Development of ML Algorithms for search of Anti-Helium in Cosmic Rays

Luca Tabarroni
Università di Roma Tor Vergata
luca.tabarroni@roma2.infn.it

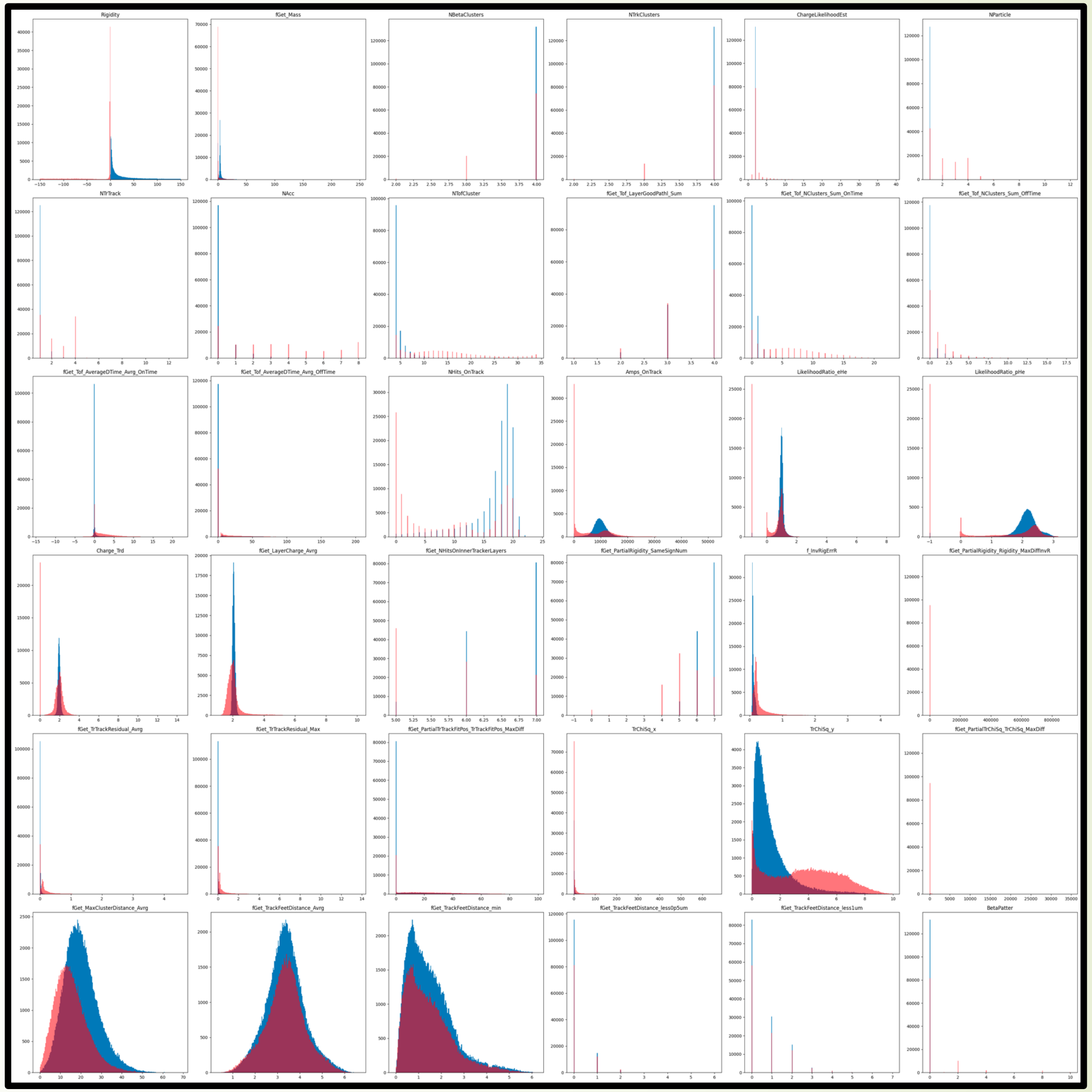
TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

INFN
Roma Tor Vergata

ASI
Agenzia Spaziale Italiana

Feature distributions

- 36 different features distribution
- Different shape for signal and background
- Rigidity and Mass are not available features for our algorithms



/storage/gpfs_ams/ams/groups/AMS-Italy/ntuples/v1.0.0/He.B1236

Boost Decision Tree

Advantages

- No need of features preparation
- Missing values do not affect the building of the tree
- A decision tree is human readable

Disadvantages

- Very sensitive to Overfitting
- Higher training time with respect to other algorithms

Boost Decision Tree

- Faster training speed:

  Light GBM use histogram based algorithm i.e it buckets continuous feature values into discrete bins which fasten the training procedure.

- Lower memory usage:

  Replaces continuous values to discrete bins which result in lower memory usage.

- Better accuracy:

  It produces much more complex trees by following leaf-wise split approach rather than a level-wise approach which is the main factor in achieving higher accuracy. However, it can sometimes lead to overfitting which can be avoided by setting the max_depth parameter.

