MACHINE-LEARNING ASSISTED MUON TOMOGRAPHY

Sadman Ahmed Shanto

Advanced Particle Detector Laboratory

Department of Physics & Astronomy, Texas Tech University

TEXAS TECH UNIVERSITY
Department *of* Physics & Astronomy™

ADVANCED~PARTICLE~DETECTOR~LABORATORY
TEXAS TECH UNIVERSITY

MACHINE-LEARNING ASSISTED MUON TOMOGRAPHY

Outline

- Muon Tomography
- Muon Detection System
- Muon Telescope
    - Prototype 1a
    - Prototype 1b (Current)
- Muon Tracking
- Analysis Software Design
- Recurrent Neural Networks
- Image Segmentation

ADVANCED~PARTICLE~DETECTOR~LABORATORY

Muon Tomography

- Muon Tomography – Technique that utilizes muon scattering and muon absorption to generate images of large objects such as buildings, volcanoes, and ancient archaeological structures
- Muon images contain both density and shape information of objects
- Non-invasive way of imaging using a natural source

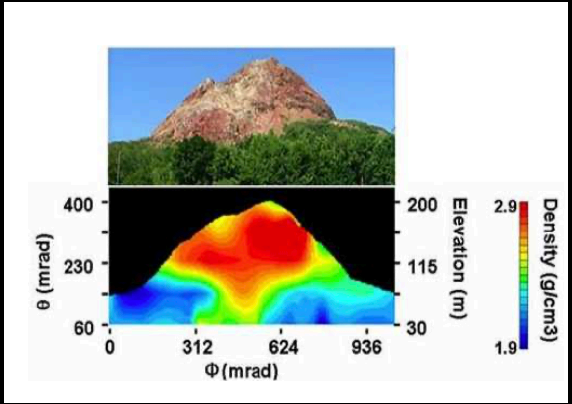


Fig 1. Top: Showa-Shinzan Lava Dome. Bottom: Density Distribution

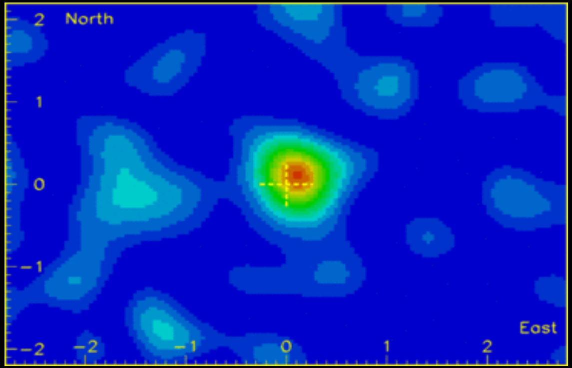


Fig 2. The Moon's Cosmic Ray Shadow detected by the Soudan II detector

Muon Detection System

- The muons are generated in cosmic ray showers
- When they pass through scintillators, the create scintillation photons
- These photons are detected by PMTs or SiPMs and converted into electrons
- The DAQ system comprises a readout electronics circuit that determines muon hits

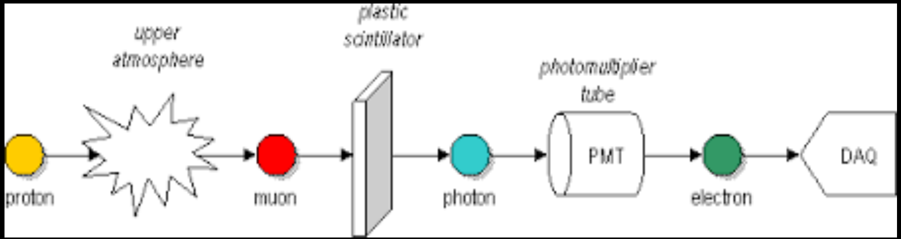


Fig 1. Schematic summarizing the muon detection process

Fig 2. EJ – 200 Scintillator Bars

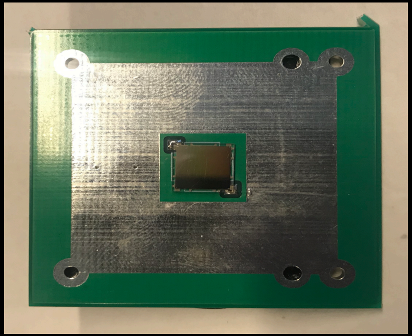


Fig 3. Silicon Photo Multiplier (SiPM)

Muon Telescope:
Prototype 1a

- Telescope - 2 layered system with each tray containing the following components
- Scintillator bars (5 x 5 x 60 cm^3), silicon photomultipliers (SiPM), Winston Cone light collectors, Readout Electronics and a network of Arduinos (DAQ).
- Size: 90 cm by 180 cm
- Reference: https://aip.scitation.org/doi/abs/10.1063/10.0002046

