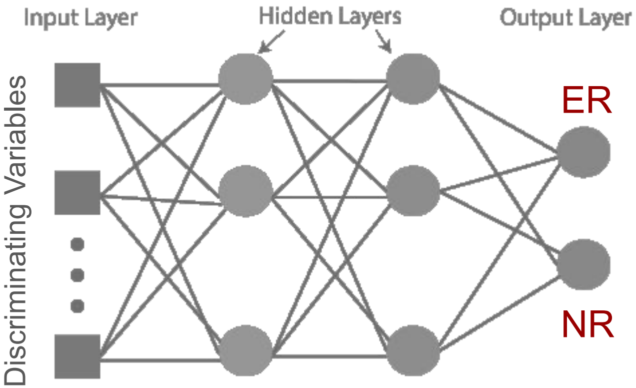
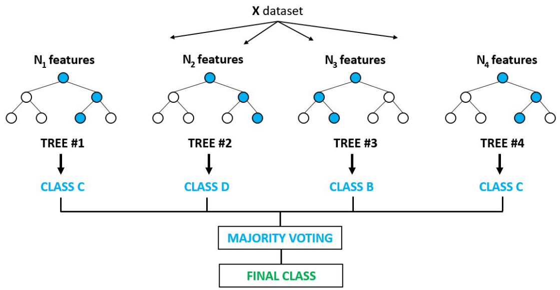
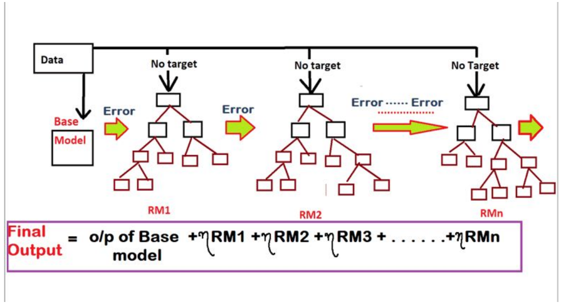
Background Rejection with Deep Learning Models and Classical Approach

Atul Prajapati 06/10/2022

Deep Learning Models







1) Deep Neural Network

- ❖ Weights of the network is optimised iteratively
- ❖ Result is the output of the last layer.

2) Random Forest Classifier

- ❖ It can build each tree independently.
- ❖ Results are combined at the end of the process.

3) Gradient Boosted Classifier

- ❖ It builds one tree at a time.
- ❖ It combines results along the way.

Classical Approach:

- Applying cuts on all the variables that I used for training.

  Signal Events (N_{signal}) = No. of NR events from the variable passing the cut

  Bkg Events (N_{bkg}) = No. of ER events from the variable passing the cut

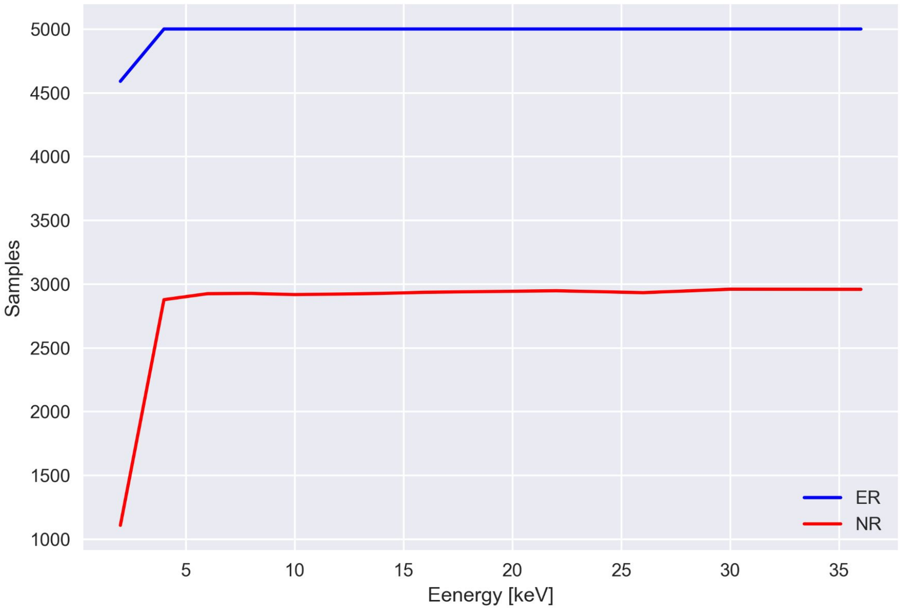
  Signal efficiency (S_{eff}) = N_{signal}/N_{total}

  Bkg. Efficiency (B_{eff}) = N_{bkg}/N_{total}

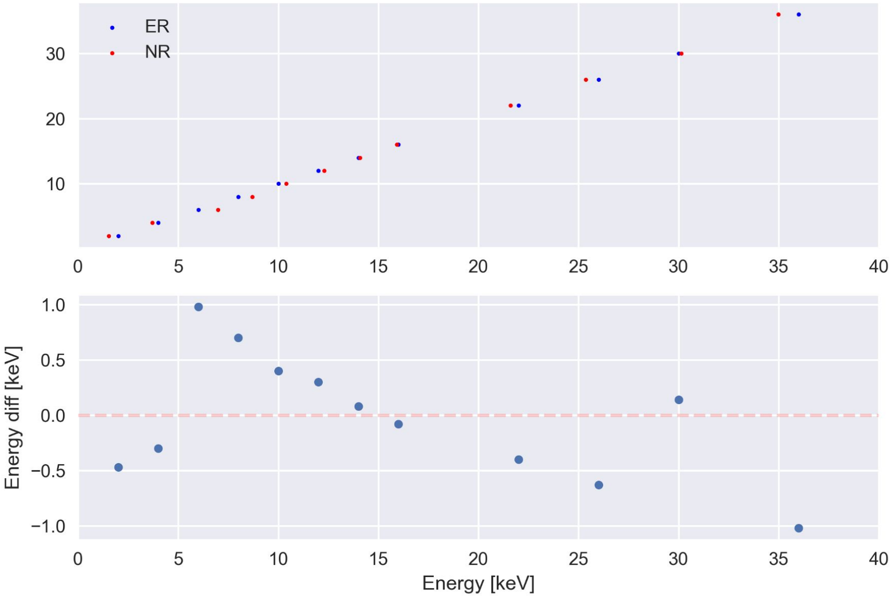
  Bkg. Rejection = N_{total}/N_{bkg}

- **Variables**: thin_track, SDCD, CylThick, ChargeUnif, LAPA, MaxDen, eta, curlyness, SC_nhits, SC_integral, SC_length, SC_width, delta, slimness

