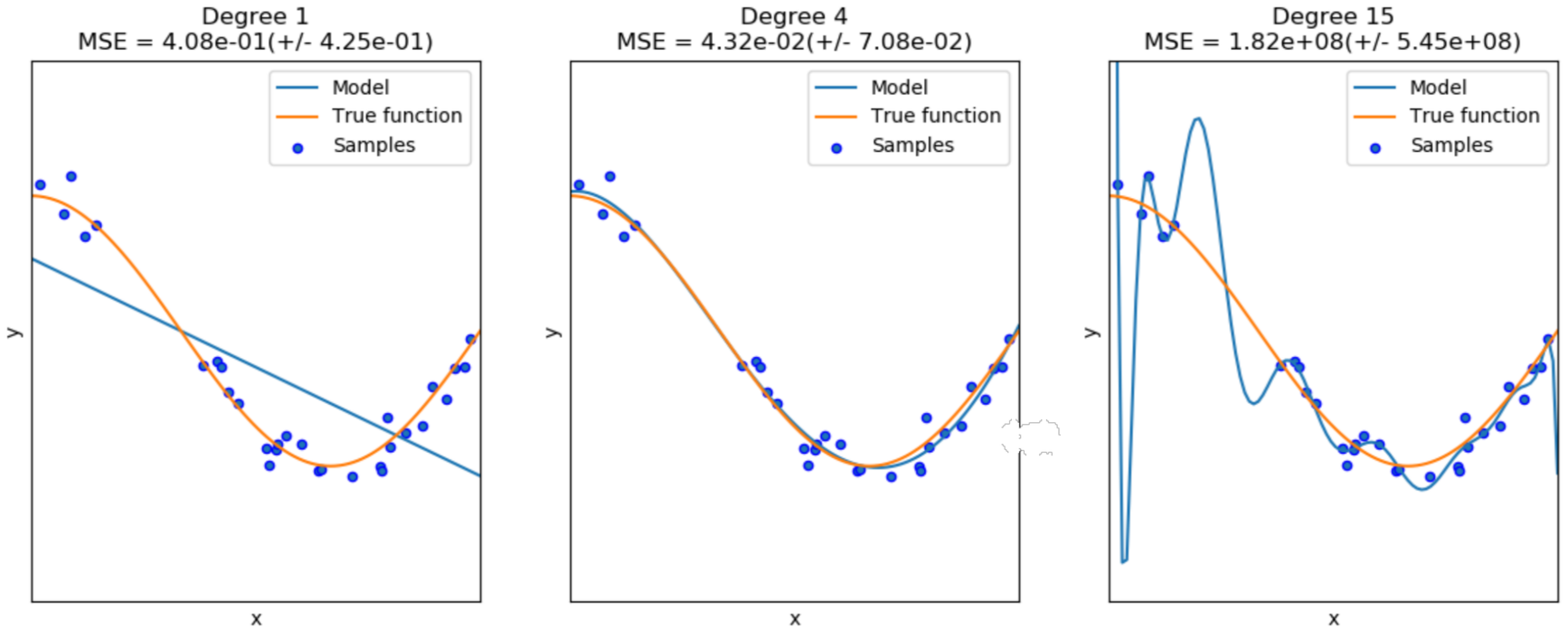




Simple models underfit

- Derivate from data (high bias)
- Do not correspond to data structure (low variance)

Training and generalization



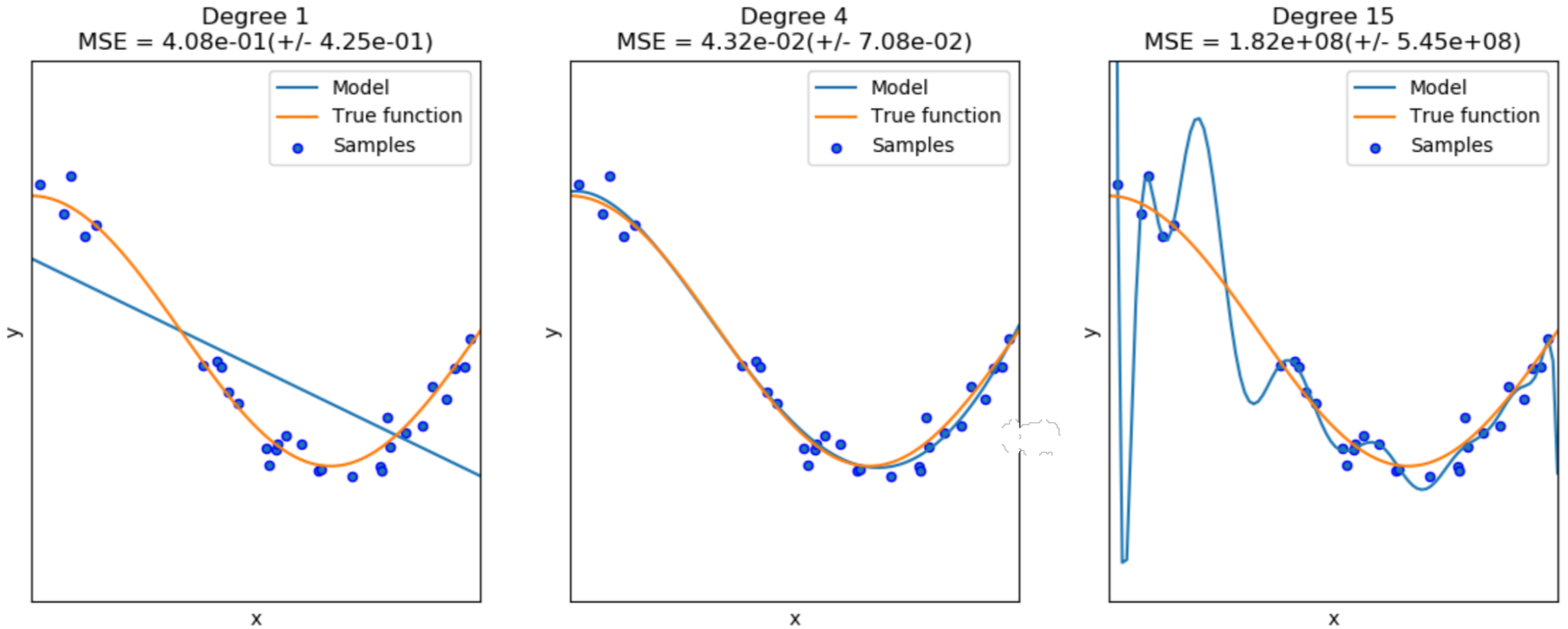
Simple models underfit

- Derivate from data (high bias)
- Do not correspond to data structure (low variance)

Complex models overfit

- Very low systematical deviation (low bias)
- Very sensitive to data (high variance)

Training and generalization



Simple models underfit

- Derivate from data (high bias)
- Do not correspond to data structure (low variance)

→ **Bias-Variance tradeoff**

- **Separate data into train and test sets**
- **Find optimal model hyperparameter e.g. with cross-validation**

Complex models overfit

- Very low systematical deviation (low bias)
- Very sensitive to data (high variance)

Where can we use ML in accelerators?

Detection of instrumentation failures

Beam control and lattice imperfection corrections

Optimization and automation of design and operation

Virtual Diagnostics

- Defining a **narrow task** (optimization of specific parameters rather than the entire machine)
- **Performance measure** of selected model (beam size, pulse energy, …)
- e.g. when no analytical solution is available, rapidly changing systems, no direct measurements are possible.

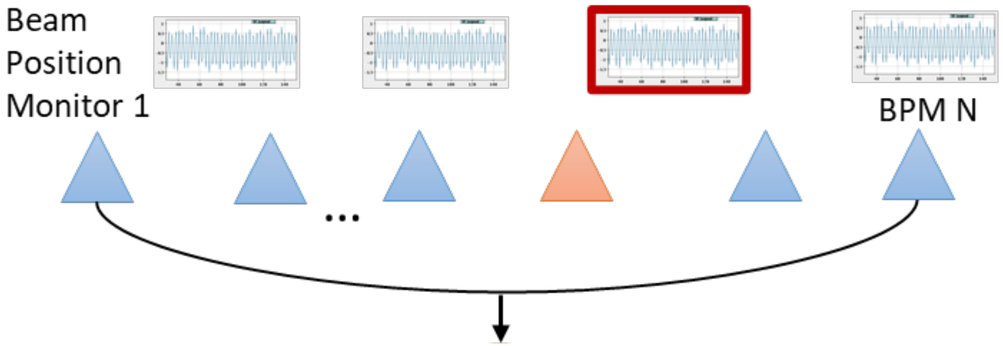
Important to identify where ML can surpass traditional methods

How much effort is needed to implement a ML solution? Is appropriate infrastructure for data acquisition available? Enough resources to perform the training?