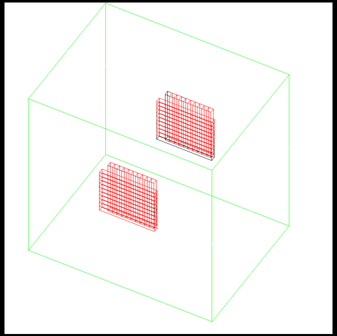


Fig 1. 2 Layers of Scintillator Bars

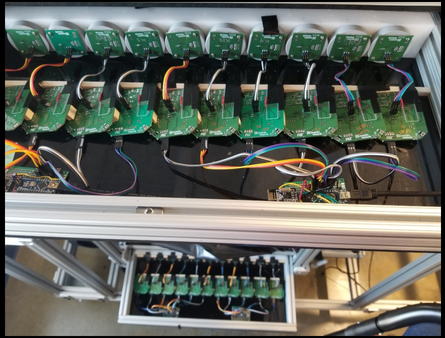


Fig 2. SiPMs and the DAQ

Fig 3. Muon Telescope

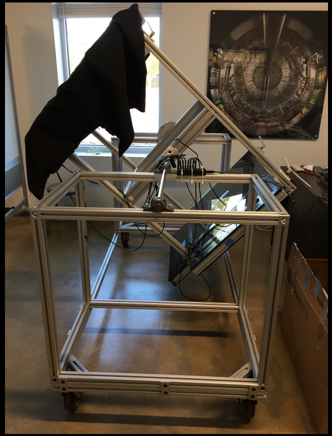


Fig 4. Muon Telescope rotated 45 degrees

Muon Telescope:
Prototype 1b

- Similar set up to Prototype 1a
- PMTs
- Optical Cookies
- CAMAC (DAQ)
- New Tomogram Generation Schema

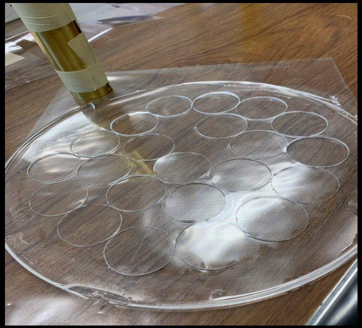


Fig 1. Optical Cookies

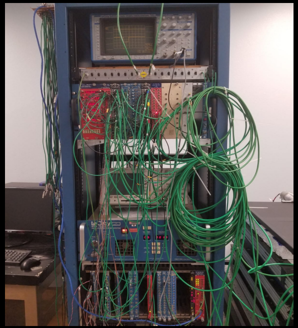


Fig 2. CAMAC DAQ

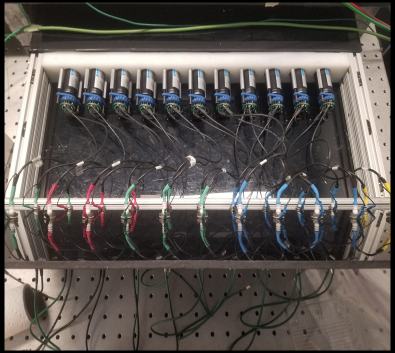


Fig 3. PMTs installed on a tray

Fig 4. Prototype 1b Test Set Up

Data Acquisition System (DAQ)

- DAQ – Communication system that allows us to efficiently transfer data from the start to finish via wireless communication.

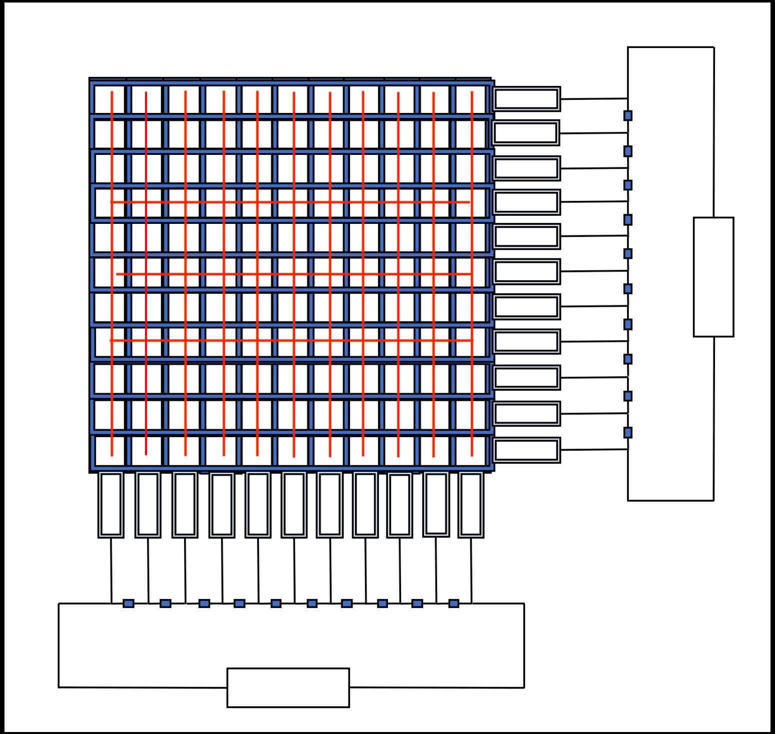


Fig 1. DAQ Pipeline.