Dataset

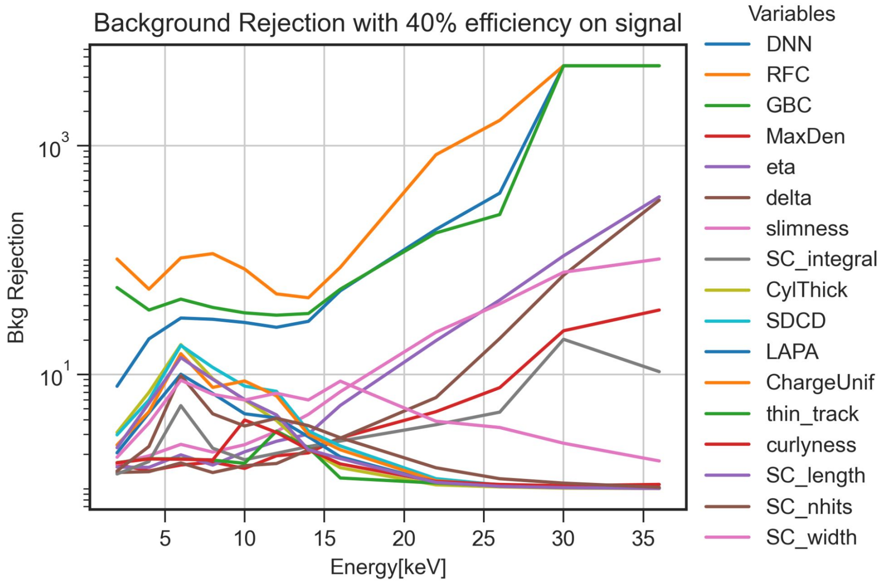


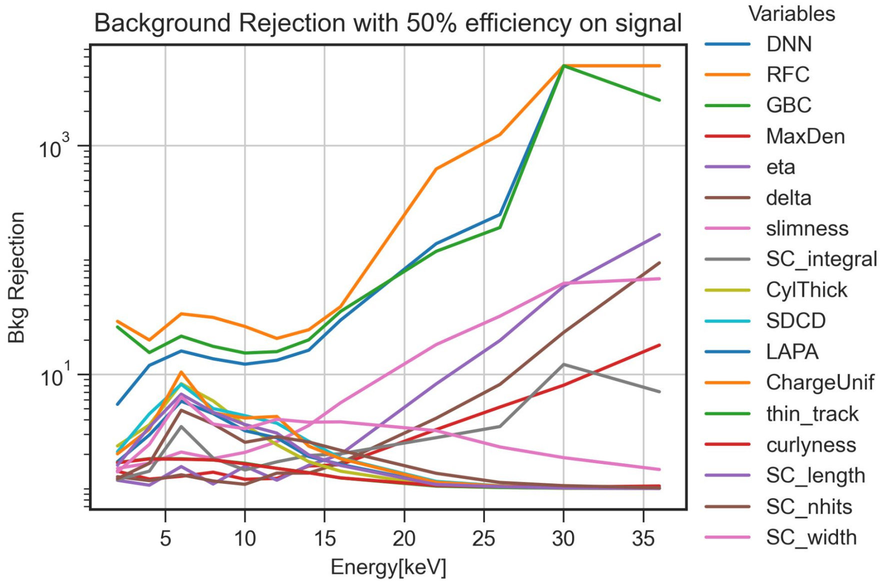
Samples
Eenergy [keV]
ER
NR
5000
4500
4000
3500
3000
2500
2000
1500
1000
5
10
15
20
25
30
35



ER
NR
Energy diff [keV]
Energy [keV]
30
20
10
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
40

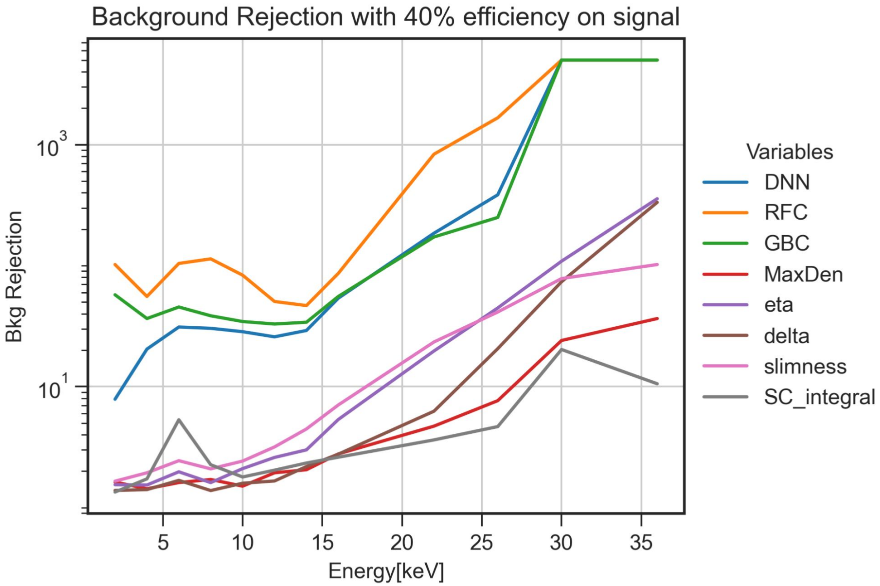
Models trained on whole dataset

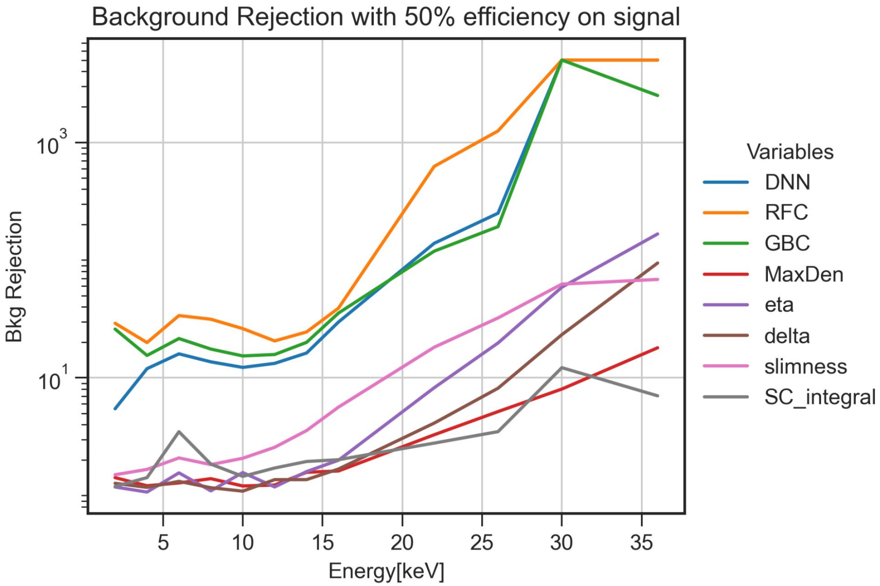




- ❖ Background Rejection is plotted with 40% and 50% signal efficiency in each energy bin.
- ❖ All the variables shown in the plot show the background rejection with classical approach.

