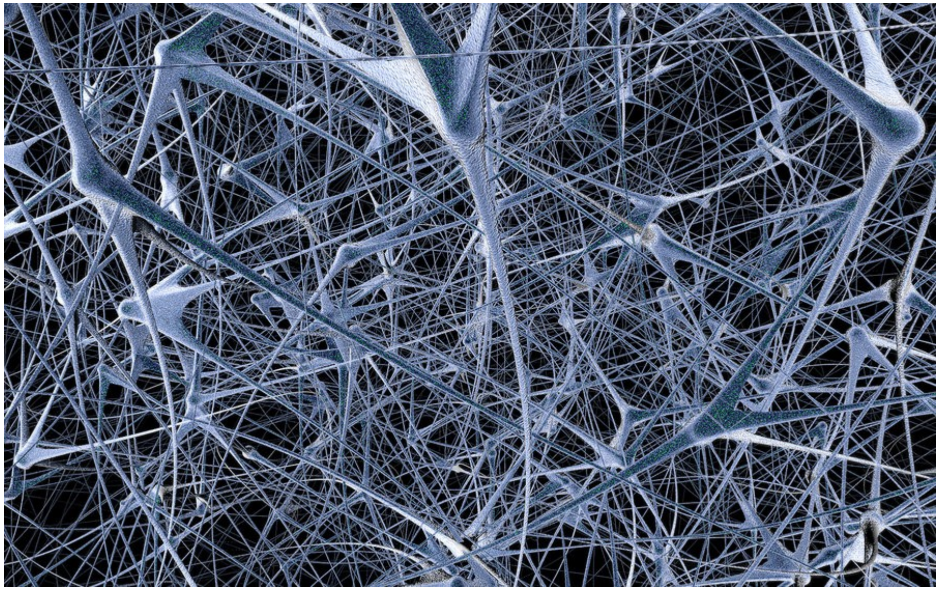
Using neural networks for gamma/neutron discrimination on NEDA data

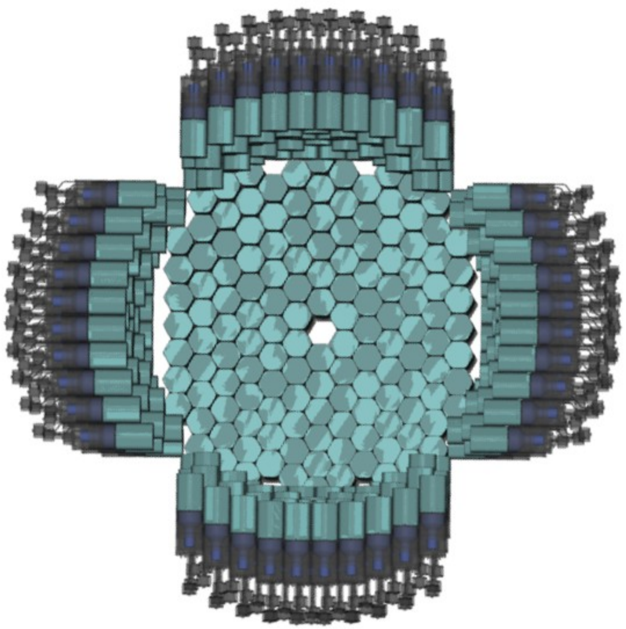


1. Context : NEutron Detector Array
2. Neural Networks
3. Results

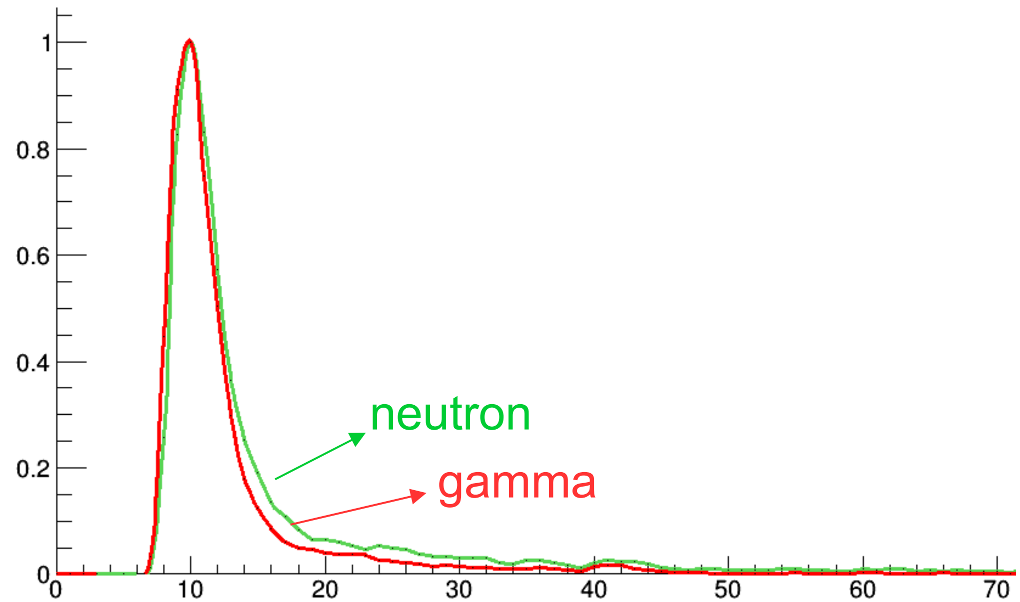
G. Baulieu, L. Ducroux, J. Dudouet, X. Fabian, O. Stezowski

IPN Lyon

→ **The NEutron Detector Array (NEDA)**

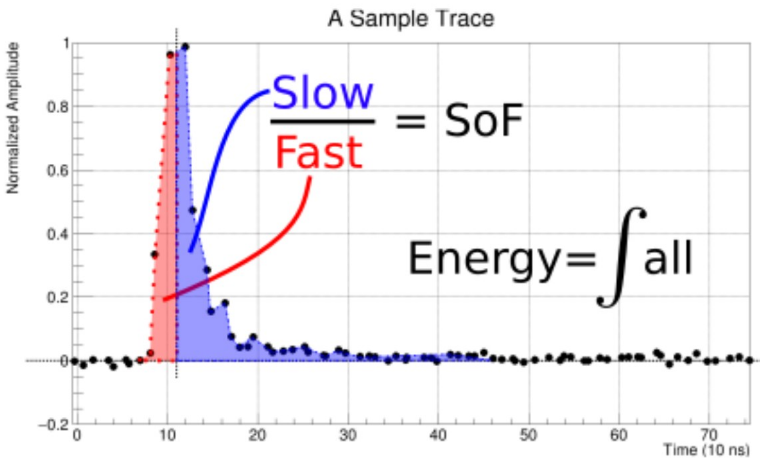


- **Neutron detector**
- First campaign in 2018 at GANIL :
  - **ancillary of AGATA** along with DIAMANT (charged particles)
    → selection of events according to neutrons, protons and alpha rays
- Made of liquid organic **scintillators**
- Reacts to neutrons and gamma rays



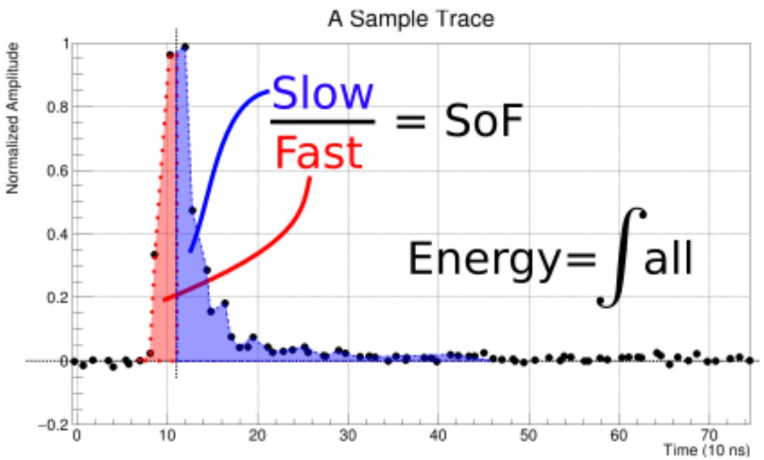
Target : to be able to distinguish neutrons and gammas by Pulse Shape Analysis