Data Acquisition System (DAQ)

- DAQ – Communication system that allows us to efficiently transfer data from the start to finish via wireless communication.
- CAMAC Crates
- ADC Modules
- TDC Modules
- Scaler Module

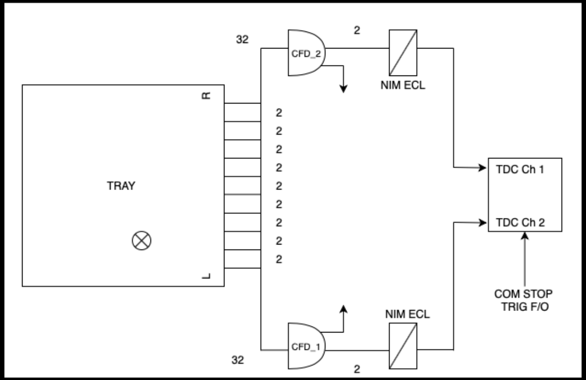


Fig 1. DAQ Signal Processing.

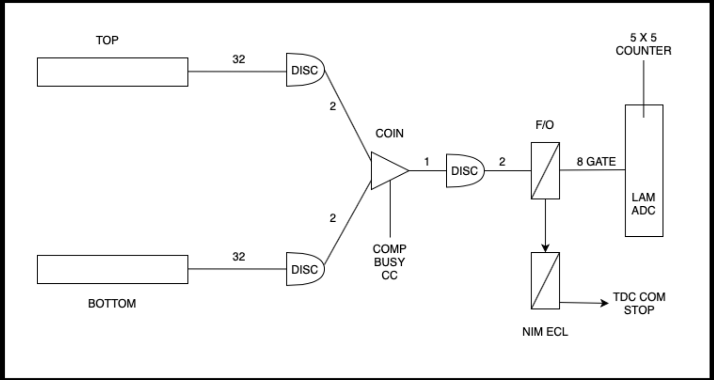


Fig 2. DAQ Trigger Process.