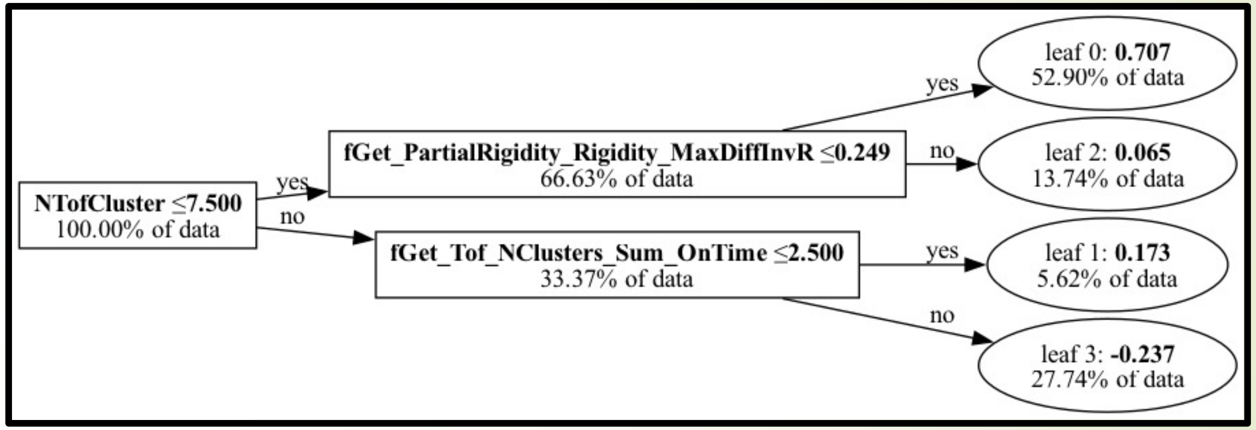
LightGBM

https://lightgbm.readthedocs.io/en/stable/

Boost Decision Tree Diagraph Example

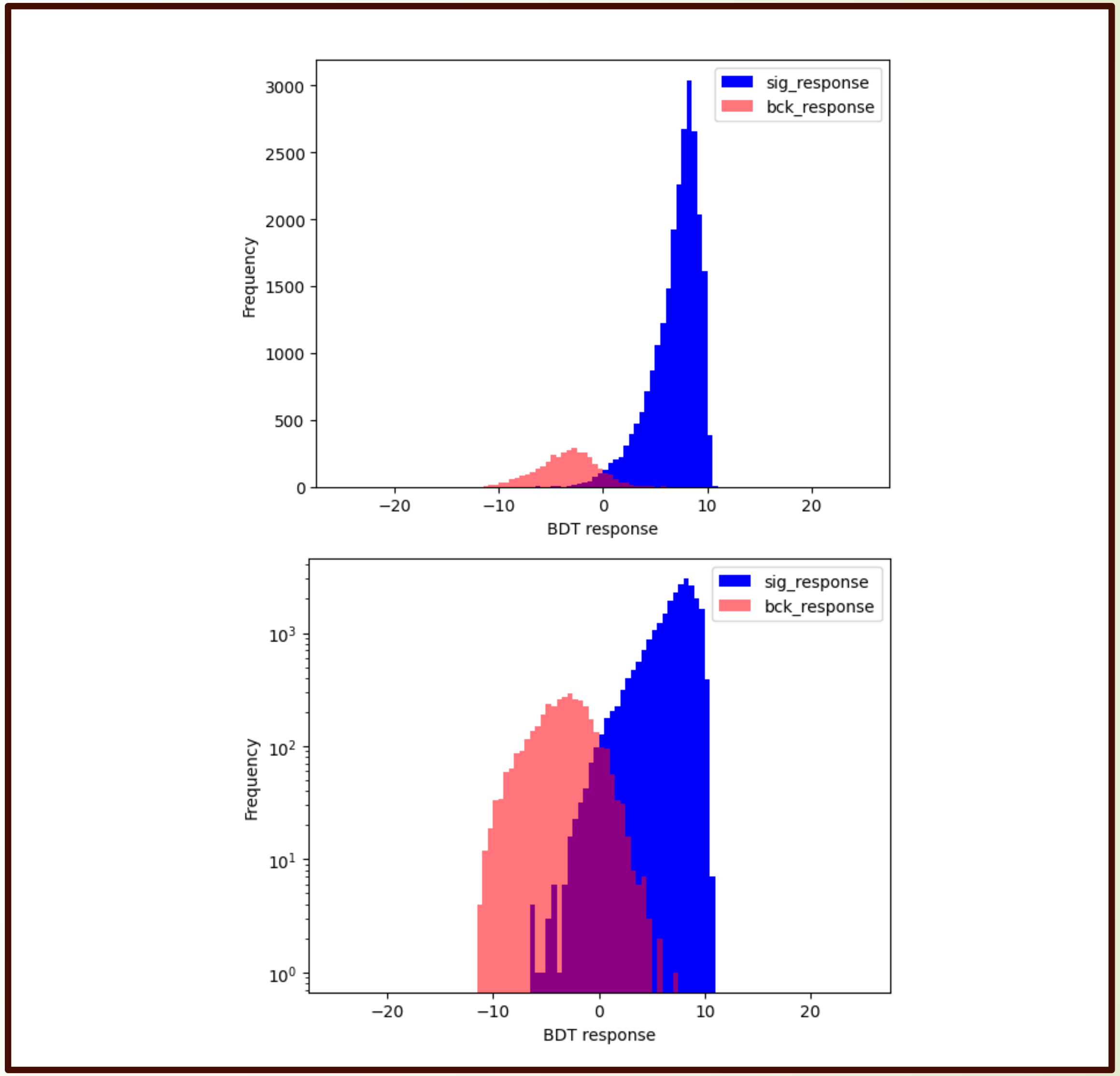
In our case, we implemented an algorithm with:

- One-hundred trees
- Stopping conditions:
  - 4 leaves
  - No depth conditions
  - No purity conditions
- Gradient Boosting
- Learning rate of 0.25



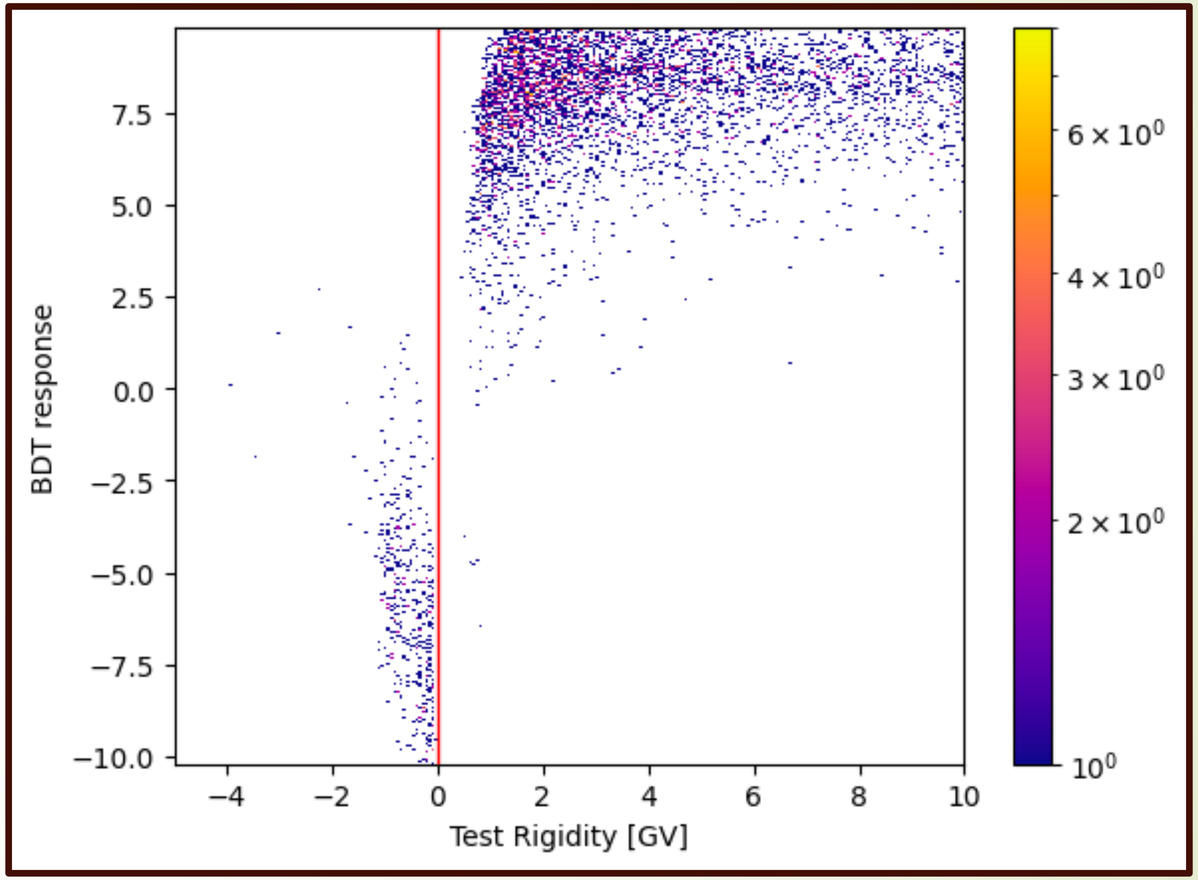
Boost Decision Tree Response

- The BDT associates a response to each leave
- The response is the ”probability” to belong to the background/signal class of an event ending in the leave
- The BDT response is used as classificator



