October 19 - 25, 2024

CHEP 2024

ML-based classification in an open-source framework for the ALICE heavy-flavour analysis

M.T. Camerlingo (INFN Bari) on behalf of ALICE Collaboration and FAIR Project*

FAIR Future Artificial Intelligence Research

Abstract: The reconstruction of charmed baryons using Machine Learning (ML) in the ALICE experiment at the CERN LHC offers a valuable use-case to develop a user-friendly and interactive open-source pytorch-based environment to test the INFN computing infrastructure and perform BDT-based multivariate analyses within the activities of FAIR, a European project synergic to the ALICE experiment.

Overview of FAIR WP6.7 Use Case (UC)

ReCaS-Bari HPC/GPU Cluster

JupyterHub for ReCaS users

Methods&Resources: The FAIR benchmark imports different ML packages (XGBoost, Sklearn and Ray) to prepare the data and configure the BDT models in Jupyter Notebooks. Currently, the training is performed on a preliminary dataset, fraction of pp collisions collected in 2022 by ALICE in Run3 at the LHC, using a partitioned-shared A100 GPU available through an Apache Mesos cluster at the ReCaS-Bari datacenter [1].

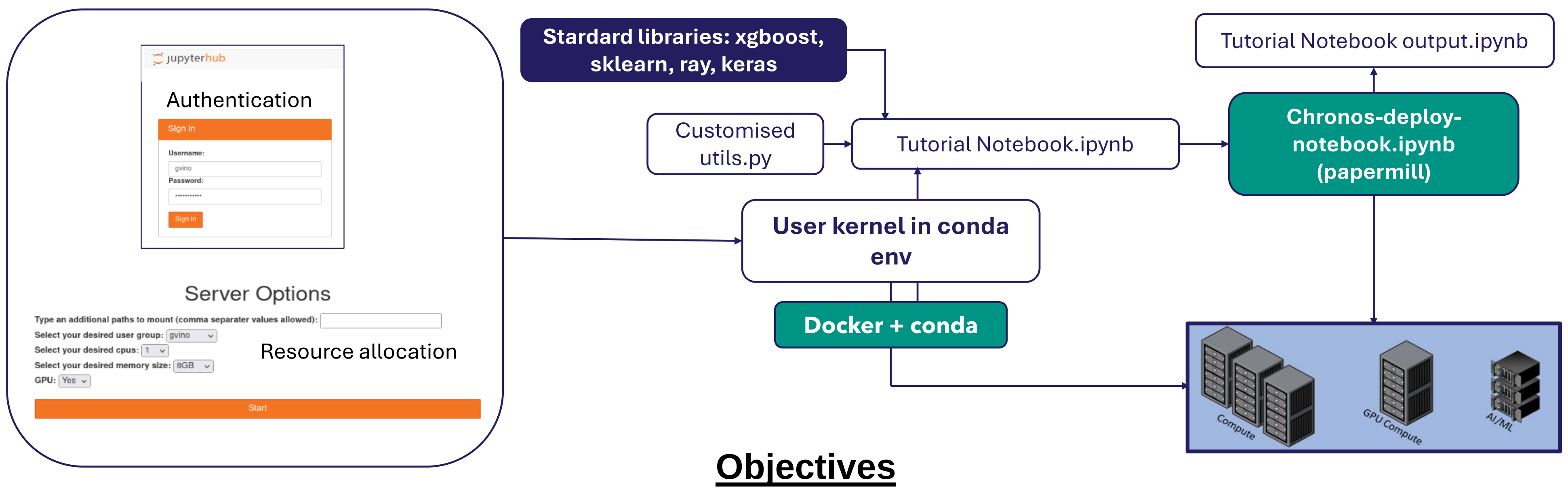
Classifier (BDT)

compared with

different search strategy, different ML tool:

Autoencoder (Anomaly detection)