Muon Tracking

- TDC Signal Distribution
- 2 Channels Per Layer

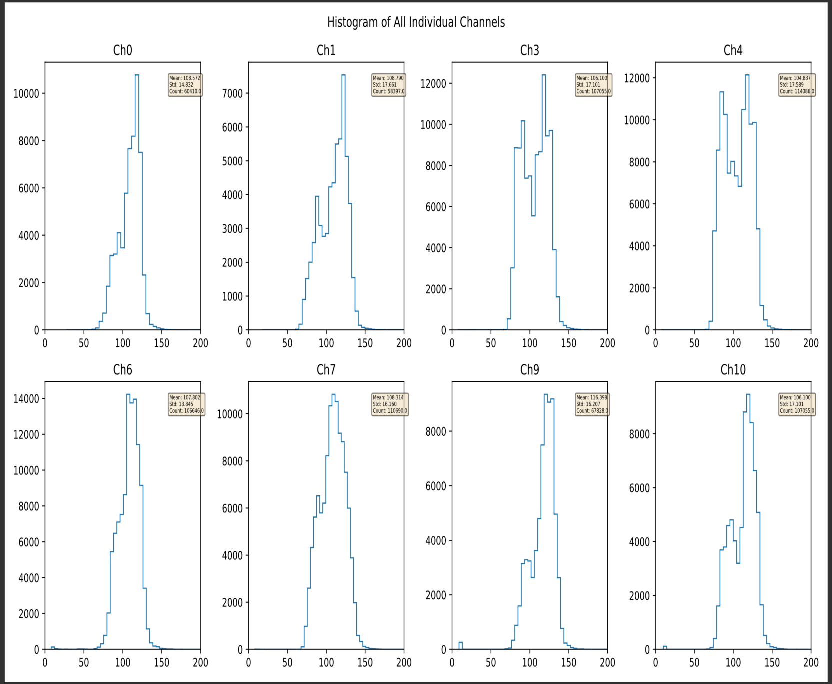


Fig 1. TDC Distribution of Channels

Muon Tracking

- TDC Difference
- Peaks ⟶ Channels

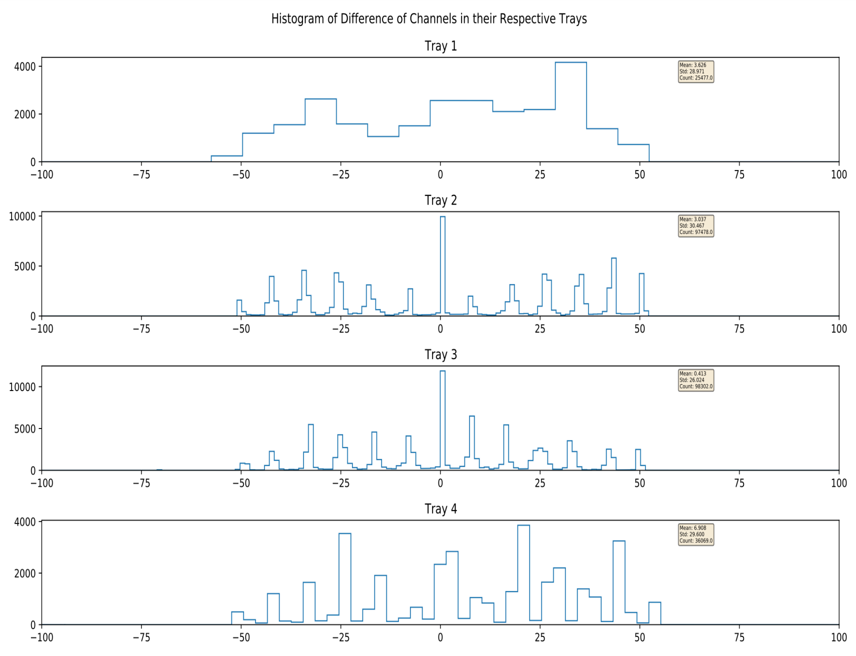


Fig 1. TDC Difference Plots for Individual Trays

Muon Tracking

- Definition of Asymmetry
- 2D Asymmetry Plot
- Asymmetry to Spatial Location

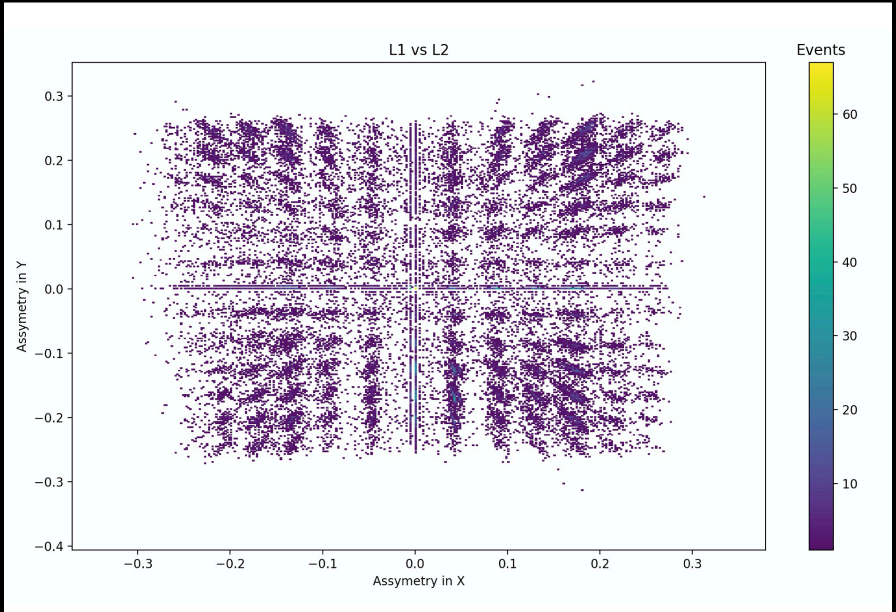


Fig 1. Asymmetry Plot of Layer 1 (Horizontal) vs Layer 2 (Vertical)