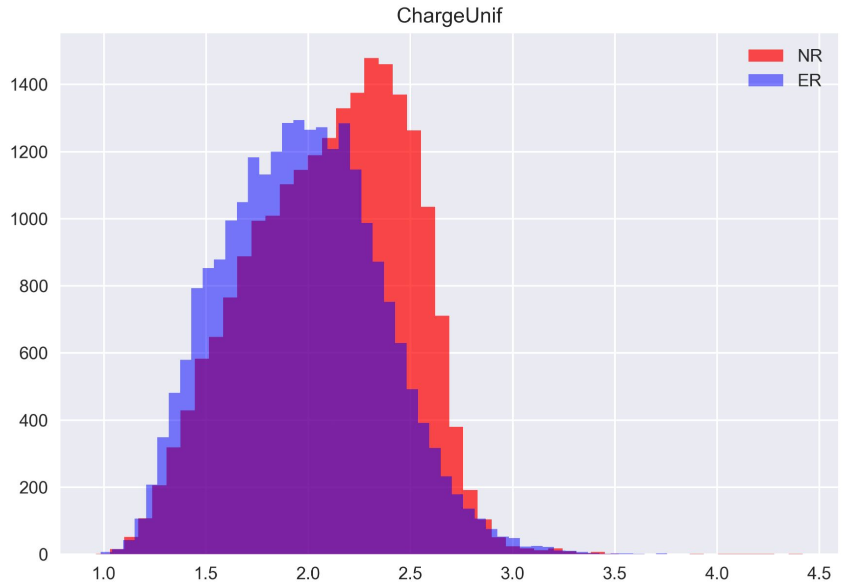
Bkg. Rejection only for Variable_5



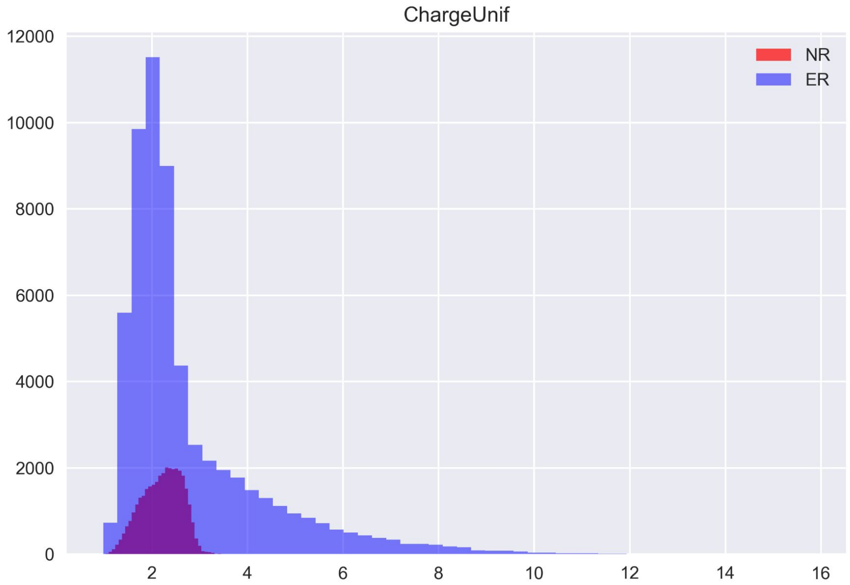


Deep Learning models are trained on the whole dataset.

Variables with decreasing rejection at higher energy



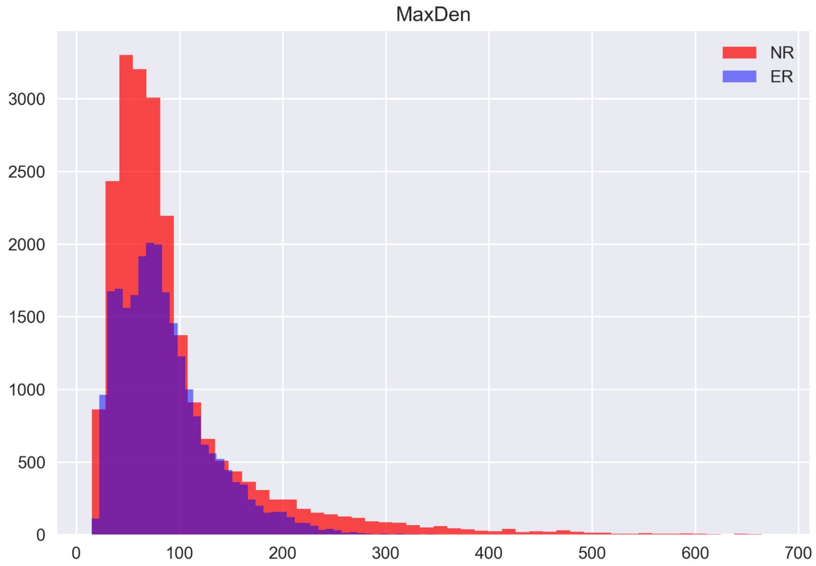
ChargeUnif till 15 keV



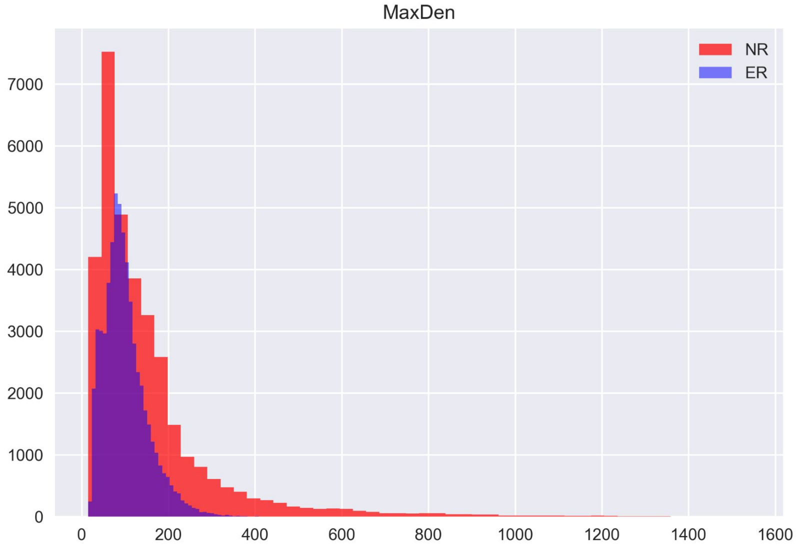
ChargeUnif till 36 keV

Variable_9: thin_track, SDCD, CylThick, ChargeUnif, LAPA, curlyness, SC_nhits, SC_length, SC_width