User-friendly and interactive pytorch-based environment

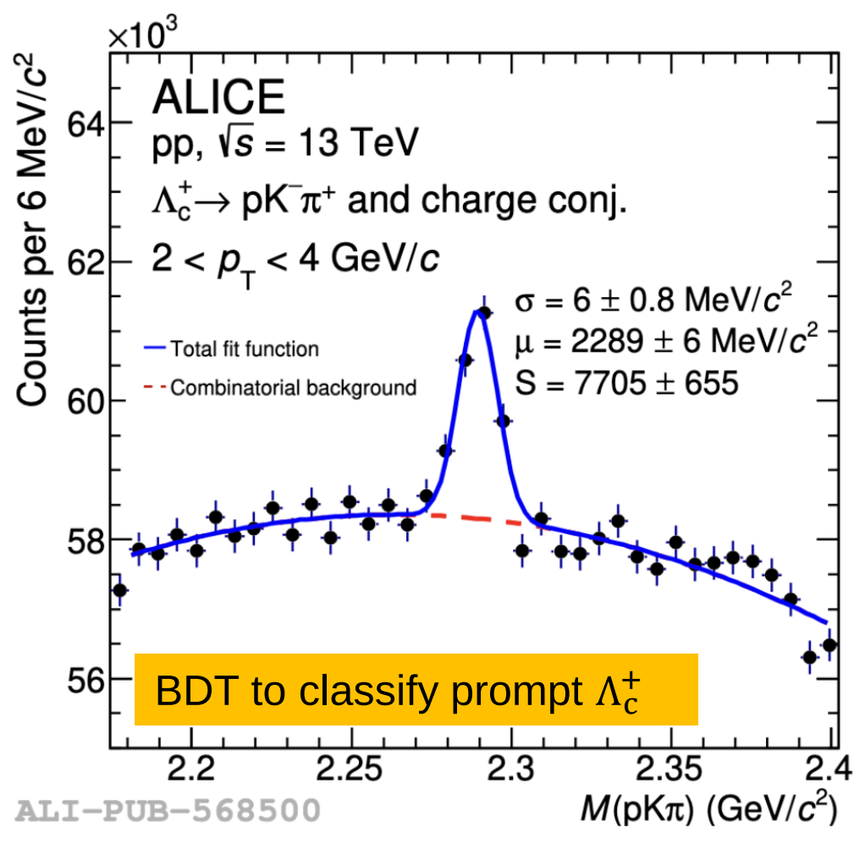

jupyterhub
Authentication
Sign in
Username:
gvino
Password:
Sign in
Server Options
Type an additional paths to mount (comma separater values allowed):
Select your desired user group: gvino
Select your desired cpus: 1
Select your desired memory size: 8GB
GPU: Yes
Start
Resource allocation
Stardard libraries: xgboost, sklearn, ray, keras
Customised utils.py
Tutorial Notebook.ipynb
Tutorial Notebook output.ipynb
Chronos-deploy-notebook.ipynb (papermill)
User kernel in conda env
Docker + conda
Compute
GPU Compute
AI/ML

Objectives

- FAIR UC to test the new machines.
- A performance comparison of the investigated ML architectures trained with simulated signal events and background Run 3 data provided by ALICE.

- When a larger dataset will be available, we intend to leverage a GPU-powered Kubernetes cluster for processing large-scale applications, including ML tool training and inference.

- ML-based analyses [2] were already performed in the ALICE HF sector.



