Predictive Learning applied to Muon Chamber Monitoring

Data Quality Monitoring with Neural Networks

Candidate: Nicolò Lai

Supervisor: Prof. Marco Zanetti

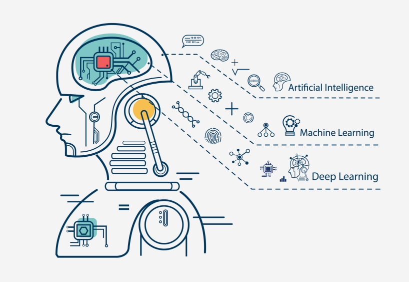
December 14, 2021

Università degli Studi di Padova

Università degli Studi di Padova

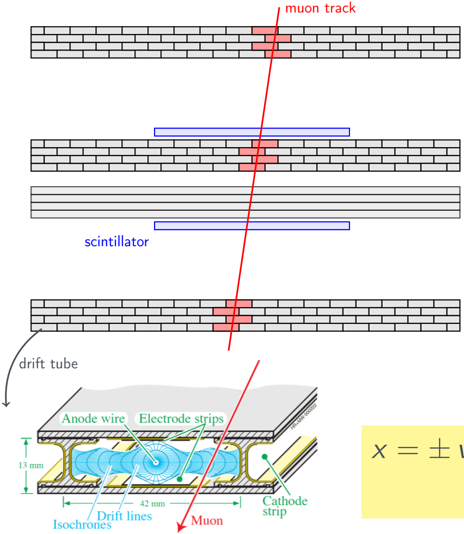
Employement of deep learning allows for

- Flexibility
- Accuracy
- Automation



The Detector

Experimental setup



- 4 muon chambers (*superlayers*)
- 4 staggered layers of 16 *drift tubes* each
- *trigger-less* data acquisition system
- 2 *scintillator* tiles

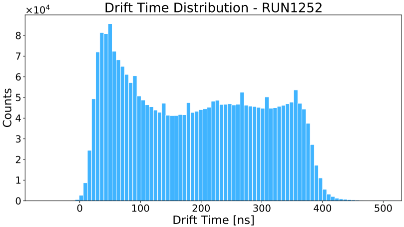
$$x = \pm v_{\mathrm{drift}} \underbrace{(t - t_0)}_{\text{drift time}}$$

Università degli Studi di Padova

Collection of the times that the electrons take to reach the anodic wire in the center of the cell

The external trigger

- allows to discriminate *muon hits* from background noise
- provides a *timing reference* t_0 for computing the drift time



Time box shape:

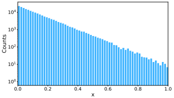
- approximately *uniform*
- $\sim$400 ns width

The Deep Learning Algorithm

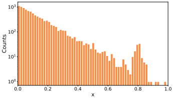
A brief overview

INPUT

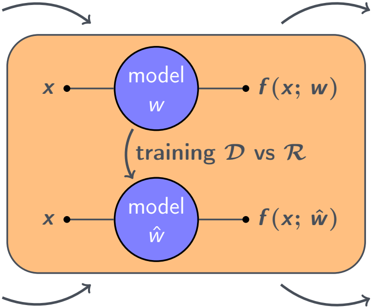
reference sample $\mathcal{R}$



data sample $\mathcal{D}$



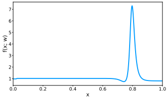
NEURAL NETWORK



OUTPUT

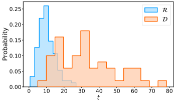
log-ratio distribution

$$f(x;\, \hat{w}) \approx \log \left[\frac{n(x\,|\,\mathcal{T})}{n(x\,|\,\mathcal{R})} \right]$$



t distribution

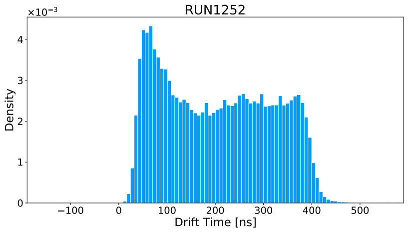
$$t(\mathcal{D}) = -2 \, \min_w L[f]$$



Tuning the Neural Network

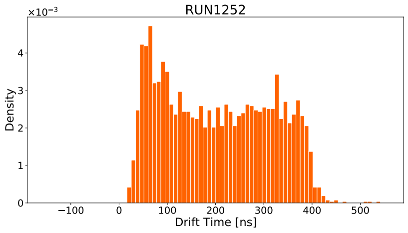
Training the network on the reference dataset