Muon Tracking

- Extracting Muon Hits
- Sum of TDC as Indicator of Muon Hit Cluster Position

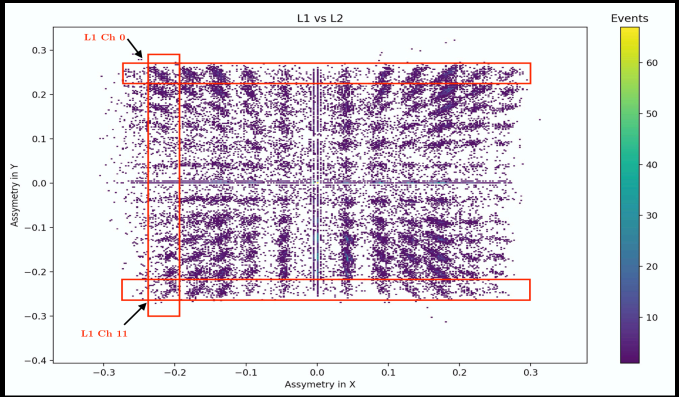


Fig 1. Isolating Events from Asymmetry Values

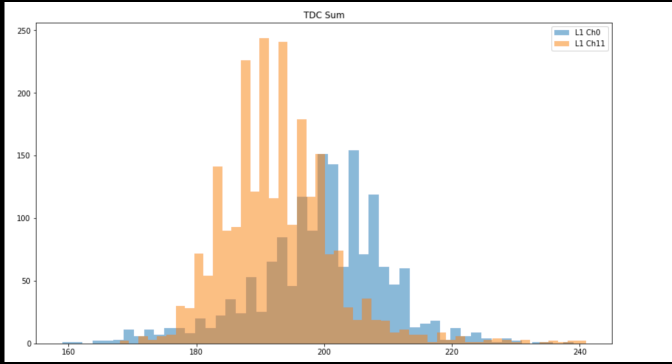


Fig 2: Sum of TDCs of Isolated Events

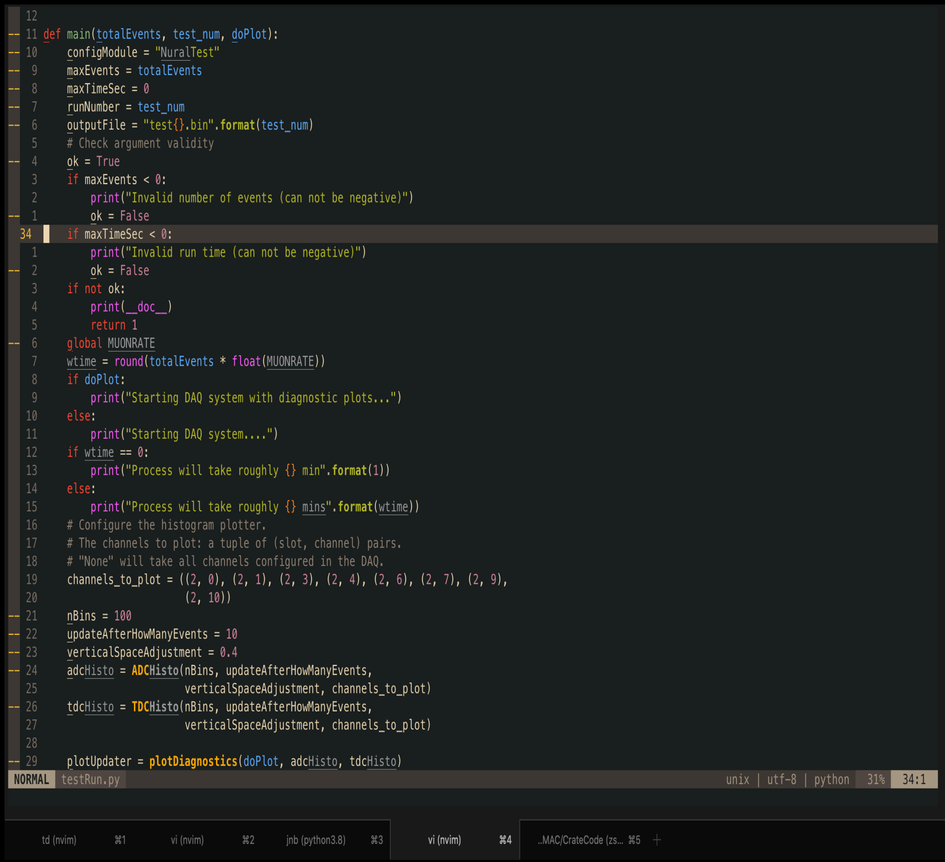
MACHINE-LEARNING ASSISTED MUON TOMOGRAPHY

Outline

- Muon Tomography
- Muon Detection System
- Muon Telescope
  - Prototype 1a
  - Prototype 1b (Current)
- Muon Tracking
- Analysis Software Design
- Recurrent Neural Networks
- Image Segmentation

Analysis Software Design:
DAQ Controller

- Interfaces with CAMAC Crates and Modules
- Automated Data Storage
- Diagnostic Tests and Plots



```python
def main(totalEvents, test_num, doPlot):
    configModule = "NuralTest"
    maxEvents = totalEvents
    maxTimeSec = 0
    runNumber = test_num
    outputFile = "test{}.bin".format(test_num)
    # Check argument validity
    ok = True
    if maxEvents < 0:
        print("Invalid number of events (can not be negative)")
        ok = False
    if maxTimeSec < 0:
        print("Invalid run time (can not be negative)")
        ok = False
    if not ok:
        print(__doc__)
        return 1
    global MUONRATE
    wtime = round(totalEvents * float(MUONRATE))
    if doPlot:
        print("Starting DAQ system with diagnostic plots...")
    else:
        print("Starting DAQ system....")
    if wtime == 0:
        print("Process will take roughly {} min".format(1))
    else:
        print("Process will take roughly {} mins".format(wtime))
    # Configure the histogram plotter.
    # The channels to plot: a tuple of (slot, channel) pairs.
    # "None" will take all channels configured in the DAQ.
    channels_to_plot = ((2, 0), (2, 1), (2, 3), (2, 4), (2, 6), (2, 7), (2, 9),
                        (2, 10))
    nBins = 100
    updateAfterHowManyEvents = 10
    verticalSpaceAdjustment = 0.4
    adcHisto = ADCHisto(nBins, updateAfterHowManyEvents,
                        verticalSpaceAdjustment, channels_to_plot)
    tdcHisto = TDCHisto(nBins, updateAfterHowManyEvents,
                        verticalSpaceAdjustment, channels_to_plot)

    plotUpdater = plotDiagnostics(doPlot, adcHisto, tdcHisto)
```