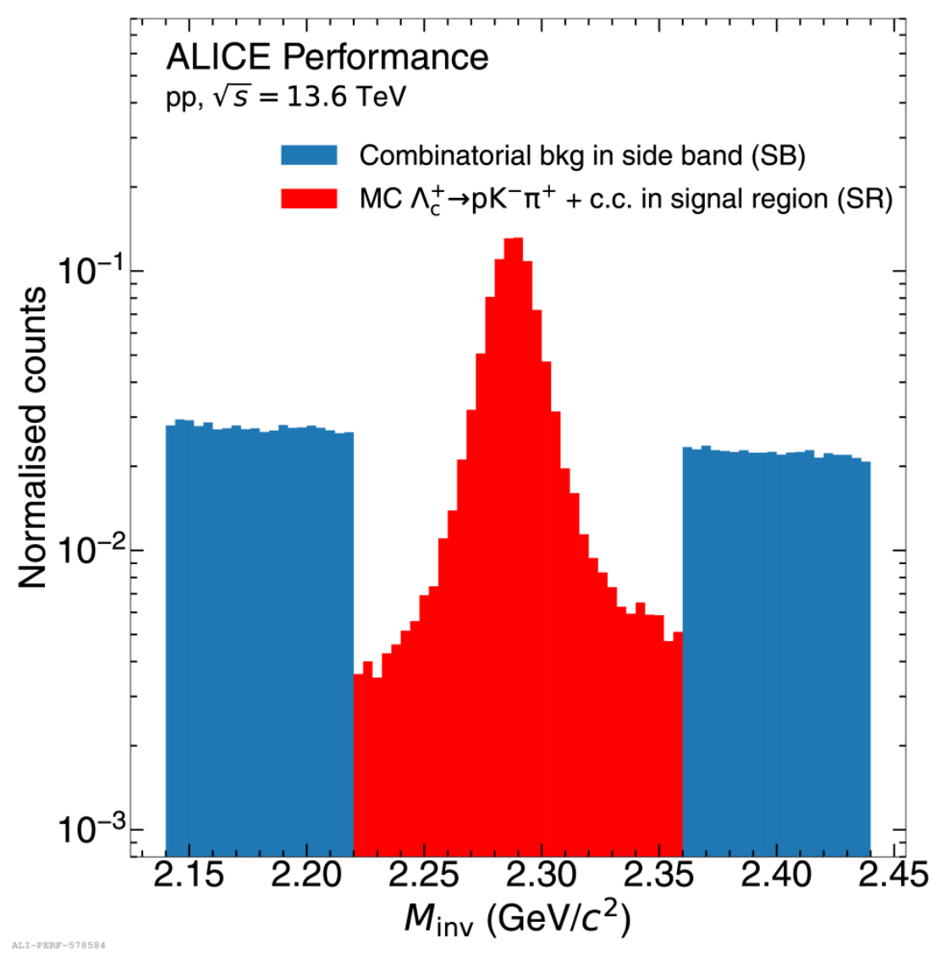
FAIR UC aims to contribute to the new ML-based analysis of Λ_c^+, $\Xi_c^+ \rightarrow \mathrm{p\,K^-\pi^+}$and their charge conjugates.

[1] M. Antonacci et al., «The ReCaS Project: The Bari Infrastructure" , https://doi.org/10.1142/9789814759717_0003

[2] ALICE Coll., «Study of flavor dependence of the baryon-to-meson ratio in proton-proton collisions at √s=13 TeV», https://doi.org/10.1103/PhysRevD.108.112003

Binary classifier workflow/search strategy

1. Define the Side Bands (SB) and Signal Region (SR) of invariant mass (M_{inv})



INPUT DATA:

- **CLASS 0 : ALICE data from LHC Run 3 pp collisions in SB**
- **CLASS 1: ALICE MC data simulating the signal $\Lambda_c^+ \rightarrow \mathrm{pK\pi}$ in SR**

   1. Search independent variables on M_{inv}
   2. MC signal in SR vs data in SB to find the most discriminant ones
   3. **Prepare the mixed dataset (S in SR+B in SB) and sample weights + splitting**

2. BDT Training and Test
   1. Monitoring figures of merits
   2. Calculate score threshold @MaxSignificance (cut)
3. Cut application on data in SR

ALICE

Anomaly detection workflow/search strategy

1. Define the side bands (SB) and signal region SR of invariant mass (M_{inv})
   1. Search independent variables on M_{inv}
   2. If MC is available, the most discriminant ones
2. Train the autoencoder to represent background from the variables that are independent on the invariant mass region
3. Calculate the MSE between the reconstructed input and original input for
   1. Test data in SB (i.e. combinatorial bkg) - **to define a data-driven MSE threshold**
   2. **MC signal in SR to cross-validation and to calculate the MSE threshold @MaxSignificance**
4. Cut application on data in SR