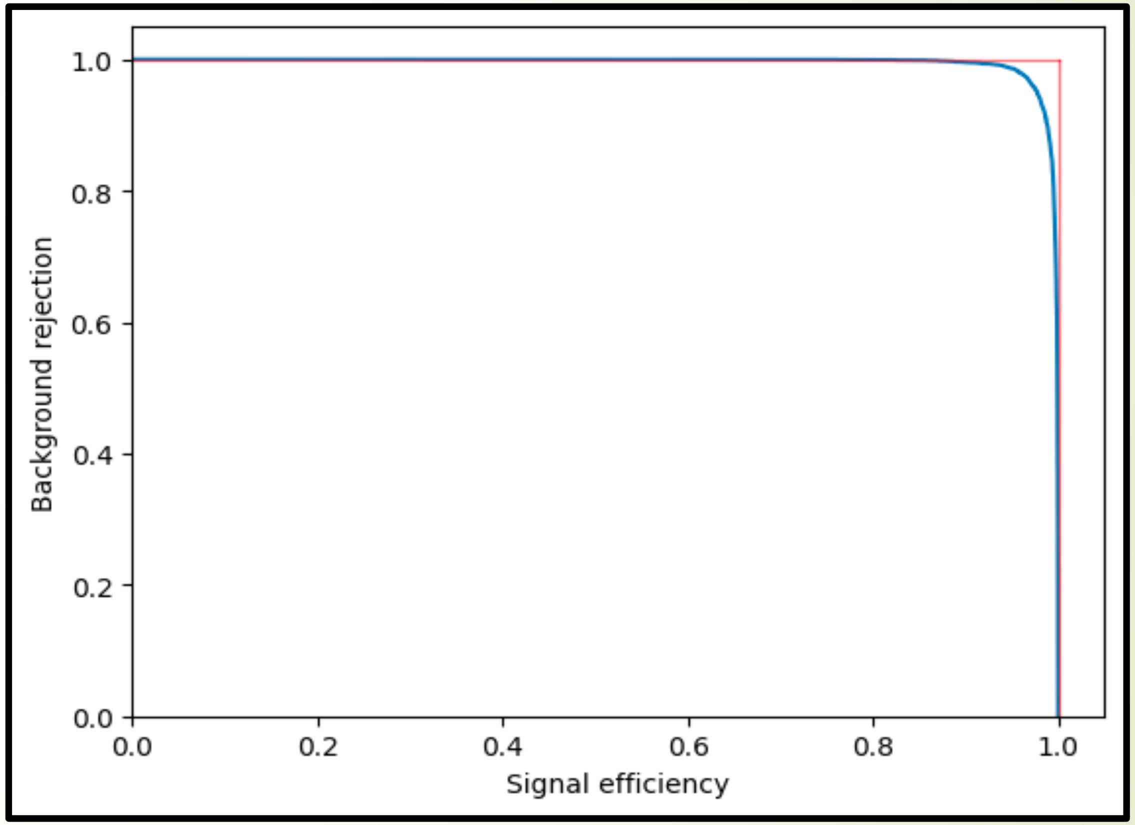
Boost Decision Tree Response vs Rigidity

- We can plot the BDT response vs the event Rigidity
- We got a higher response for the signal and a lower for the background
- A clear distinction is evident, but more background events are necessary



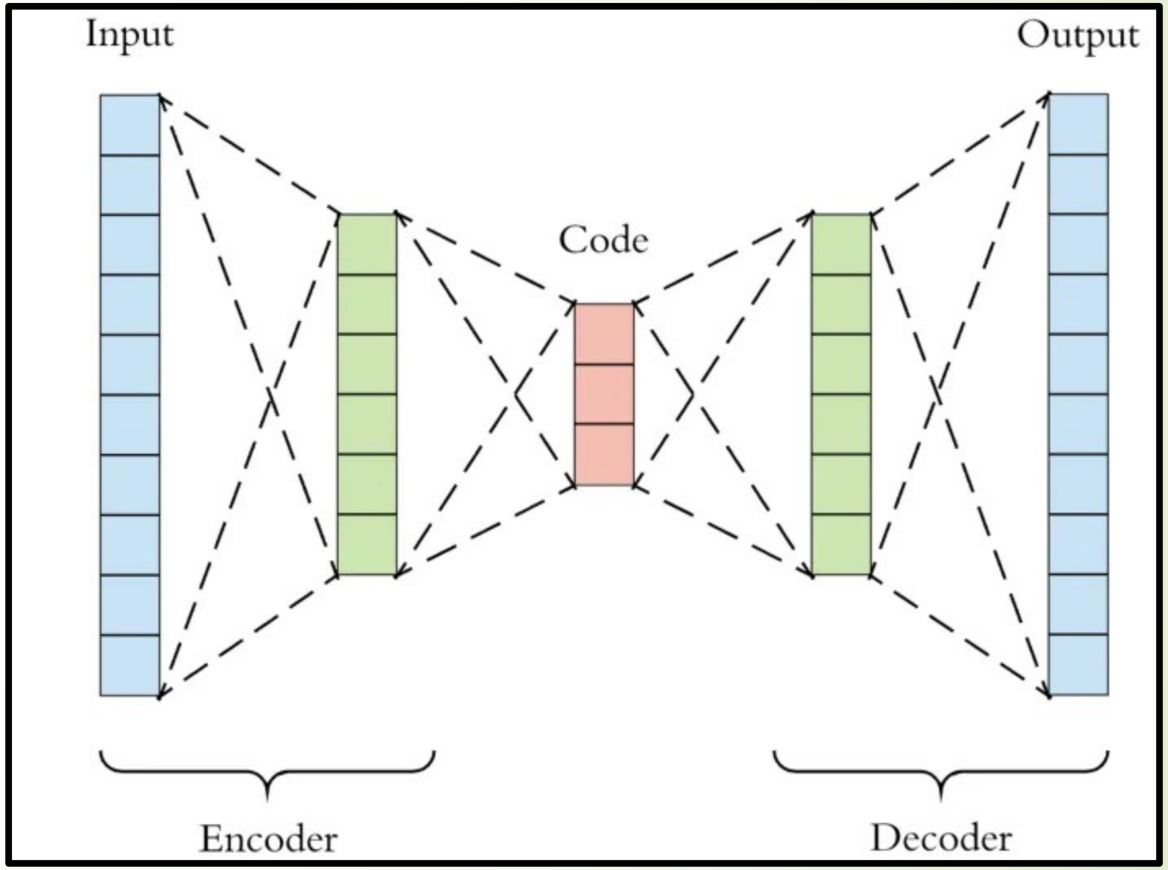
Boost Decision Tree ROC Curve

- The starting point, the baseline
- We are going to implement ML algorithms to improve this result



Auto-Encoder
General introduction

- Main purpose: image compression
  - Image reduction to a lower dimension object: Code space
  - Image reconstruction from the Code space back to the original dimension
  - Convolutional layers
  - Max-Pooling layers
  - Up-Sampling layers