Variables with increasing rejection at higher energy



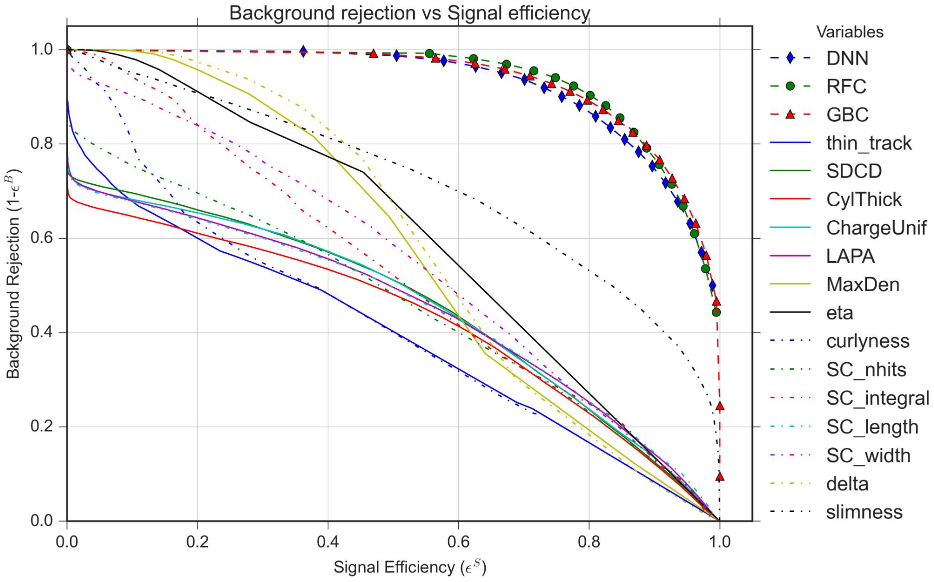
MaxDen till 15 keV

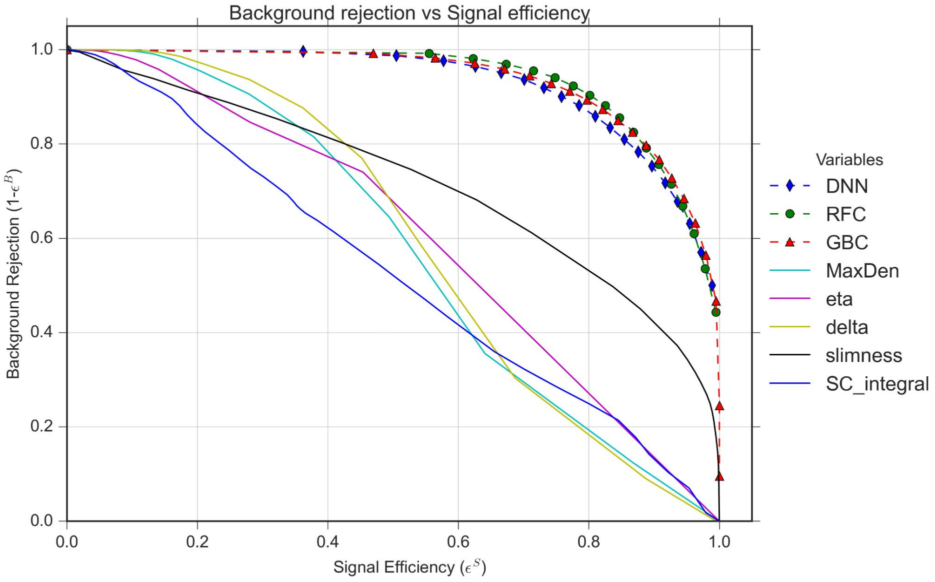


MaxDen till 36 keV

Variable_5: MaxDen, eta, SC_integral, delta, slimness

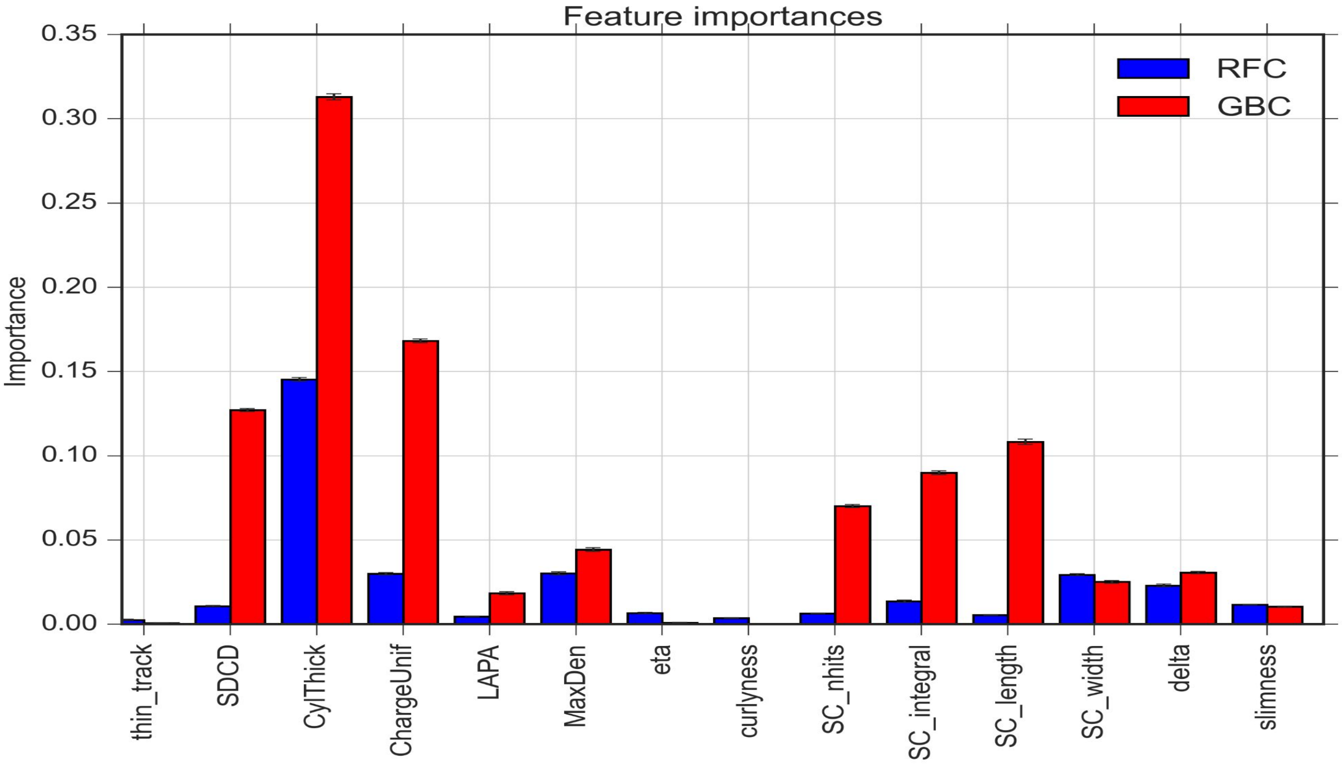
Bkg. Rejection vs Sig. Eff.