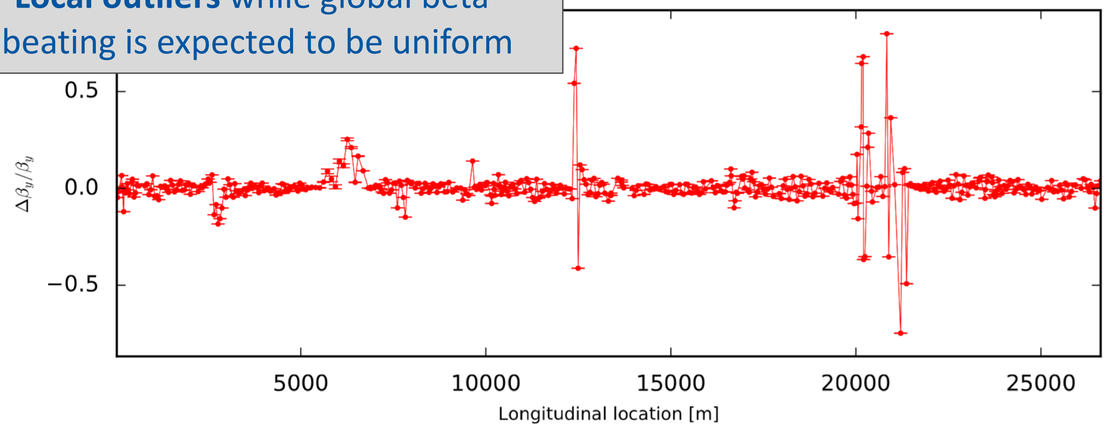
Examples: ML for LHC optics measurements and corrections

Detection of faulty Beam Position Monitors

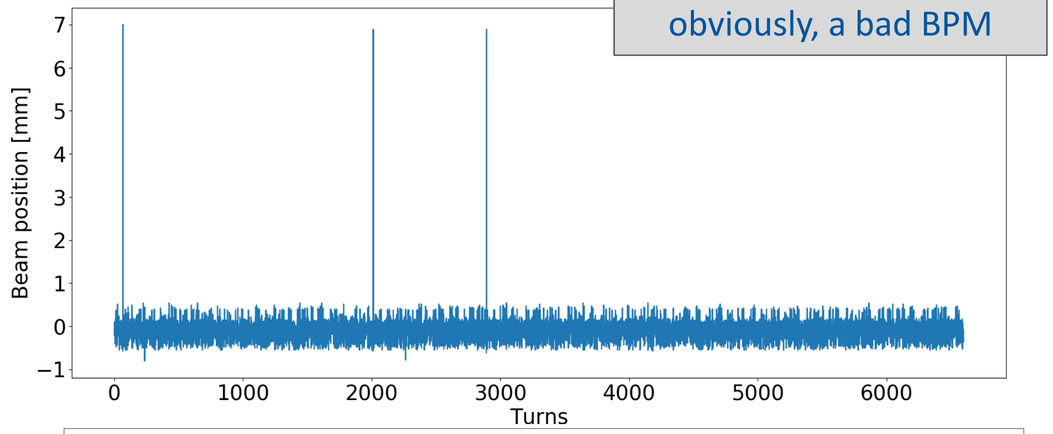
- Faulty BPMs **are a-priori unknown**: no ground truth → **Unsupervised Learning**
- Anomaly detection using **Isolation Forest algorithm** implemented with *Scikit-Learn.*



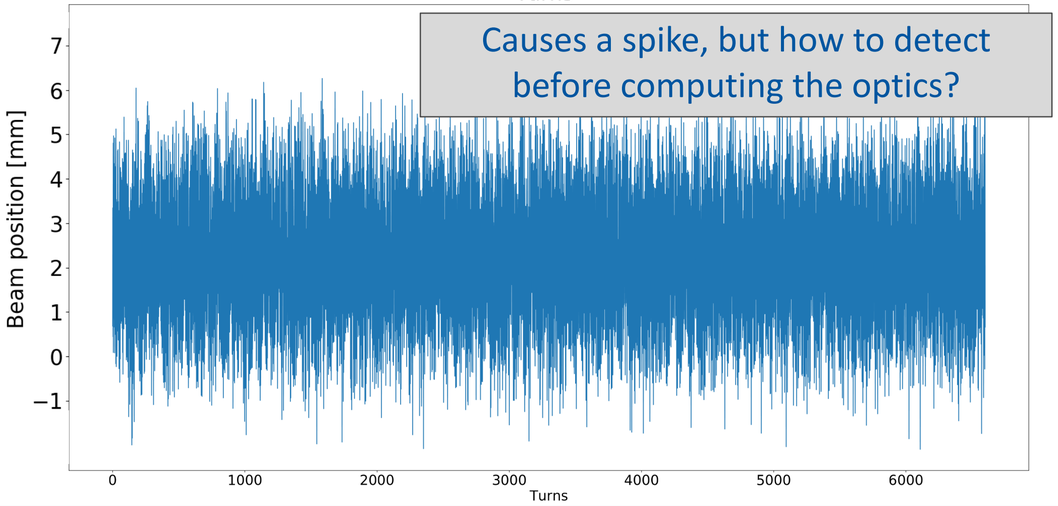
Local outliers while global beta-beating is expected to be uniform



Causes a spike, obviously, a bad BPM



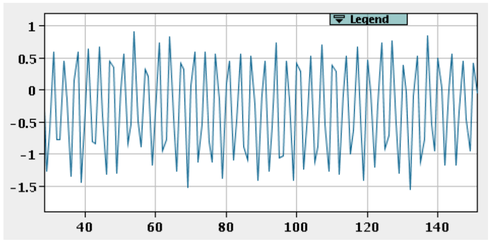
Causes a spike, but how to detect before computing the optics?



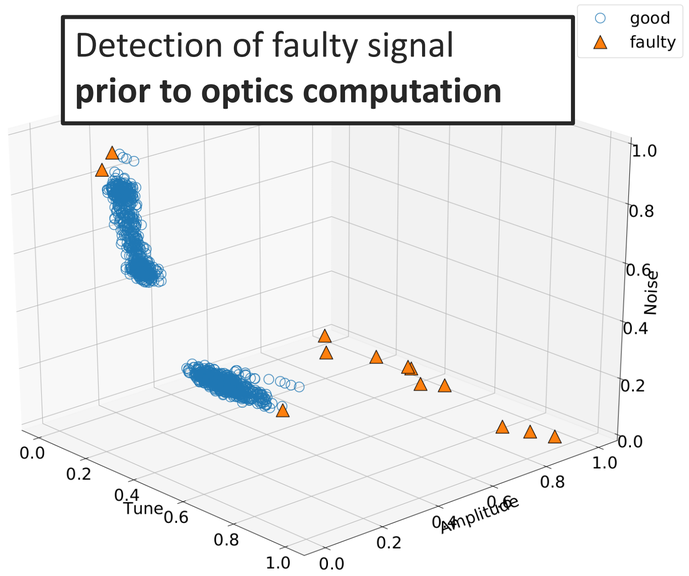
Detection of faulty Beam Position Monitors

- Faulty BPMs **are a-priori unknown**: no ground truth → **Unsupervised Learning**
- Anomaly detection using **Isolation Forest algorithm** implemented with *Scikit-Learn.*

Harmonic analysis of BPM turn-by- turn data

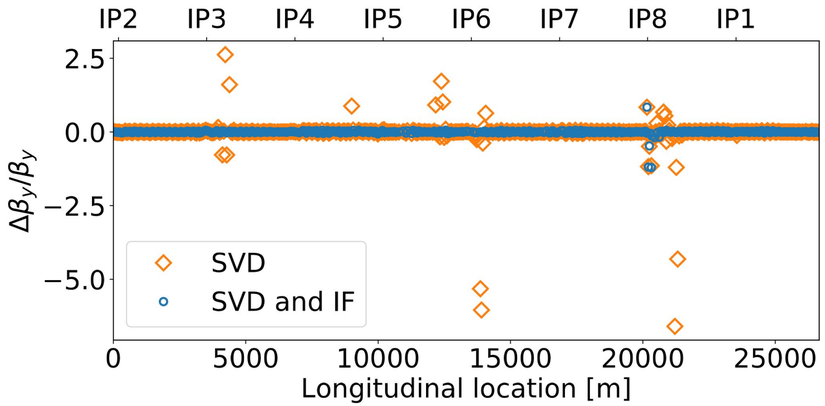


Detection of faulty signal **prior to optics computation**



- Outlier detection based on combination of several signal properties
- Immediate results

Avoid the appearance of erroneous optics computation



LHC Commissioning 2022

✓ Statistical analysis of historical data

✓ Identified **116 critical faulty BPMs out of more than a thousand BPMs** in the LHC.

