6th International Workshop on Deep Learning in Computational Physics DLCP-2022

MLIT, JINR, Dubna, July 6-8, 2022

Decision trees as an alternative for particle identification with TPC and ToF detector system

A. Ayriyan, H. Grigorian, A. Mudrokh, V. Papoyan

06.07.2022

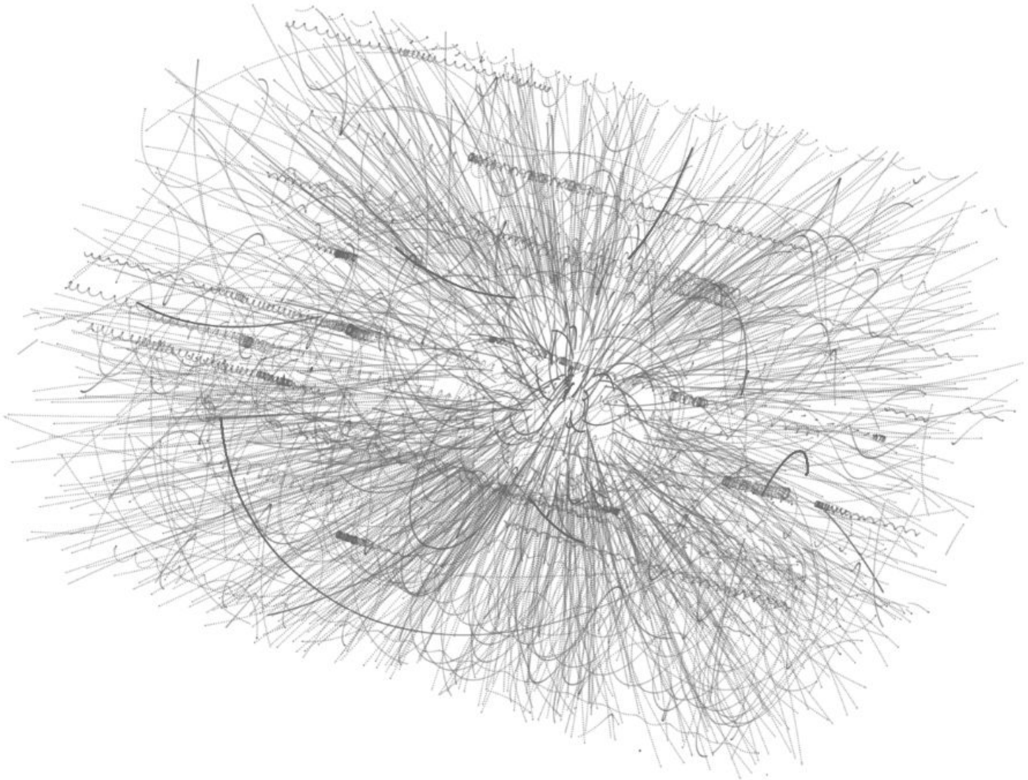
TPC and ToF detectors

Time Projection Chamber (TPC) is an electronically read gaseous detector delivering direct three-dimensional track information: for each point on the particle track, x-, y- and z-coordinates are measured simultaneously [1].

Time of Flight (ToF) determine charged particle velocity by measuring the time required to travel from the interaction point to the time of flight detector.

Particle identification can be achieved by using information about **momentum**, **charge**, **energy loss** (TPC) and **mass squared** (TPC + TOF).

ion collision:



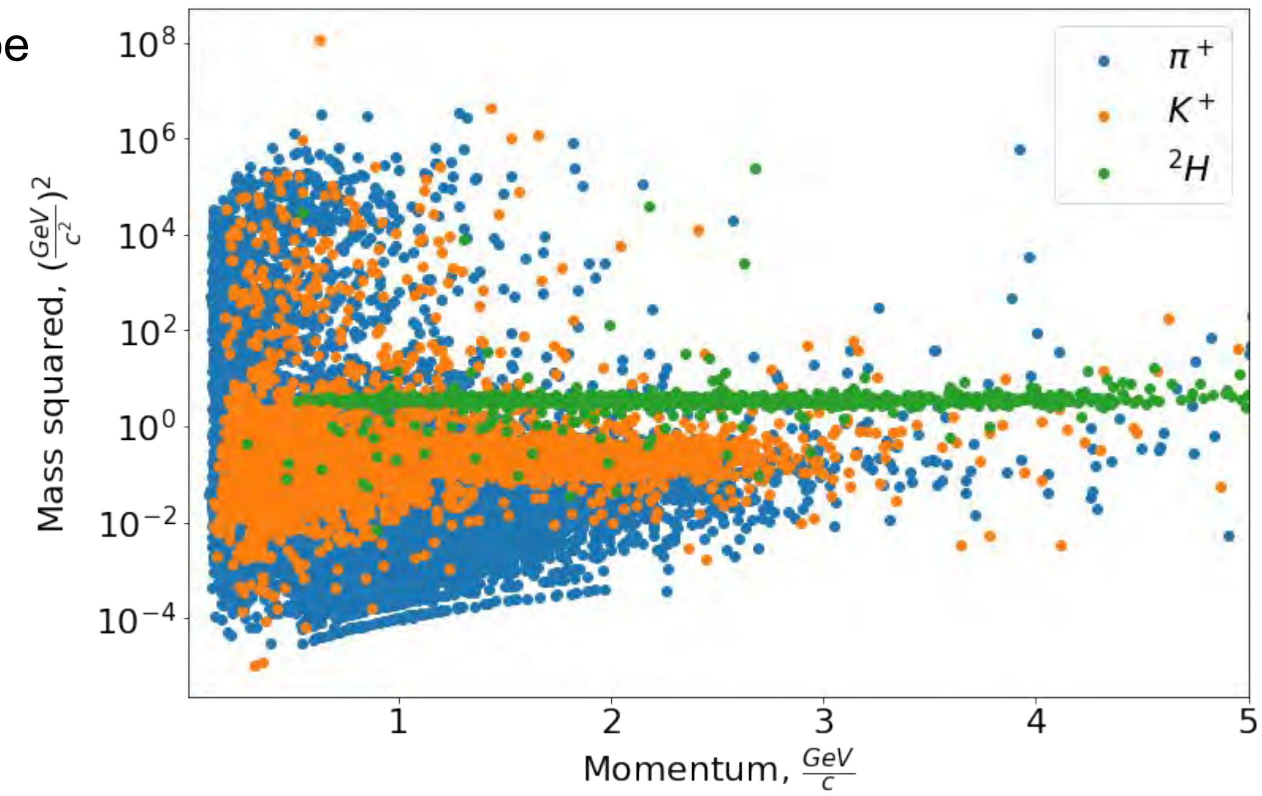
[1] Hilke H. J. Time projection chambers //Reports on Progress in Physics. – 2010. – T. 73. – №. 11. – C. 116201.

Particle Identification

Particle IDentification (**PID**) is the task of identifying the particle type associated with a given track.

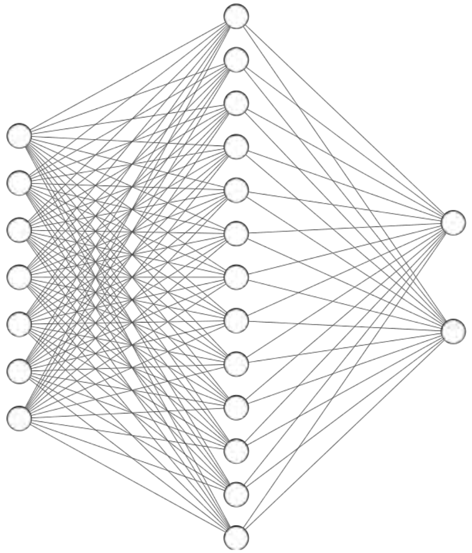
In Machine Learning terms, PID can be considered as:

1. multiclass classification problem;
2. binary classification problem
   a. one-vs-rest;
   b. one-vs-one.



Machine Learning in PID

Present time ML methods for PID are widely used.

