BM@N workshop
27 March 2024

Gradient Boosted Decision Tree for Particle Identification at BM@N

V. Papoyan[1,3]

Coauthors: A. Ayriyan[1,3], K. Gertsenberger [2], H. Grigorian[1,3]

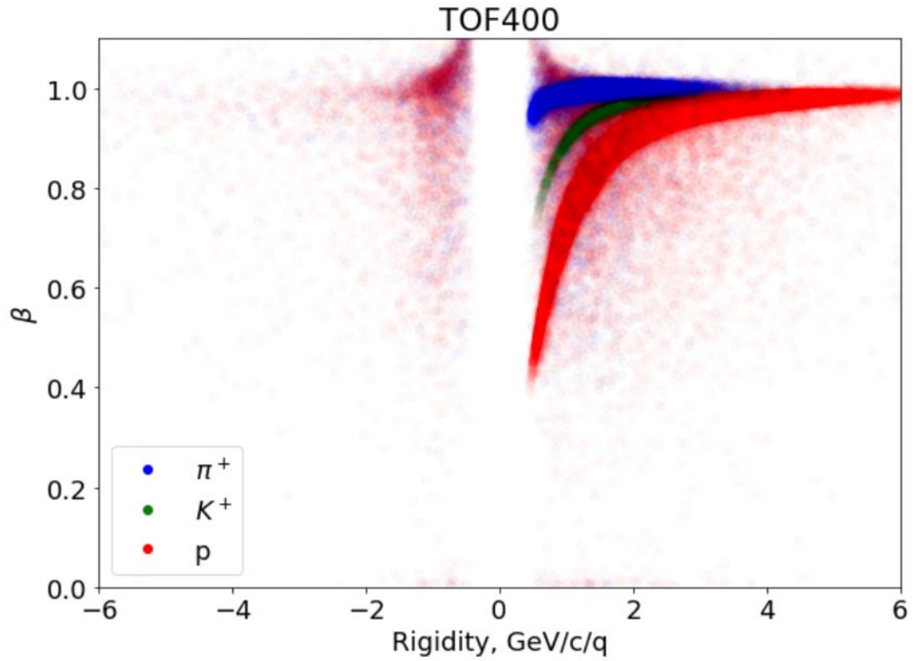
[1]MLIT JINR, [2]VBLHEP JINR, [3]AANL (YerPhi)

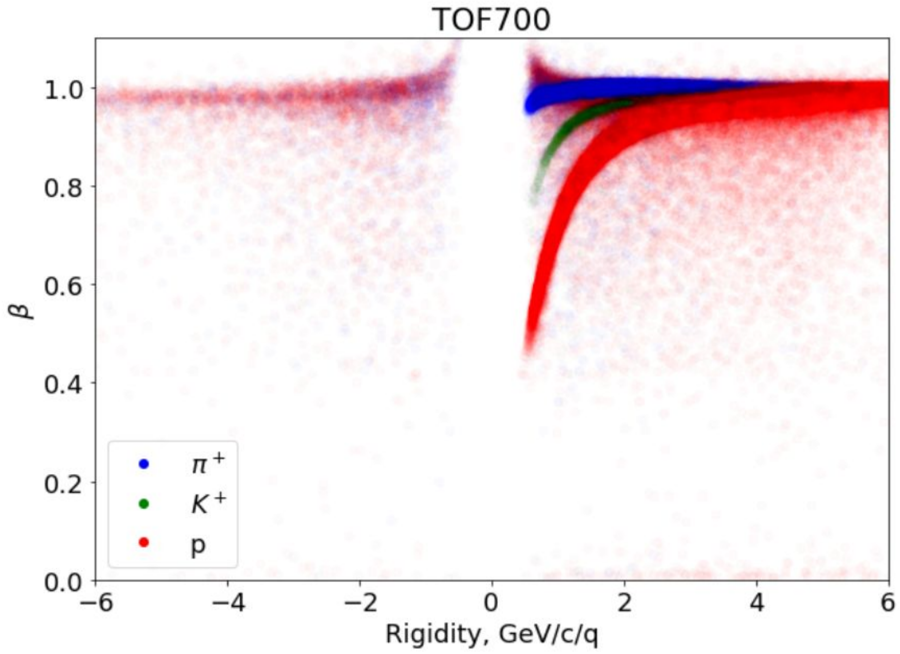
Particle Identification at BM@N experiment

BM@N particle identification (PID) is based on two **Time-of-Flight** (TOF400 and TOF700) chambers

A ToF measures the particle flight **time** (t) over a given **distance** (L) along the track trajectory;

$$\beta = \frac{L}{ct}$$



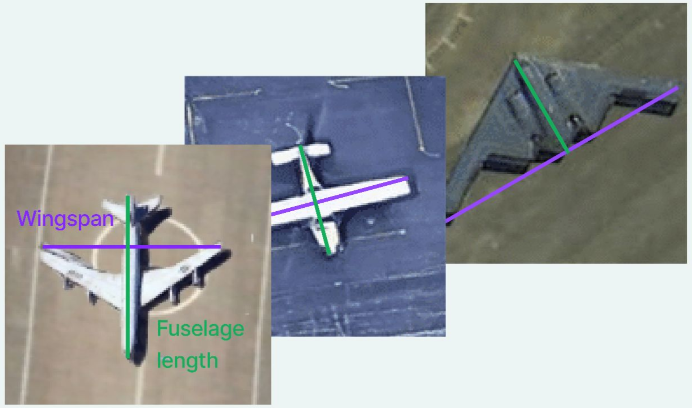


Knowing the particle flight **time** one obtains the **relative velocity** and thus identity of the particle.