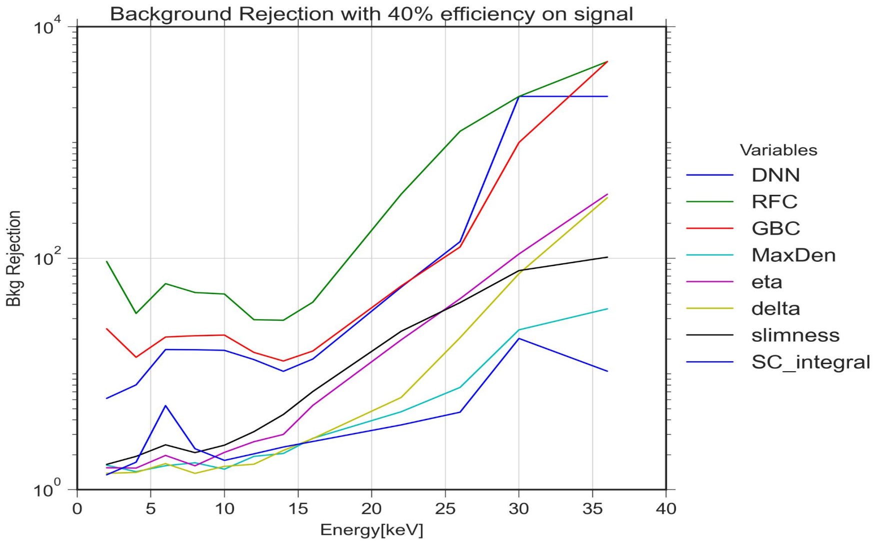


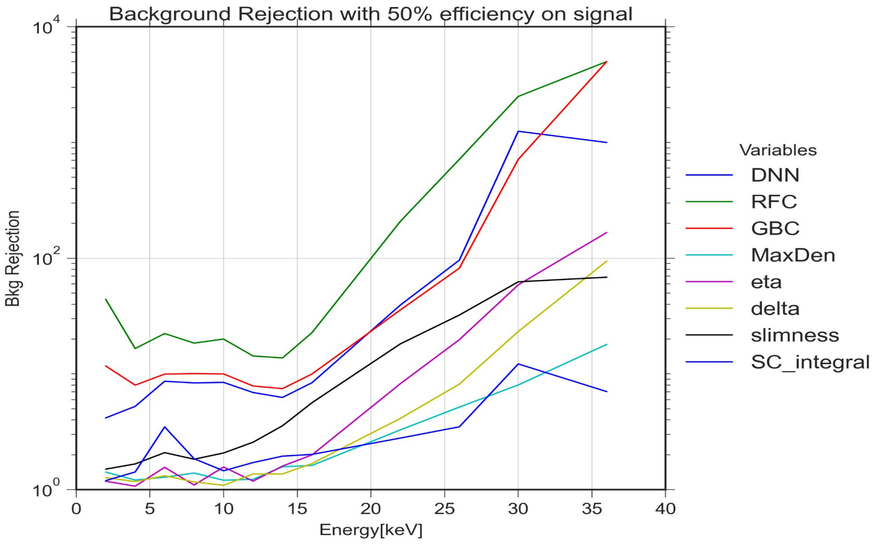
Deep Learning models are trained on the whole dataset.