✓~60 BPMs reveal **actual instrumentation issues**, which **otherwise stay hidden.**

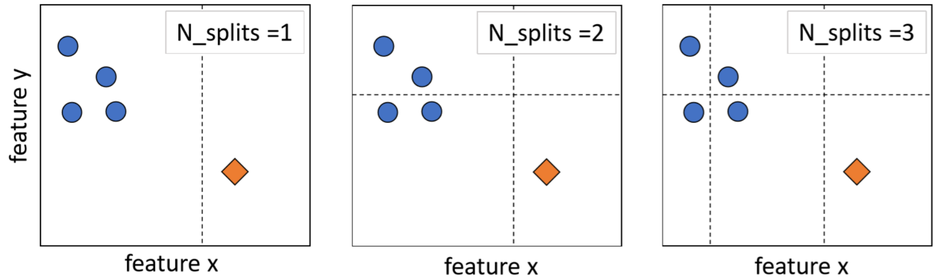
E. Fol et al., "Detection of faulty beam position monitors using unsupervised learning", Phys. Rev. Accel. Beams 23, 102805, 2020

Isolation Forest Algorithm

- Forest consists of several **decision trees***
- **Random splits aiming to “isolate” each point**
- The less splits are needed, the more “anomalous”
- **Contamination factor**: fraction of anomalies to be expected in the given data

→ First obtained empirically from the past measurements

→ Refined on **simulations introducing expected BPM faults**.



Conceptual illustration of Isolation Forest algorithm

*Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. “Isolation forest.” Data Mining, 2008. ICDM‘08.

Isolation Forest Algorithm

- Forest consists of several **decision trees***
- **Random splits aiming to “isolate” each point**
- The less splits are needed, the more “anomalous”
- **Contamination factor**: fraction of anomalies to be expected in the given data

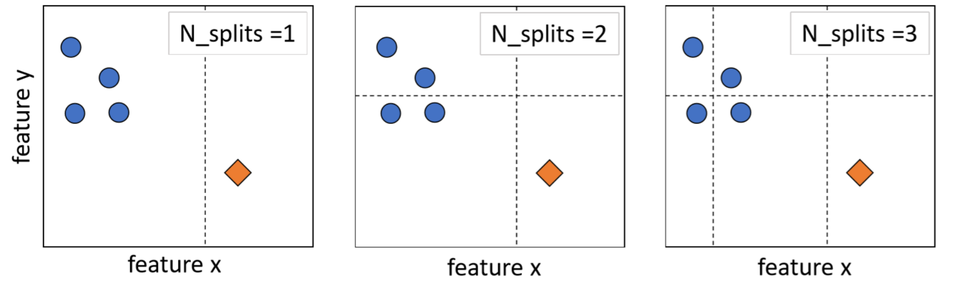
→ First obtained empirically from the past measurements

→ Refined on **simulations introducing expected BPM faults**.

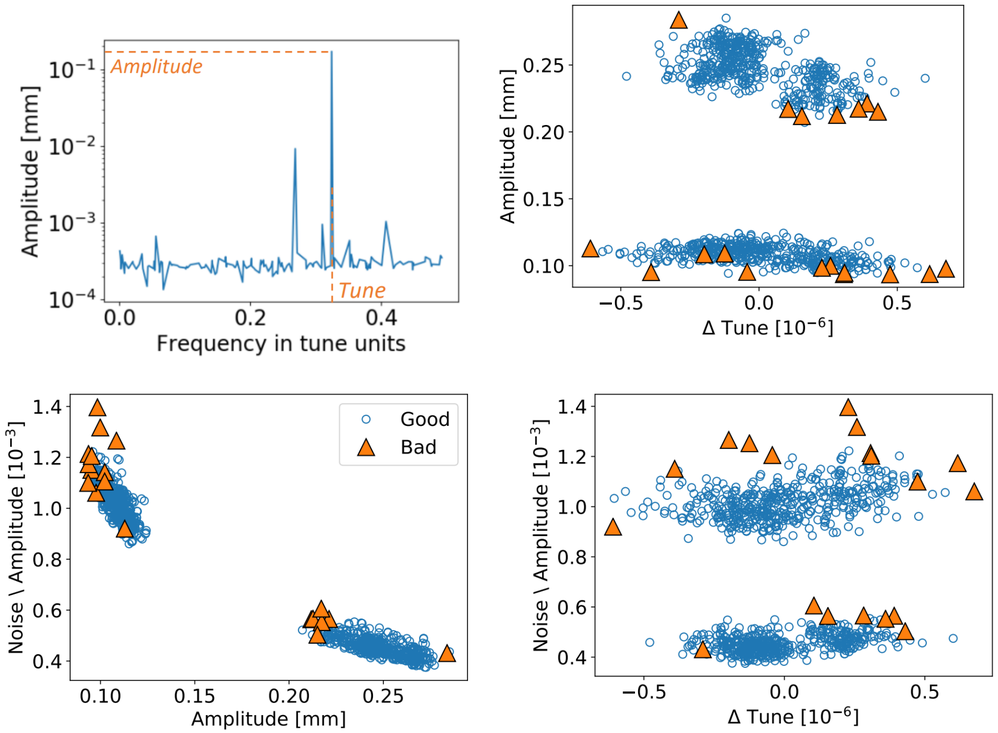
- **Input data: combination of several signal properties** obtained from harmonic analysis of BPM turn-by-turn measurements

→ No additional data handling needed.

→ No training, applied directly on the currently taken measurements data

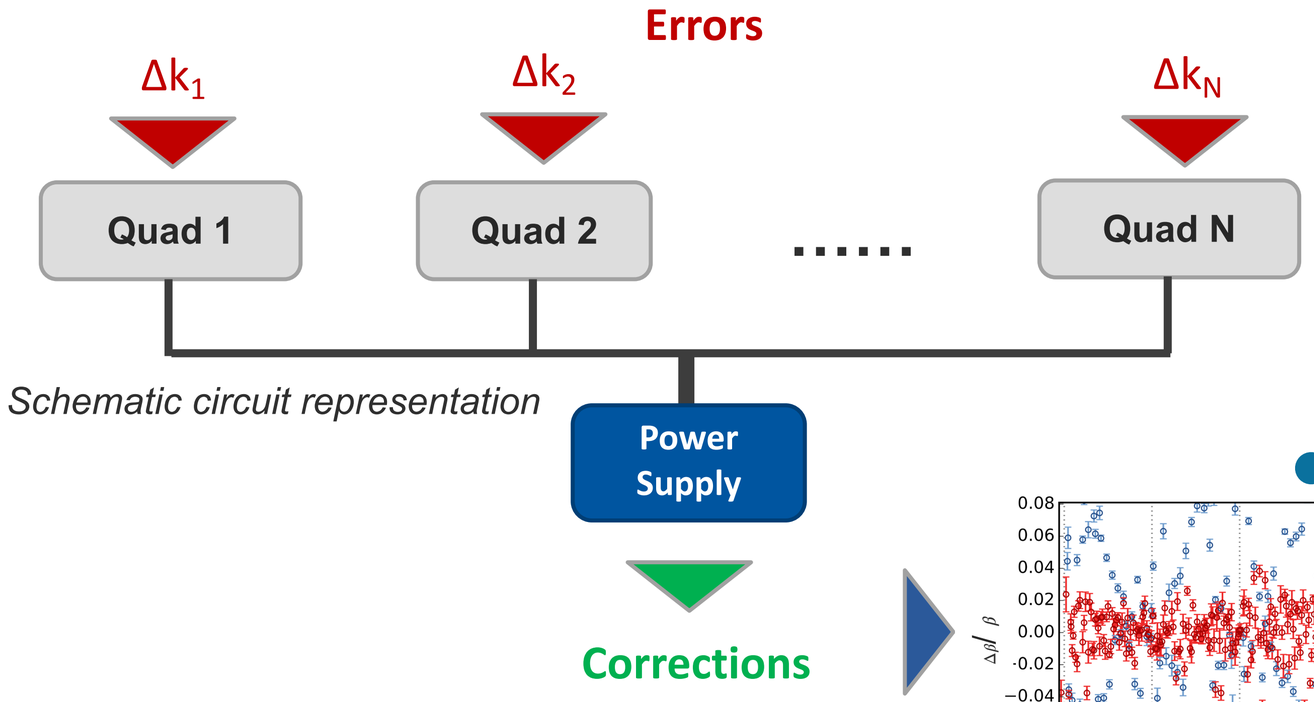


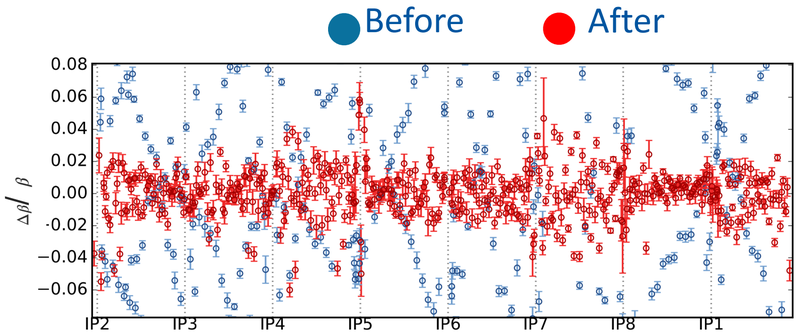
Conceptual illustration of Isolation Forest algorithm



*Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. “Isolation forest.” Data Mining, 2008. ICDM‘08.