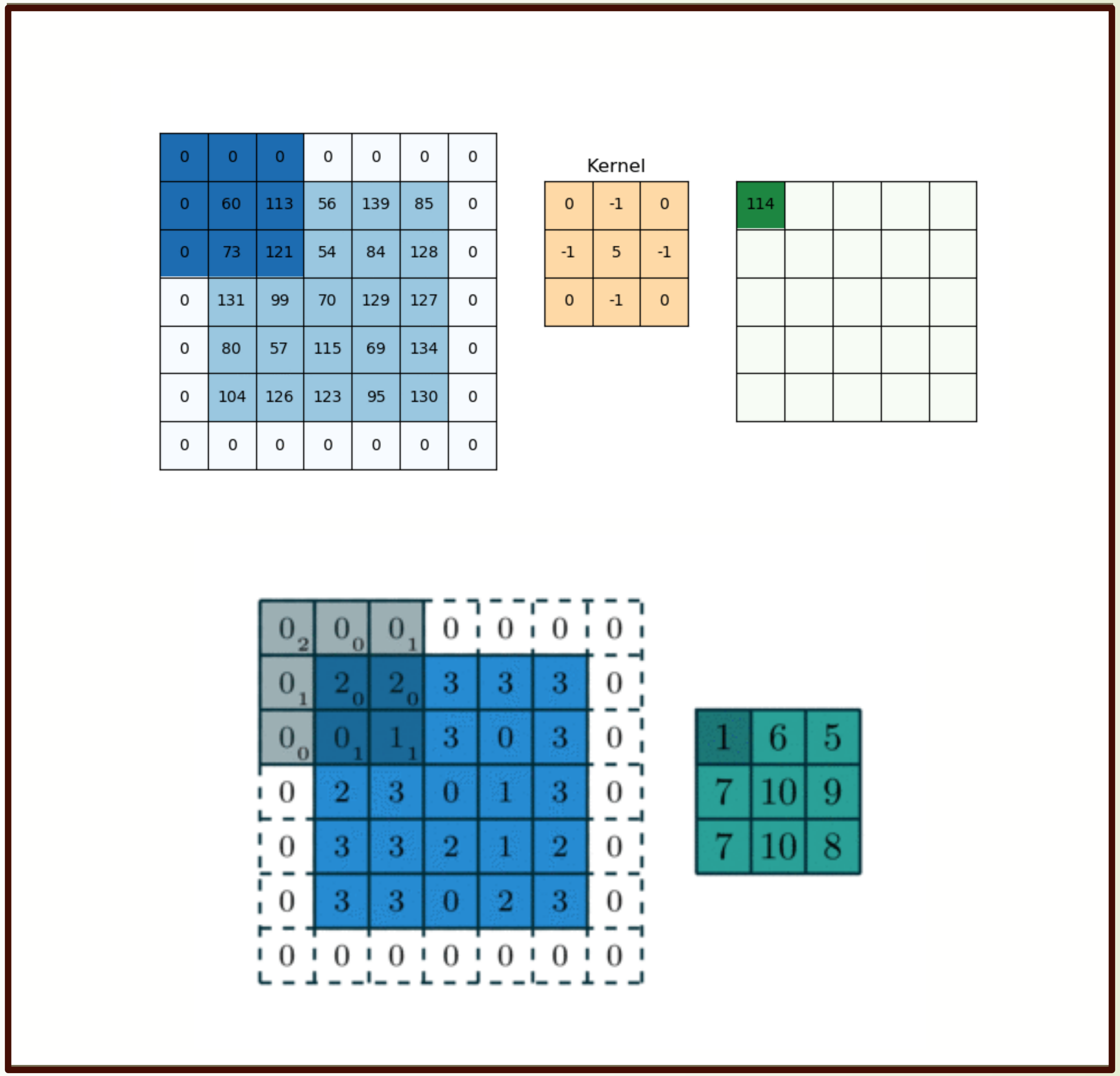


Auto-Encoder Convolutional layers

- Kernel, matrix that swipes on the whole input image. Each ConvLayer may have an arbitral number of kernel
- Stride, number of entries of the input images along which the kernel skip
- Padding, parameter that add zeros around the input image to regulate the output dimension

Output dimension:

$$Output = \frac{Input - Kernel + 2 * Padding}{Stride} + 1$$



Auto-Encoder
Max-Pooling Layers

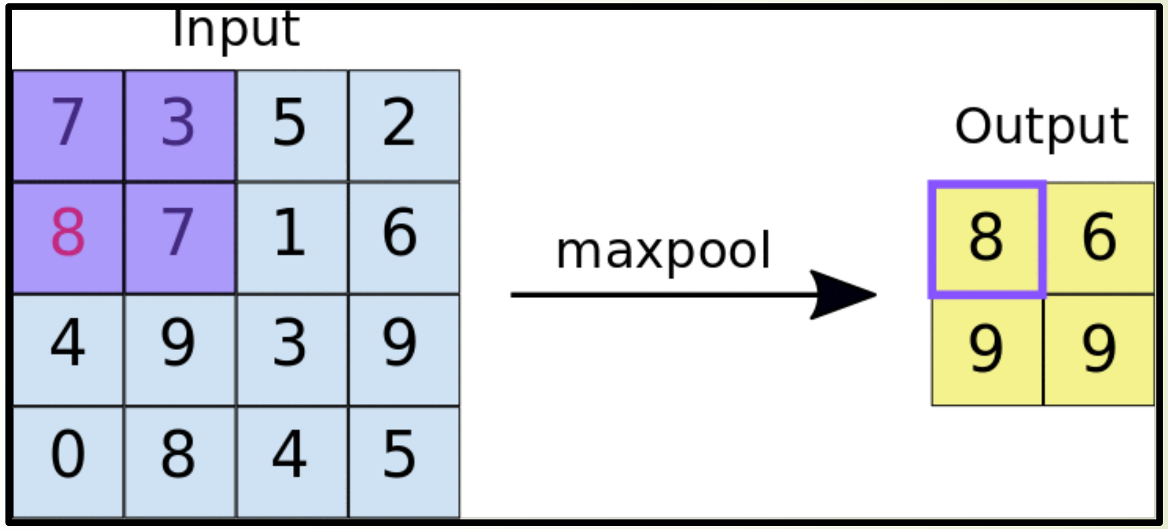
- No learning parameters, only devoted to the reduction of the dimensionality
- Pooling kernel, swipe the original input keeping only the maximal value of those considered
- Stride, number of entries of the input images along which the kernel moves
- Padding, parameter that regulates the output dimension

Padding='same'

$$Output = \left\lceil \frac{Input - 1}{Stride} \right\rceil + 1$$

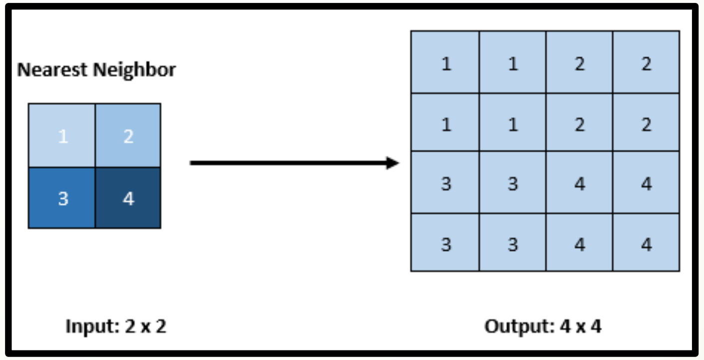
Padding='valid'

$$Output = \left\lceil \frac{Input - Pooling}{Stride} \right\rceil + 1$$



Auto-Encoder
Up-Sampling Layers

- Devoted to the increase of the dimensionality
- Size, upsampling factor along rows and columns
- Interpolation, parameter defining how the new values are evaluated
  - Nearest: the new pixel takes the value of the nearest pixel in the original image