Feature Importance

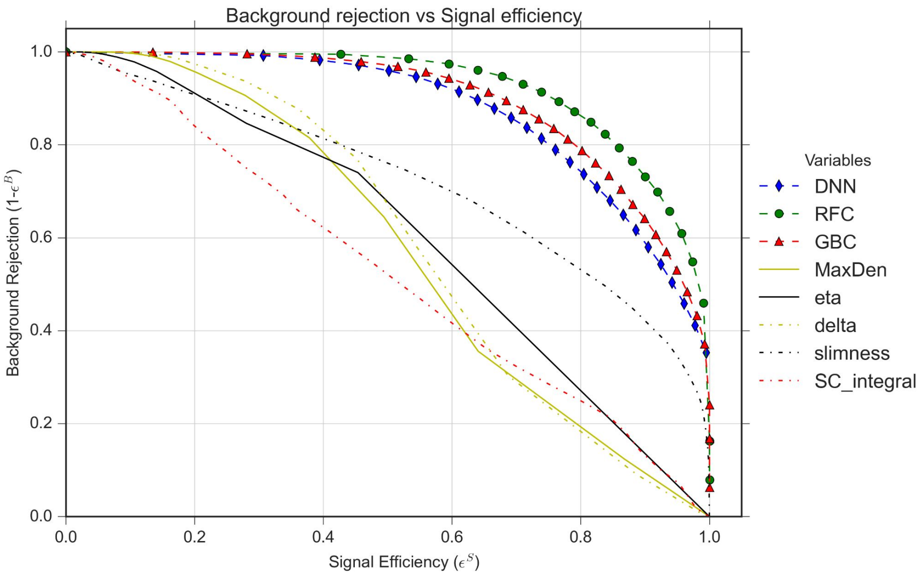


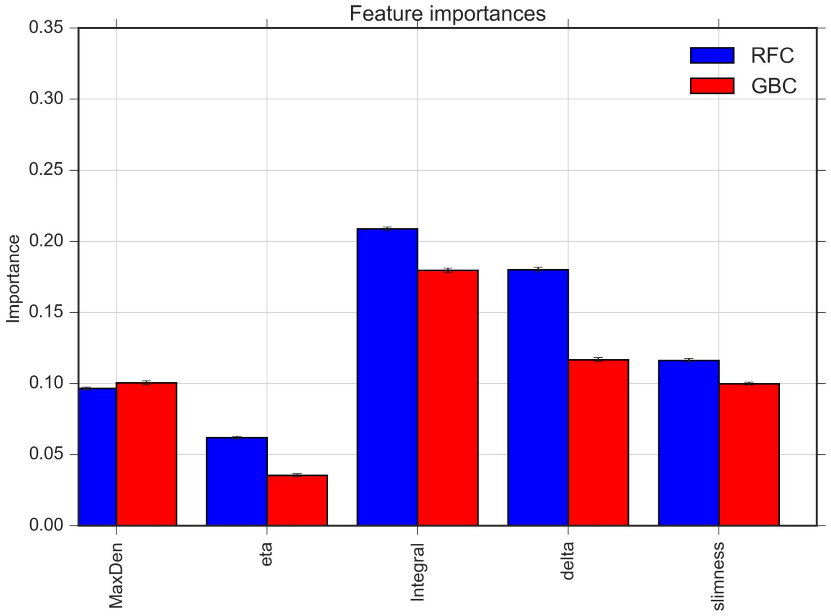
Models trained on Variable_5



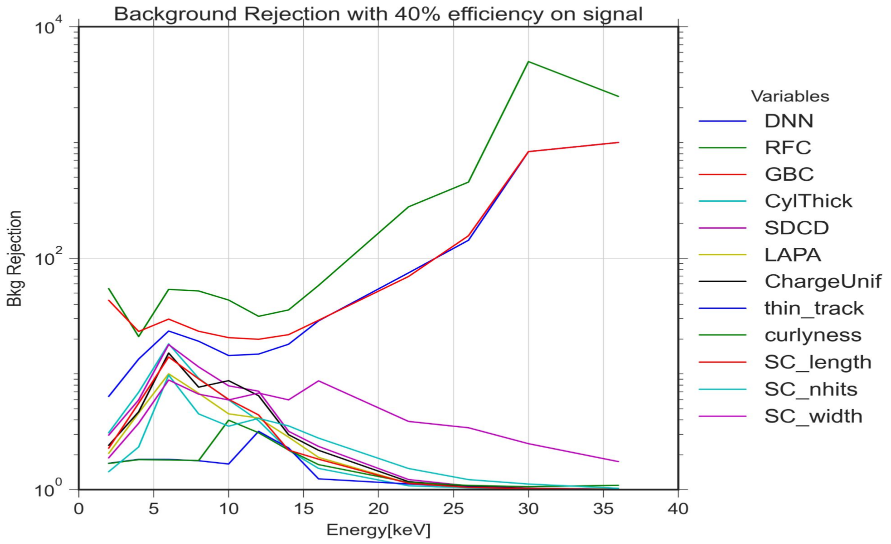


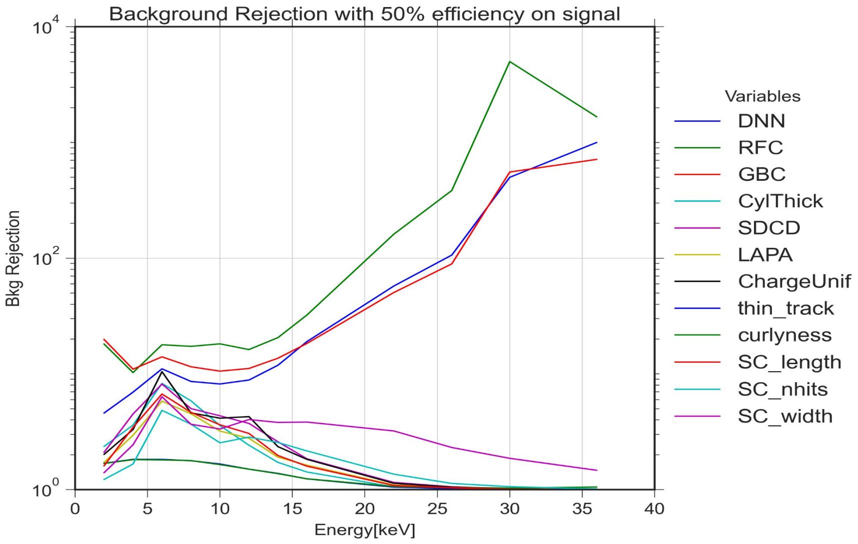
Bkg. eff. Vs Sig. eff. And Feature importance for Variable_5