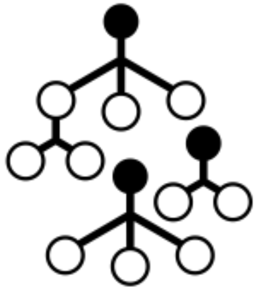


ProbNN (Shallow Neural Networks):

one-particle-vs-rest strategy; One shallow neural network for the each particle type

DNN (Deep Neural Network):

multiclass strategy; Deep NN with three hidden layers



XGboost & **CatBoost** (Boosted Decision Trees):

multiclass strategy; CatBoost uses **oblivious** trees (robust to noise) [1]

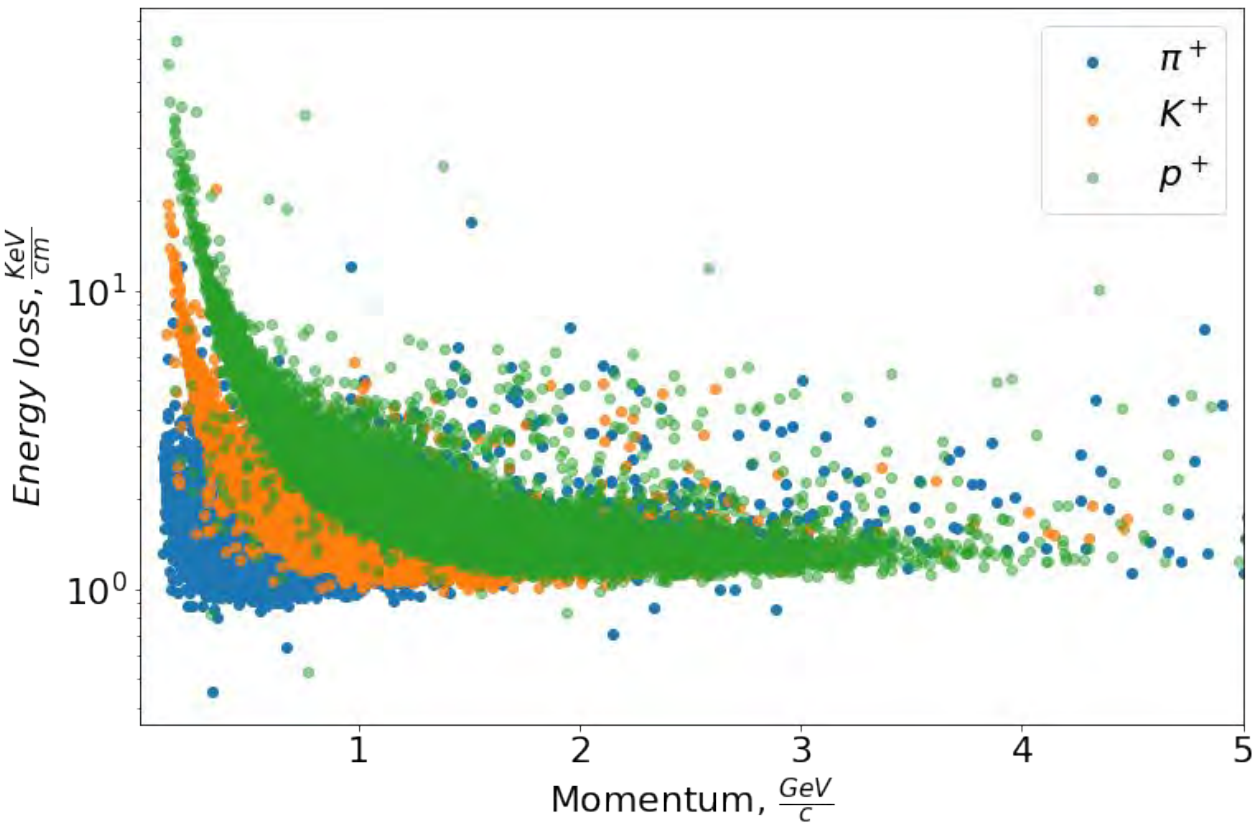
In this research, the preliminary results were obtained by application of **Decision tree** model.

