LHC beam intensity lifetime optimization: ramp losses and machine learning applications

Speaker: Loïc Coyle

Master Thesis exchange program Grenoble INP Phelma-EPFL-LPAP

Supervisors: Belen SALVACHUA (CERN-LHC Operation), Tatiana PIELONI, Leonid RIVKIN (EPFL-LPAP)
Motivation

Characterize systematic losses occurring during the ramp

Help to shed light on the impact of parameters on the proton losses

Help optimize and determine operational setups

Goal:

First trial of applying machine learning models to the LHC optimization

Try to predict from initial setting the beam lifetimes over a physical cycle and possibly feedback online corrections to optimize it.

For use in collider optimization → maximising integrated luminosity reach and reducing proton losses

Improve the understanding of proton losses to feed to numerical models for future projects studies
Ramp losses
Signals from BLMs closest to primary collimators at IRs 7 and 3.

The **vertical dashed** lines are the **optic matching** times.

Consistent losses **on or between** these times.
Ramp losses 2018 - BLM

Signals from BLMs closest to primary collimators at IRs 7 and 3.

The **vertical dashed** lines are the **optic matching** times. Consistent losses on or between these times.
Ramp losses 2018 - position at IP7 collimator

Absolute offset from ideal orbit interpolated at IP7

**Horizontal offset:**
Same correlation with optic matching points

No major difference between both beams

No significant drifts

**Vertical offset:**
No correlation with optic matching points
Machine Learning
The Data

Input features:

- **Tune** Horizontal/Vertical/B1/B2
- **Chromaticity via Sextupole magnet circuit currents** B1/B2
- **Beam intensity** B1/B2
- **Beam transverse position** B1/B2
- **Number of bunches**
- **Heat load**
- **Landau Octupole powering** B1/B2
- **Beam losses** B1/B2
- **Transverse Emittance** H/VB1/B2
- **Time elapsed** since start of PRERAMP

Target:

- **Intensity lifetime** B1/B2

Time period:

- 2017-09-04 to 2017-12-31
- 2018-01-01 to 2018-07-31

Beam mode: **PRERAMP**

Input features: **26**

Overview: **175 fills, 45000 samples**

Split: **40% test & 60% training.**
Dataset visualization (2017) - Beam 1
Physical Correlations

As expected as IR7 BLM used for lifetime calculations

Vertical tune doesn’t impact lifetime.
High horizontal tune → Better lifetime
Feature Analysis

Spearman correlation coefficient: Measure of ‘fitting’ with monotonous function.

Lifetimes strongly correlated with:

- Losses
- H tunes of B1 & B2
- Emittance
Choosing the model

<table>
<thead>
<tr>
<th>Model</th>
<th>Beam 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-2</td>
<td>MSE</td>
</tr>
<tr>
<td>Linear</td>
<td>0.666</td>
<td>9482.435</td>
</tr>
<tr>
<td>Ridge</td>
<td>0.664</td>
<td>9521.182</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.655</td>
<td>9773.588</td>
</tr>
<tr>
<td>ElasticNet</td>
<td>0.141</td>
<td>24376.165</td>
</tr>
<tr>
<td>Multi-Layer Perceptron</td>
<td>0.833</td>
<td>4856.423</td>
</tr>
<tr>
<td>RandomForest</td>
<td>0.996</td>
<td>101.971</td>
</tr>
<tr>
<td>GB Decision Trees</td>
<td>0.997</td>
<td>76.577</td>
</tr>
</tbody>
</table>

**Linear models:**
Determine the $\omega$ coefficients which minimise:

$$\min_w \left\| \sum_{i=1}^w X\omega - y \right\|_2^2.$$  

With varying degrees of regularization


**Fully connected neural network**


**Decision Tree** based methods

GBDT from https://github.com/Microsoft/LightGBM
Model performance

Performance of trained model on unseen data.

Orange: model prediction
Blue: True value

Model can **accurately predict the value of lifetime**.

Presence of a few outliers

Same for beam 2

Robustness of model?
Performance of trained model on unseen data.

Orange: model prediction
Blue: True value
Green: 2.5 - 97.5% percentile range

200 models trained on Bootstrap samples of training dataset
Performance of trained model on unseen data.

**Orange** : model prediction

**Blue** : True value

**Green** : 2.5 - 97.5% percentile range

200 models trained on **Bootstrap** samples of training dataset

Beam 2 less robust at lower lifetime values...
Feature importance: number of times the splits in the decision trees occur on each feature

Correlation with tunes from both beams?

Tunes changed together → both tunes redundant?

Additional data with independently changed tunes
Feature Importance

**Feature importance:**
number of times the splits in the decision trees occur on each feature

Correlation with tunes from both beams?

**Tunes changed together → both tunes redundant?**

Additional data with independently changed tunes
Bare data set

Second Input Feature set: **bare bones**

Extract more subtle trends

Only features we can control:

- Tunes H/V/B1/B2
- Chromaticity H/V/B1/B2
- Octupole current B1/B2
- Time elapsed since start of PRERAMP

Same time span & number of fills.

**Total: 15 features**
Model Performance

Weaker model

Still predicts with decent accuracy

Seems to have learned the trends between the limited input features and the lifetimes
The main trends are consistent between the 2 beam models, interestingly the **time elapsed** is of greatest importance, followed by **tunes**.