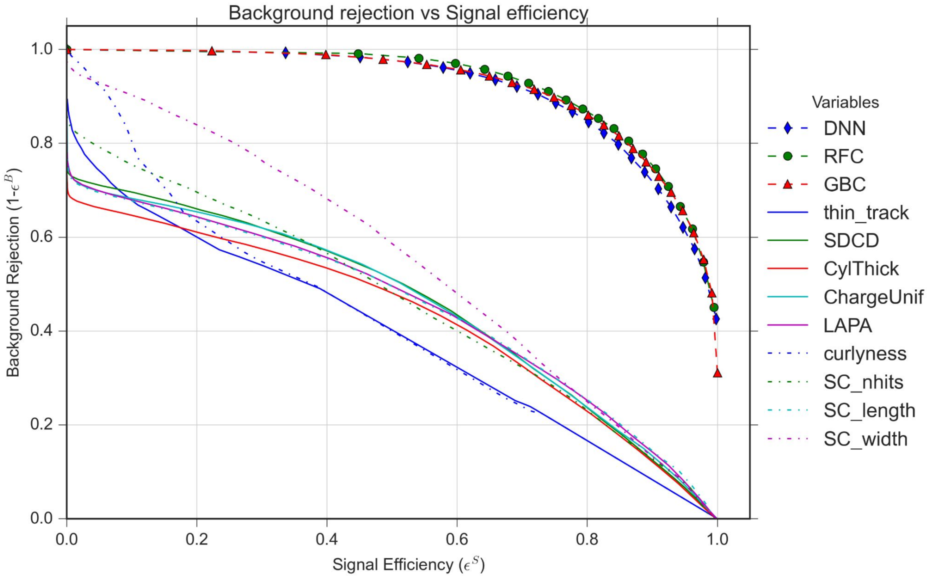


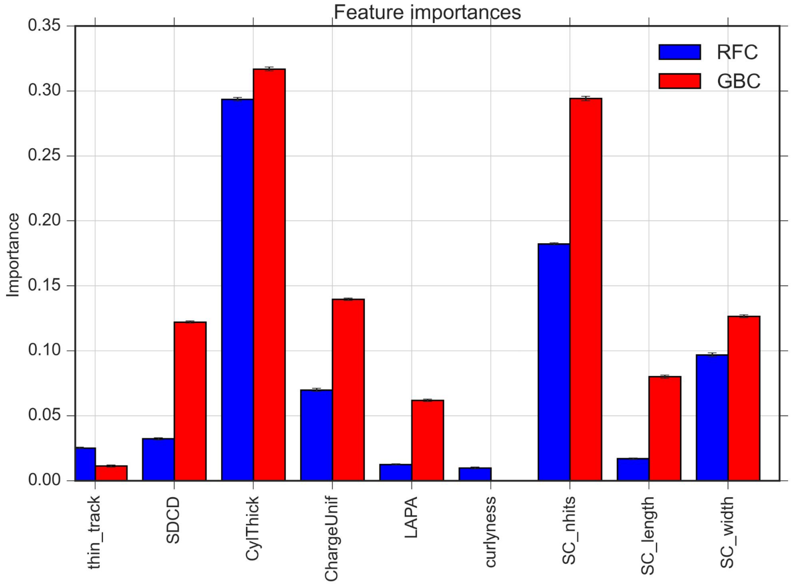
Models trained on Variable_9



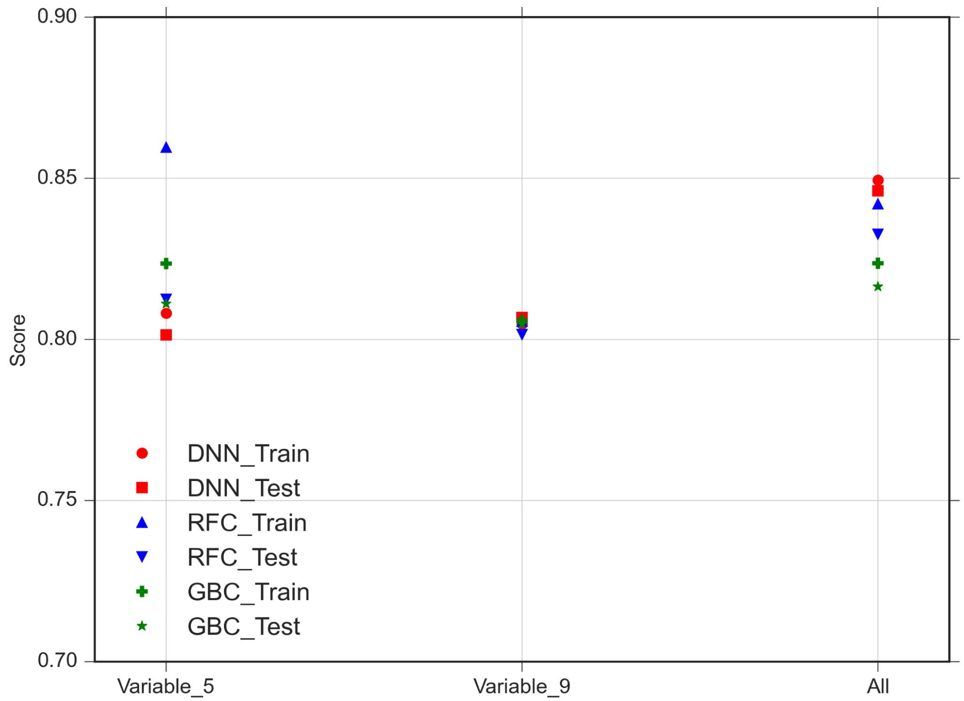


Bkg. eff. Vs Sig. eff. And Feature importance for Variable_9





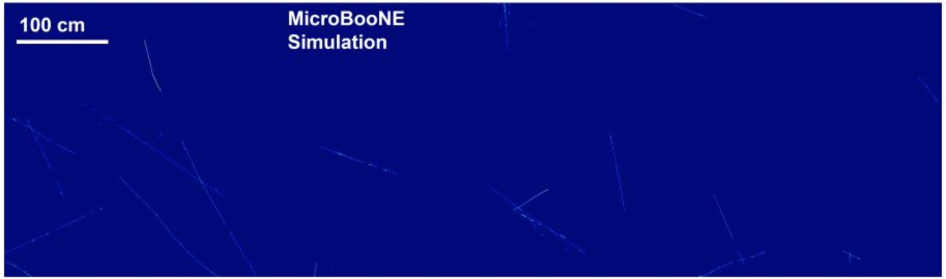
Training and Testing Scores for all 3 models



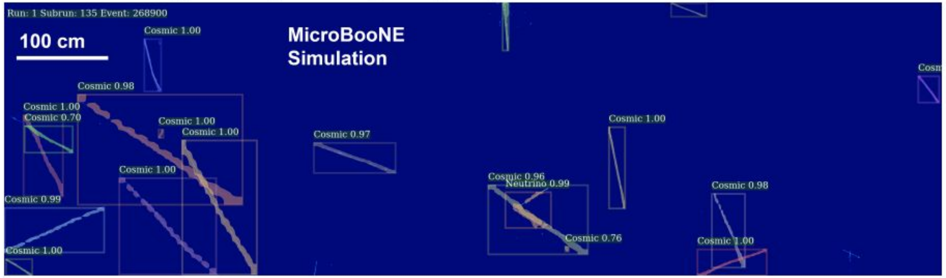
- All the 3 models were trained on all 3 different datasets namely: Variable_5, Variable_9 and variable_5+ Variable_9.
- Score of training and testing of all the models is plotted for the different datasets.

Mask-RCNN studies

Mask-RCNN Model



(a) An example of an input image as given to sMask-RCNN to process



(b) A simulated neutrino interaction overlaid on cosmic ray muons from data, labeled by sMask-RCNN

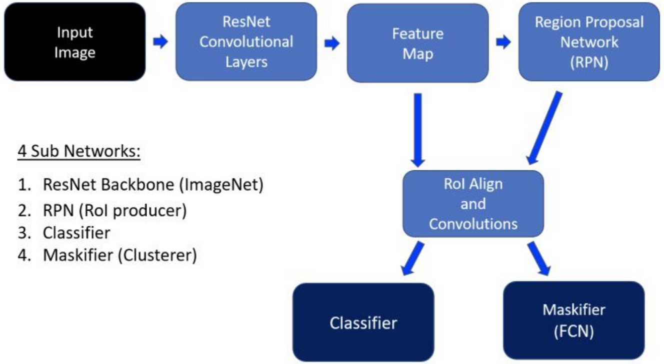


Figure 3: Network Architecture for Mask-RCNN in MicroBooNE.

Cosmic ray muon clustering for the MicroBooNE liquid argon time projection chamber using sMask-RCNN
https://arxiv.org/pdf/2201.05705.pdf

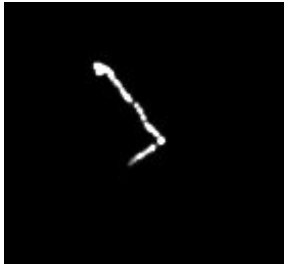
Data for Mask-RCNN

a) Digitized Image
(512x512)

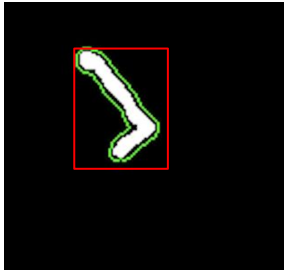
b) Pedmap
(512x512)

c) Pedestal subtracted
(Blurred -> Thresholded)