[1] Derkach D. et al. Machine-Learning-based global particle-identification algorithms at the LHCb experiment //Journal of Physics: Conference Series. – IOP Publishing, 2018. – T. 1085. – №. 4. – C. 042038.

Data set

There are 10 **types** of particle:

Protons (p^+ , p^-); Kaons (K^+ , K^-); Pions (π^+ , π^-); Triton (t); Deuterium (^2H); Helium-3 (^3He); Helium-4 (^4He).



Feature vector:

- momentum;
- charge;
- energy loss;
- mass squared;
- number of hits in TPC;
- pseudorapidity;
- dca.

Train and Test Samples

The Decision tree model is trained on Monte-Carlo data (24M tracks in total).

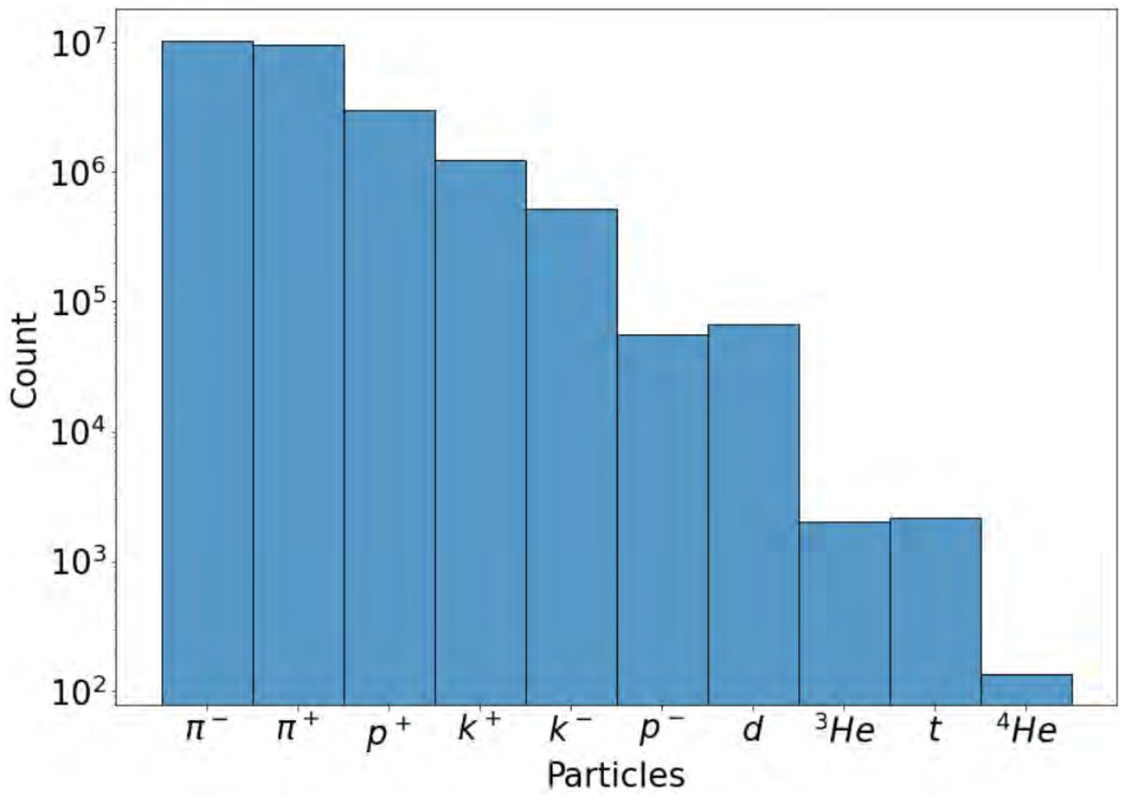
Train sample: random 70% tracks from Monte-Carlo data.

Test sample: remaining 30% tracks.

Classes are **imbalanced** - not having enough tracks for the minority classes (^4He, t, ^3He).

PID efficiency reduction for minority classes. Balanced data are better for training.

