DRS4 signal/noise classification algorithms

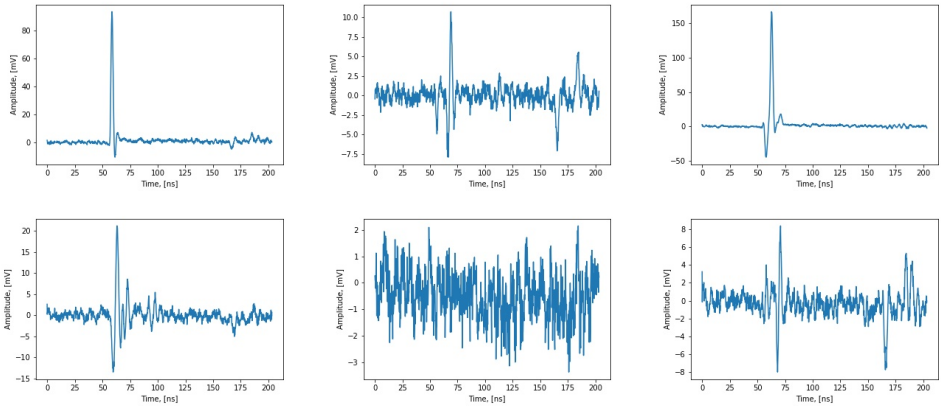
for the NA61/SHINE experiment at CERN

Aizat Daribayeva
BVLHEP, JINR
1 November, 2023

Outline

Motivation

Examples of DRS4 electronics response:



Distinguishing signals from noise can be tricky and challenging. A good signal/noise classification algorithm should be developed due to its significant impact on detector performance assessment.

Why do we need a reliable algorithm for signal/noise classification?

An existing methodology was based on the selection of the signal if there was a significant rise in the amplitude of the signal. Not reliable, since for example, cross talks would be considered as signals.

FFT based algorithm

The algorithm based on *Fast Fourier Transform* was developed for this important task [Credit: Sasha] and already was used to analyze DRS4 data collected during the tests of mRPC with the positron beam at "PAKHRA" accelerator (LPI RAS Troitsk). Details and test beam results can be found here:
V.Babkin et al., Beam test results of the MRPC prototype for the new NA61/SHINE ToF system, NIM A, Vol. 1034

FFT based algorithm

This algorithm is based on the approach that there should be some signal waveform that will be considered as reference one, and the convolution procedure would be applied between this reference signal and the real signal. At this point, the waveform of the real signal wouldn't be taken into account. The resulting signal would be classified as either signal or not.

Decision Tree algorithm

As an alternative to FFT, one of the ML algorithms - the *Decision Tree* was chosen and implemented. The main criteria for this algorithm:

- better/same performance as FFT
- robust to unseen data
- way too fast

Decision Tree algorithm

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes, and leaf nodes.

From ML perspective, signal/noise classification is a *binary classification* problem!

Binary classification

is the classification of the elements of a set into two groups (each called class) on the basis of a classification rule.

- *Dataset preparation* -> Troitsk beam test data was used as the dataset. Single data sample -> amplitude values corresponding to DRS4 time bins(1024)!

  [0.717486 -108.240036 1.052743 ... -0.100482 -0.576893 0.204639]

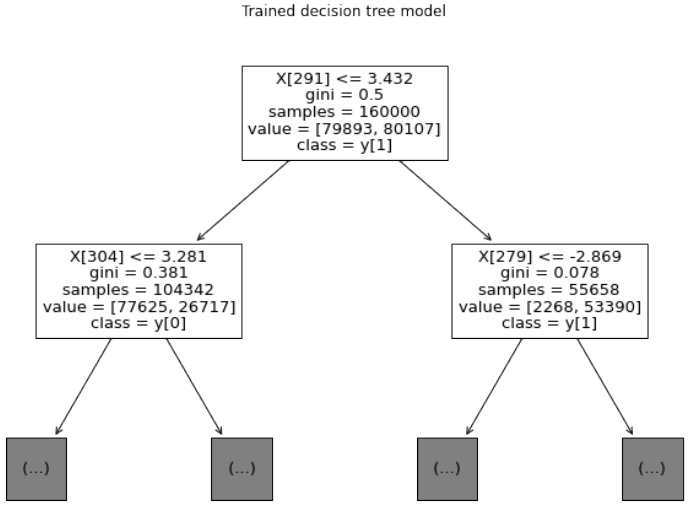
  200000 data samples were randomly chosen among run files: **6**, **7**, **9**, **14**, **15**, **21**, **25**, **30**, **41**.

- *Prior classification* -> Prior to training the model, FFT + data mapping was applied to dataset in order to distinguish signal and noise samples, i.e. labeling them as two different classes.