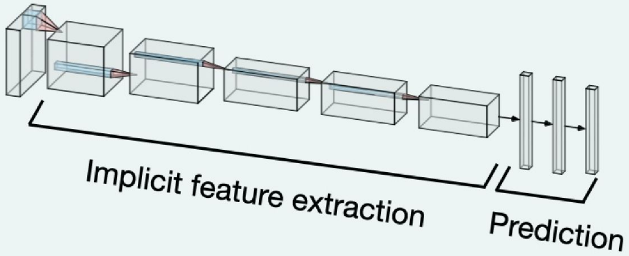
Klempt W. Review of particle identification by time of flight techniques

Tabular Data: Deep Learning vs Gradient Boosting

Unstructured data



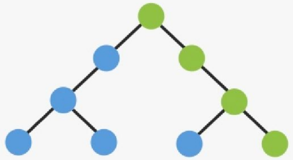
Deep learning



Structured data

	Fuselage length	Wingspan
Boeing 707	44,07	39,9
Cessna 172	8,28	11
B-2 Spirit	20,90	52,12

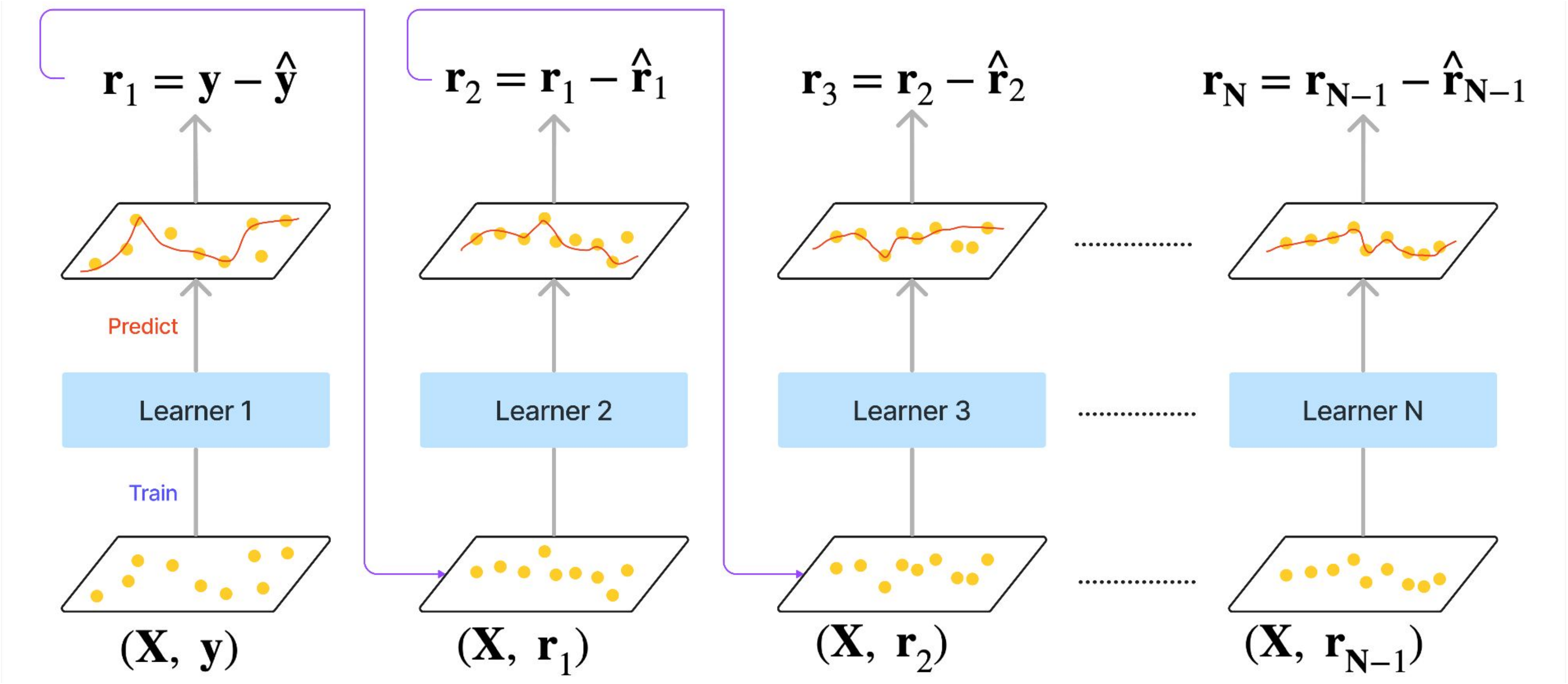
Gradient
Boosting



https://sebastianraschka.com/blog/2022/deep-learning-for-tabular-data.html

Gradient Boosting

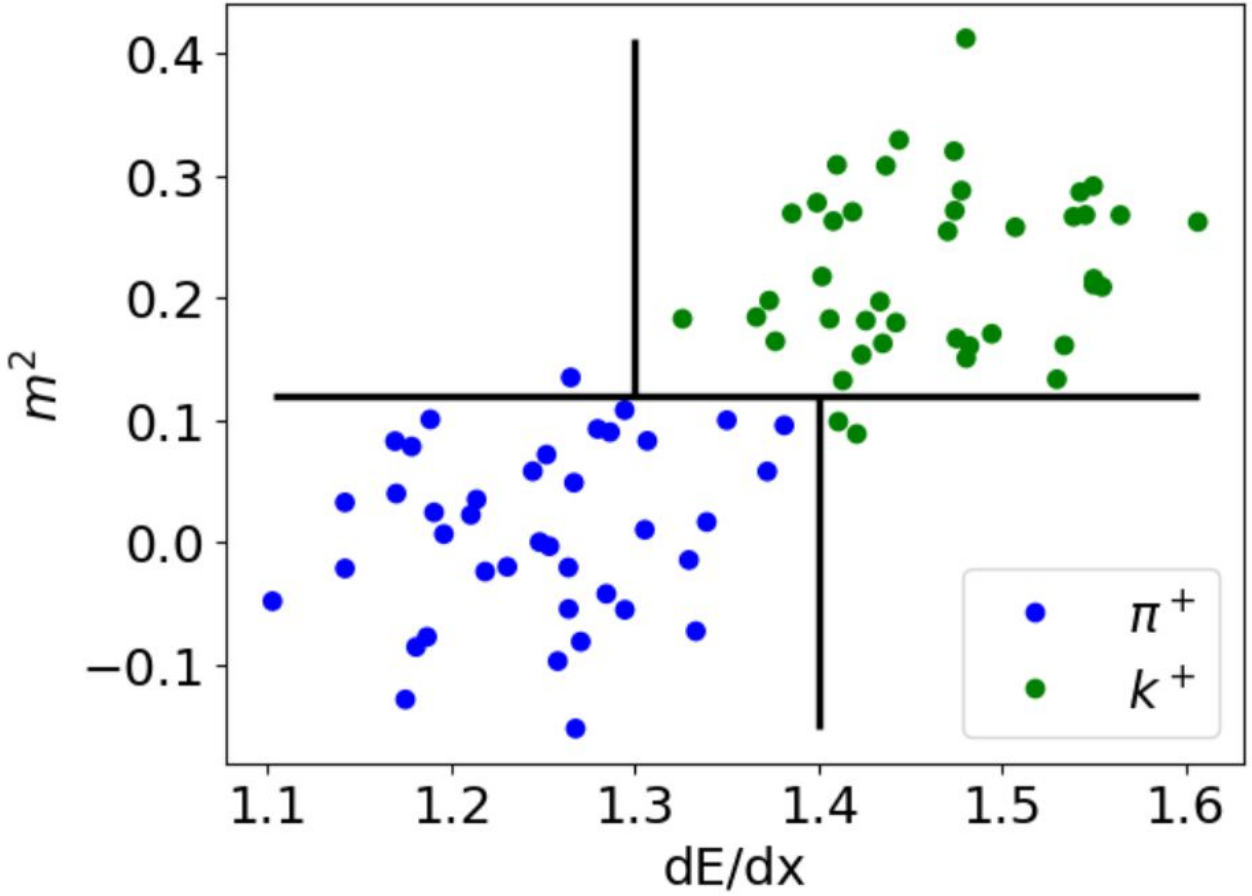
Gradient boosting is a machine learning technique which combines weak learners into a single strong learner in an iterative fashion