Fig 1. Screen Capture of DAQ Code

Analysis Software Design:
Event Display

- Identify Event Channels
- Approximate Locations
- Estimate and Create Tracks

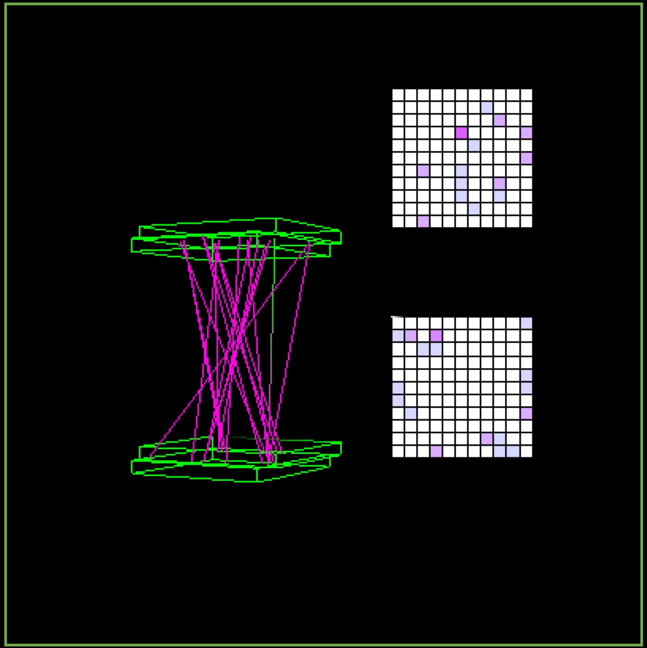


Fig 1. Figure Showing Event Display for a Sample Run

Analysis Software Design:
Data Pipeline

- Raw Data - *.bin*
- Processed Data - *.ftr*
- Data Structure – *Modin DataFrame*

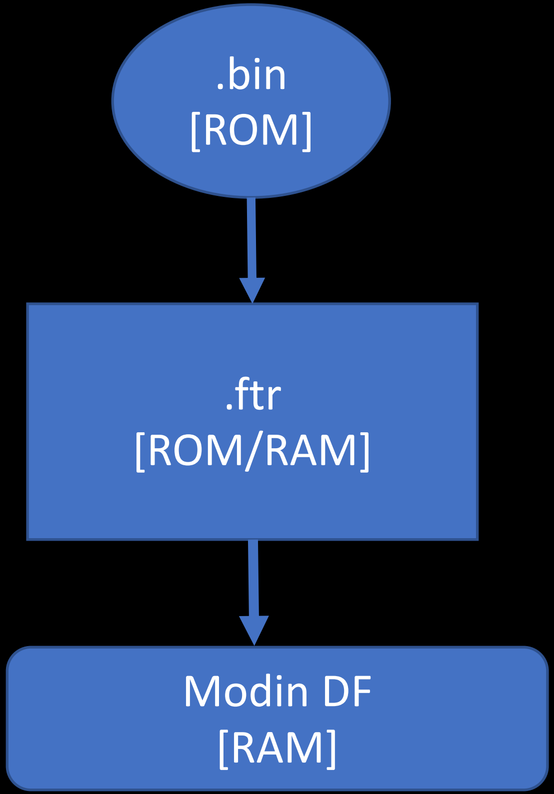
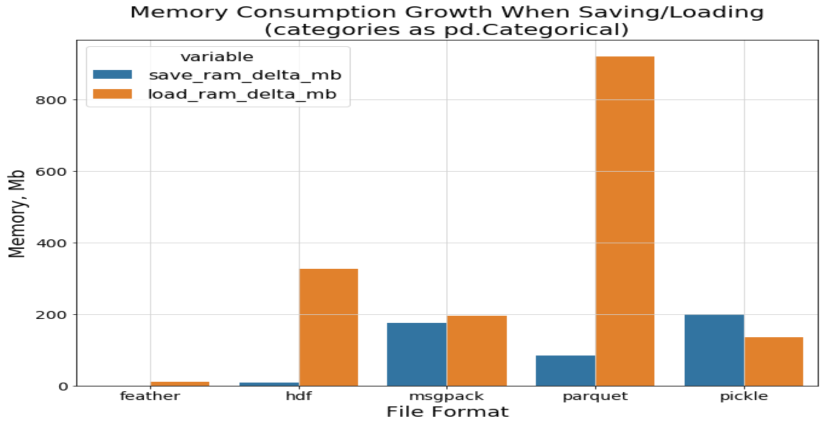