$\Lambda_c^+ \rightarrow \mathrm{p\,K^-\pi^+}$ with a Tight preselection BDT vs AE (ONGOING)

Common preselection on input features (Tight to improve the quality of input dataset)

Boosted Decision Tree (BDT)

- CLASS 0 (background) : ALICE data from LHC Run 3 pp collisions in SB
- CLASS 1: ALICE MC data simulating the signal in SR

Hyperparameter	Value	Hyperparameter	Value
input features	12	gradient boosting library	XGBoost
tree_method	hist	device	cuda
objective	binary logistic	eval_metric	auc (rmse, error)
colsample_bytree	0.952	subsample	0.941
max_depth	7	min_child_weight	3
η	0.168	num_round	100 (default in fit)
α	1 (default)	γ	0 (default)

Autoencoder (AE)

- Ordinary events (background) : ALICE data from LHC Run 3 pp collisions in SB
- Test labelled sample: ALICE data from LHC Run 3 pp collisions in SB + MC simulations in SR

Layer (type)	Output Shape	Param #	Activation
In Encoded (InputLayer)	[(None, 13)]	0	regularizers.L2(lr=0.0687)
Encoded (Dense)	(None, 239)	3346	ReLU
Latent (Dense)	(None, 20)	4800	ReLU
Decoded (Dense)	(None, 239)	5019	ReLU
Out decoded (Dense)	(None,13)	3120	sigmoid
Hyperparameters	**Optimizer**	**Batch**	**epochs**
	Adam	1996	30

Computing performance:

- As expected, BDT has faster times but heavy resident memory size (RES) usage than AE.

BDT	Wall time (s)	RES (GiB)	Class 0 train/val stat	Class 1 train/val stat	# of trainings
Hyp opt	(52 ± 1)*10	56	4*class 1 stat	32k/8k	100
Cross-val	(9 ± 1)*10	5.5	200k/-		5 folds (x 20 rounds)
Training	8.0 ± 0.5	3.9	207k/52k	32k/8k	1
Prediction	0.060 ± 0.005	3.9	-/52k	-/8k	-