Reference sample $\mathcal{R}$



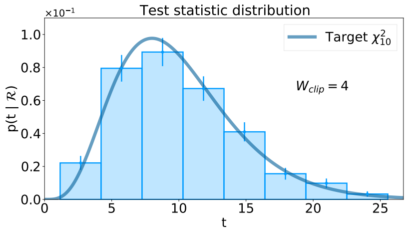
$\mathcal{N}_{\mathcal{R}} = 200000$

Data sample $\mathcal{D}$

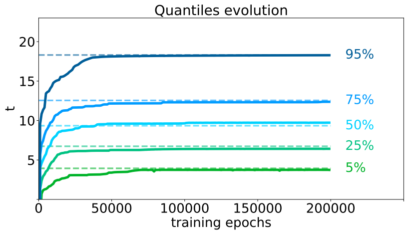


$\mathcal{N}_{\mathcal{D}} = 3000$ sampled from $\mathcal{R}$

Test statistic distribution



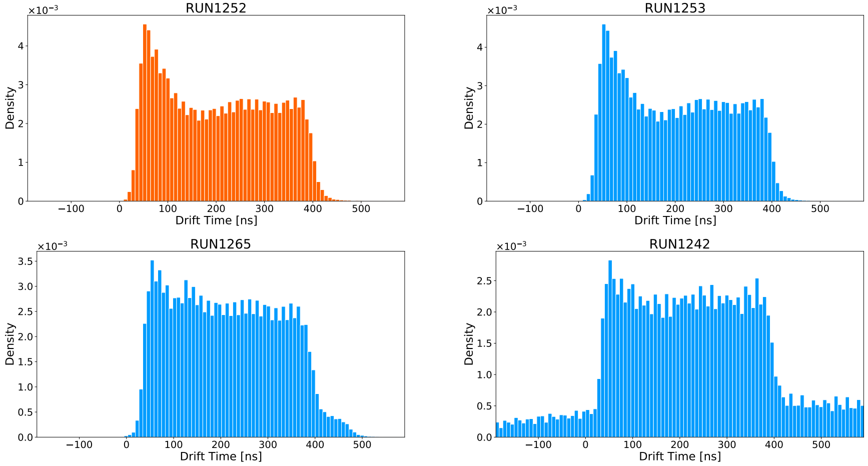
Quantiles evolution



400 samples of $\mathcal{N}_{\mathcal{D}} = 3000$ each

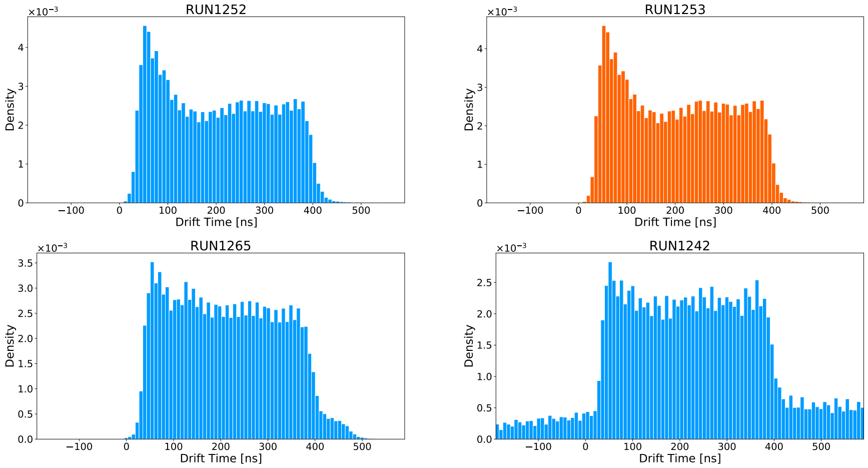
Testing the algorithm performance

Drift time distributions



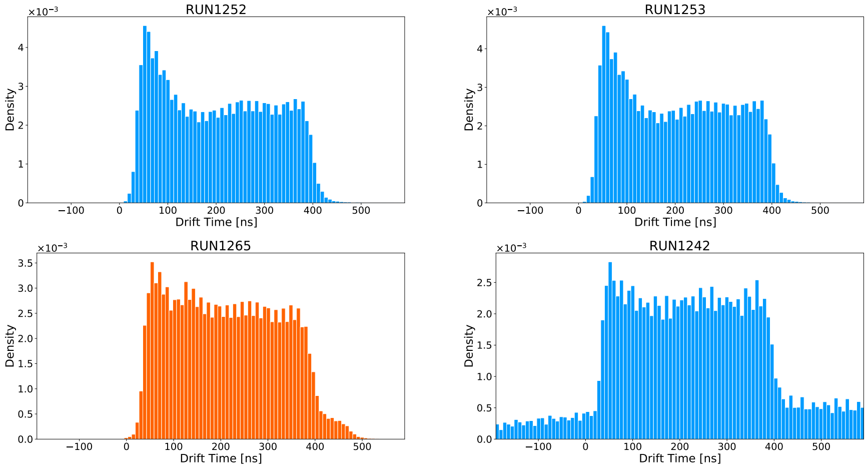
Testing the algorithm performance

Drift time distributions



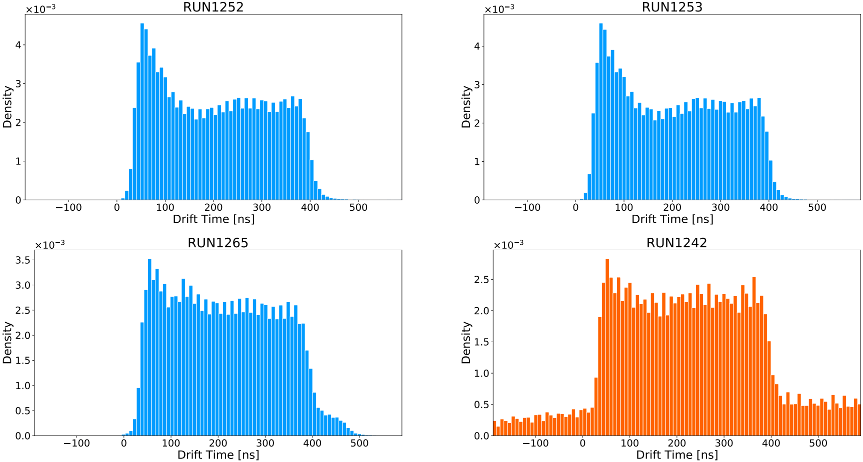
Testing the algorithm performance

Drift time distributions



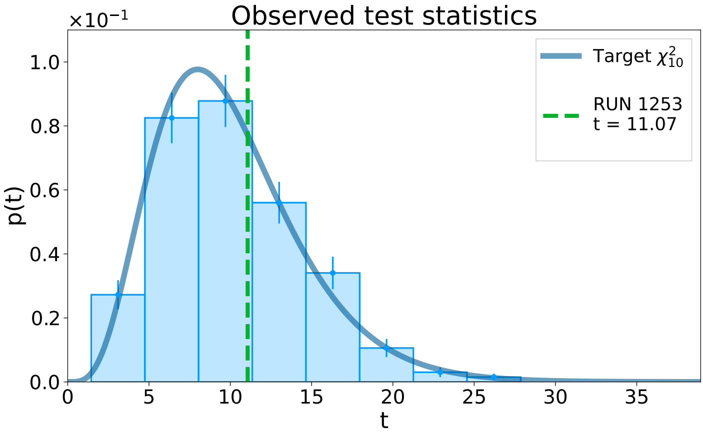
Testing the algorithm performance