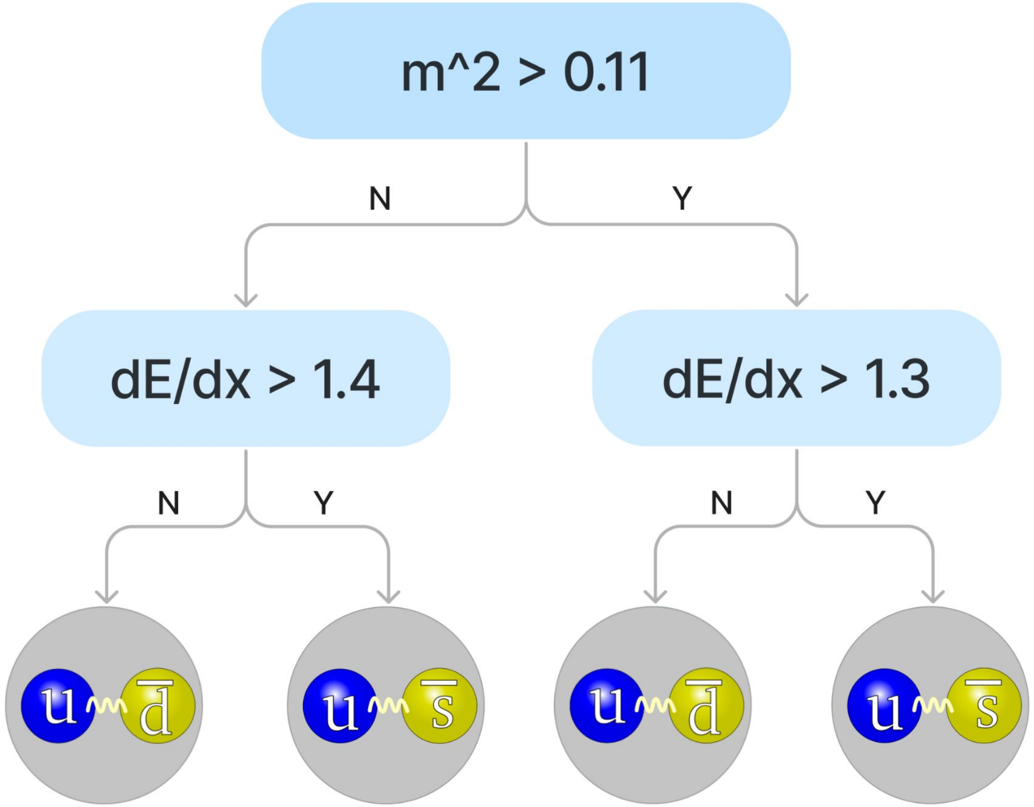


$$F(X) = \sum_{n=1}^{N} \gamma_n l_n(X)$$

Gradient Boosted Decision Tree

Gradient Boosted Decision Tree (GBDT) uses decision trees as weak learner. They can be considered as automated multilevel **cut-based** analysis