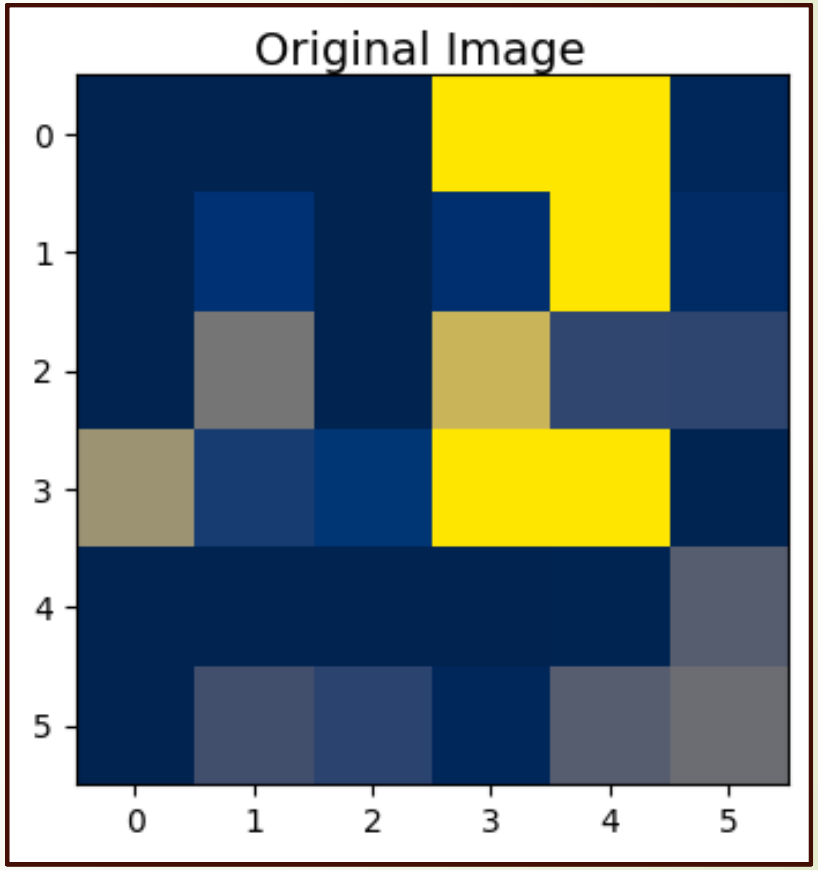


  - Bilinear: the new pixel is the result of a weighted mean of the four nearest pixels of the original image

Auto-Encoder
Input and Output

Input

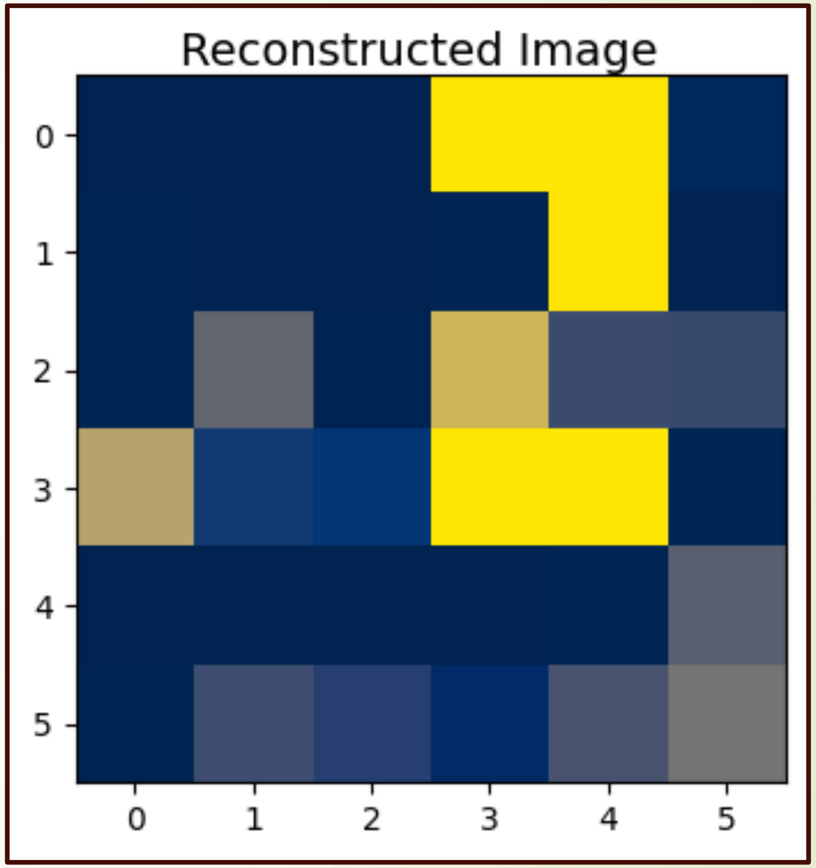
- We arrange the whole events dataset as an array of 6x6 matrices
- Each matrix is a single event
- Each value of a matrix is a normalized feature of the event
- Each feautre becomes a pixel of the input image



Auto-Encoder Input and Output

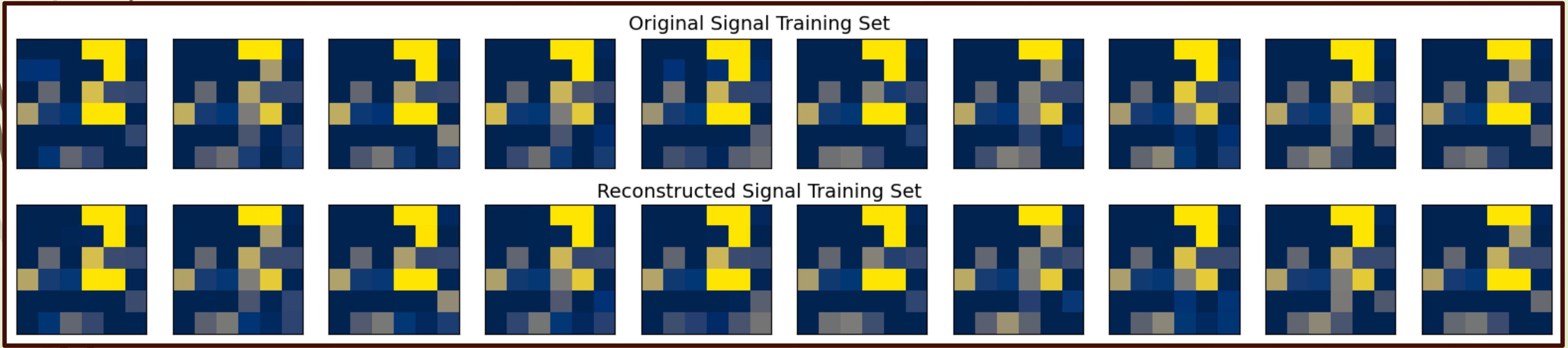
Output

- The Auto-Encoder reconstructs a corrispondent 6x6 image for each event
- The Auto-Encoder is trained to reconsttuct correctly ONLY THE SIGNAL
- We expect good perfomances for the signal
- We expect bad perfomances or the background