Correcting the optics



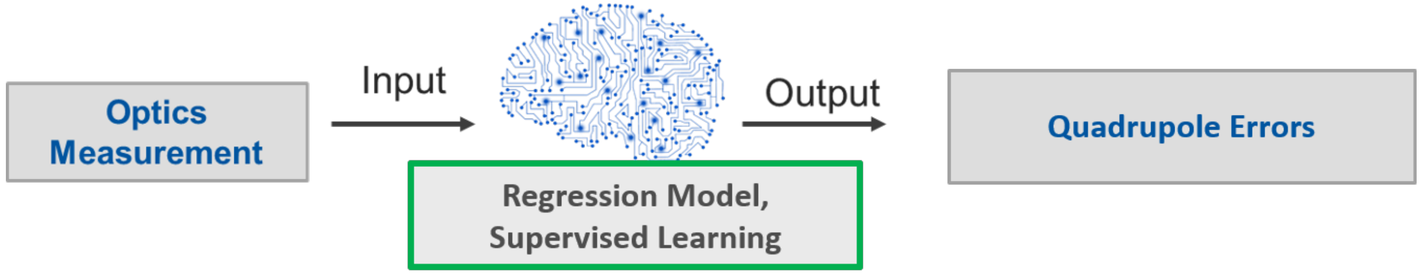


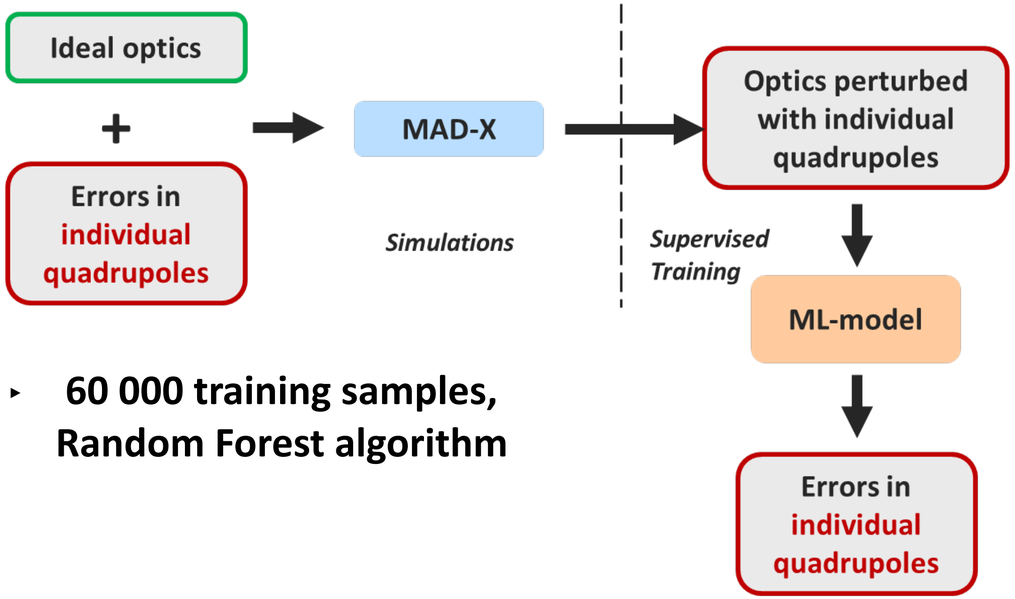
- Corrections are implemented by changing the strength of **circuits**
- Optics perturbations are caused by all **individual magnets.**

➢ What are the **actual errors of individual quadrupoles?**

➢ How to obtain the **full set of errors in one step**?

Estimation of quadrupole errors





- **60 000 training samples, Random Forest algorithm**

Local optics corrections in LHC optics commissioning 2022:

- **~2200 input** variables: phase advance deviations from nominal optics
- **32 output targets**: triplet quadrupole errors in Interaction Regions

E.Fol et al.,"Supervised learning-based reconstruction of magnet errors in circular accelerators", European Physical Journal Plus volume 136, Article number: 365 (2021) ,

Random Forest Regression

Supervised Learning approach:

➡ generalized model **explaining relationship between input and output** variables in **all training samples**.

➡ Capturing the “physics”: correlation between cooling performance and parameters of the cooling channel.

Decision Trees:

- Partition data based on a sequence of thresholds
- Continuous target y, in region estimate:

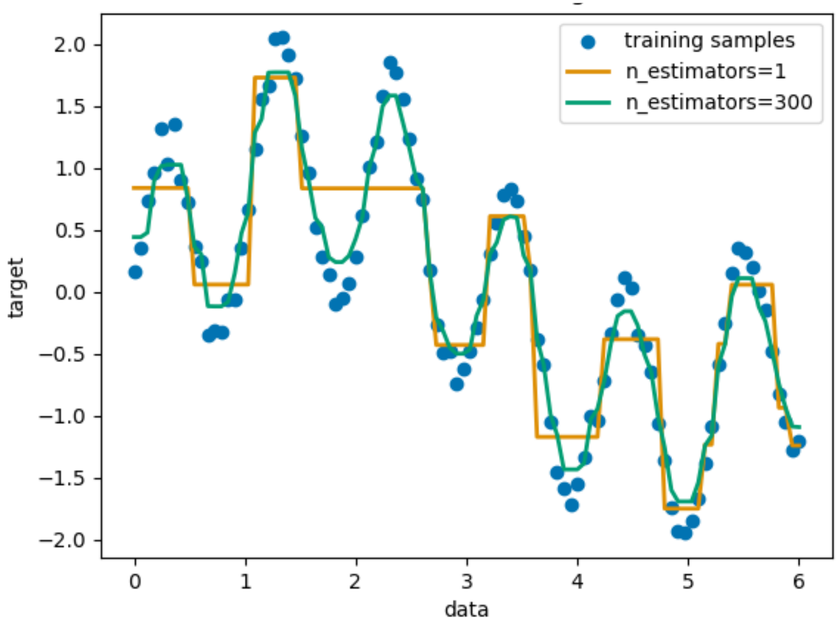
$$c_m = \frac{1}{N_m} \sum_{i \in N_m} y_i$$

- Mean Square Error:

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - c_m)^2$$

Random Forest:

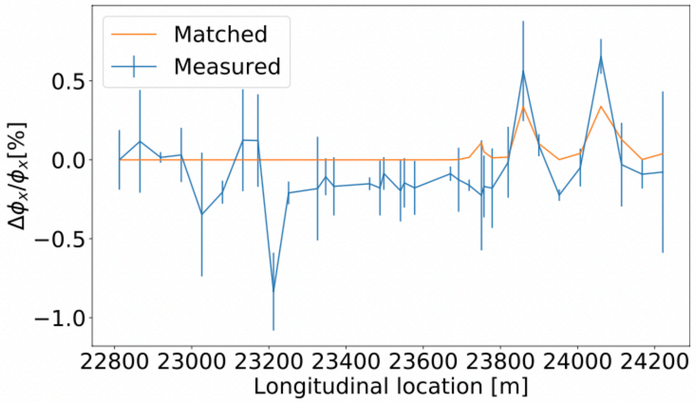
- Random subset of examples, train separate model on each subset
- Only random subset of features is used at each split
- Increases variance, tend not to overfit

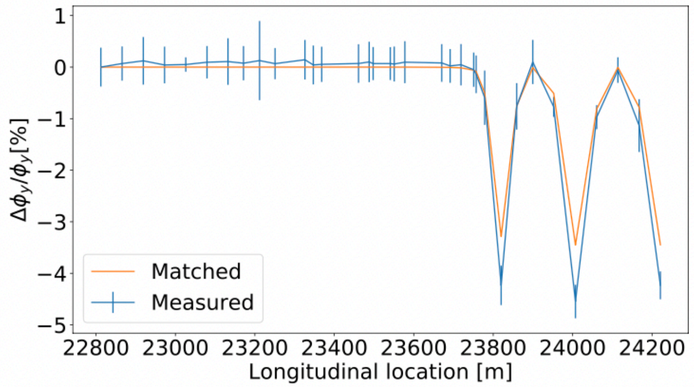


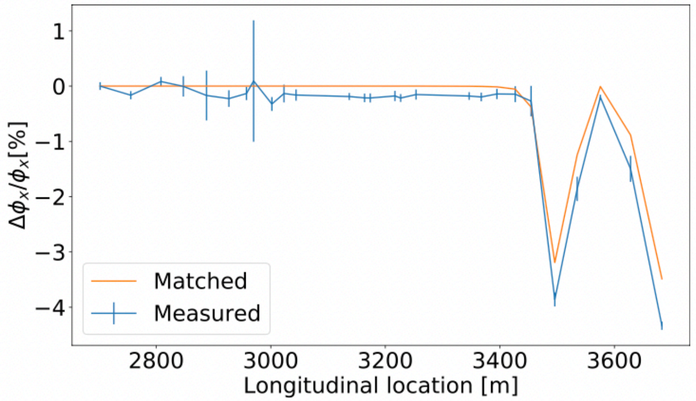
LHC commissioning 2022: predicting local quadrupole errors