→ **NEDA PSA**

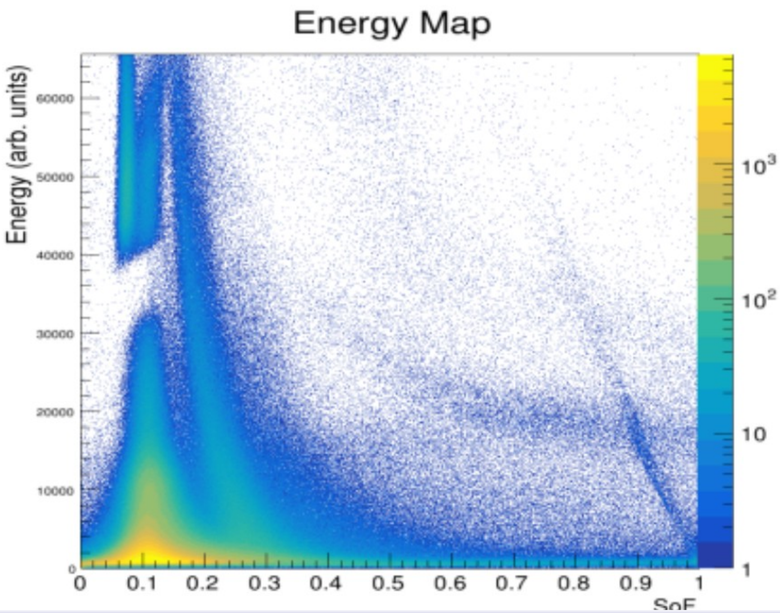


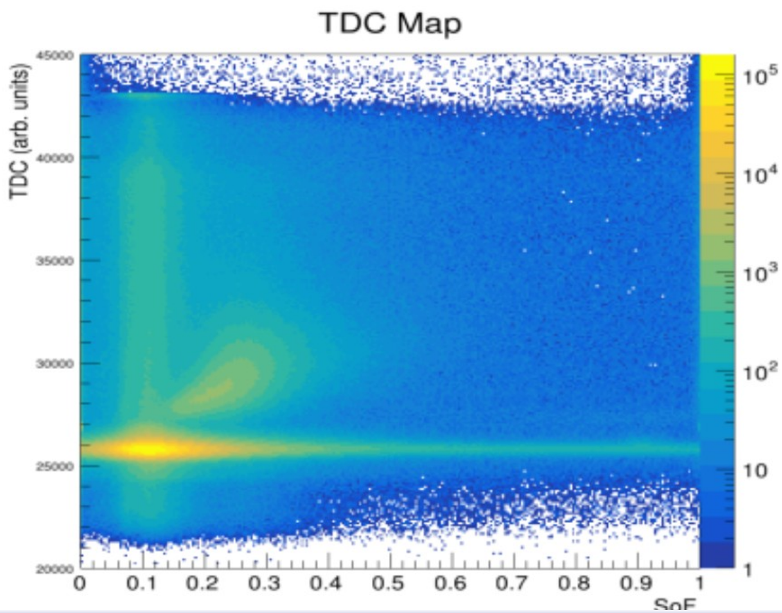
- Comparison of fast and slow components

→ **NEDA PSA**

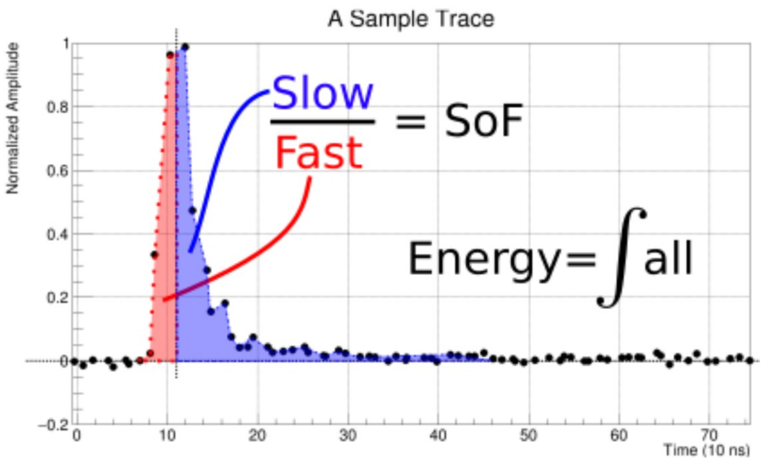


- Comparison of fast and slow components

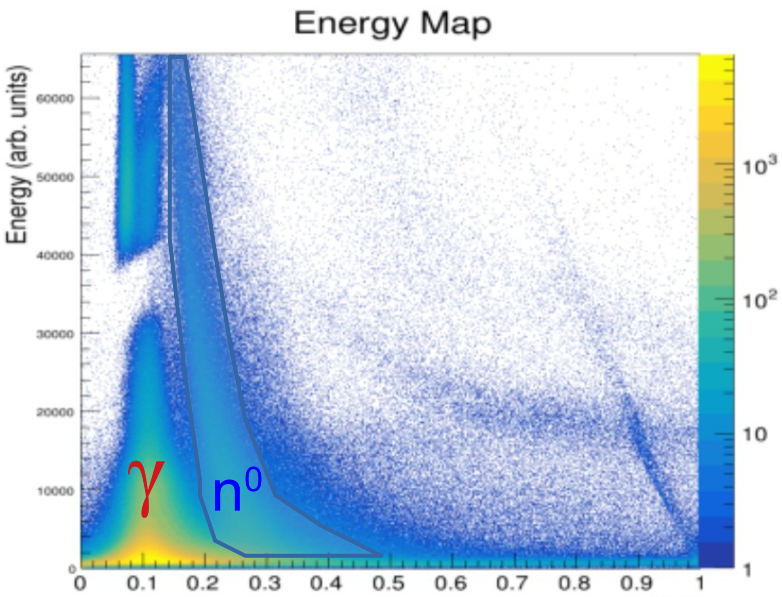


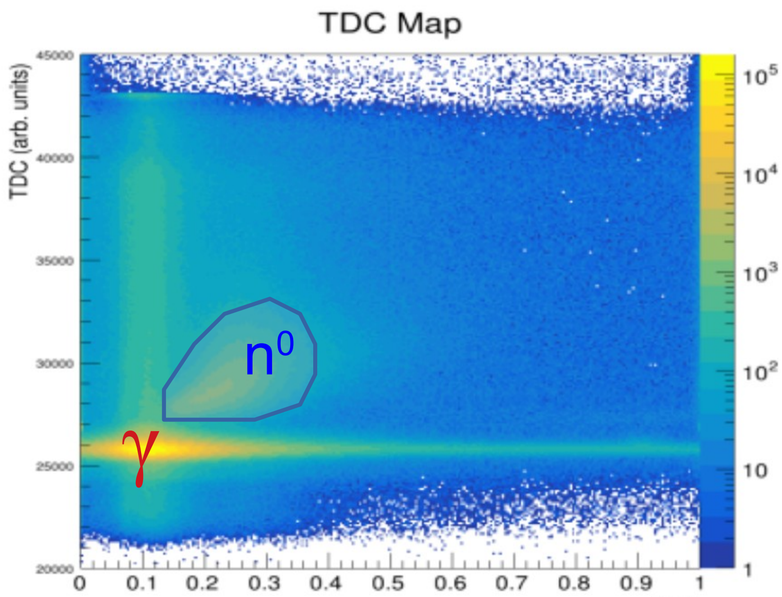


→ **NEDA PSA**



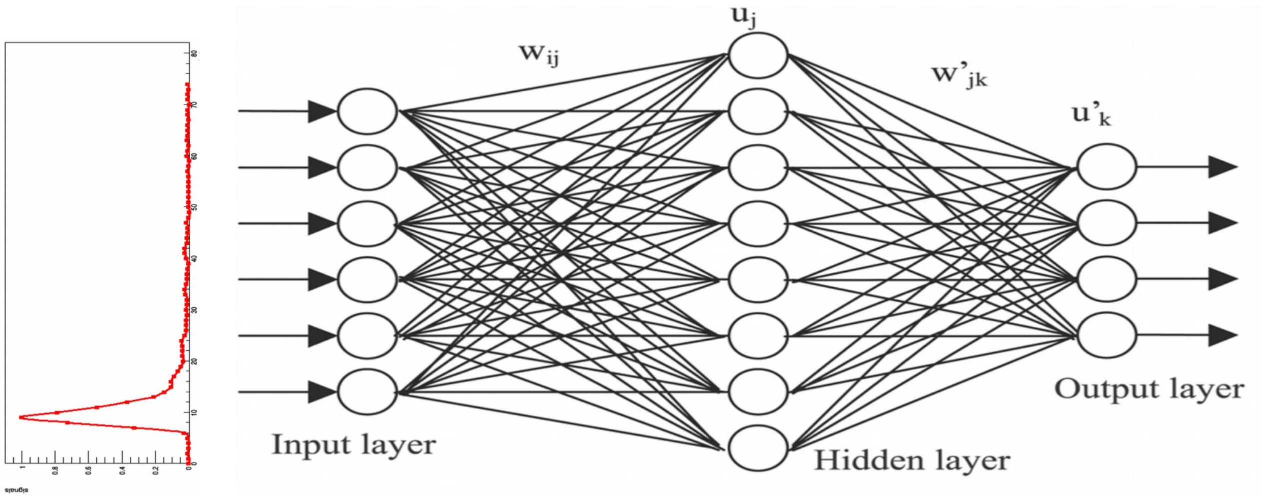
- Comparison of fast and slow components
- Cuts for final selection