Drift time distributions



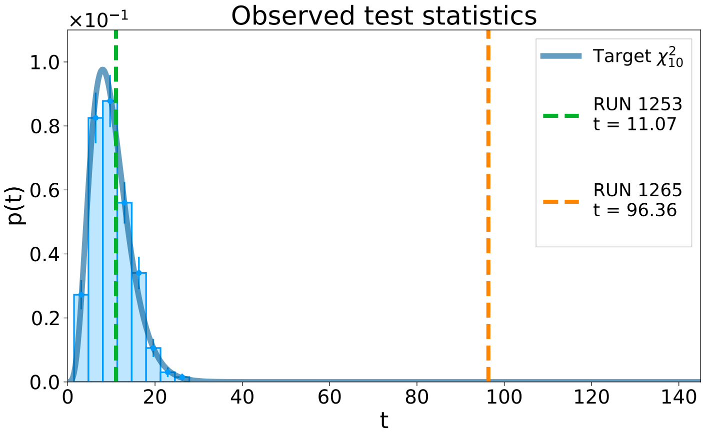
Testing the algorithm performance

Discrepancy assessment



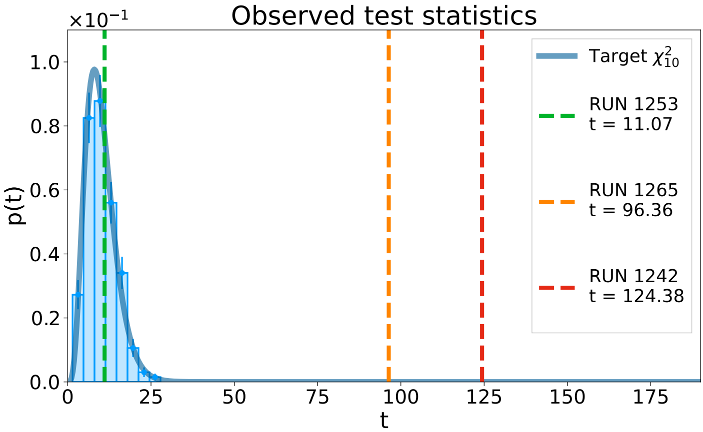
Testing the algorithm performance

Discrepancy assessment



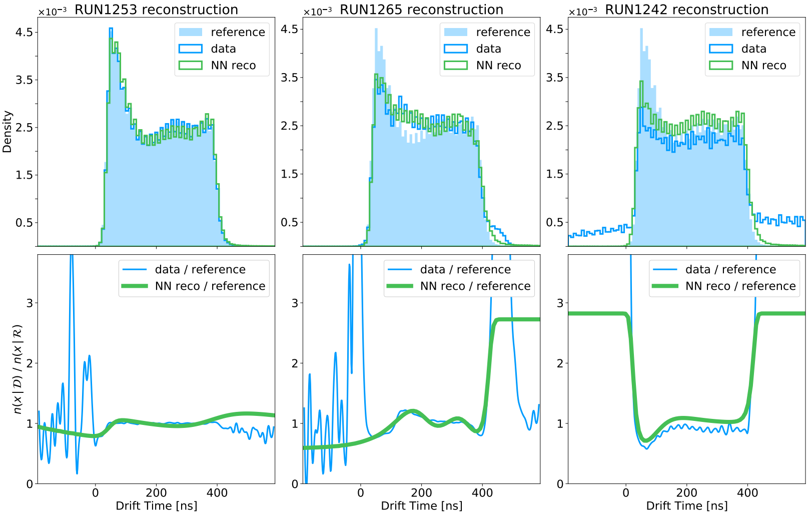
Testing the algorithm performance

Discrepancy assessment



Testing the algorithm performance

Neural network's data reconstruction



The current implementation of the algorithm

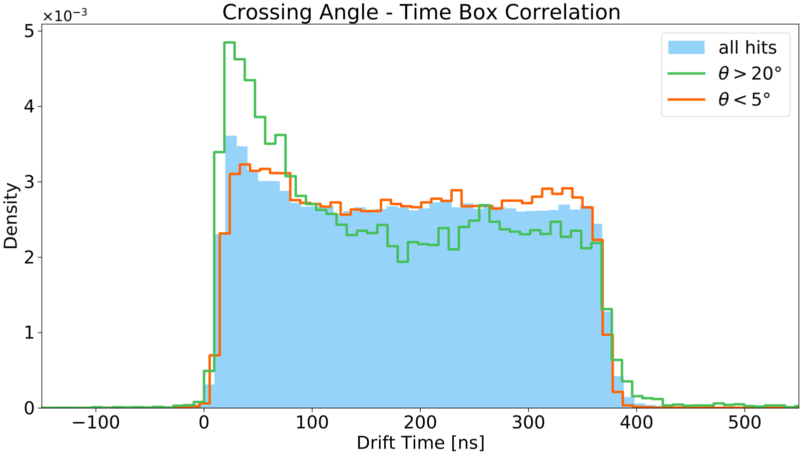
- correctly detects discrepancies between $\mathcal{R}$ and $\mathcal{D}$
- returns larger t_{obs} when the anomalies are more evident

However

- the most critical anomalies are not always the most obvious
- the time box shape may vary for other reasons
  (e.g. due to correlations with other observables)

Example of Correlated Observables

Correlation between time box shape and muon crossing angle



Future Outlook

Procedure improvements

1. Gathering correlated observables to improve flexibility
   - multi-dimensional input datasets

2. Mapping different data anomalies to the corresponding detector failures
   - systematic study of discrepant data collected with well-known detector setups

3. Building an *online* DQM framework for full automation
   - the current implementation of the algorithm is too slow
   - FalkonML[1] can be exploited to emulate our algorithm
   - training time lowers from $\sim$ 47 min to $\sim$2 sec

[1] GitHub repository: https://github.com/FalkonML/falkon

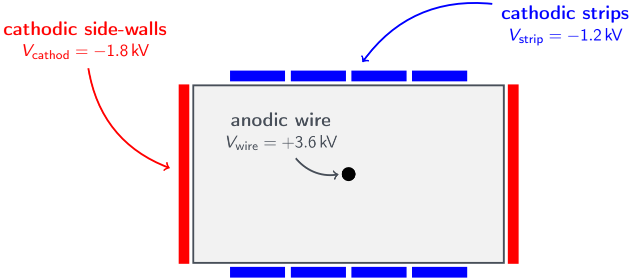
Thank you for your attention!