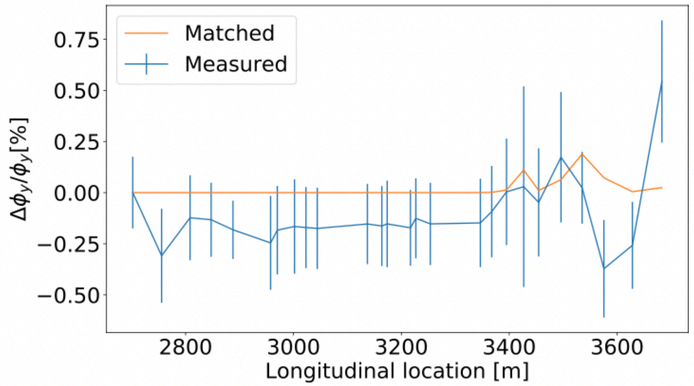
Example: Corrections in Interaction Region 1, squeeze to β^* = 30 cm

➡ Correction test propagating the predicted errors within a selected local region:







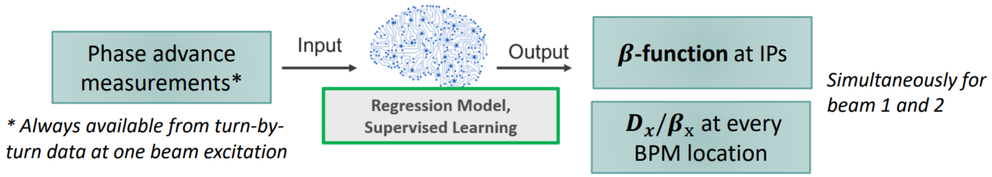


✓ Phase errors can be corrected applying the errors with opposite sign as correction settings

✓ **Simultaneous local correction in all IRs within seconds.**

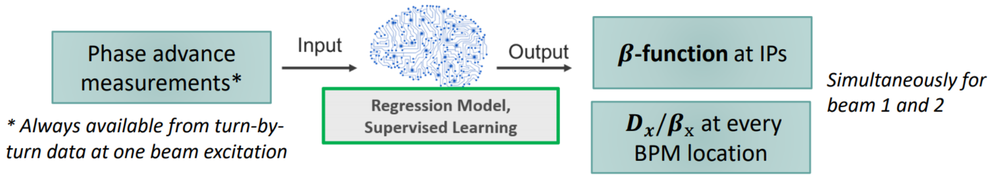
Virtual Optics Measurements

Predict advanced optics observables from phase advances:



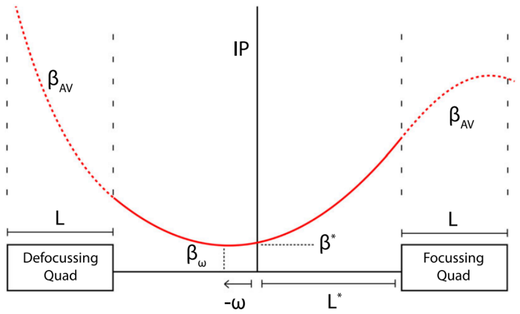
Virtual Optics Measurements

Predict advanced optics observables from phase advances:



Measuring beta-function in Interaction Regions:

Traditional technique: k-modulation:



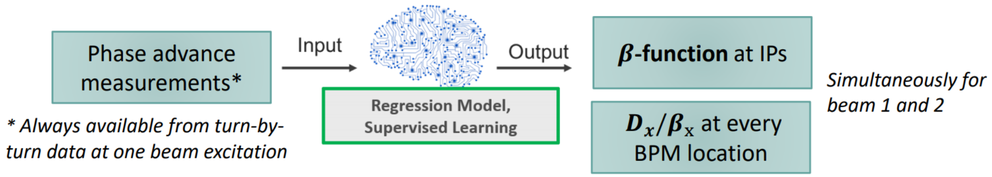
- Based on modulation of quadrupole current
- Time consuming
- Accuracy varies depending on tune measurement uncertainty, magnet errors and β^* settings.

✓ β-functions left and right from IPs **within a few seconds vs. several minutes for k-modulation**

✓ Average accuracy: **5 % for β^* = 30 cm**.

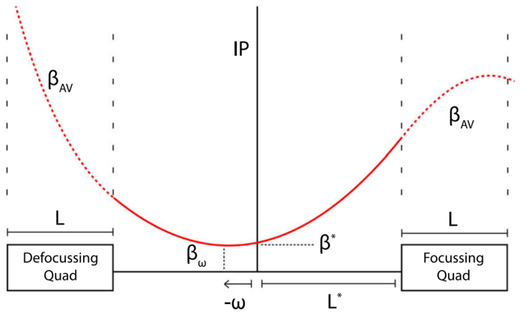
Virtual Optics Measurements

Predict advanced optics observables from phase advances:



Measuring beta-function in Interaction Regions:

Traditional technique: k-modulation:



- Based on modulation of quadrupole current
- Time consuming
- Accuracy varies depending on tune measurement uncertainty, magnet errors and β^* settings.

✓ β-functions left and right from IPs **within a few seconds vs. several minutes for k-modulation**

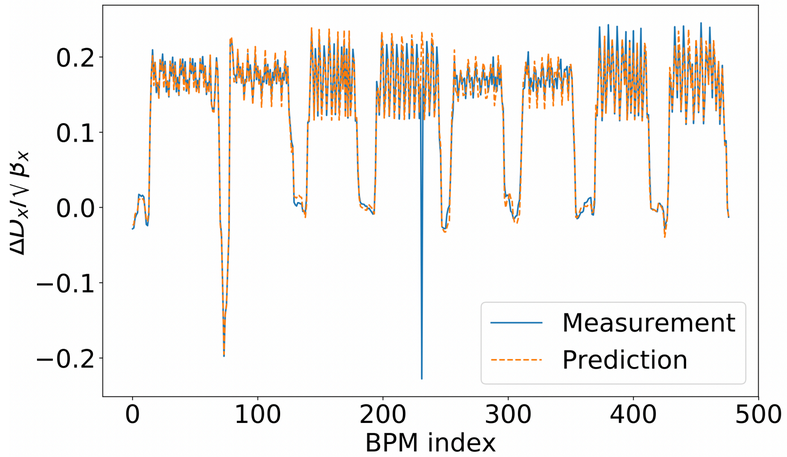
✓ Average accuracy: **5 % for β^* = 30 cm**.

Horizontal Dispersion reconstruction:

- Computed by acquiring turn-by- turn data from **several beam excitations, shifting the momentum.**

✓ Simultaneous reconstruction of normalised dispersion in beam 1 and beam 2 requires only a few seconds.

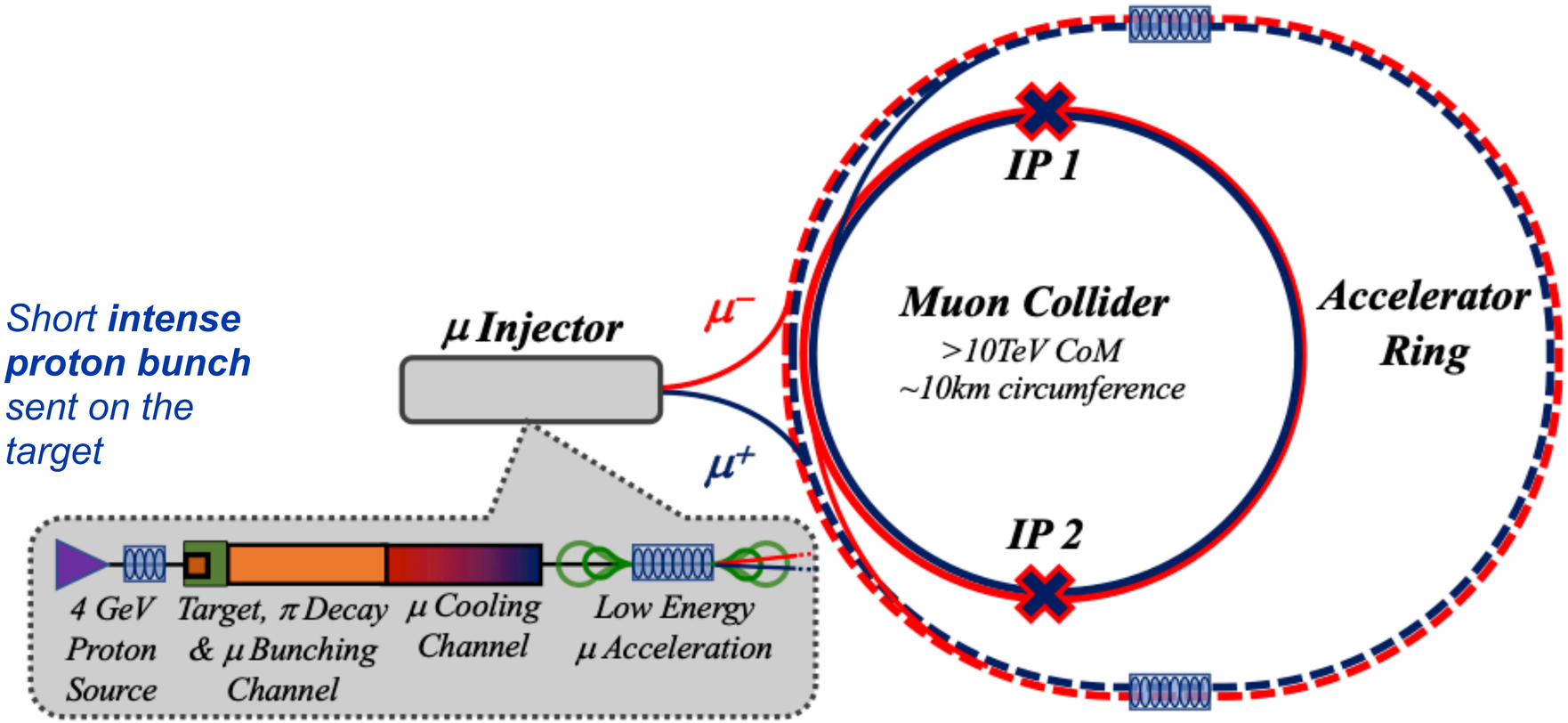
✓ **The relative error of prediction is 5% (beam 1) and 7% (beam 2).**