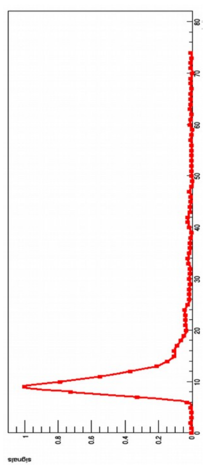
→ **Using neural networks for PSA**

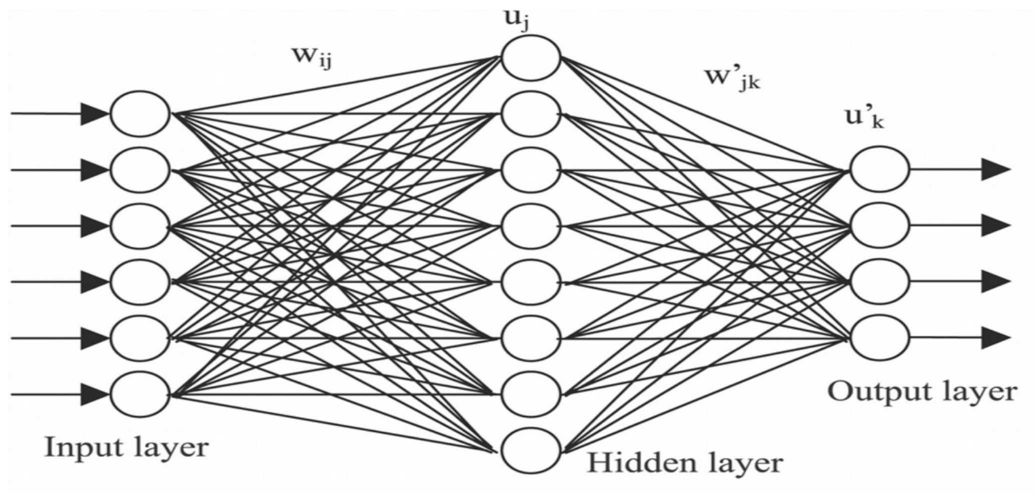
First studies by P-A Söderström et al using a Multi Layers Perceptron in Root



→ Using neural networks for PSA

First studies by P-A Söderström et al using a Multi Layers Perceptron in Root

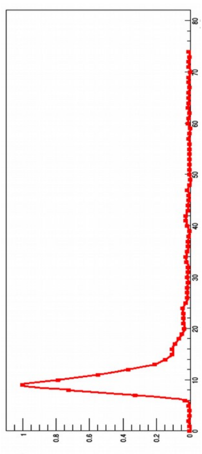


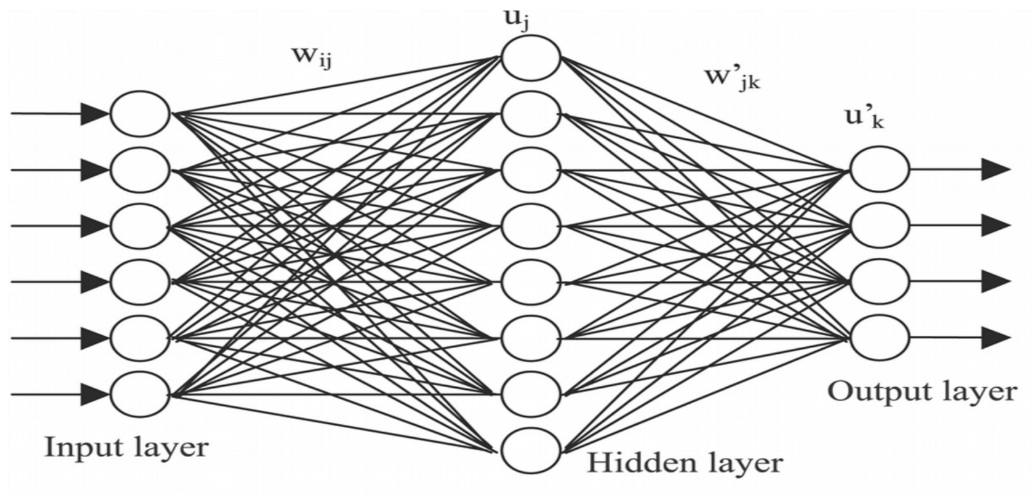


Interesting results but too slow → Migration to TensorFlow

→ Using neural networks for PSA

First studies by P-A Söderström et al using a Multi Layers Perceptron in Root





Interesting results but too slow → Migration to TensorFlow

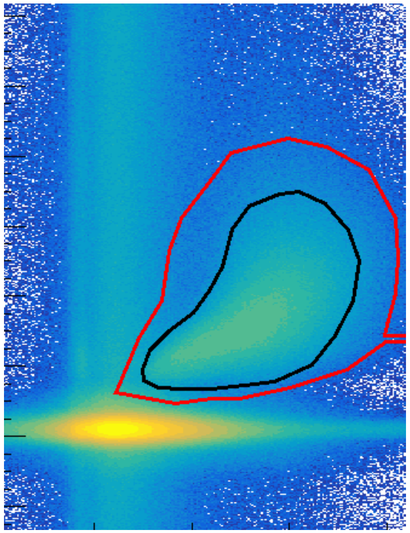
Training in Python → Freeze the model → Load the model for inference in C++ (Ganpro)

→ **Creation of a training set**

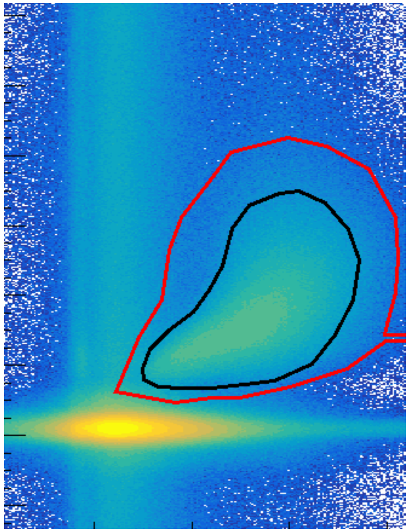
- No full simulation → need to use real data

→ **Creation of a training set**

- No full simulation → need to use real data
- Use very conservative cuts and let some areas as unknown



→ **Creation of a training set**

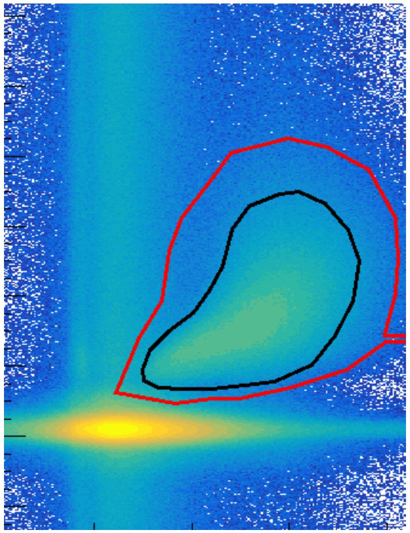


- No full simulation → need to use real data
- Use very conservative cuts and let some areas as unknown

→ **Tests on 2 network architectures**

- **Multi-Layers perceptron** : legacy, easy to setup and compute

→ **Creation of a training set**



- No full simulation → need to use real data
- Use very conservative cuts and let some areas as unknown

→ **Tests on 2 network architectures**

- **Multi-Layers perceptron** : legacy, easy to setup and compute
- **Recursive Neural Network** (Long Short-Term Memory) : interesting to analyze time-series...

→ **Setup on GPU Farm**

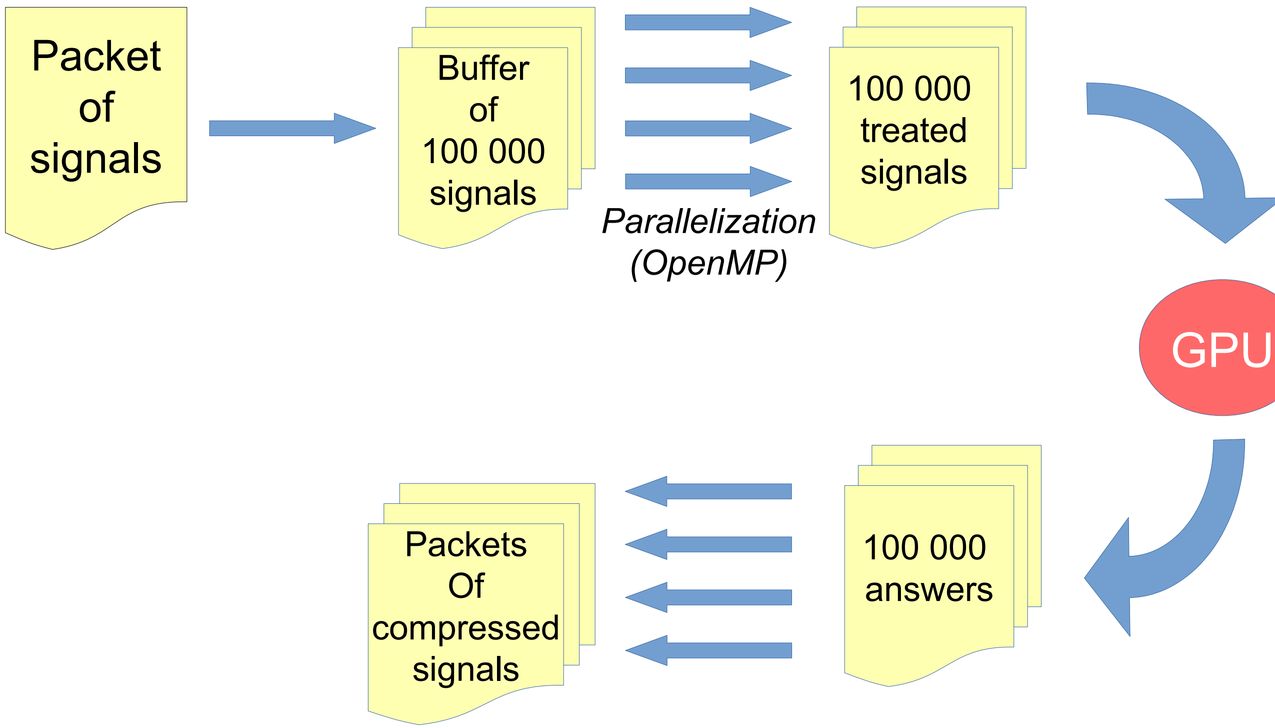
- Existing Docker image used in Gitlab-CI (Ubuntu 16.04)
  → Conversion to **Singularity**
- Compilation of **TensorFlow-GPU** (v1.11) on the cluster with **Cuda 9.2**
- Each job starts a **singularity container** before launching the computing process
- NEDA data are naturally split in 6 (number of acquisition cards) : **6 analysis jobs** per run