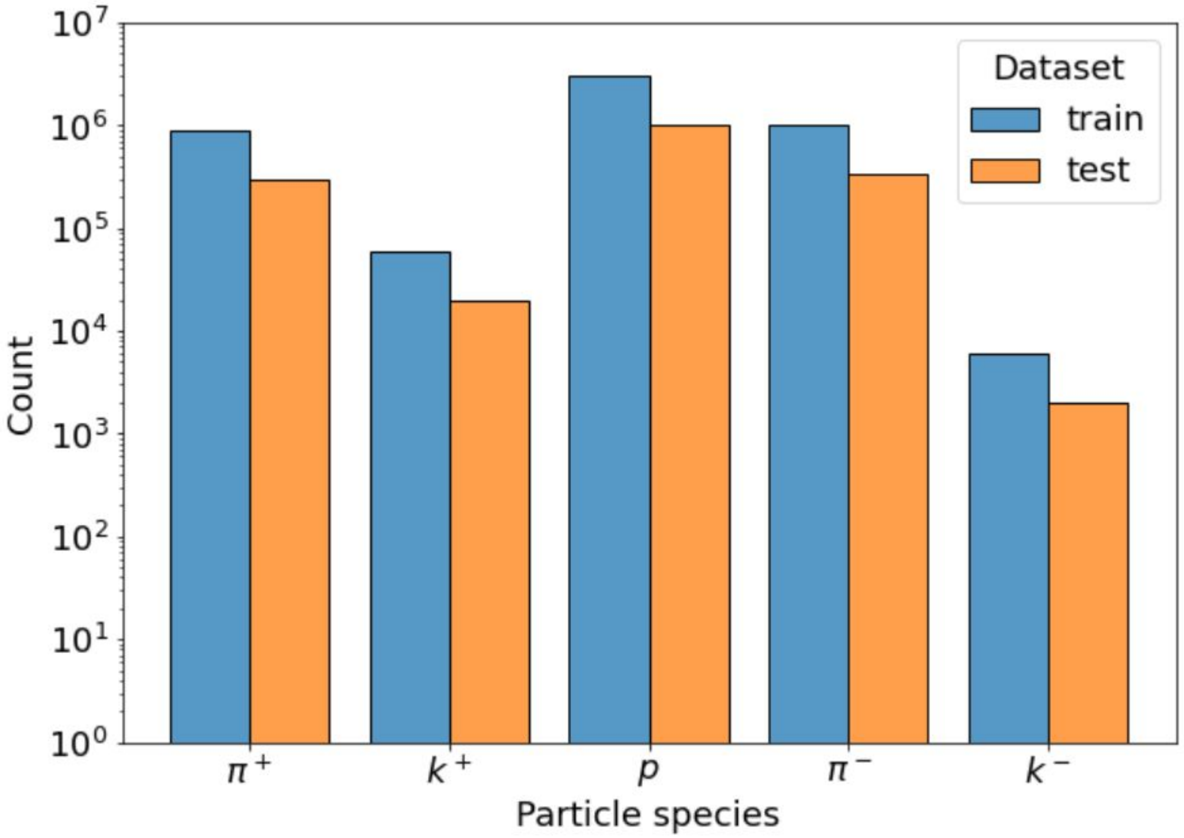




Dataset

Subsample of the Monte-Carlo production has been used

Event generator	DCM-SMM
Colliding system	Xe+CsI
Energy	3.9 A GeV

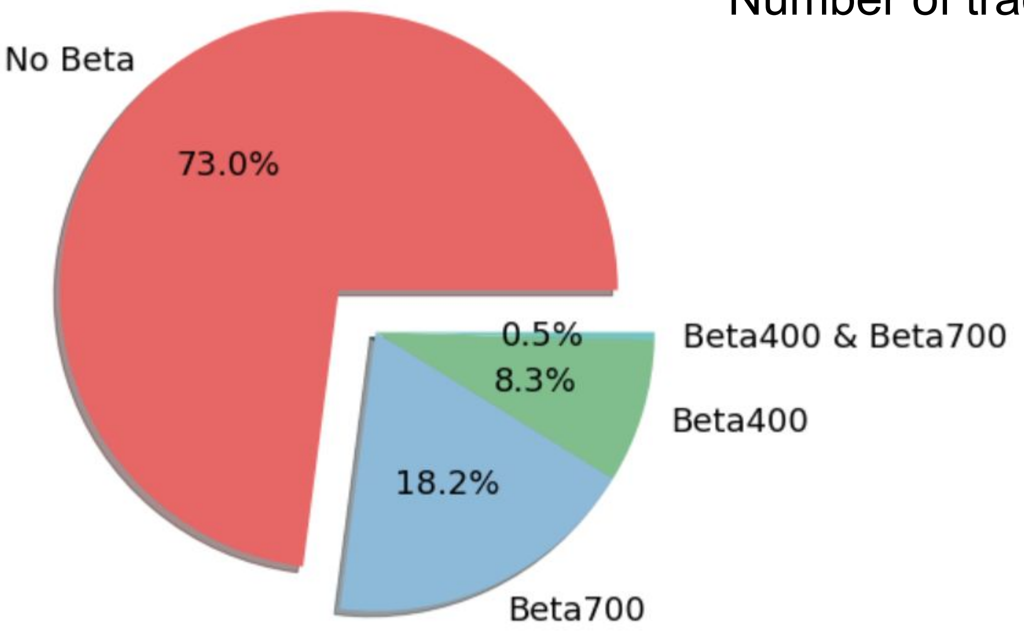


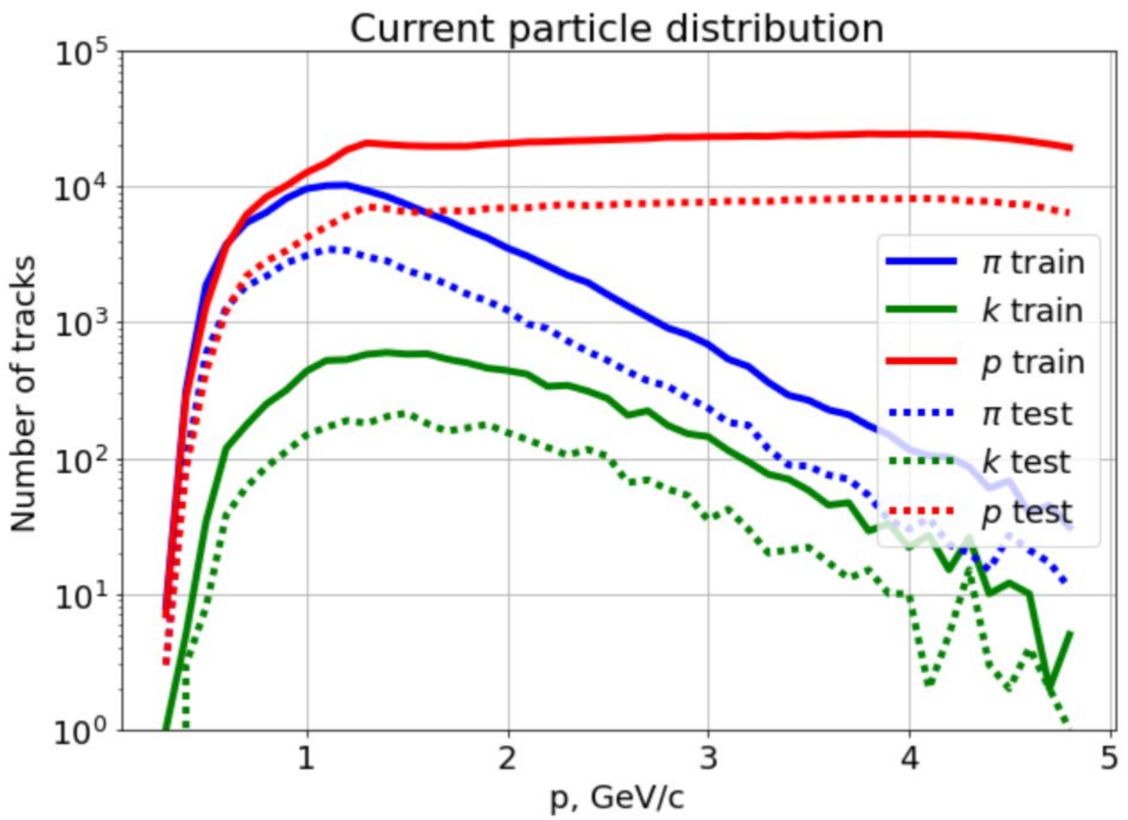
track selection criteria are default within BM@N reconstruction