Measurement taken during LHC commissioning, β^* = 30 cm

ML applied to the design of future accelerators: Final Cooling for the Muon Collider

Muon Collider: overview



*Short **intense proton bunch** sent on the target*

Interaction with the target produces pions

➡ ***decay into muons***

*Muons are captured and **cooled to lower emittance***

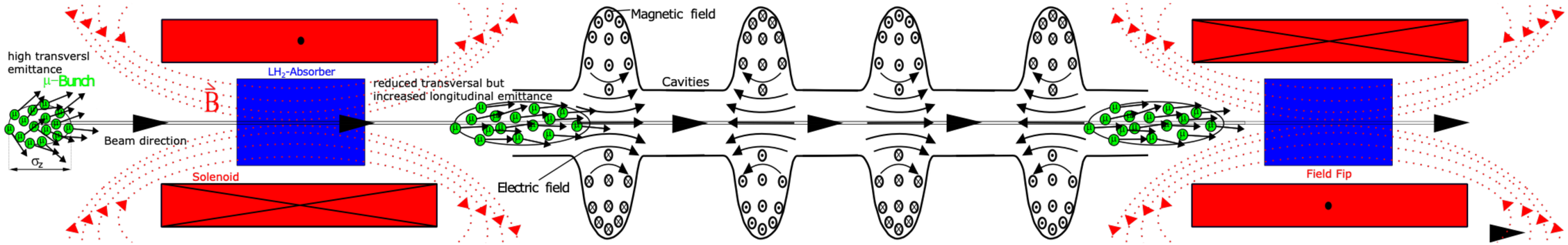
*Acceleration to **high collision energy***

[1]: https://muoncollider.web.cern.ch

Technology and challenges of Final Cooling

Ionisation cooling: the only technique that works on the **timescale of the muon lifetime**

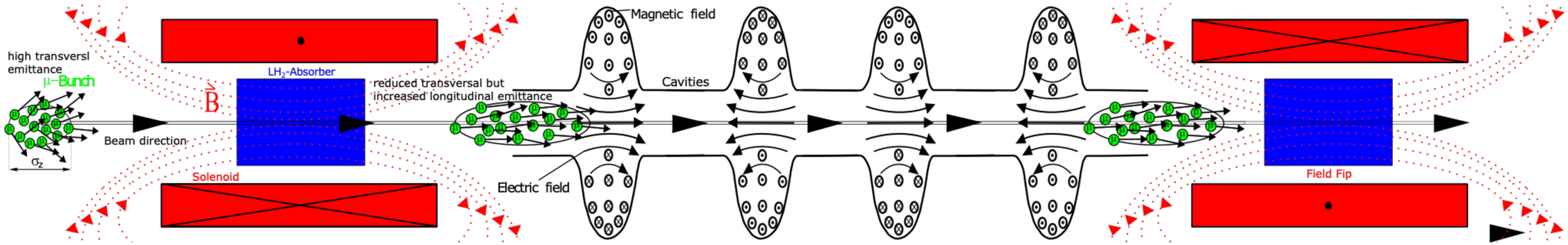
- Muons passing through a material —> energy loss due to the interaction with absorber material
- Reduction of normalised beam emittance
- Re-accelerating the beam to restore the longitudinal momentum



Technology and challenges of Final Cooling

Ionisation cooling: the only technique that works on the **timescale of the muon lifetime**

- Muons passing through a material —> energy loss due to the interaction with absorber material
- Reduction of normalised beam emittance
- Re-accelerating the beam to restore the longitudinal momentum



Optimization routines executing tracking of thousands of muons at each step

→ How to **speed up** optimization?

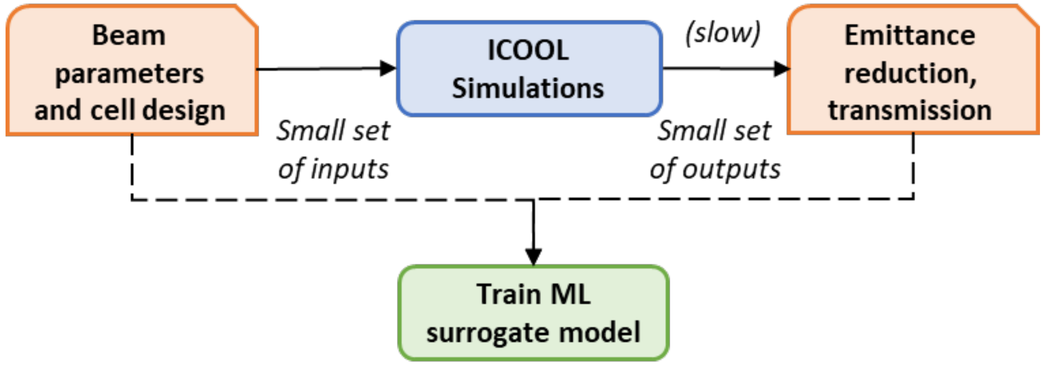
→ How to **find starting parameters**?

✓ Application of **Supervised Learning to model cooling performance** from saved simulation data

✓ Inverse models for the **initial parameters estimation**

Potential speed-up of simulation studies

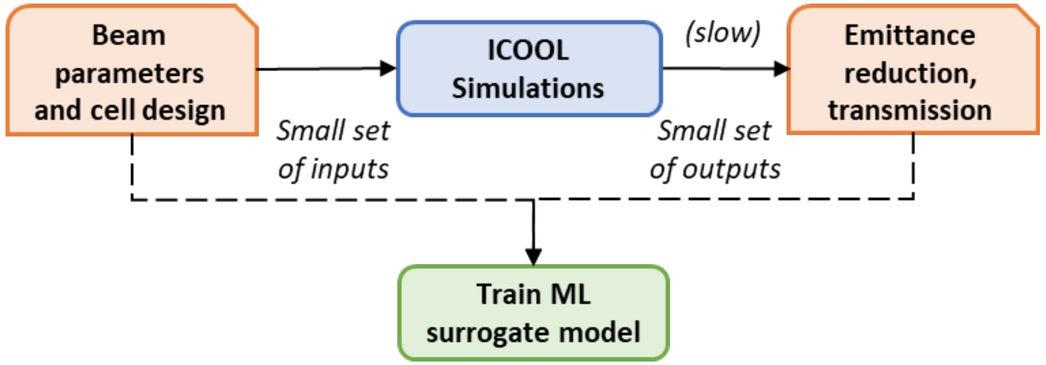
- Systematically storing produced simulation data:
  allows automatic **mapping between initial conditions and simulation results**
- "**Surrogate modelling**": models based on existing data (*Supervised Learning*)



- E. Fol „Evaluation of Machine Learning Methods for LHC Optics Measurements and Corrections Software“, CERN-THESIS-2017-336, 2017
- A. Edelen et al. „Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems“ , Phys. Rev. Accel. Beams 23, 044601, 2020