Fig 1. Water Tower at
Reese Technology Center

Analysis Software Design: Computational Efficiency

- Python (Cythonized)
- Numba
- Feather file format
- Modin vs PyROOT
- Parallelization
- Vectorization



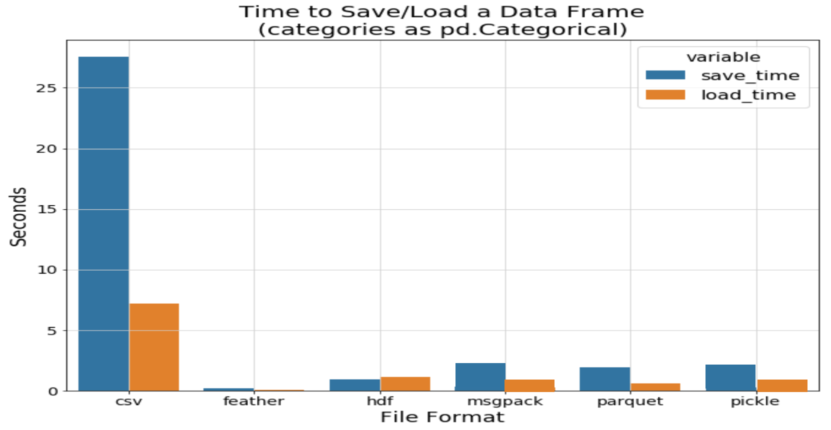


Fig 1. I/O Speed comparisons

Analysis Software Design:
GUI Analysis Program

- Customized Functionality
- Automated Report Generation
- Fast and Dirty Analysis

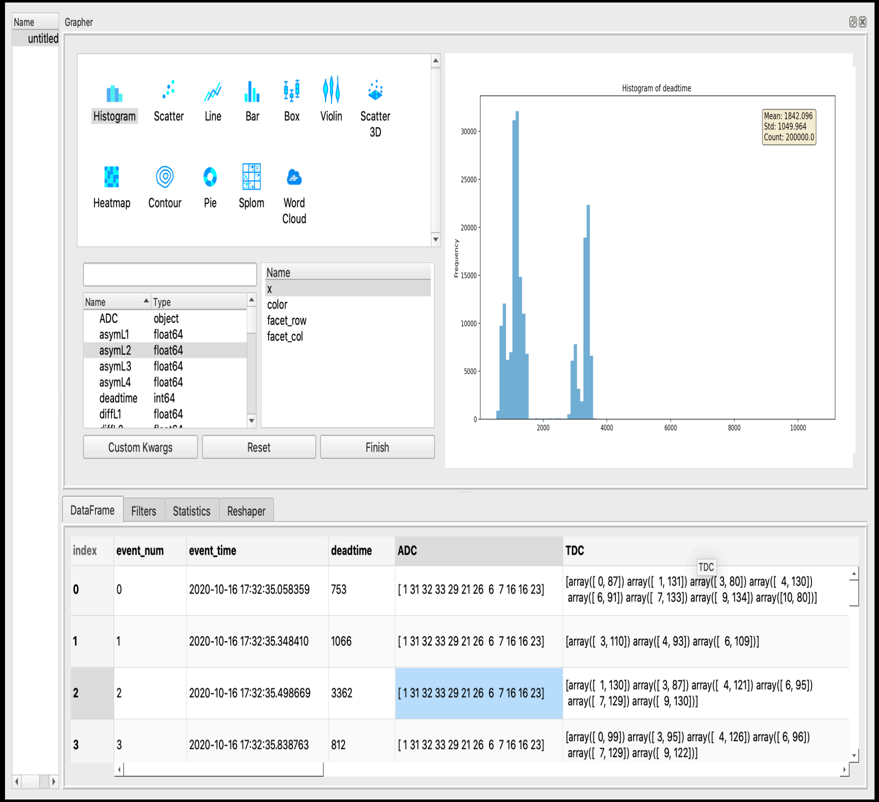


Fig 1. Screen Capture of GUI

Recurrent Neural Networks

- Ideal for Sequential Data
- Reposes next-step hit predictions as regression problem
- Uses LSTM (Long Short Term Memory) network for powerful non-linear sequence modeling capabilities
- Reference: arXiv:1810.06111v1