→ **GPU usage**

- Not efficient to send a single signal to the GPU → buffering of the signals

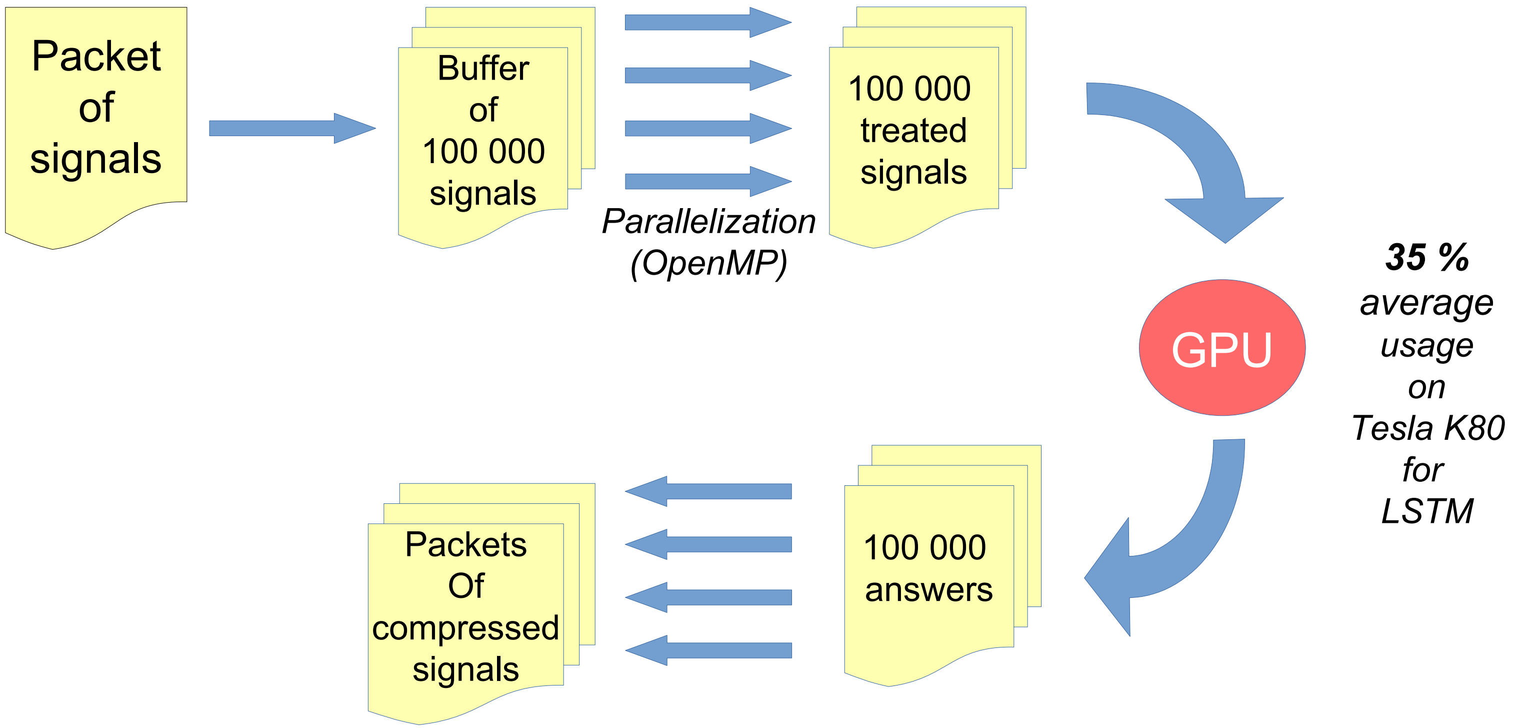
→ **GPU usage**

- Not efficient to send a single signal to the GPU → buffering of the signals

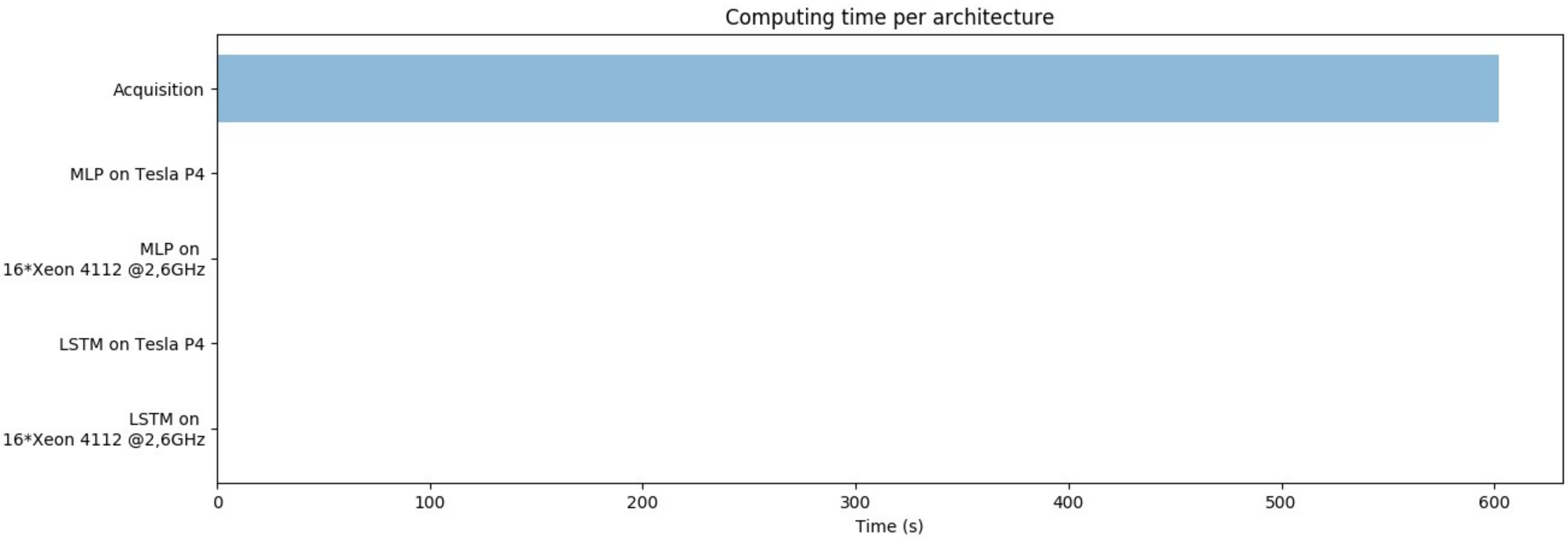


→ **GPU usage**

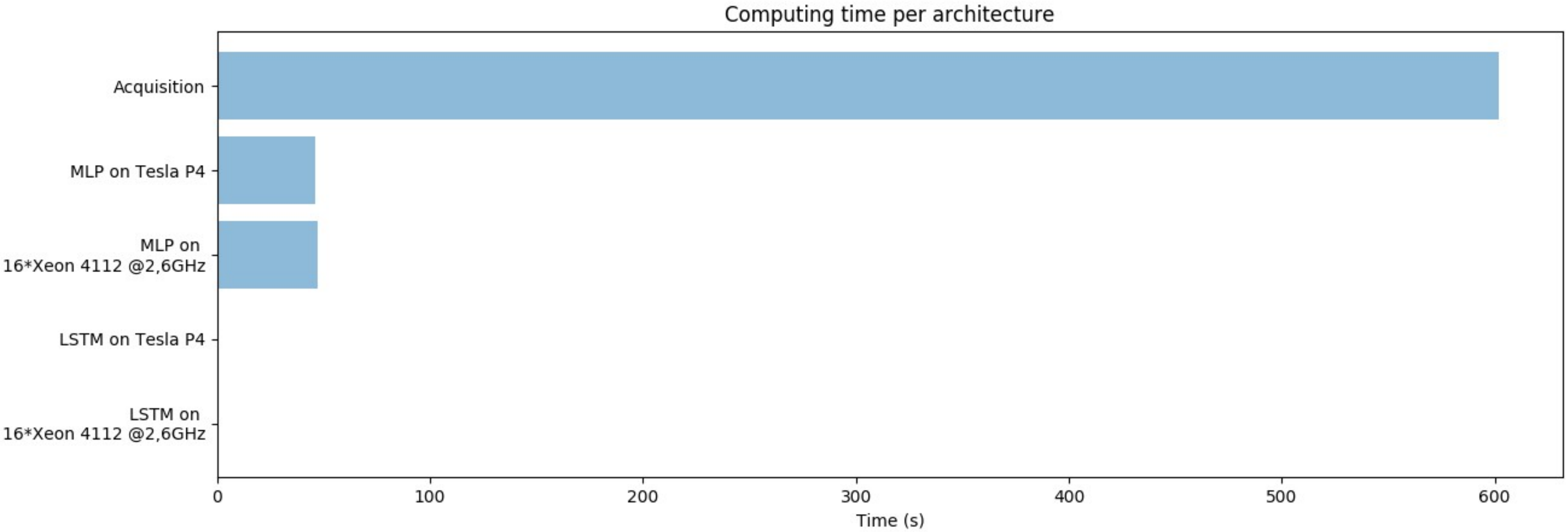
- Not efficient to send a single signal to the GPU → buffering of the signals