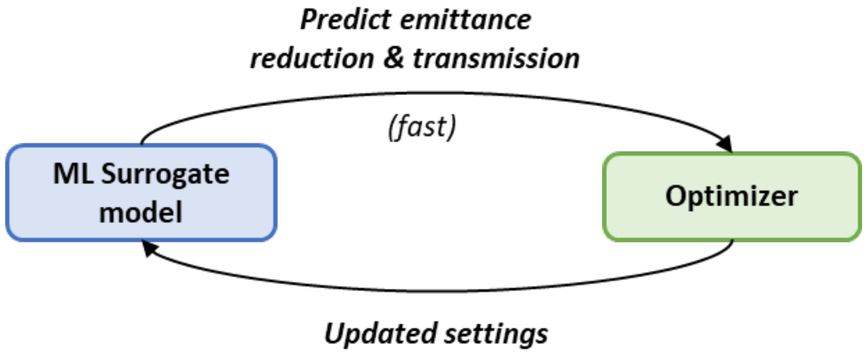
Potential speed-up of simulation studies

- Systematically storing produced simulation data:
  allows automatic **mapping between initial conditions and simulation results**
- "**Surrogate modelling**": models based on existing data (*Supervised Learning*)

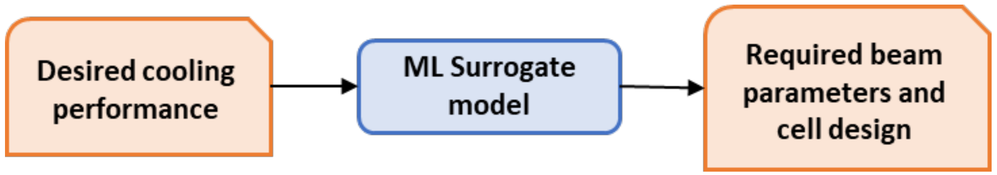


- E. Fol „Evaluation of Machine Learning Methods for LHC Optics Measurements and Corrections Software“, CERN-THESIS-2017-336, 2017
- A. Edelen et al. „Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems“ , Phys. Rev. Accel. Beams 23, 044601, 2020

1. Speeding up optimization:

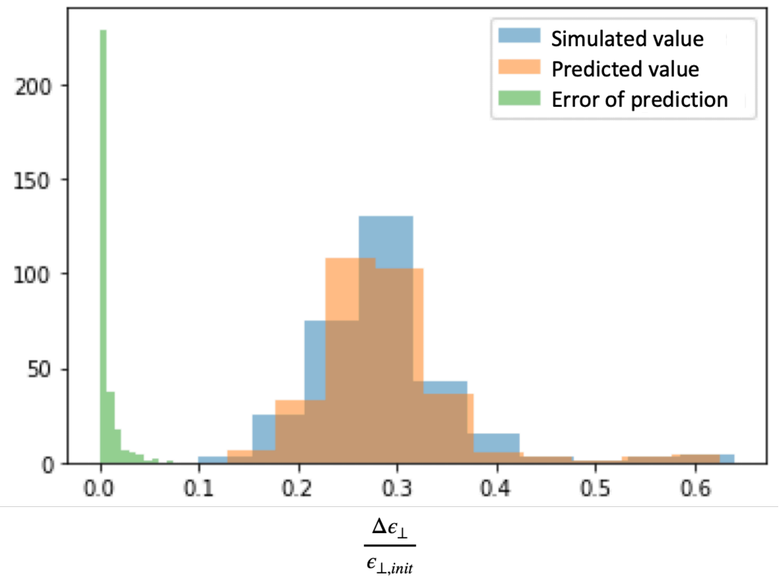


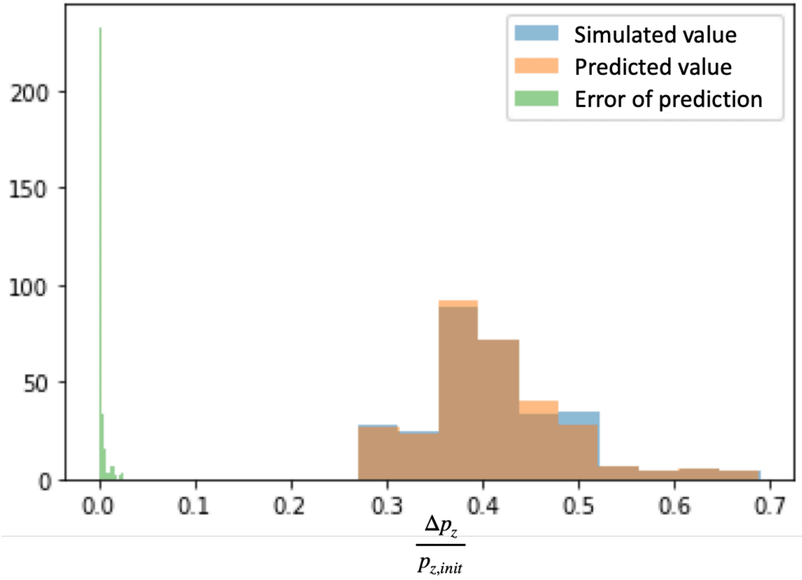
2. "Inverse" design:

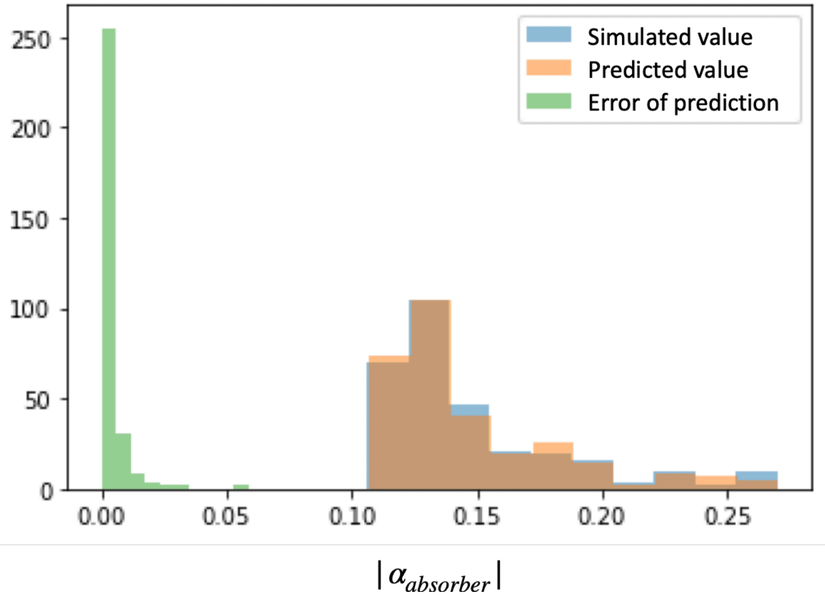


Prediction of simulation output

- Random Forest regressor, 1200 simulations
- **98.3%** accuracy on a test set (300 simulations)







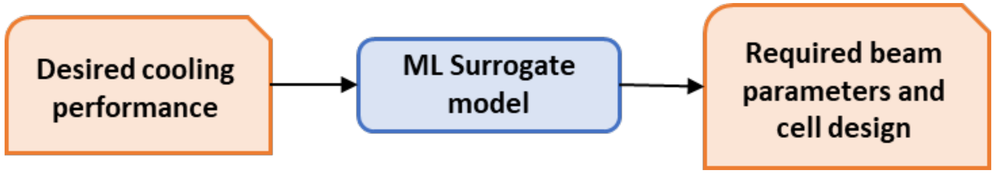
- Evaluate objective function during optimization:

$$\min_p \frac{\Delta\epsilon_\parallel}{(\Delta\epsilon_\perp \Delta N)} + \bar{\alpha}$$

- ✓ **Compute optimization function from ML-model prediction**
- ✓ **Optimization in a few minutes instead of ~1.5 hours for 200 steps using simulations**

Inverted models

- **Estimate the initial parameters to achieve a desired cooling performance**



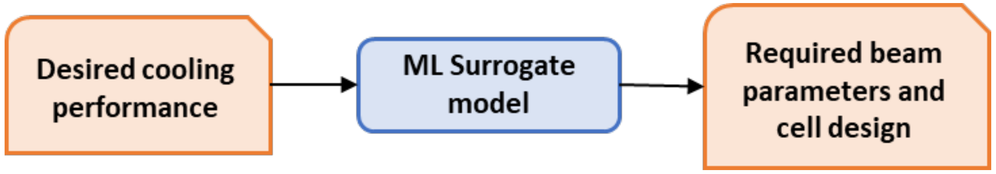
Input: Emittance reduction, momentum reduction, transmission
Output: required start energy, beta, absorber densities in **2 consecutive cells**

Example: aiming for $\Delta\epsilon$=50%, Δpz = 60%, ΔN=90%

1. **Predict design parameters**: Ekin = 0.0714GeV, beta = 0.846, absorber densities = 1.3, 1.1
2. **Tracking using configuration predicted by ML model:**

Inverted models

- **Estimate the initial parameters to achieve a desired cooling performance**



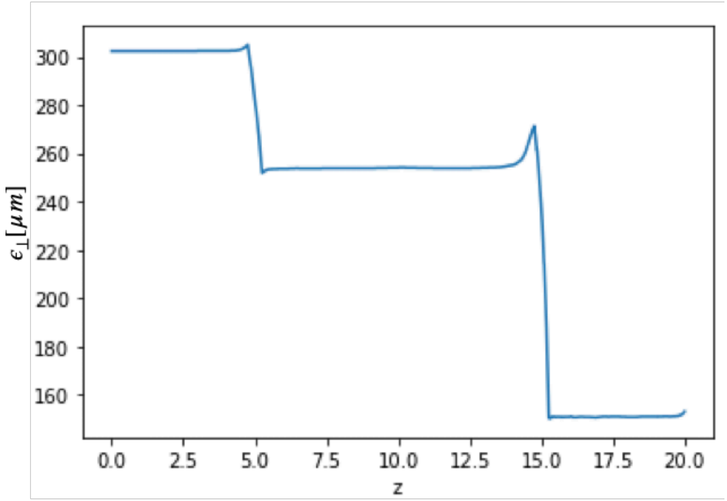
Input: Emittance reduction, momentum reduction, transmission
Output: required start energy, beta, absorber densities in **2 consecutive cells**

Example: aiming for Δϵ=50%, Δpz = 60%, ΔN=90%

1. **Predict design parameters**: Ekin = 0.0714GeV, beta = 0.846, absorber densities = 1.3, 1.1
2. **Tracking using configuration predicted by ML model:**

Δϵ=0.49%, Δpz = 0.61%, ΔN=0.98%

➡ ***Warm start for optimization or final solution?***



Conclusions

How can we profit from ML?