Feature vectors by Beta

Number of tracks: around **5M**

Number of tracks with at least one ToF: approx. **1.3M** (27%)

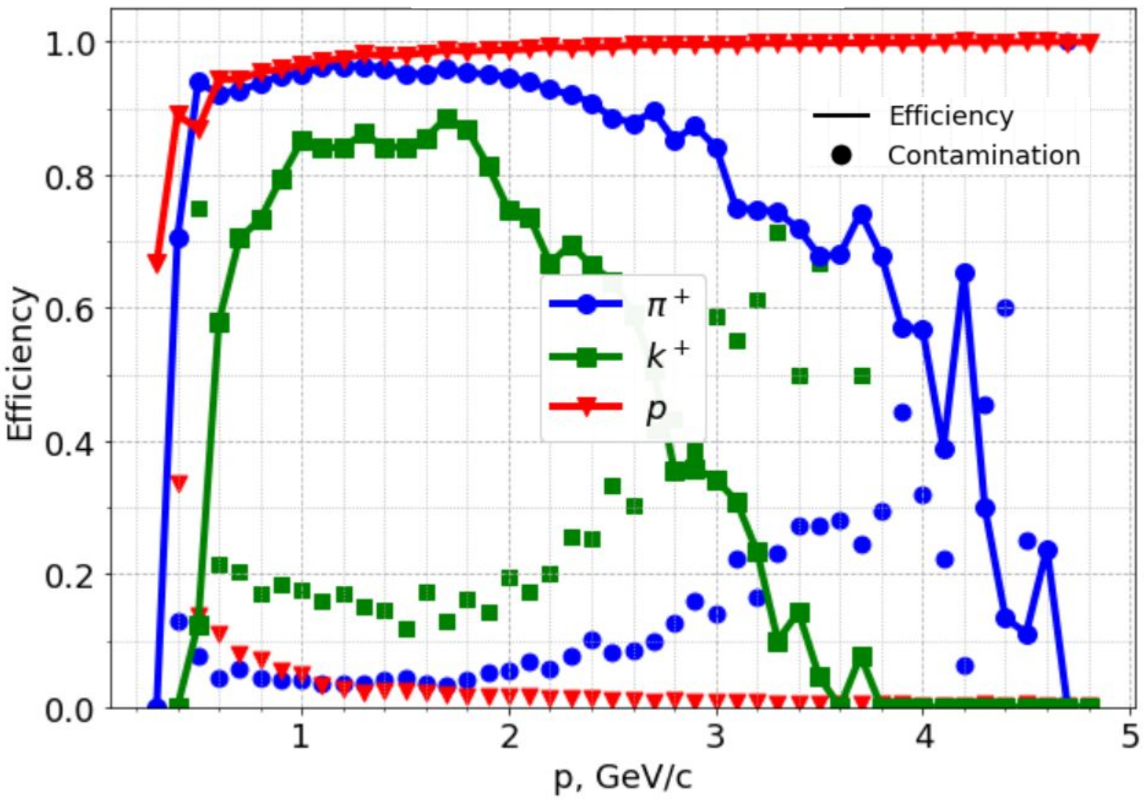


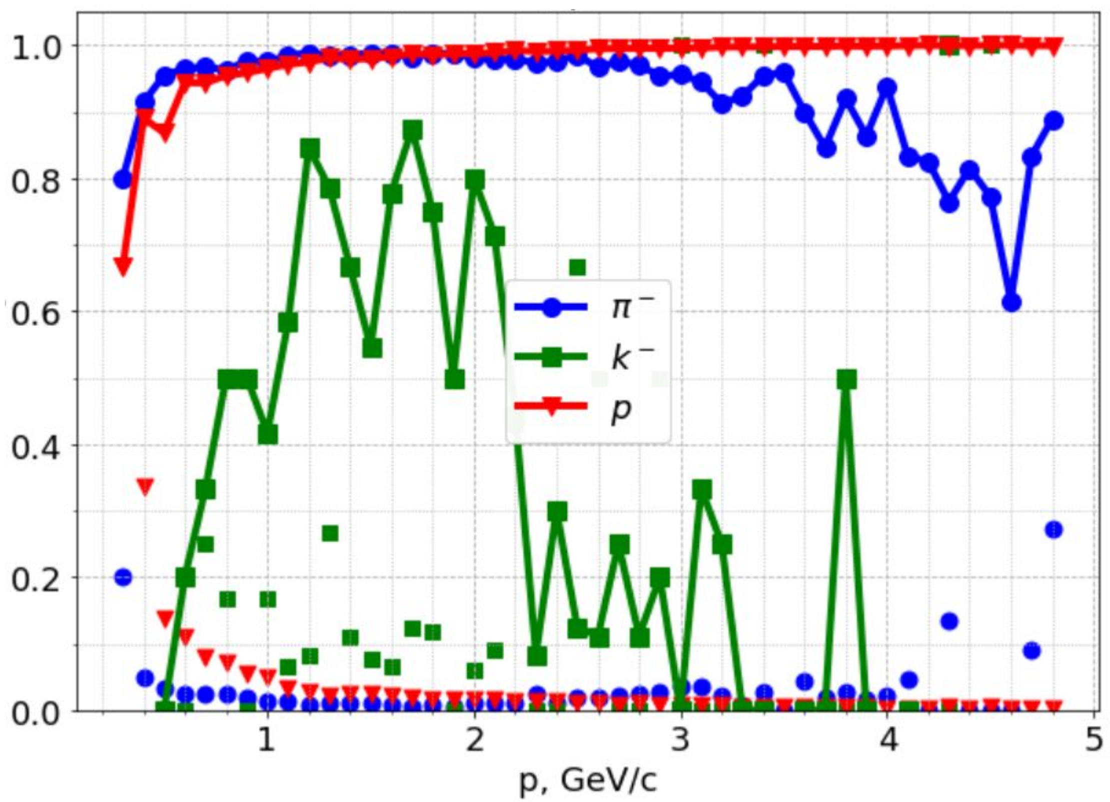


Preliminary results

$$E^s = \frac{N^s{}_{corr}}{N^s{}_{true}}$$

$$C^s = \frac{N^s{}_{incorr}}{N^s{}_{corr}+N^s{}_{incorr}}$$

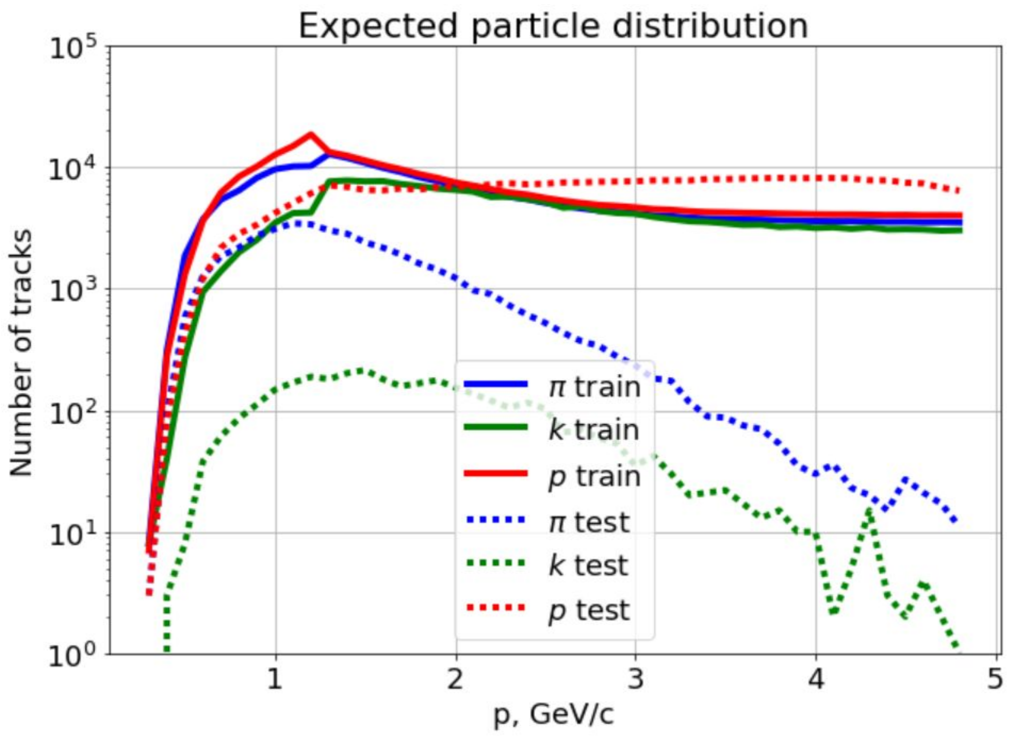


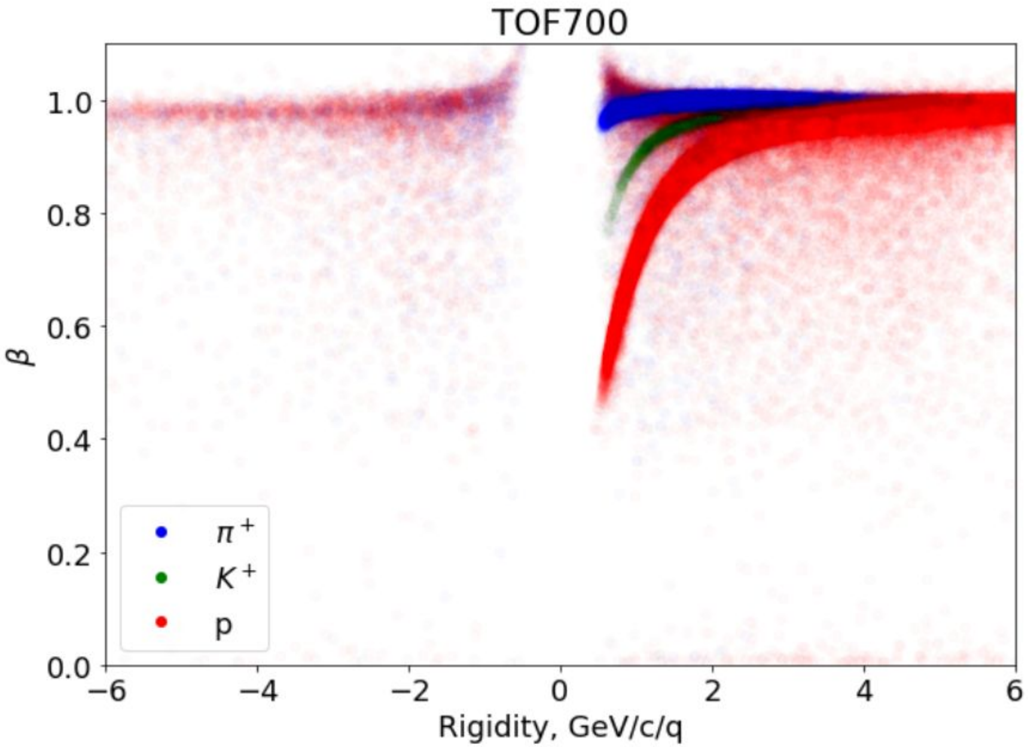


E_{total} = 98.3%

Class imbalance problem

Next we are going to investigate the class imbalance problem





Backup

Classification of Charged Particles

In Machine Learning terms PID can be considered as **classification** task (**Supervised** learning).

Let

X - is the input space (particle characteristics such as: dE/dx, m^2, β, q, etc)

Y - is the output space (particle species such as: π, k, p, etc)

Unknown mapping exists

$$\mathbf{m : X \to Y},$$

for values which known only on objects from the finite training set

$$\mathbf{X^n = (x_1, y_1), \ldots, (x_n, y_n)},$$

Goal is to find an algorithm **a** that classifies an arbitrary new object $\mathbf{x \in X}$