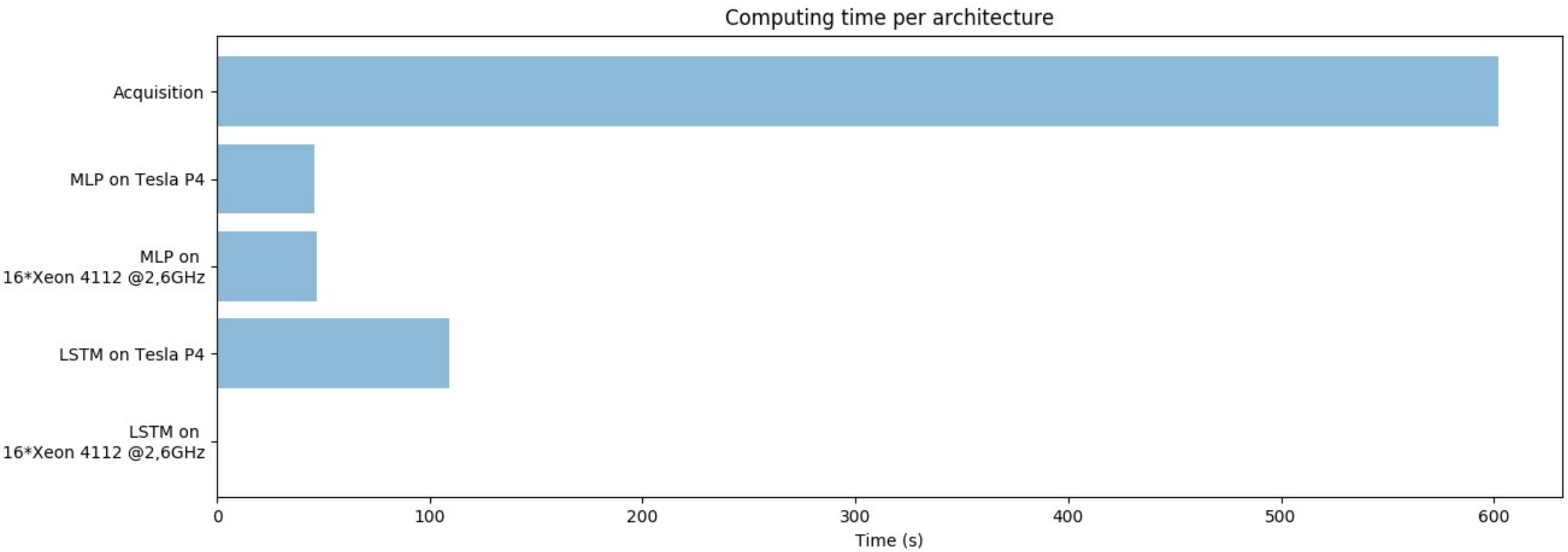
→ **Computing Time**



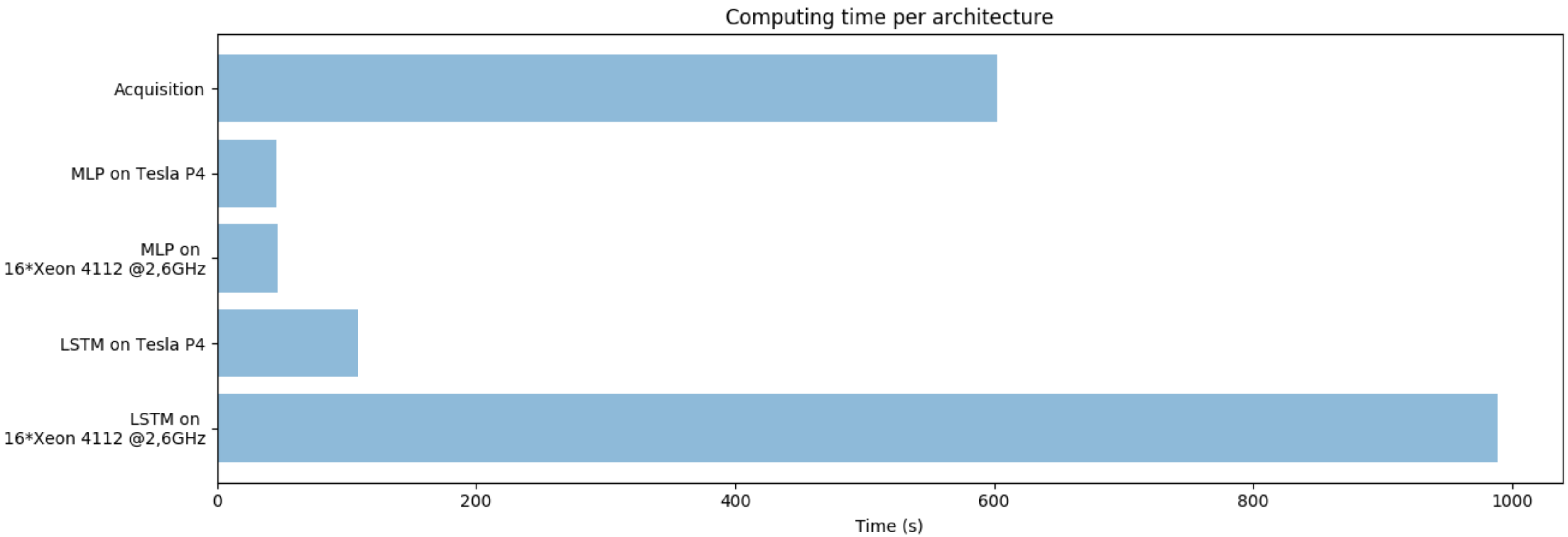
→ Computing Time



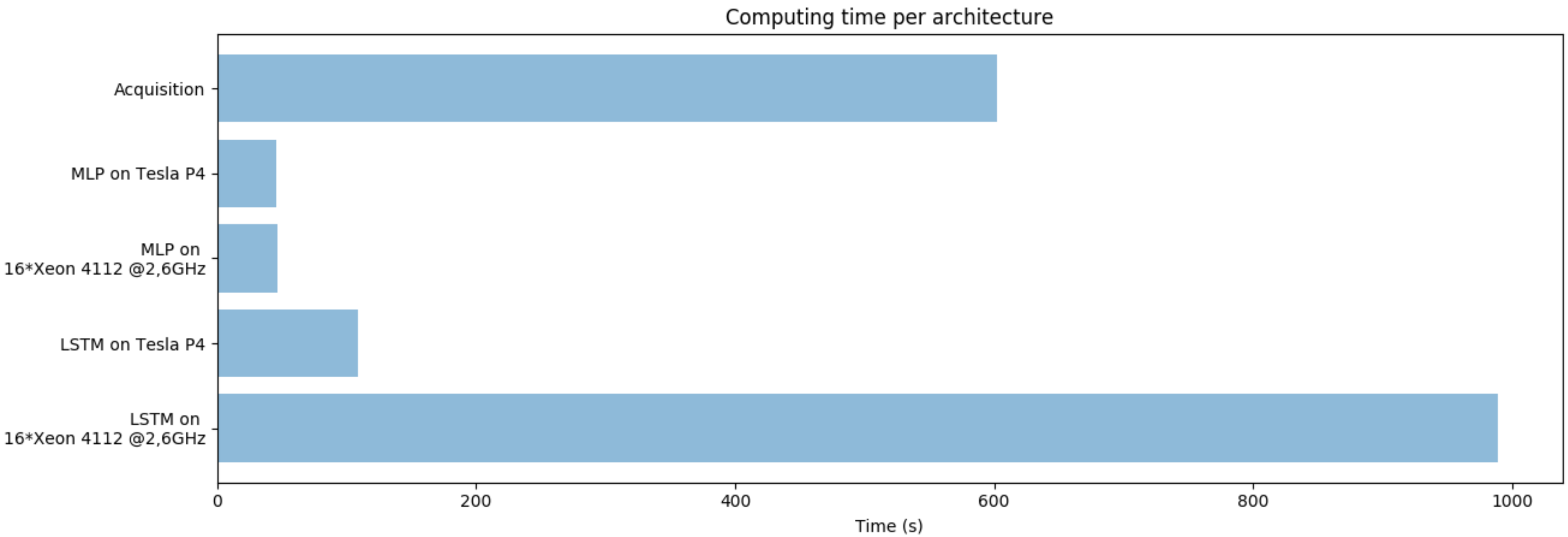
→ **Computing Time**



→ Computing Time



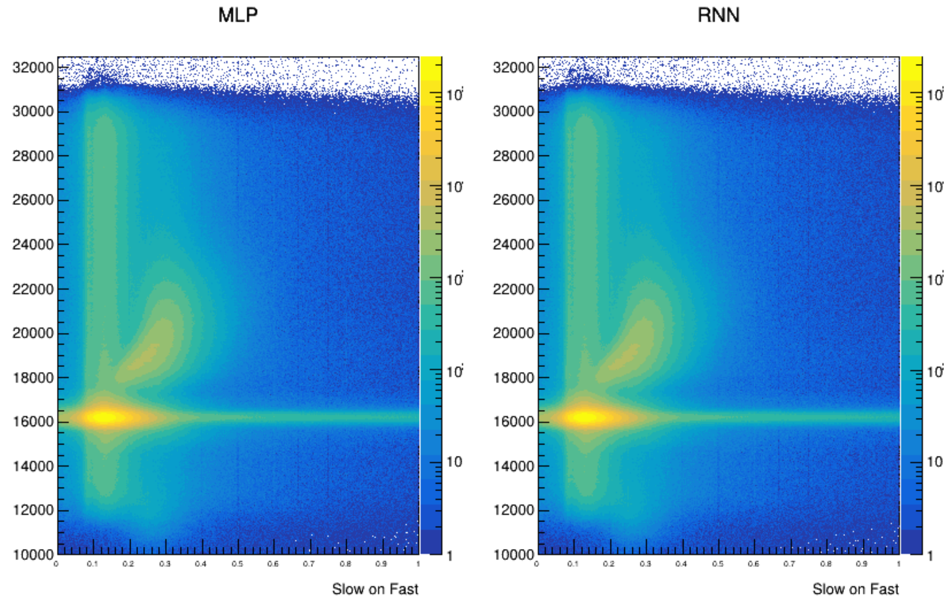
→ Computing Time