Accelerators

- Operation
- Diagnostics
- Beam Dynamics Modeling

Which limitations can be solved by ML with reasonable effort?

Machine Learning:
✓ *Learn arbitrary models*
✓ *Directly from provided data*

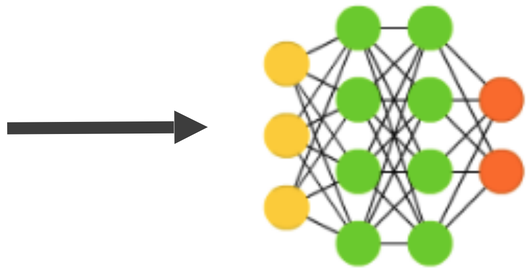
➢ large amount of optimization targets
➢ computationally expensive simulations
➢ direct measurements are not possible
➢ previously unobserved behaviour
➢ non-linear interacting sub-systems, rapidly changing environment.

ML in accelerators: summary

Accelerator Problem	ML methods	Benefits	To be considered
• Automation of particular components	Supervised techniques for classification: Decision Trees, SVR, Logistic Regression, NN	Saving operation time, reducing human intervention, preventing subjective decisions	Dedicated machine time usually required to collect training data and to fine tune developed methods.
• Online optimization of several targets which are coupled • Unexpected drifts, continuous settings readjustment needed to maintain beam quality	Reinforcement Learning, Bayesian optimization, Gaussian Process, Adaptive Feedback	Simultaneous optimization targeting several beam properties, automatically finding trade-off between optimization targets, allows faster tuning offering more user time.	Ensuring that all important properties are included as optimization targets.
• Detection of anomalies	Unsupervised methods: clustering, ensembles of decision trees (e.g. Isolation Forest), supervised classification, Recurrent NN for time-series data.	Preventing faults before they appear, no need to define rules/ thresholds, no training is needed and can be directly applied on received data	In unsupervised methods, usually no “ground truth” is available → methods can be verified on simulations.

ML in accelerators: summary

Accelerator Problem	ML methods	Benefits	To be considered
• Computationally heavy, slow simulations • Reconstruct unknown properties from measurements	Supervised Regression models, NN for non-linear problems	Learning underlying physics directly from the data, faster execution	100% realistic simulations are not possible → the model performance will be as good as your data is.
• Reduction of parameter space e.g. for optimization	Clustering, Feature Importance Analysis using Decision trees	Speed up of available methods, simpler defined problems, easier to interpret	Parameter selection and combination (feature engineering) can have significant impact on ML methods performance
• Missing or too noisy data	Autoencoder NN	Robust models, data quality	Significant information should not be removed from the signal.



Cat!

Thank you for your attention!

Back-up slides

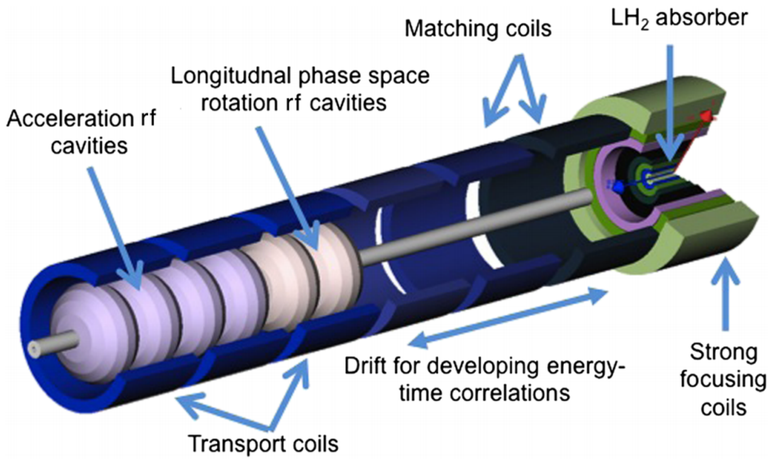
Further References

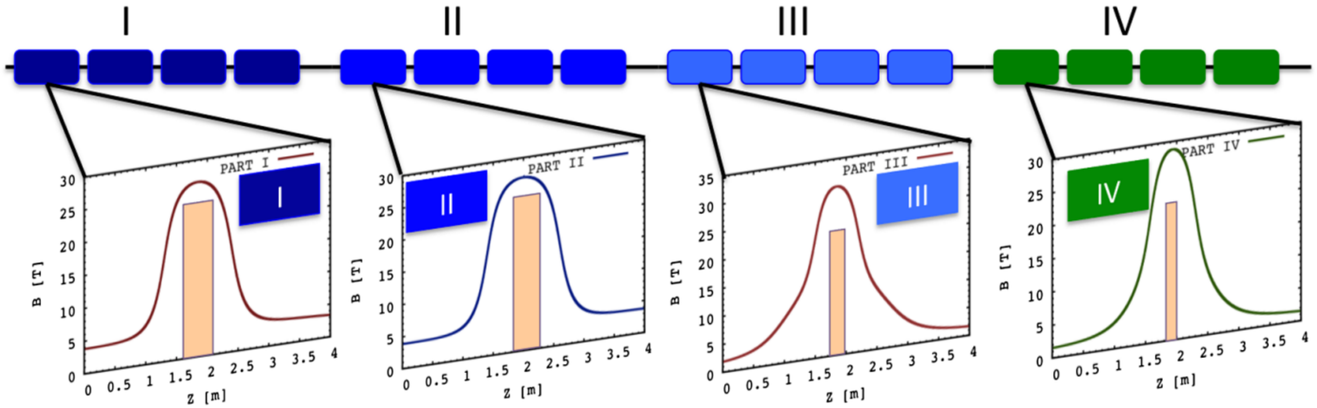
- **Machine learning for beam dynamics studies at the CERN Large Hadron Collider**
  https://doi.org/10.1016/j.nima.2020.164652
- **Opportunities in Machine Learning for Particle Accelerators**
  https://arxiv.org/abs/1811.03172
- **Optimization and Machine Learning for Accelerators (USPAS course)**
  https://slaclab.github.io/USPAS_ML/

Final Cooling baseline

- A Gaussian input beam with **$\varepsilon_{\perp}$=300 µm and $\varepsilon_{\parallel}$ = 1.5mm**
- For final cooling, the beam momentum is reduced initially to **135 MeV/c**

- High-field magnets limited to 25—32 T, and the cooling beam momenta ranged from 135 MeV/c to 70 MeV/c (40 to 20 MeV kinetic energy)
- Cooled to $\varepsilon_{\perp}$ = 55 µm, with $\varepsilon_{\parallel}$ = 70 mm and transmission of 50%
- Preferred $\varepsilon_{\perp}$ = 25µm, should be possible to achieve with stronger focusing fields, alternative absorber configuration and further optimization.





High field – low energy muon ionization cooling channel
Hisham Kamal Sayed, Robert B. Palmer, and David Neuffer
Phys. Rev. ST Accel. Beams **18**, 091001 – Published 4 September 2015

Practical advice

- **Feature engineering** is highly important! Rescaling, denoising, outlier elimination...
  ➢ data vizualisation can help
- Start with **simple models** (increase the model complexity (e.g. applying Neural Networks) only if really needed.
- Well **structured** data, **extendable architecture** of existing frameworks
  → possibility for the integration of ML tools.
- Estimate model generalization (split into training, test and validation sets)

Frameworks to use:

- Prototyping, fast and easy implementation (very good documentation): http://scikit-learn.org/
- High-level package for Neural Networks: – https://keras.io/
- Deep Learning, specific complex model architectures: https://www.tensorflow.org/ http://deeplearning.net/software/theano/
- Reinforcement Learning: OpenAI Gym https://gym.openai.com/