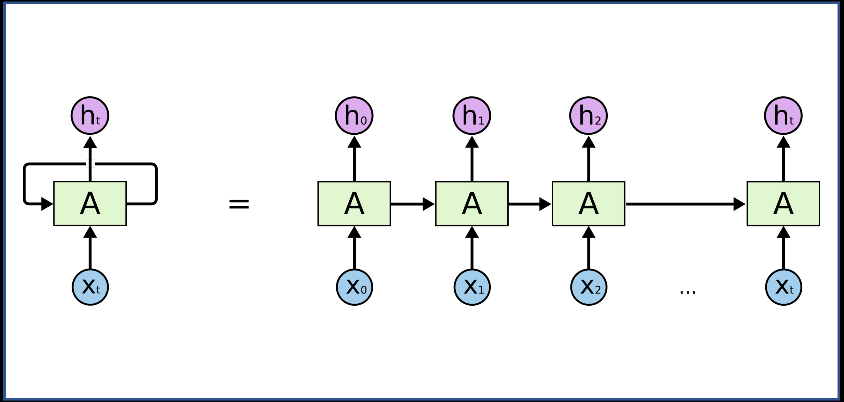


Fig 1. RNN Structure

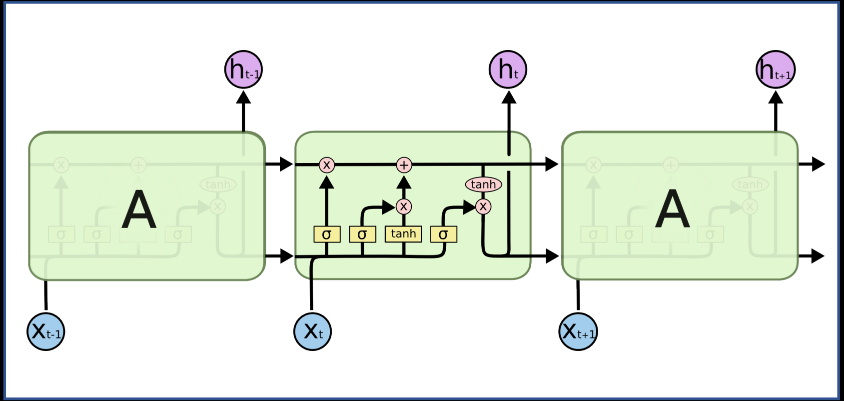


Fig 1. LSTM Cell Structure

Recurrent Neural Networks

- Input: *Sequence of hit coordinate*
- Processing: *For every element, a prediction of the position of the next hit conditioned on its position and the preceding hit positions.*
- The learning problem: *multi- target regression problem*
- The model is trained with a mean-squared-error loss function.
- We train model on tracks that hit all 4 detector layers.
- Reference: arXiv:1810.06111v1

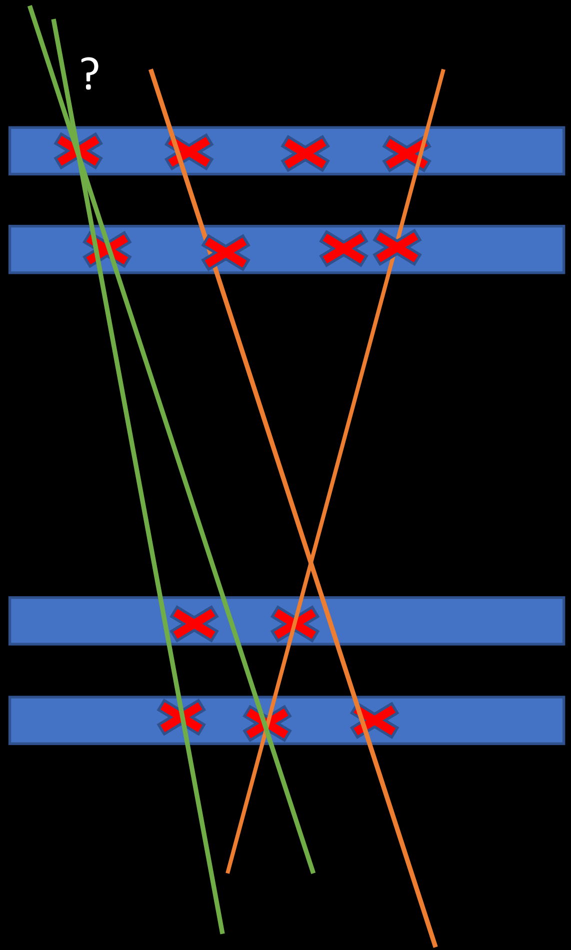


Image Segmentation (IS)

- Purpose: *train a neural network to output a pixel-wise mask of the image*
- Idea:
  - Extract shape of Object of Interest using digital image
  - Leverage IS to transform Tomogram data to enhance resolution
- Popular Technique in Medical Imaging

Fig 1. Application of IS on digital image data

Looking Forward

- Implementation and Analysis of ML
- Lead Brick Experiment
- Comprehensive Software Package Development
- Muon Parity Study

Questions?