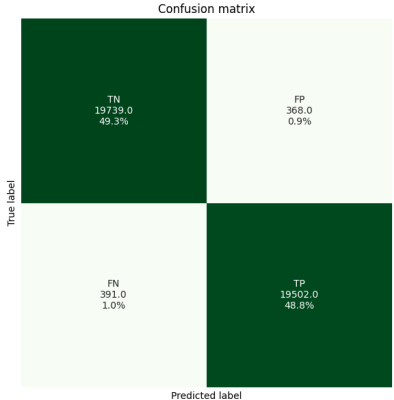
Data mapping

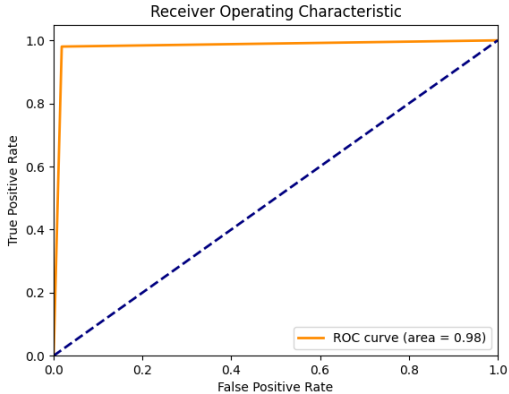
is the correspondence of the electronics channel to a specific strip and side of a specific detector.

- *Training and test the model* - Decision tree algorithm from mlpack package was trained using 80% of the dataset. The remaining 20% is used to test the algorithm's performance.


Trained decision tree model
X[291] <= 3.432
gini = 0.5
samples = 160000
value = [79893, 80107]
class = y[1]
X[304] <= 3.281
gini = 0.381
samples = 104342
value = [77625, 26717]
class = y[0]
X[279] <= -2.869
gini = 0.078
samples = 55658
value = [2268, 53390]
class = y[1]
(...)
(...)
(...)
(...)

Model's performance on the test data:





DRS4 signal/noise classification
Comparison of results

Run files: **6**, **7**, **9**, **14**, **15**, **21**, **25**, **30**, **41** were used for dataset. The robustness of the model and its performance were tested on the following files:

Numerical results on unseen data				
run file №	time resolution (FFT, ps)	time resolution (DT, ps)	efficiency (FFT, %)	efficiency (DT, %)
10	51.2	51.5	99.5	99.2
11	51.5	52.0	99.3	99.4
16	51.8	51.2	99.6	99.4
17	51.6	51.7	99.8	99.6
18	52.2	52.1	99.6	99.5
37	47.3	47.2	99.8	99.6
38	48.8	49.0	99.7	99.5
39	51.1	51.4	99.7	99.4

Comparison of elapsed time for both algorithms, tested over 400k data samples:

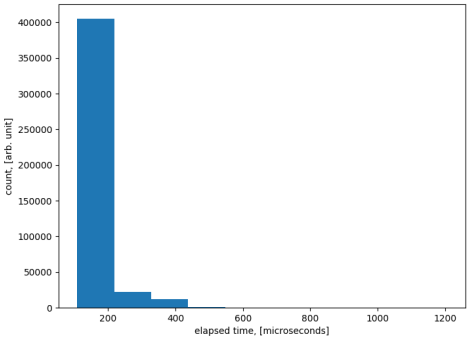


Figure: elapsed time, FFT

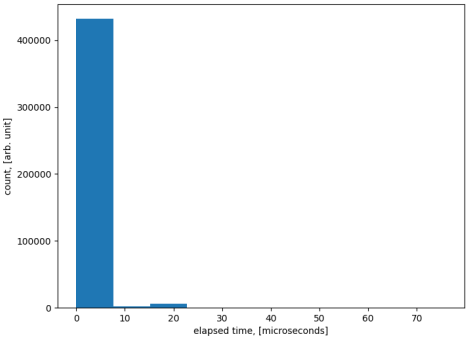


Figure: elapsed time, Decision Tree

Elapsed time was measured with two STL libraries: chrono::high_resolution_clock and ctime:clock_monotonic.

Closer look at time distributions:

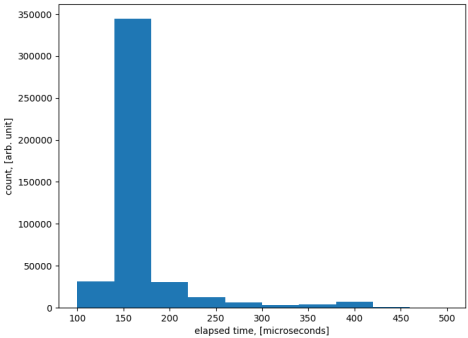


Figure: elapsed time, FFT

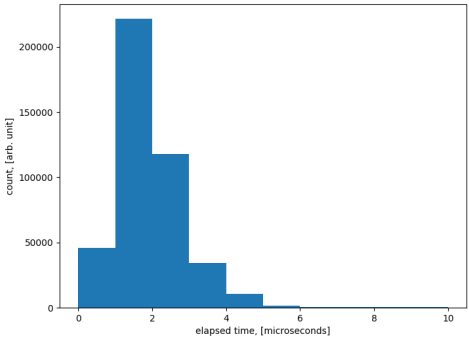


Figure: elapsed time, Decision Tree

Summary I

Two algorithms were investigated for DRS4 signal/noise classification:

- The FFT-based algorithm is well-established and already has shown good results, please see the provided article.
- The Decision Tree algorithm has shown competitive results in terms of performance and is much faster in terms of run time,
- DT model is planned to be implemented to pteroDAQtyl and tested -> possibility to reduce the data occupancy of mRPC (56%)!!!
- mlpack is lightweight, header only, cross-platform, fast C++ ML library -> seamless interoperability between C-style arrays and mlpack data containers -> just a few lines of codes!

Thank you!