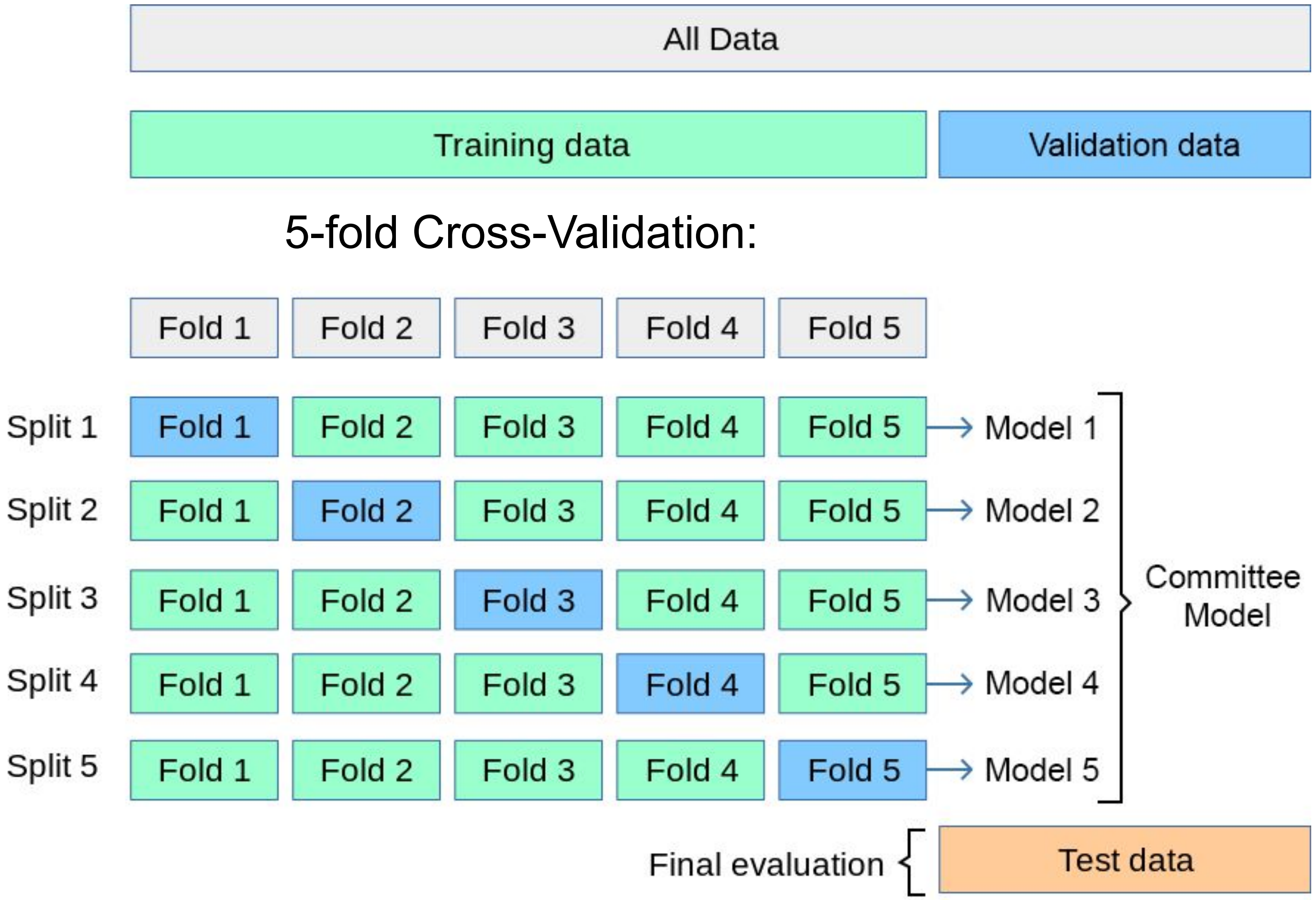
$$\mathbf{a : X \to Y}.$$

Formulas

$$m^2 = \frac{p^2}{c^2}\left[\frac{t^2c^2}{L^2} - 1\right] \qquad \beta = \frac{L}{ct}$$

$$-\left(\frac{dT}{dx}\right) = \frac{4\pi n_e z^2 e^4}{m_e v^2}\left[\ln\frac{2m_e v^2}{I} - \ln(1-\beta^2) - \beta^2 - \delta - U\right],$$