Analysis time for a 1 To run (6x15h) on GPU Farm (Tesla K80) : ~ 5 hours

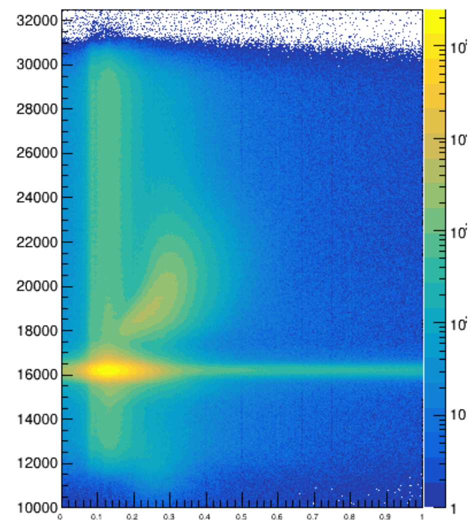
Results

→ Results on discrimination

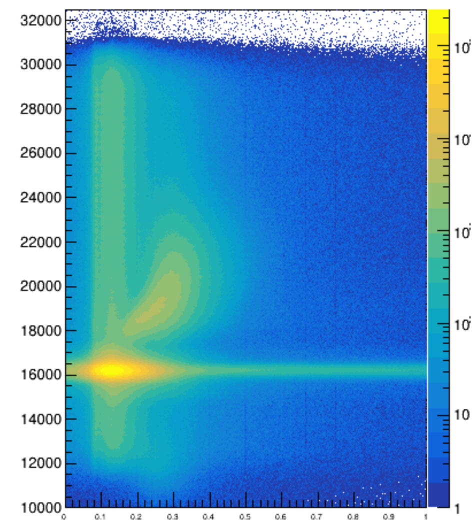


→ Results on discrimination

MLP

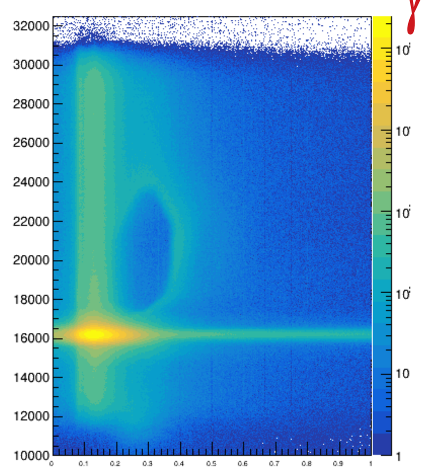


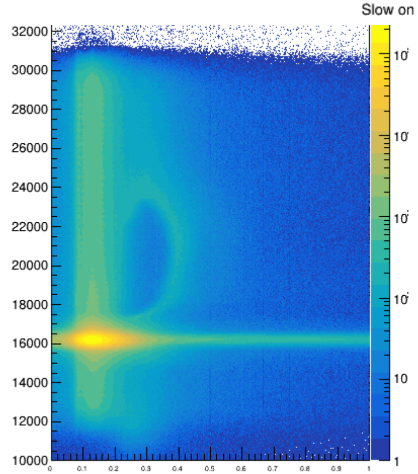