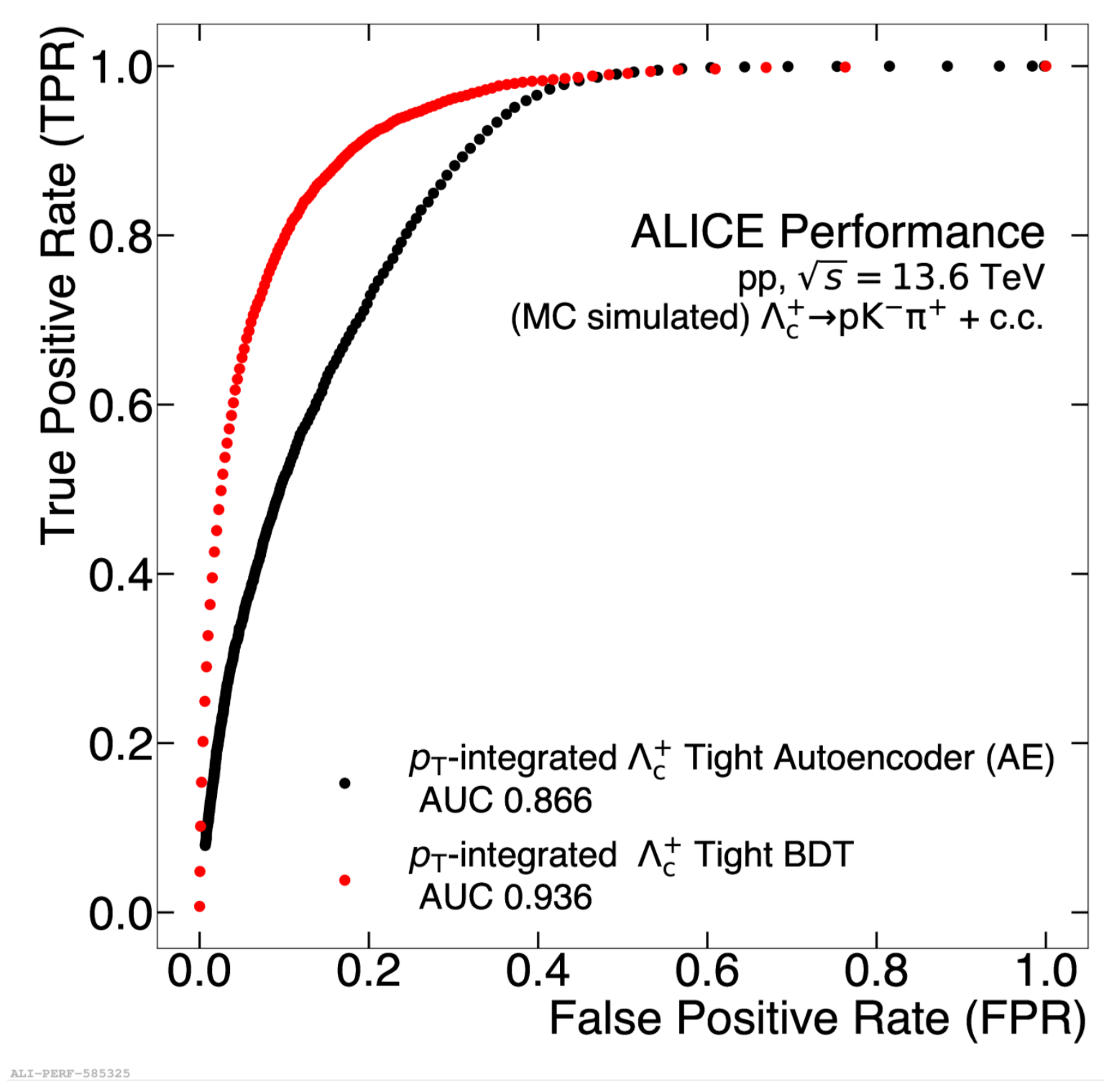
AE	Wall time (s)	RES (GiB)	Class 0 train/val stat	# of training
Hyp opt	(115 ± 6)*10	57.5	150k/10k	100
Training	40 ± 2	1.6	207k/30k	1
Prediction	1.5±0.5	1.6	-22k	-
MC Prediction	2.0±0.5	1.6	-/40k	-

Computing performance during interactive execution of the Use Case Jupyter Notebooks in which 16 CPUs and 0.1 partitioned shared GPU were allocated (standard privileges)

Model performance:

Test ROC curves

- BDT shows a larger Area Under Curve (AUC) than AE

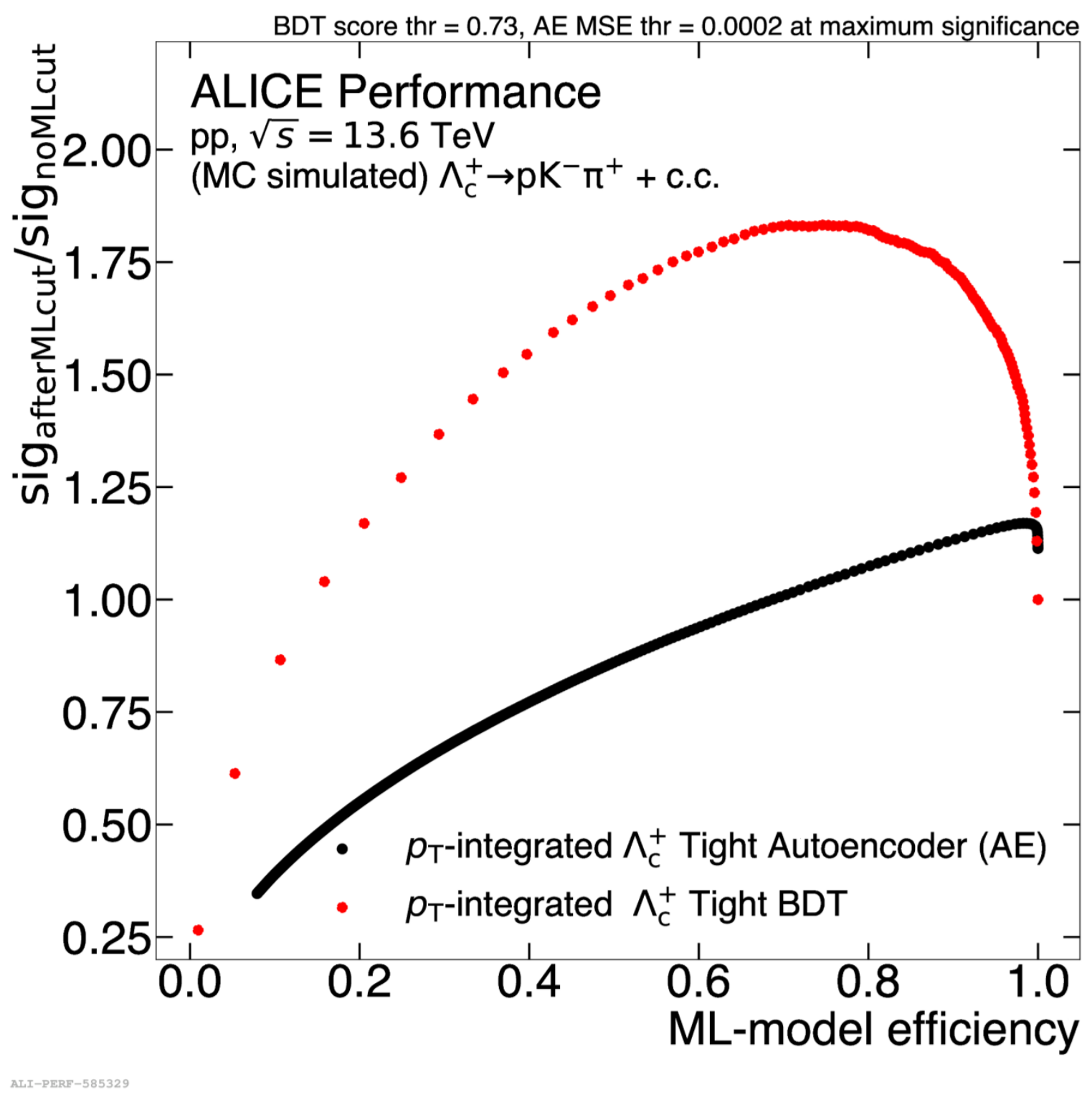


Test significance (sig)

$$\mathrm{sig}_{\mathrm{afterMLcut}} = \frac{\mathrm{Signal}}{\sqrt{\mathrm{Signal+Background}}}$$

after a cut on the min BDT class 1 score (or AE MSE)

BDT score thr = 0.73, AE MSE thr = 0.0002 at maximum significance



- BDT also shows a larger plateau at max significance.
- Significance after the cut on BDT score improves more but it corresponds to a lower efficiency.
- AE has a better efficiency after the MSE-cut.

The next project activities are:

- To study on the $\Xi_c^+ \rightarrow \mathrm{p\,K^-\pi^+}$ (and its charge coniugate) process ✓ already started
- To monitor the new resources/new infrastructure as varying the dataset statistics, ML models, and computing resources input and fill the equivalent tables.

Conclusions: The UC is in a mature state to present the computing performance and the training outputs of two different approaches, the binary classification and anomaly detection, for the signal ML-based discrimination in the ALICE experiment. These preparatory studies led the decision to keep both approaches to exploit their respective advantages in the future infrastructure tests and in the physics analysis performance.

ACKNOWLEDGEMENTS: Thanks to the ReCaS-Bari team for the useful inputs and advices to carry out these studies.

*FAIR (Future Artificial Intelligence Research), funded by the NextGenerationEU program (Italy), https://fondazione-fair.it/en/