Particles frequency



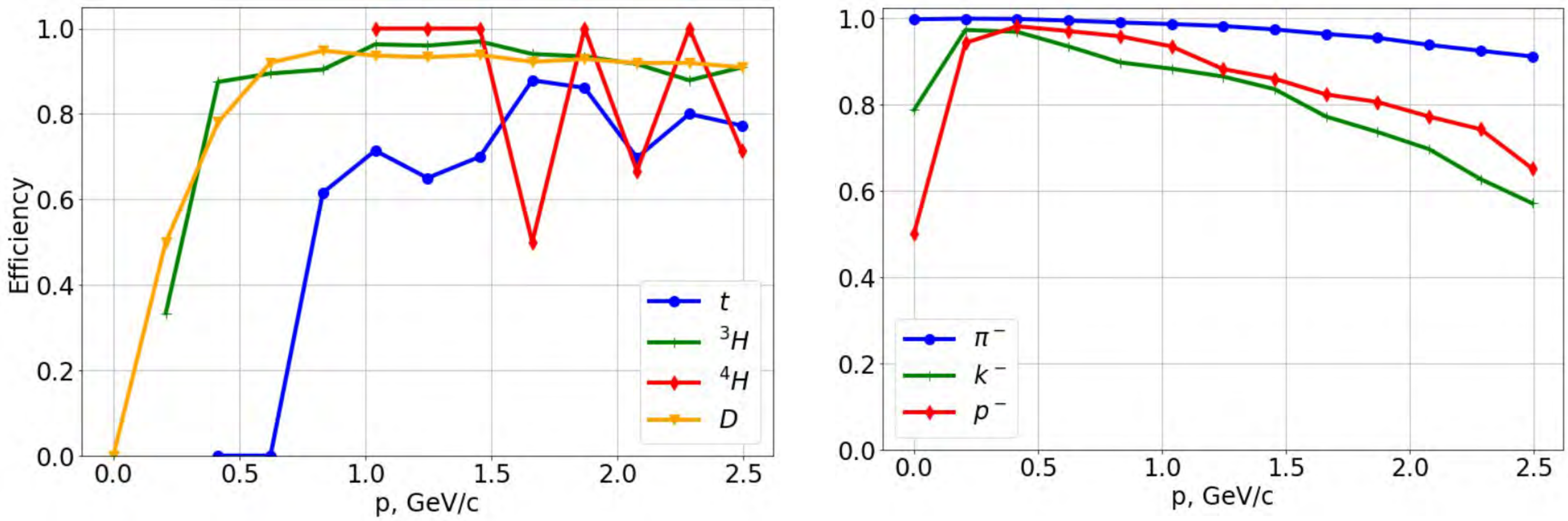
The preliminary results

Decision tree parameters:

- **criterion** : gini;
- **depth** : 7.