RNN



MLP

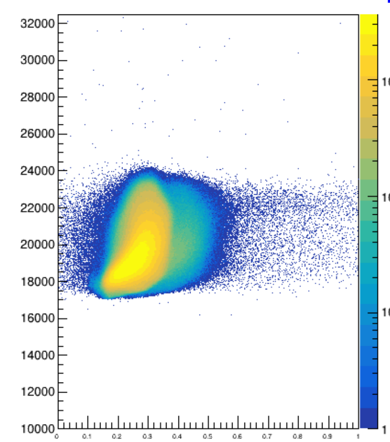
γ

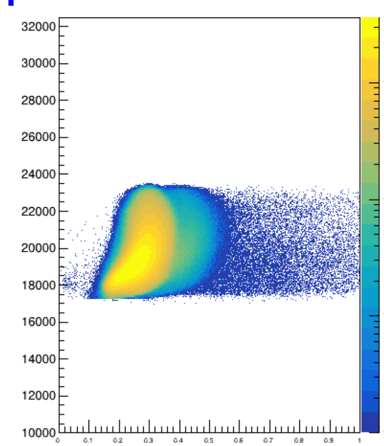




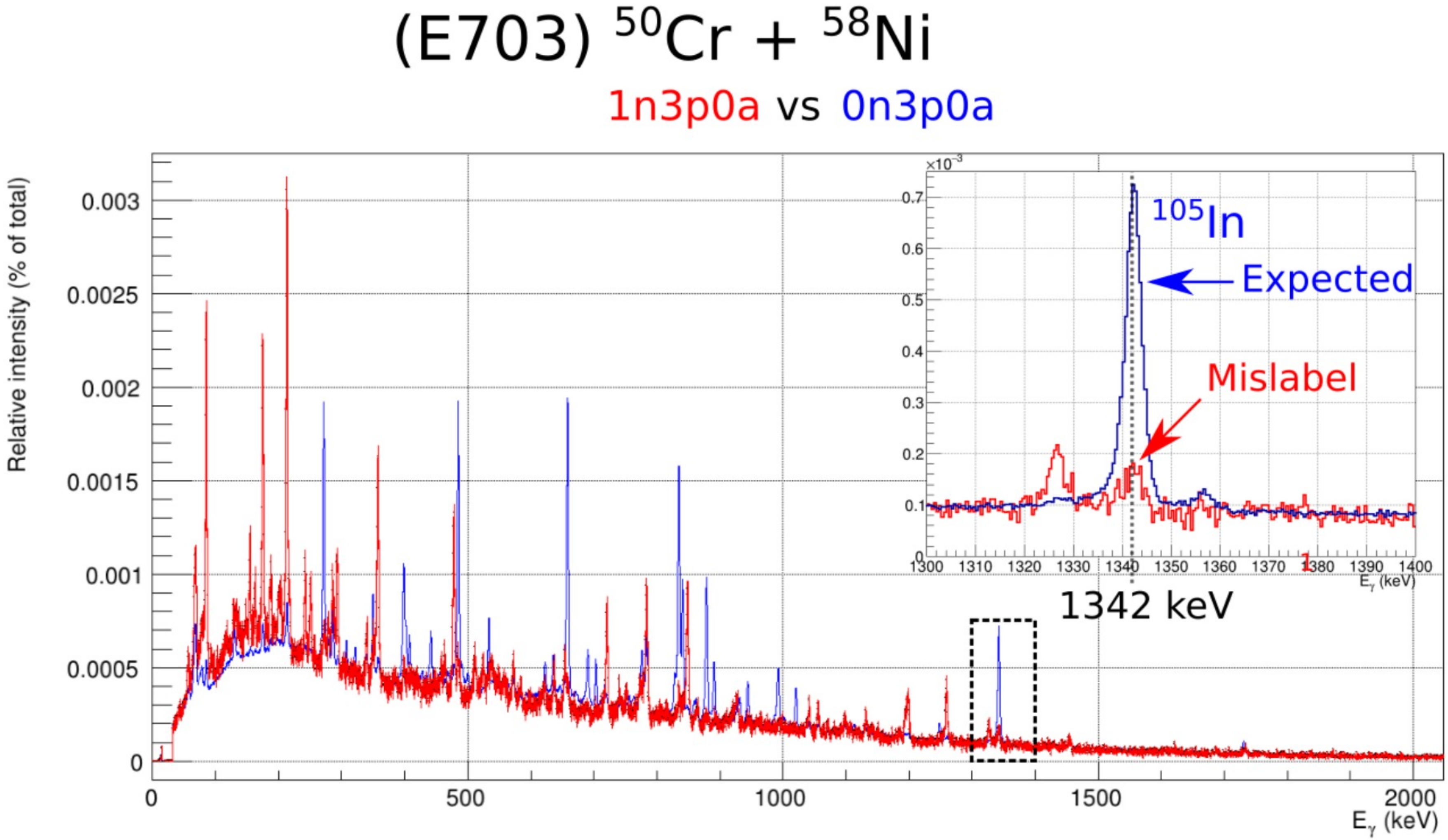
n^0

RNN

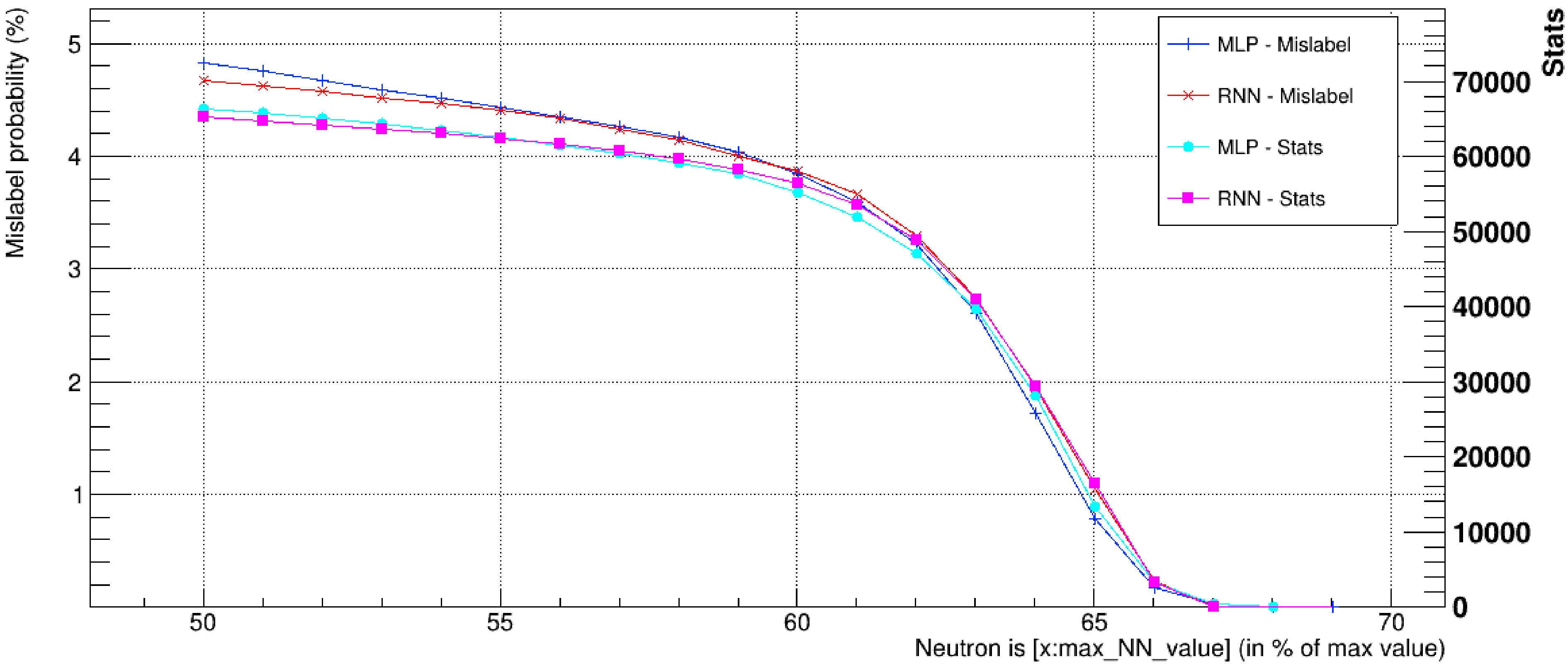




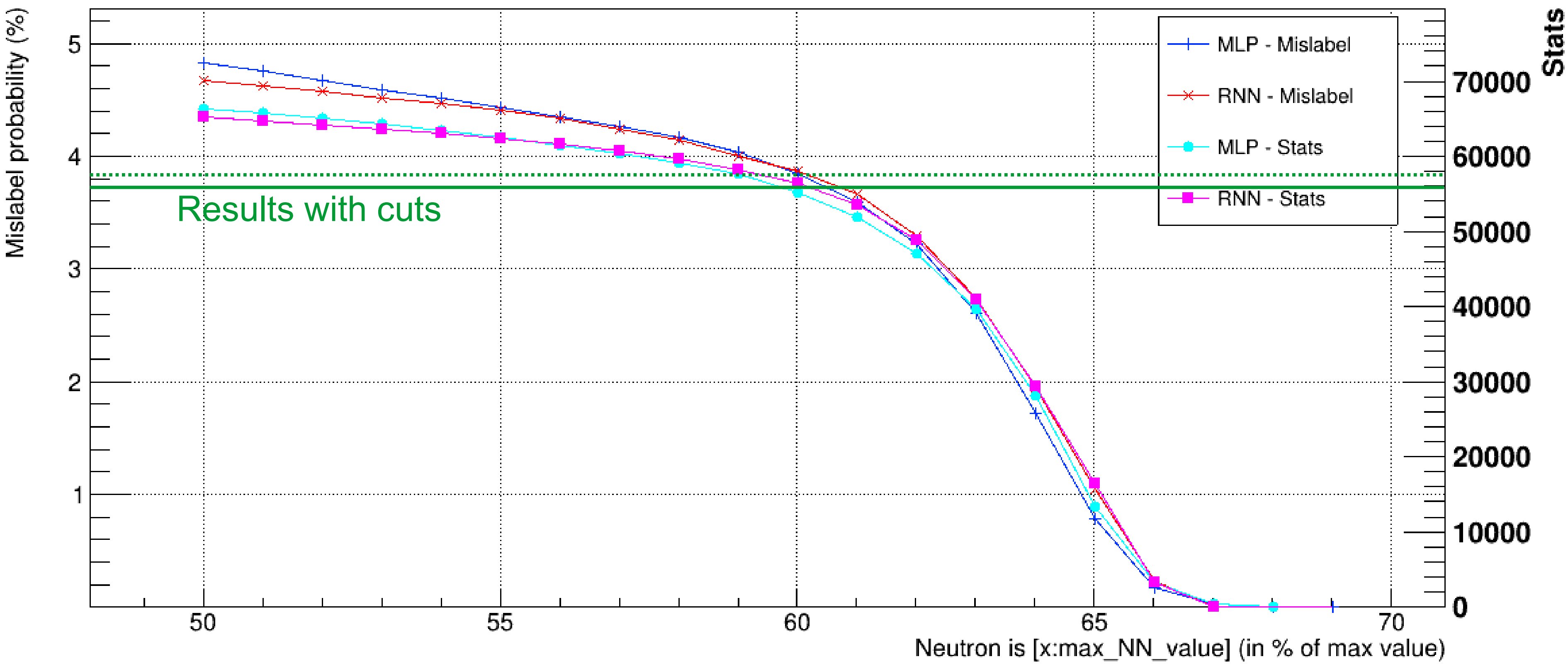
→ **Quantification of results using AGATA (1/2)**