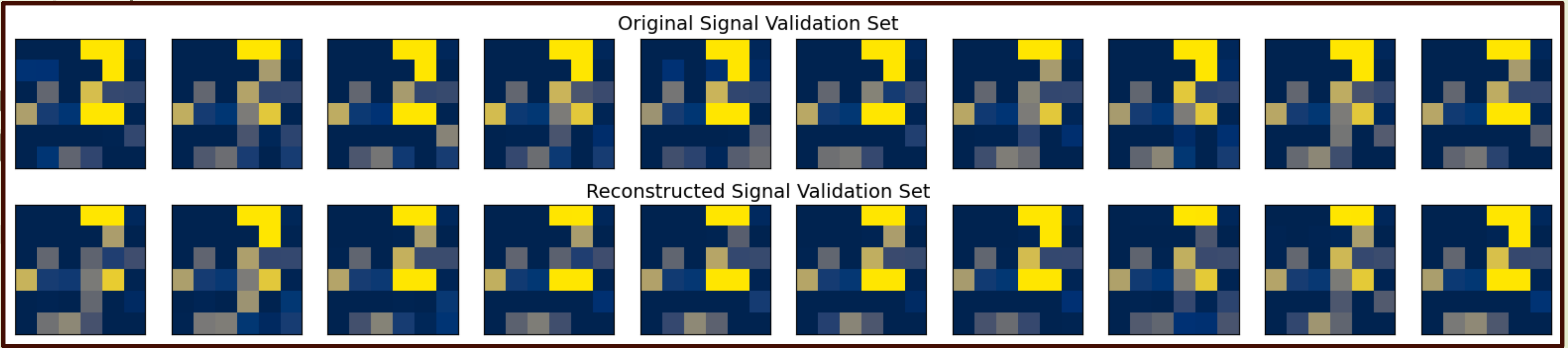
Auto-Encoder Training

- We submit to the AEC a training set of images taken from the original signal dataset
- The AEC will perform its training confronting the output images with the input ones
- The AEC goal is to reach the maximal accordance between the input and the output



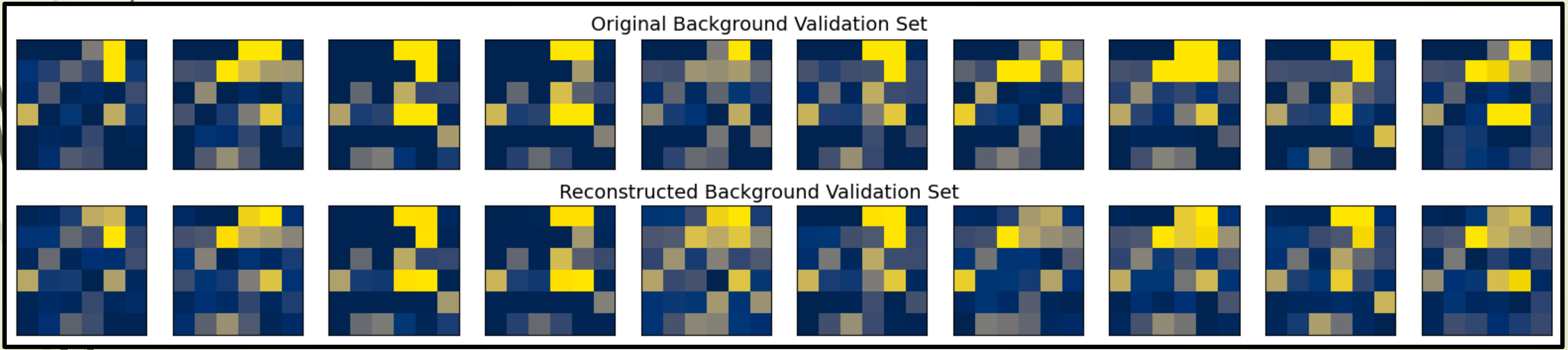
Auto-Encoder Validation

- We submit to the AEC a validation set of images taken from the original signal dataset
- The AEC has never seen this set of images
- The AEC is expected to perform well on the signal validation set since it has been trained on the signal



Auto-Encoder Validation

- We submit to the AEC a validation set of images taken from the original background dataset
- The AEC has never seen this set of images
- The AEC is expected to perform poorly on the background validation set since it has been trained on the signal



Auto-Encoder
Error in Reconstruction

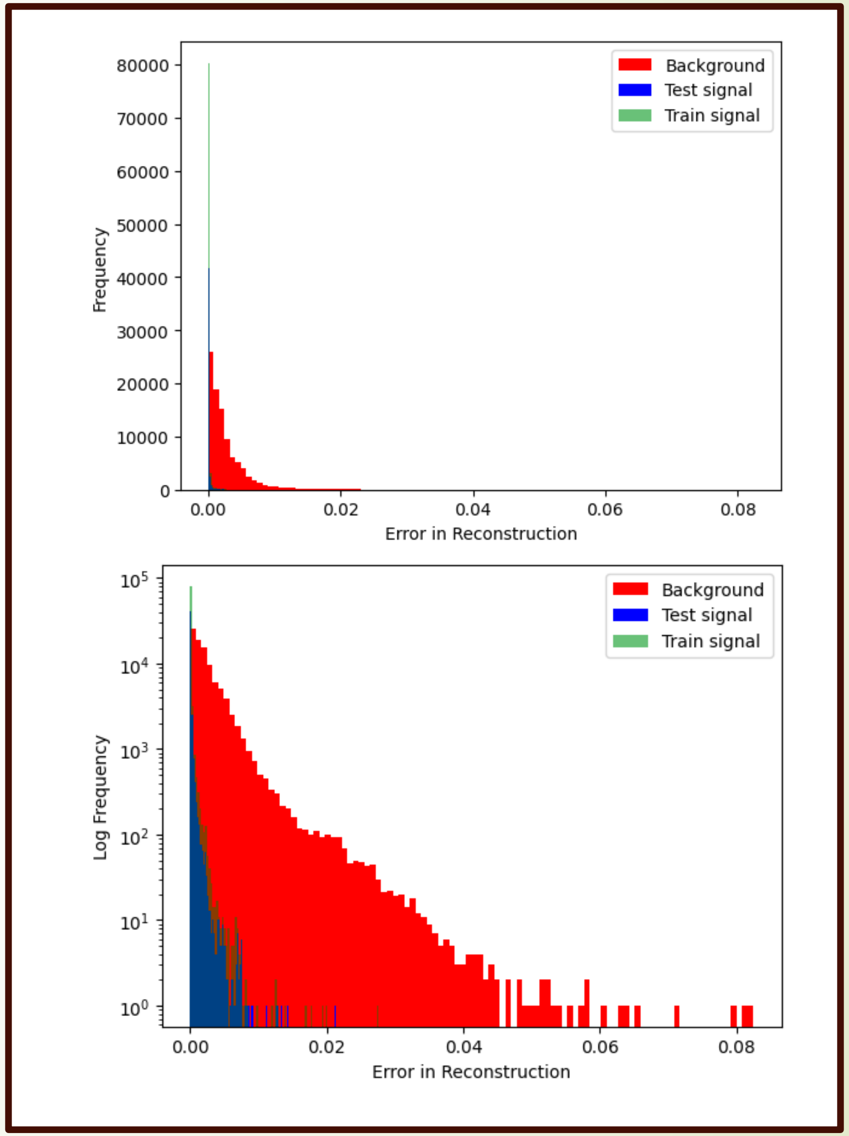
- We can evaluate the difference between an original and a reconstructed image, pixel by pixel:

$$E_1 = \frac{(R_1 - O_1)^2}{36}$$

$$\vdots$$

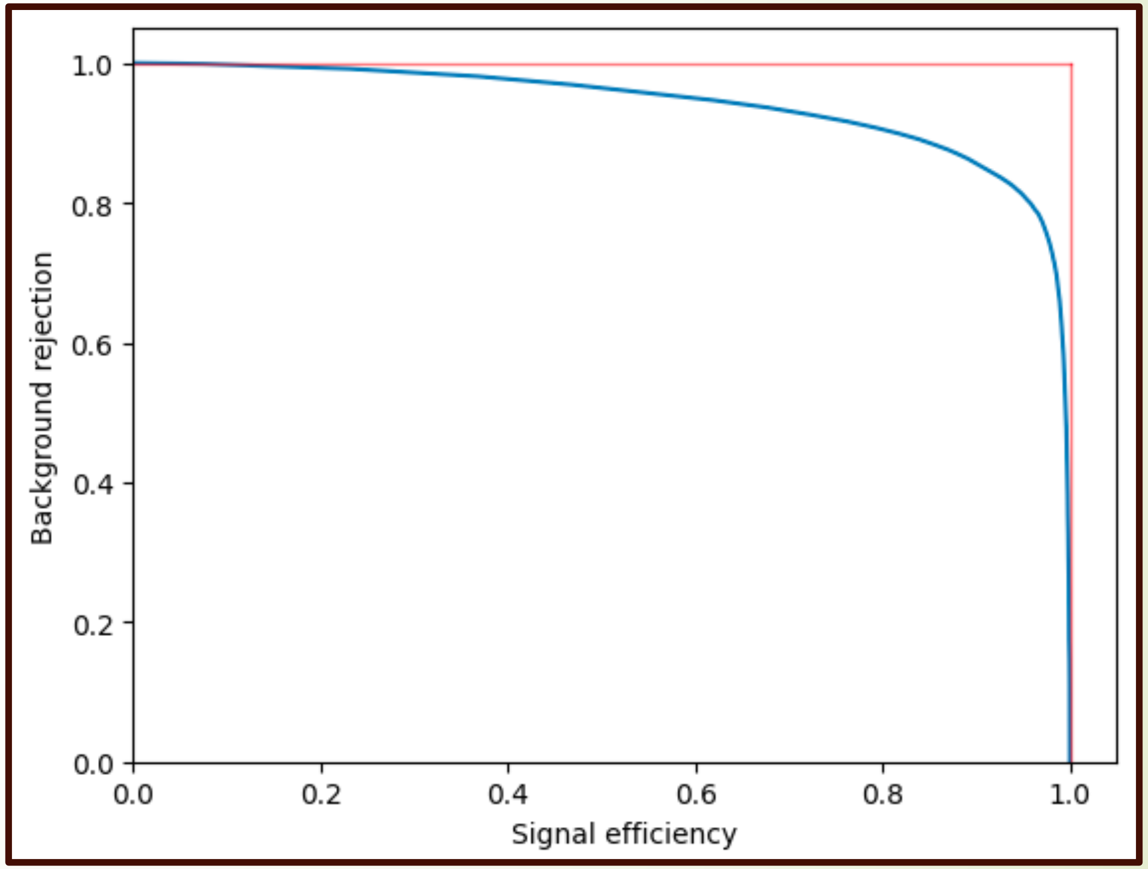
$$E_{36} = \frac{(R_{36} - O_{36})^2}{36}$$

- The total error will be the AEC selector



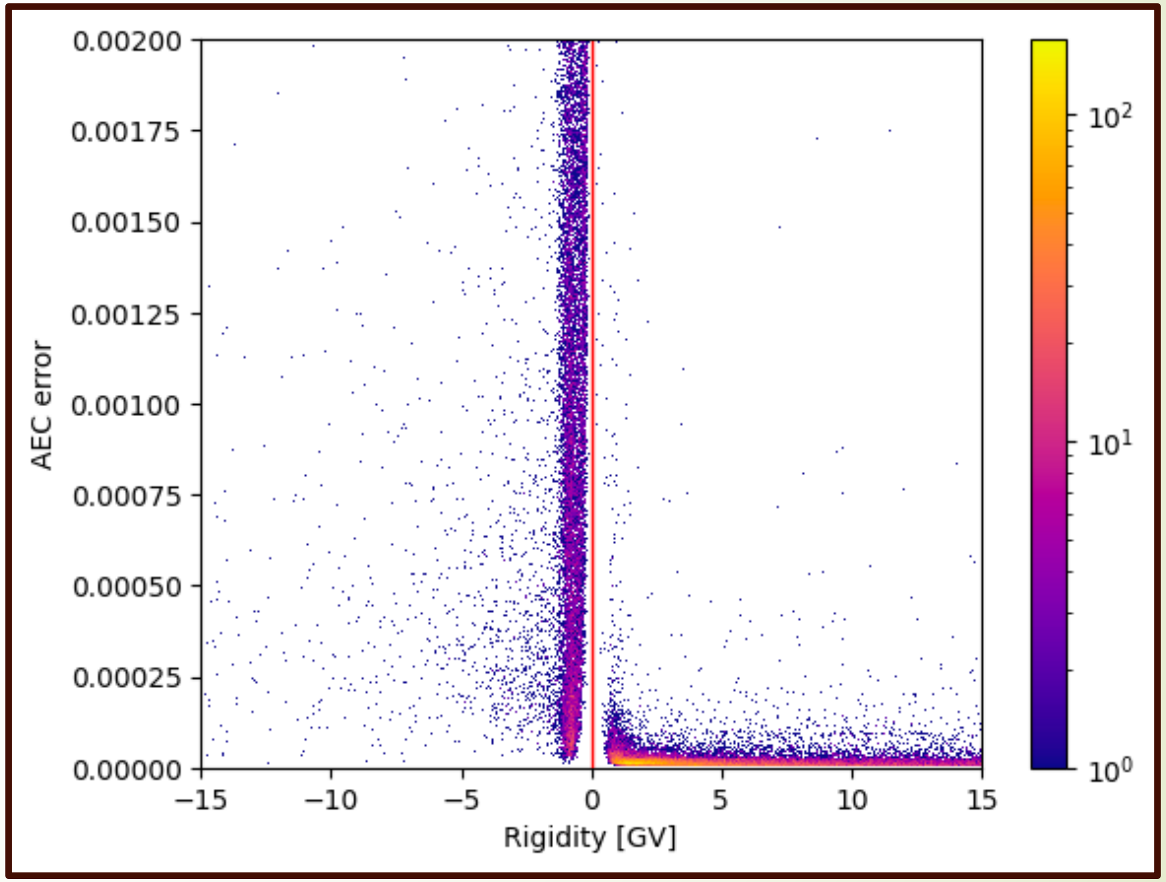
Auto-Encoder
ROC Curve

- Less separation power w.r.t. the BDT
- We are not really interested in the actual result
- We want to know if the AEC is "learning" something different from the data with respect to the BDT