Università degli Studi di Padova

Drift Tubes

Specifics of the detector elementary cells

- DTs' transverse cross section is $L \times h = 42 \times 13\,\mathrm{mm}^2$
- Filled with an Ar-CO_2 (85/15 %) gas mixture at $\sim 1\,\mathrm{atm}$
- Precise electrode configuration ensures $\vec{E} \simeq$ uniform inside DTs
- Almost constant drift velocity $v_{\mathrm{drift}} \approx 54\,\mu\mathrm{m/ns}$



- Null hypothesis H_0:
  $n\,(x \,|\, \mathcal{R}) \longrightarrow$ data following the reference model $\mathcal{R}$
- Alternative hypothesis H_1:
  $n\,(x \,|\, \boldsymbol{w}) = n\,(x \,|\, \mathcal{R})\, e^{f\,(x;\, \boldsymbol{w})} \longrightarrow$ parametrized by the NN

The most powerful statistical test is the likelihood-ratio test (Neyman-Pearson lemma)

The algorithm compares $n\,(x \,|\, \mathcal{R})$ with $n\,(x \,|\, \widehat{\boldsymbol{w}})$ where $\widehat{\boldsymbol{w}}$ is the parameters configuration that maximizes the likelihood

The Wilks' Theorem

Asymptotic distribution of the log-likelihood ratio statistic

The test statistic is given by

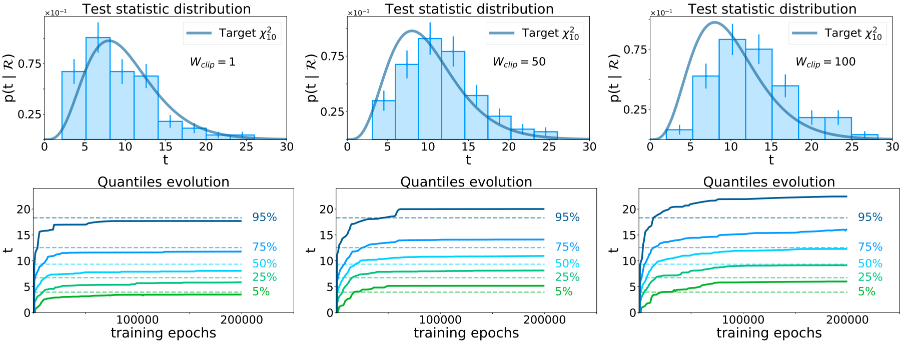
$$t(\mathcal{D}) = 2\,\log\left[\frac{e^{-\mathcal{N}(\widehat{\boldsymbol{w}})}}{e^{-\mathcal{N}(\mathcal{R})}}\prod_{x\in\mathcal{D}}\frac{n(x\,|\,\widehat{\boldsymbol{w}})}{n(x\,|\,\mathcal{R})}\right]$$

$$= -2\,\min_{\boldsymbol{w}}\left[\frac{\mathcal{N}(\mathcal{R})}{\mathcal{N}_{\mathcal{R}}}\sum_{x\in\mathcal{R}}\left(e^{\,f(x;\,\boldsymbol{w})}-1\right)-\sum_{x\in\mathcal{D}}f(x;\,\boldsymbol{w})\right]$$

Wilks' theorem states that the $t(\mathcal{D})$ distribution asymptotically approaches a χ^2 distribution under the null hypothesis H_0

Tuning the Neural Network

Testing different weight clipping values



FalkonML

Data quality monitoring with kernel methods

Kernel methods:

- $\widehat{f}(x) = \sum_i \alpha_i k(x, x_i)$ where $k(x, x') = \exp(\frac{||x-x'||^2}{2\sigma^2})$

FalkonML:

- implements the Nyström approximation
  $\widehat{f}(x) = \sum_i \alpha_i k(x, \widetilde{x}_i)$ where $\{\widetilde{x}_i\}_{i=1}^m \subset \{x_i\}_{i=1}^n$
- uses the logistic loss function
  $L[f] = \frac{\mathcal{N}(\mathcal{R})}{\mathcal{N}_{\mathcal{R}}}(1-y)\log(1+e^{f(x;\,\boldsymbol{w})}) + y\log(1+e^{-f(x;\,\boldsymbol{w})})$
- the test statistic $t(\mathcal{D})$ is computed using $f(x;\,\widehat{\boldsymbol{w}})$