$$Efficiency = \frac{TP_{tracks}}{N_{tracks}}$$

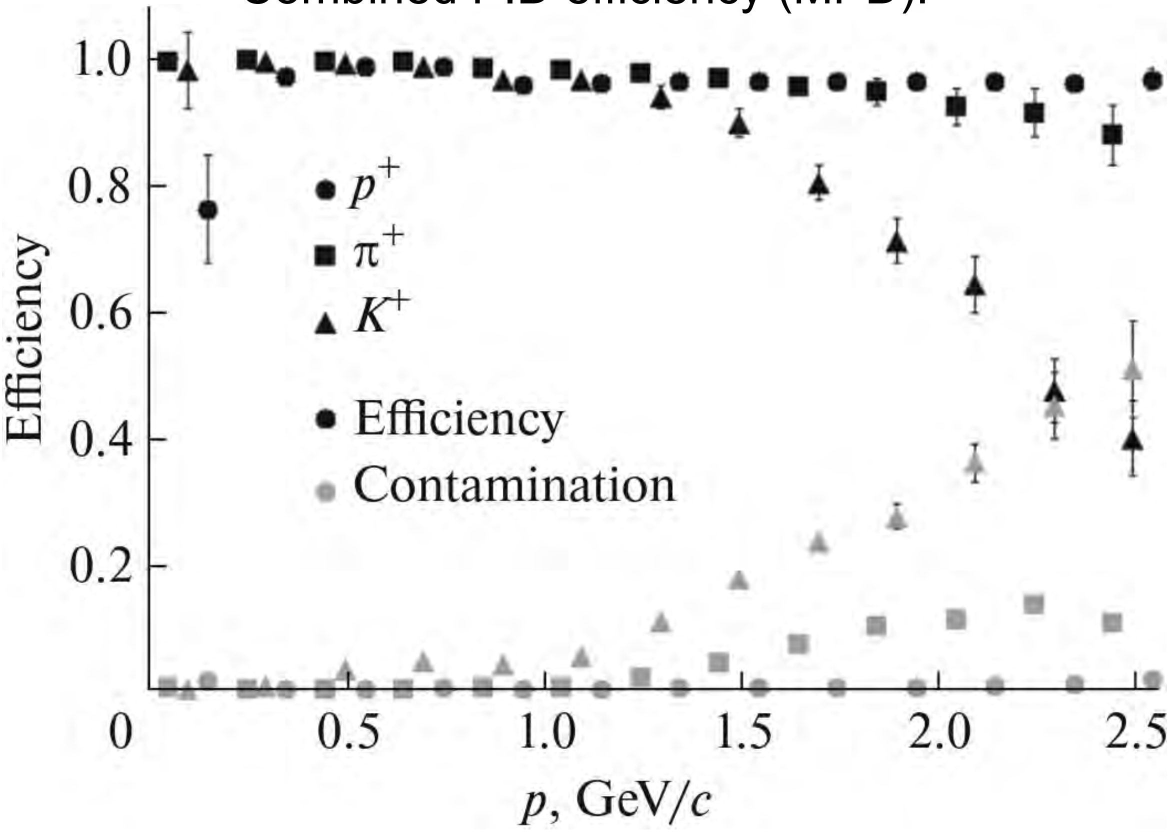
Combined PID efficiency:



Current PID results in MPD

PID results for the MPD experiment within the TPC and the TOF detector [1].

Combined PID efficiency (MPD):