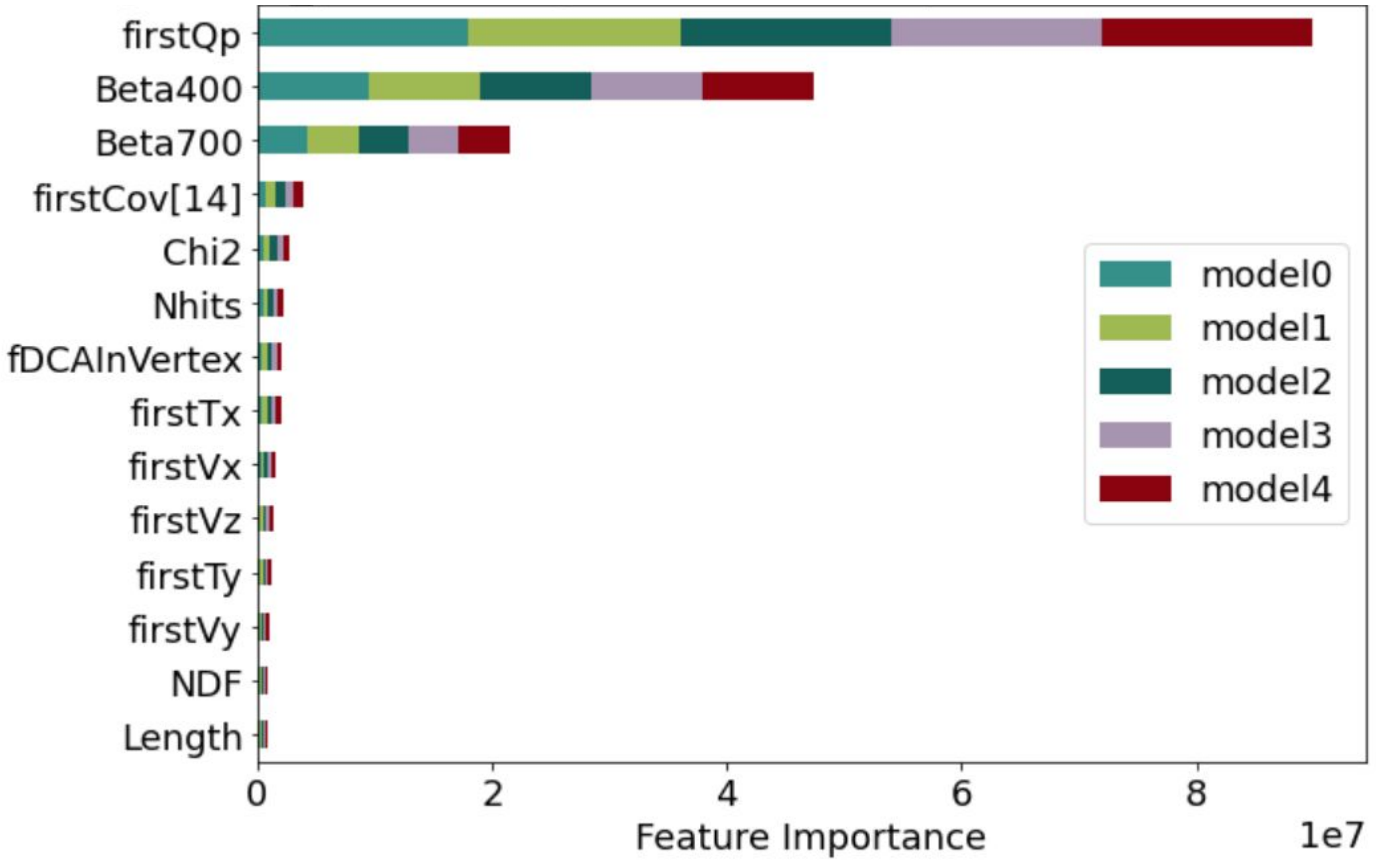
Experiment design



All classifiers have been trained using the Nvidia Tesla V100-SXM2 NVLink 32GB HBM2 within the ecosystem for tasks of machine learning, deep learning, and data analysis at **HybriLIT** platform

XGBoost Model Interpretation. Feature Importance

Importance type can be defined as the total gain across all splits the feature is used in



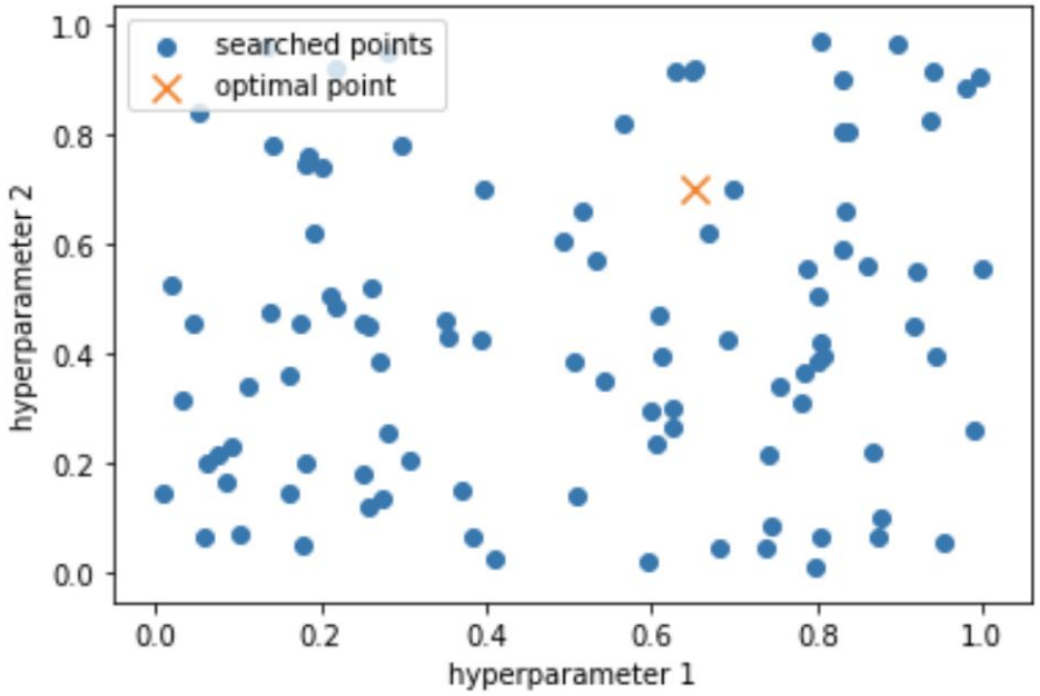
This approach are sensitive when input variables are correlated, and may lead for instance to unreliability in the importance ranking

Hyperparameters tuning

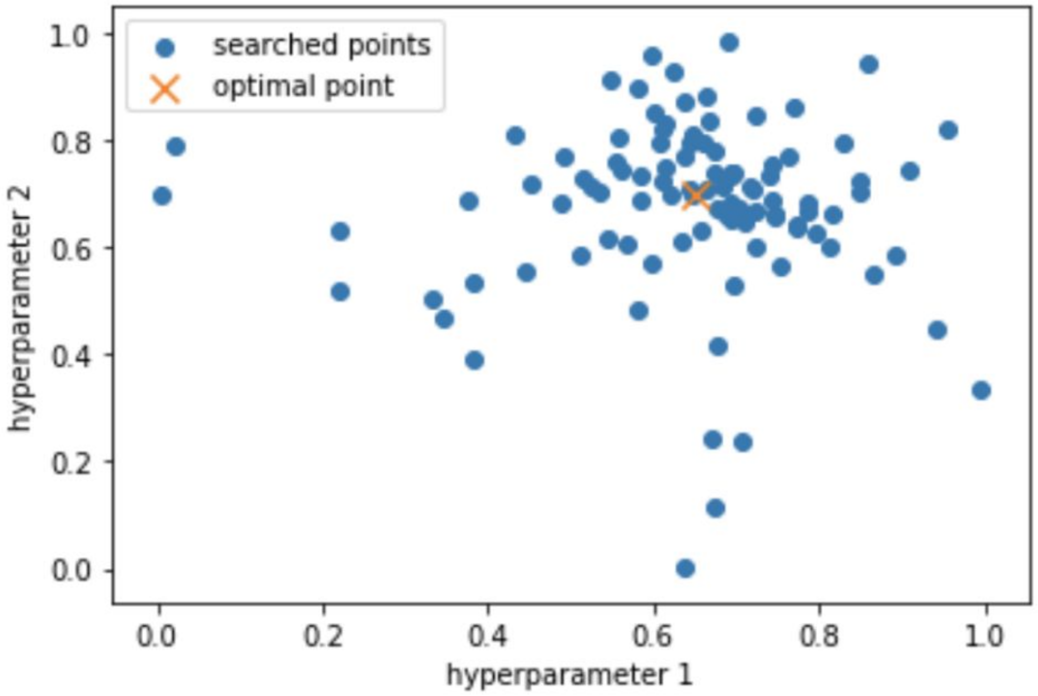
Tree-structured Parzen Estimator (TPE) was used to find the optimal hyperparameters;

TPE is a form of Bayesian Optimization.

Random search



TPE search



https://miro.medium.com/max/4800/1*tYWqO5BwNDVaM3kP3w1IAg.png