**Time elapsed** importance still **unexplained**…

Model is substituting a missing time dependent parameter with time elapsed ?
Importance of the Time elapsed

Time elapsed forcefully removed
Model performs significantly worse…
Cannot distinguish between data points especially Beam 2
Proof of concept: using ML model for optimization

Very erratic evolution of parameters, Expected as model has no concept of smoothness.

Large lifetime gain early on tapers off during the fill.

Model recommends higher horizontal tunes, agrees with the observed correlation.
The strong correlations between H tunes and lifetimes disappear when using the measured BBQ tunes as opposed to the feedback tunes.
The BOFSU (feedback) signal is acquired by applying a filtering process to the raw BBQ signal.

Thought to be caused by the a systematic under evaluation of the tune by the BBQ, due to the signal jumping to unwanted peaks.

Preprocessing fundamental
The BOFSU (feedback) signal is acquired by applying a filtering process to the raw BBQ signal.

Thought to be caused by the a systematic under evaluation of the tune by the BBQ, due to the signal jumping to unwanted peaks.

**Preprocessing fundamental**
The BOFSU (feedback) signal is acquired by applying a filtering process to the raw BBQ signal.

Thought to be caused by the a systematic under evaluation of the tune by the BBQ, due to the signal jumping to unwanted peaks.

Preprocessing fundamental
Bare 2017 on 2018 data
Model does not perform well

Cannot use the trends learnt on 2017 data to successfully predict the lifetimes on 2018 data
Bare 2018

Need to train models every year

Perhaps can be solved with a cross year data set?
Simulations

Using simulation to hopefully explain the high horizontal tune → high lifetime trend

Long term particle tracking with Sixtrack

Extension to machine learning:
- Use simulations to produce additional machine learning dataset.
- Can explore at will input parameter space

Trained model could be used instead of time consuming tracking simulations.
Simulations

Using simulation to hopefully explain the high horizontal tune $\rightarrow$ high lifetime trend

Long term particle tracking with Sixtrack

Extension to machine learning:
- Use simulations to produce additional machine learning dataset.
- Can explore at will input parameter space

Trained model could be used instead of time consuming tracking simulations.
Predicting RAMP from PRERAMP

First crude attempt at predicting ramp lifetimes from PERRAMP data
Models given continuous data from PRERAMP to STABLE
Trained to predict lifetime 1h later…
Perhaps sequence to sequence type model would work better?
Summary and outlook

- A first attempt to apply machine learning techniques to the LHC
- Model successfully learns the expected trends
- Preprocessing and quality of data is crucial
- Interesting results:
  - Difference of tune measurement devices
  - Proof of concept of fill optimization → a newer set-up of parameters to improve lifetimes
  - 2017 model not applicable to 2018 data
  - Unexpected dependency with time elapsed under investigation
- First attempt at predicting RAMP lifetimes from PRERAMP
- Numerical models to support observations have been set-up and will help physicists understand the mechanisms behind the ML results.

Outlook

- Larger dataset, introduce parameters at the bunch level, larger ranges of parameters must be explored (operation & MD ?)
- Improve the diagnostic of the system and the pre-processing of the available data: many quantities need to be introduced
- Go beyond just PRERAMP
- Define a possible “online” use to help operators with operational choices
Ramp losses 2018 - BLM

B1 losses occur at the optic matching points

B2 losses occur in between the optic matching points
Reminder

PRERAMP beam mode:
Dataset visualization (2017) - Beam 2
BBQ vs Feedback - Tune Diagrams

Tune diagrams of LHC data. Time period: 2017-09-04 till end of year. BBQ tunes data has ~8x more points and is much noisier. Presence of some unexpected structures in the BBQ tunes, perhaps due to some noise/inconsistent peak detection/higher order resonances?
Both beams show higher lifetimes for larger horizontal tunes, the vertical tune doesn’t matter as much. This goes hand in hand with the correlation matrices.
[will corroborate with simulation footprints]
Choosing model

<table>
<thead>
<tr>
<th>Model</th>
<th>Beam 1</th>
<th>Beam 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-2</td>
<td>MSE</td>
</tr>
<tr>
<td>Linear</td>
<td>0.666</td>
<td>9482.435</td>
</tr>
<tr>
<td>Ridge</td>
<td>0.664</td>
<td>9521.182</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.655</td>
<td>9773.588</td>
</tr>
<tr>
<td>ElasticNet</td>
<td>0.141</td>
<td>24376.165</td>
</tr>
<tr>
<td>Multi-Layer Perceptron</td>
<td>0.833</td>
<td>4856.423</td>
</tr>
<tr>
<td>RandomForest</td>
<td>0.996</td>
<td>101.971</td>
</tr>
<tr>
<td>GB Decision Trees</td>
<td>0.997</td>
<td>76.577</td>
</tr>
</tbody>
</table>
Gradient Boosted Decision Trees

Which feature to split on determined with information gain of split.

$$E(T, X) = \sum_{c \in X} P(c)E(c) \quad E(T, X) = \sum_{c \in X} P(c)E(c) \quad Gain(T, X) = Entropy(T) - Entropy(T, X)$$