→ **Quantification of results using AGATA (2/2)**

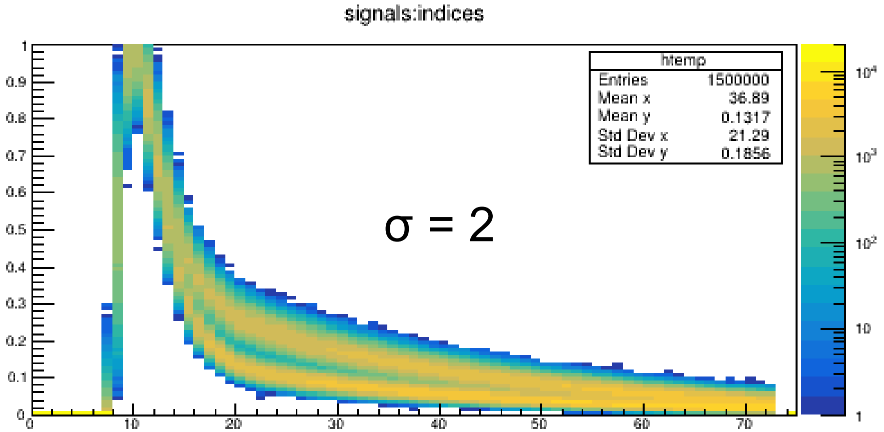


→ **Quantification of results using AGATA (2/2)**



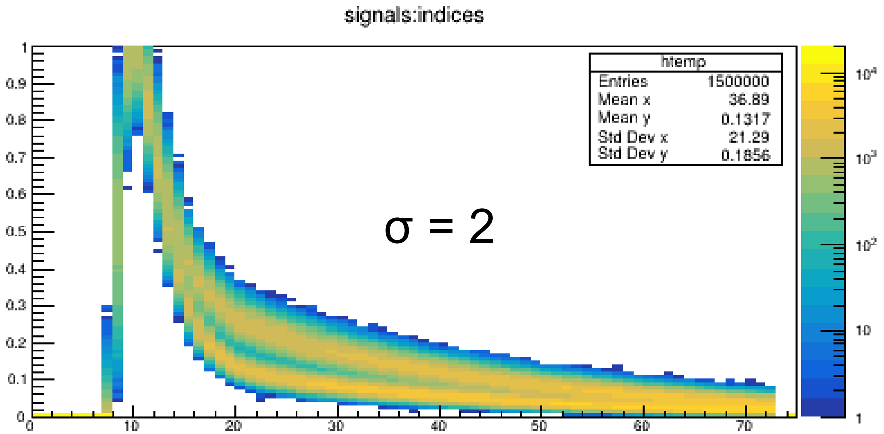
→ Robustness to desynchronization

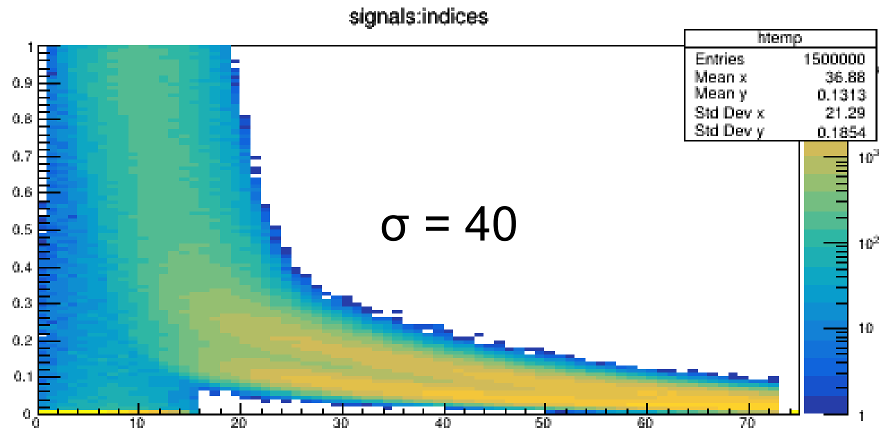
Simulation of signals with a gaussian T0 distribution



→ Robustness to desynchronization

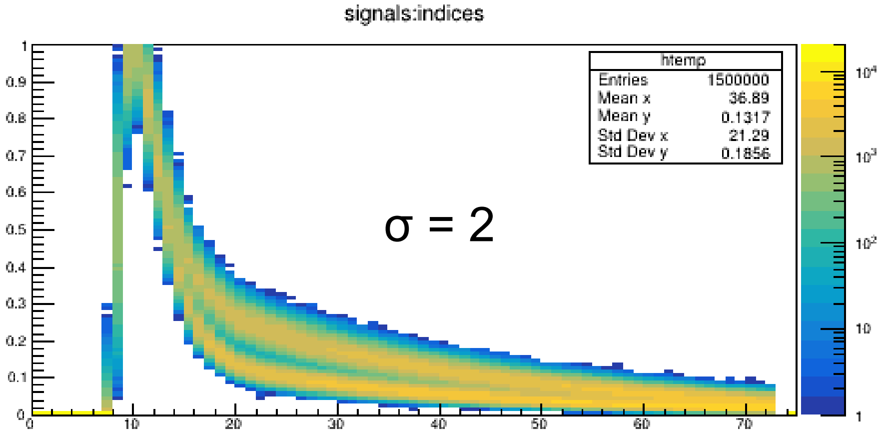
Simulation of signals with a gaussian T0 distribution

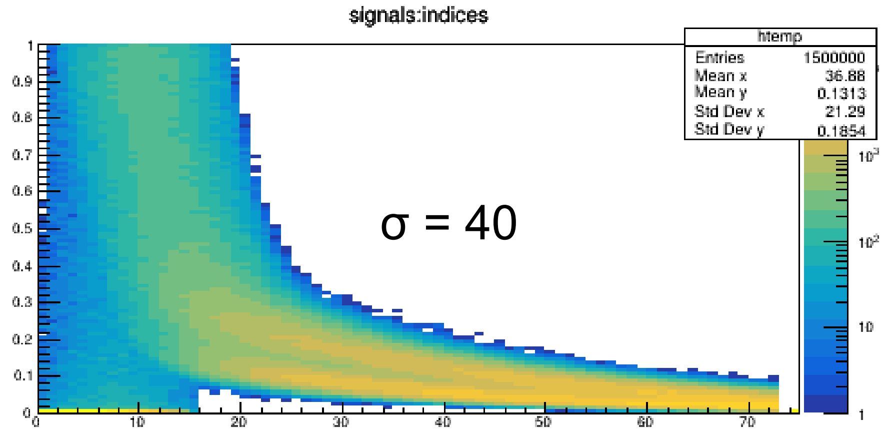


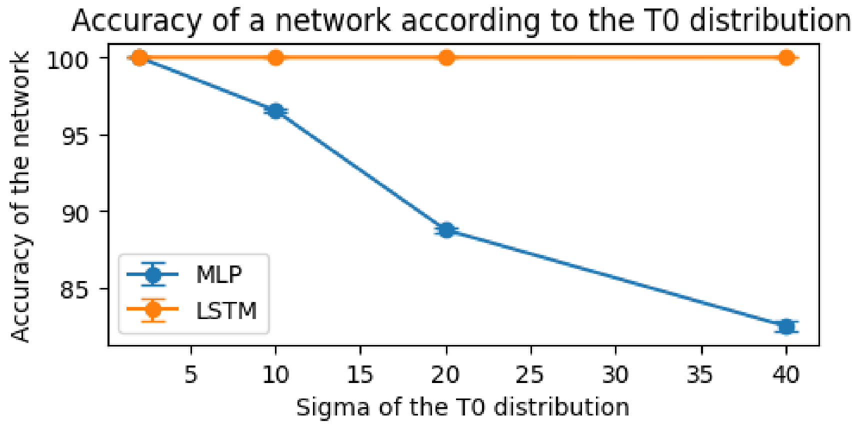


→ **Robustness to desynchronization**

Simulation of signals with a gaussian T0 distribution







LSTM is very robust to T0 shifts!

→ **Conclusion**

- Functional neural networks for gamma/neutron discrimination on NEDA data

→ **Conclusion**

- Functional neural networks for gamma/neutron discrimination on NEDA data
- Computing time compatible with online acquisition (GPU required for LSTM)

→ **Conclusion**

- Functional neural networks for gamma/neutron discrimination on NEDA data
- Computing time compatible with online acquisition (GPU required for LSTM)
- Faster offline analysis on the CC-IN2P3 GPU Farm (~5H for a big run)

→ **Conclusion**

- Functional neural networks for gamma/neutron discrimination on NEDA data
- Computing time compatible with online acquisition (GPU required for LSTM)
- Faster offline analysis on the CC-IN2P3 GPU Farm (~5H for a big run)
- LSTM robust to T0 shifts

→ **Conclusion**

- Functional neural networks for gamma/neutron discrimination on NEDA data
- Computing time compatible with online acquisition (GPU required for LSTM)
- Faster offline analysis on the CC-IN2P3 GPU Farm (~5H for a big run)
- LSTM robust to T0 shifts
- Network's output values usable for fine tuning of neutrons selection (better quality or more statistic)