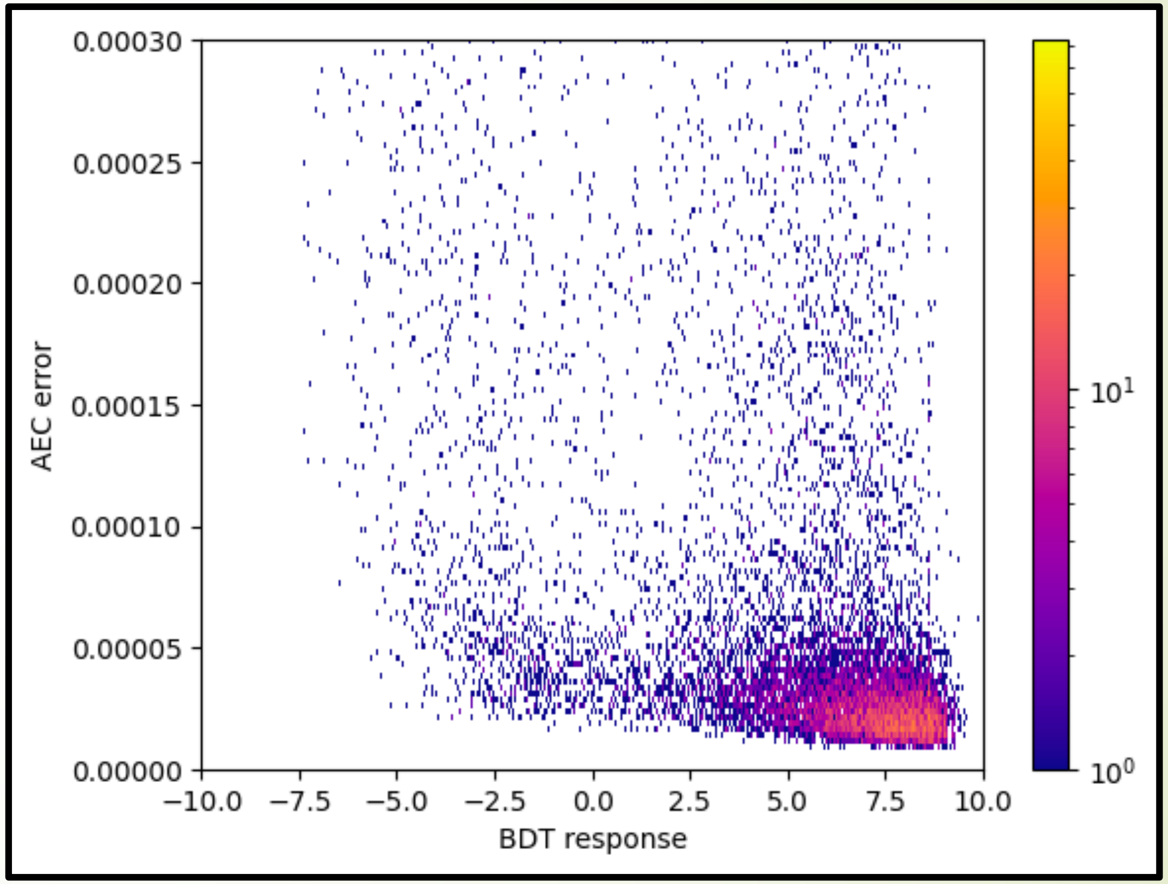
Auto-Encoder Error vs Rigidity

- The error on the signal is clearly concentrated on the lower values, as expected
- The error on the background is distributed on a large range of values
- A cut value is not evident



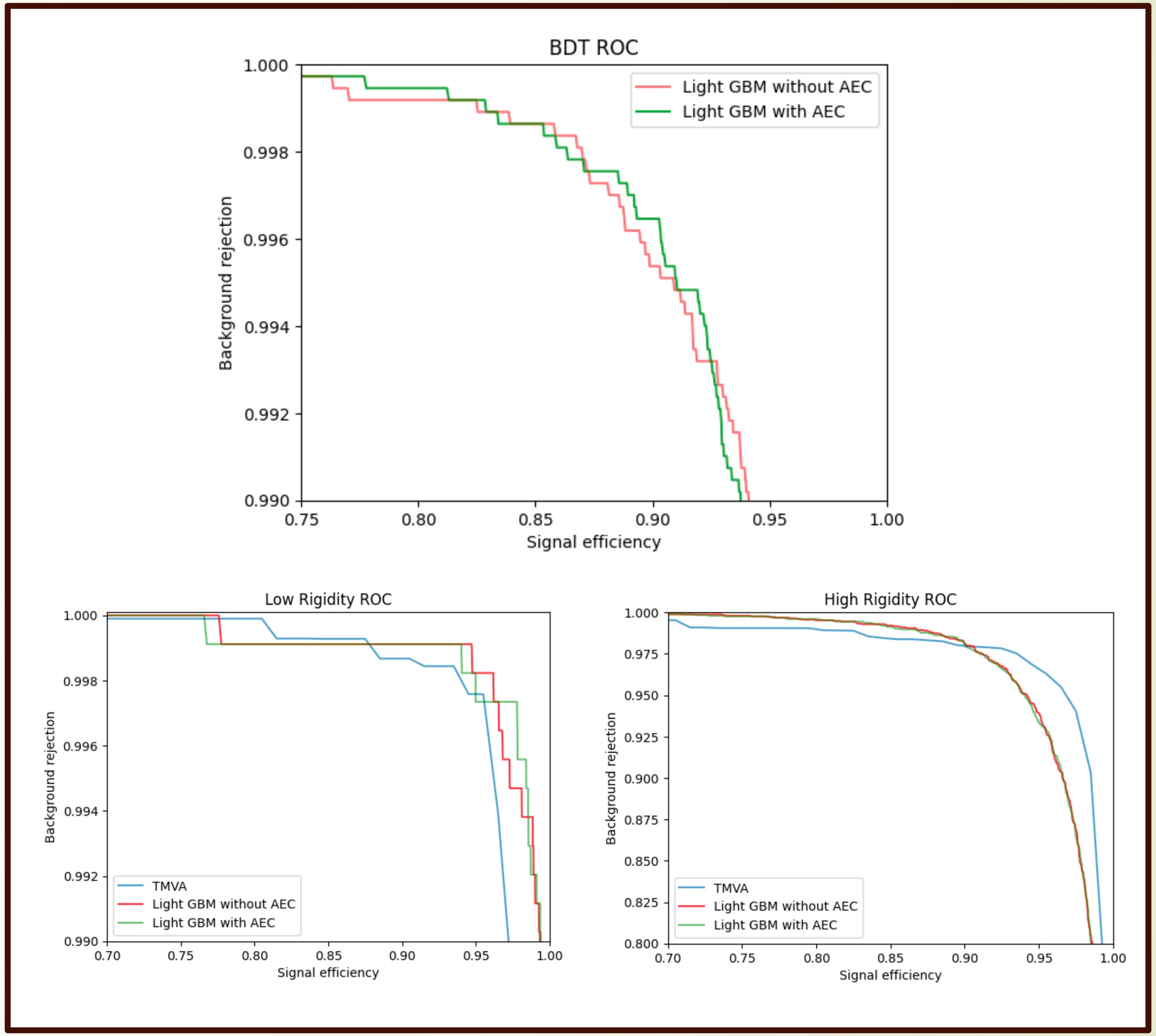
Auto-Encoder
AEC Error vs BDT Response

- The signal is characterised by a lower AEC error and a higher BDT response
- The background is characterised by a lower BDT response but no actual range for the AEC error
- The two selectors, the BDT response and the AEC error, are not tightly correlated



BDT combined with AEC ROC Curve

- The AEC error in reconstruction becomes an extra feature of each event
- A new Boost Decision Tree is trained once again, but now the events have an extra feature
- The final result for the separation power of the new BDT can be confronted with the previous result for the original BDT



Comparison TMVA vs LGBM

- At low Rigidity LGBM is more powerful

- At high Rigidity the results are similar with TMVA slightly better at high signal selection and LGBM slightly better at high background rejection