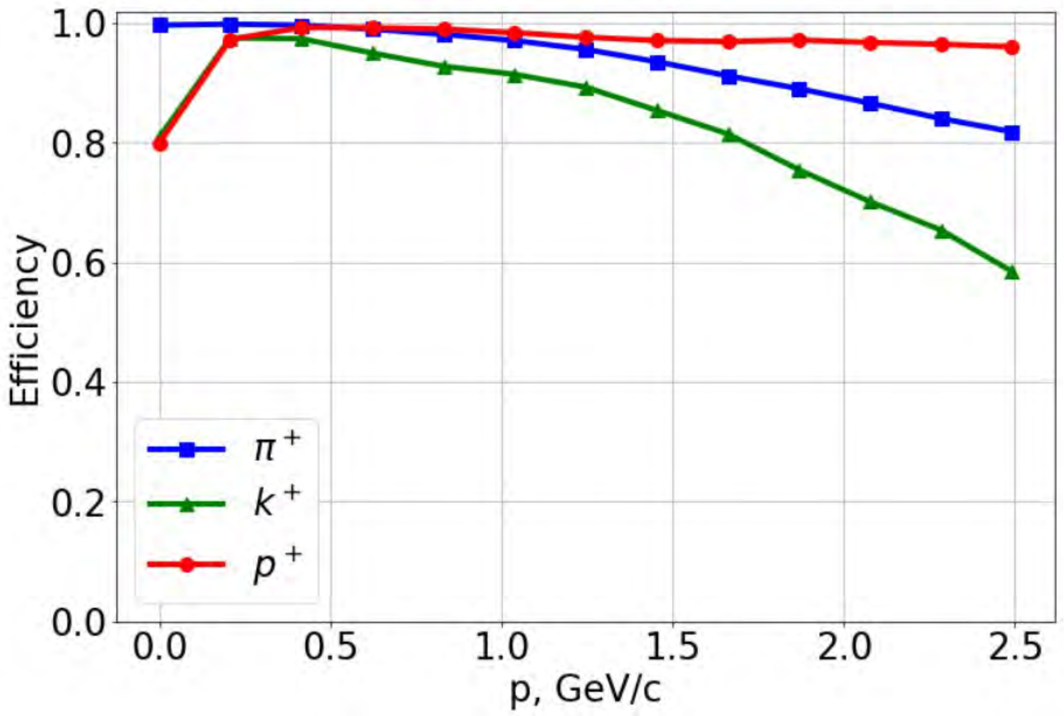


[1] Kolesnikov V. et al. Towards a realistic Monte Carlo simulation of the MPD detector at NICA //Physics of Particles and Nuclei Letters. – 2019. – T. 16. – №. 1. – C. 6-15.

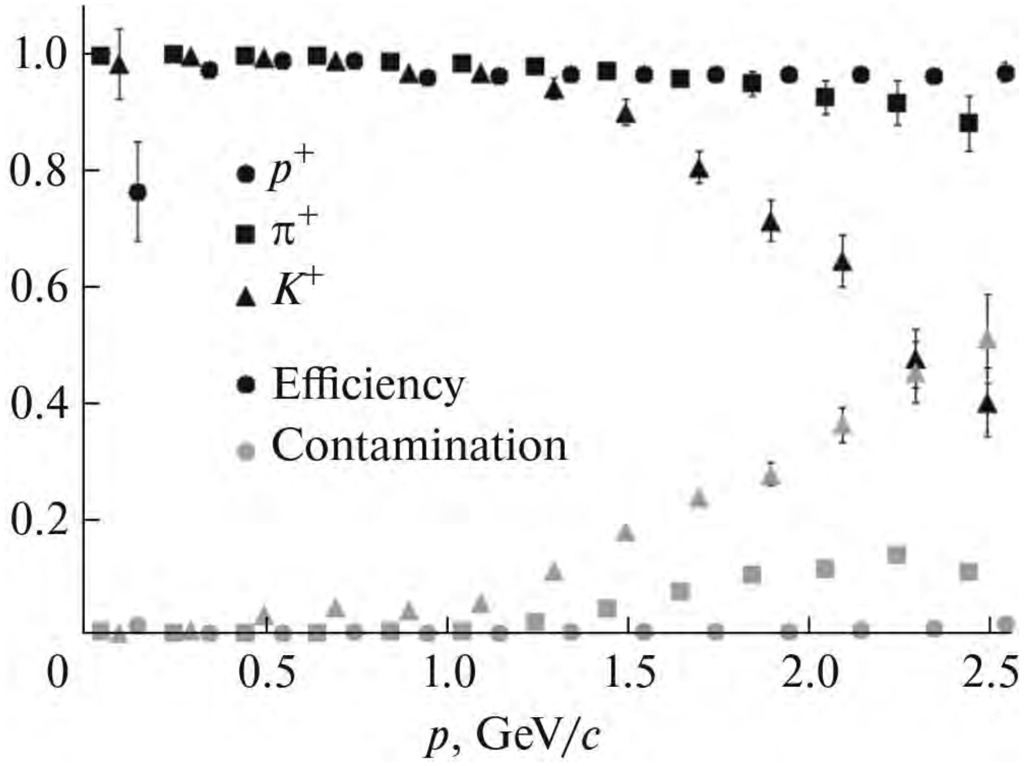
The preliminary results

$$Efficiency = \frac{TP_{tracks}}{N_{tracks}}$$

Combined PID efficiency (Decision Tree):



Combined PID efficiency (MPD) [1]:



[1] Kolesnikov V. et al. Towards a realistic Monte Carlo simulation of the MPD detector at NICA //Physics of Particles and Nuclei Letters. – 2019. – T. 16. – №. 1. – C. 6-15.

Conclusions and Outlook

1. Application of simple Decision Tree approach allowed **to reproduce** the properties of the PID MPD results. For some of particles the efficiency becomes **even better**.
2. A new **balanced training data set** will be generated for all particle classes and all momentum range. Such dataset is expected to **increase** the PID efficiency.
3. Decision Tree approach will be **replaced** to Boosting Decision Tree and Random Forest algorithms.