Input for Network



Digitized image
w/o noise



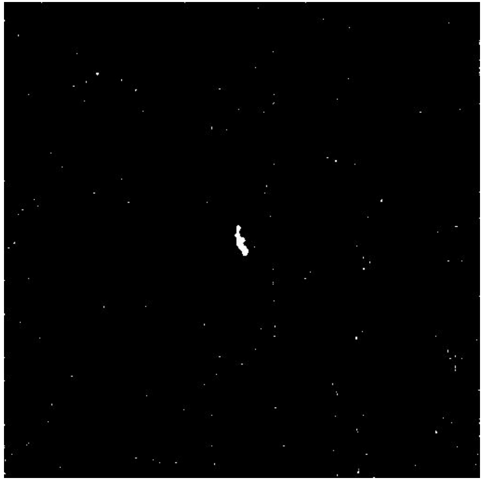
Selecting the
cluster using OpenCV

- Shape Attributes
  - Polygon
  - Coordinates of polygon
  - bbox
  - area
- Region Attributes
  - ER or NR (category)
  - Name & index

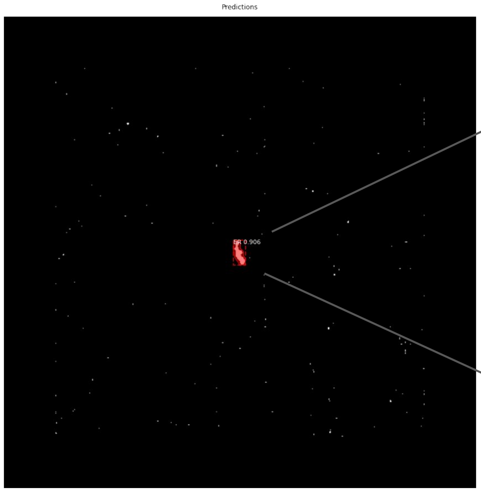
Target for Network

Annotations are done using OpenCV.

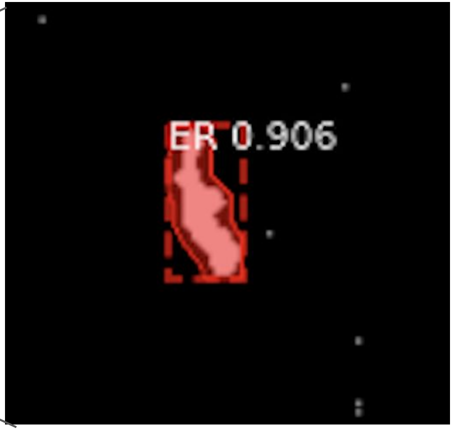
Preliminary Result:



Input Image

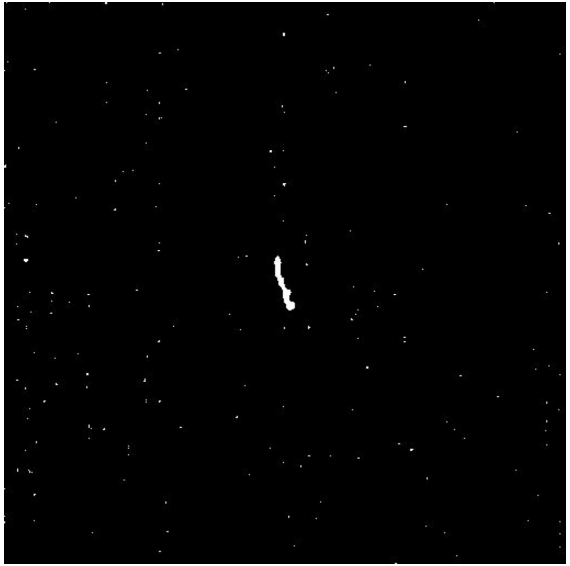


Prediction

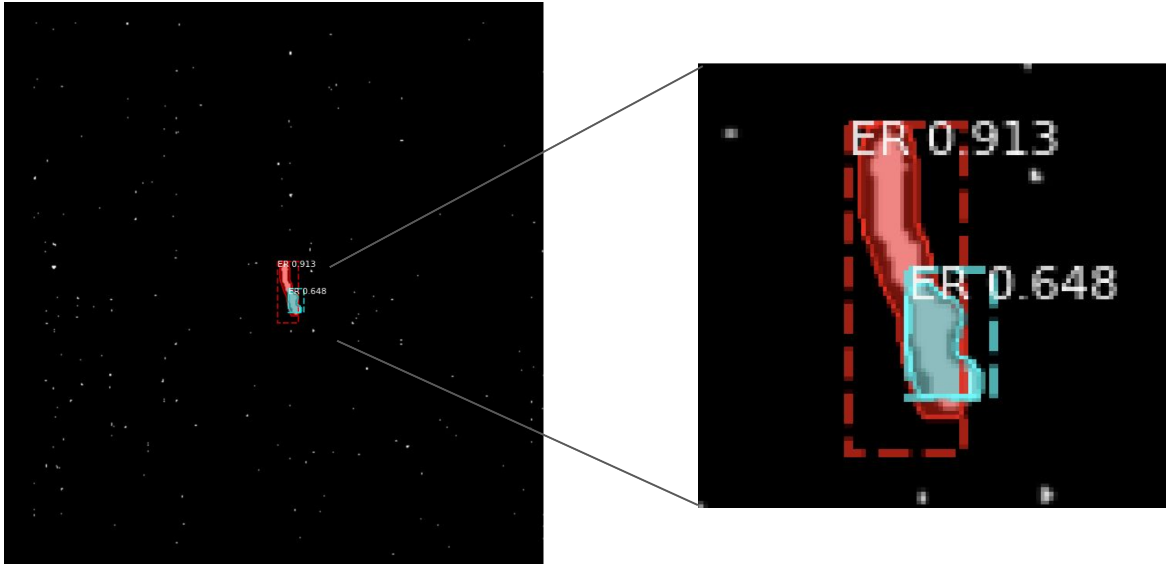


- ❖ 60 keV ER track.

60 keV ER track

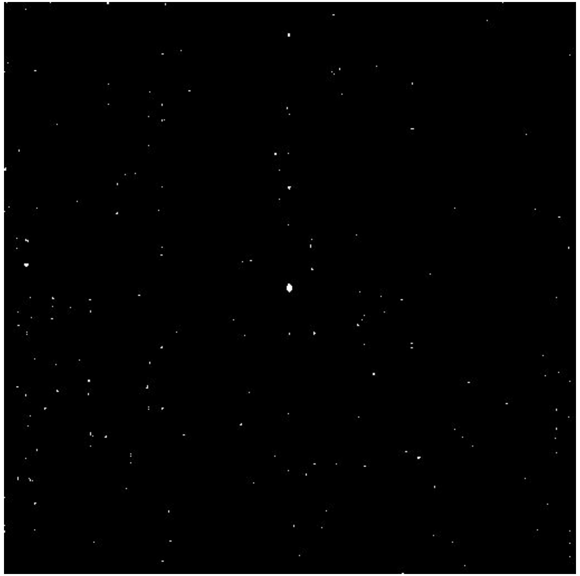


Input Image

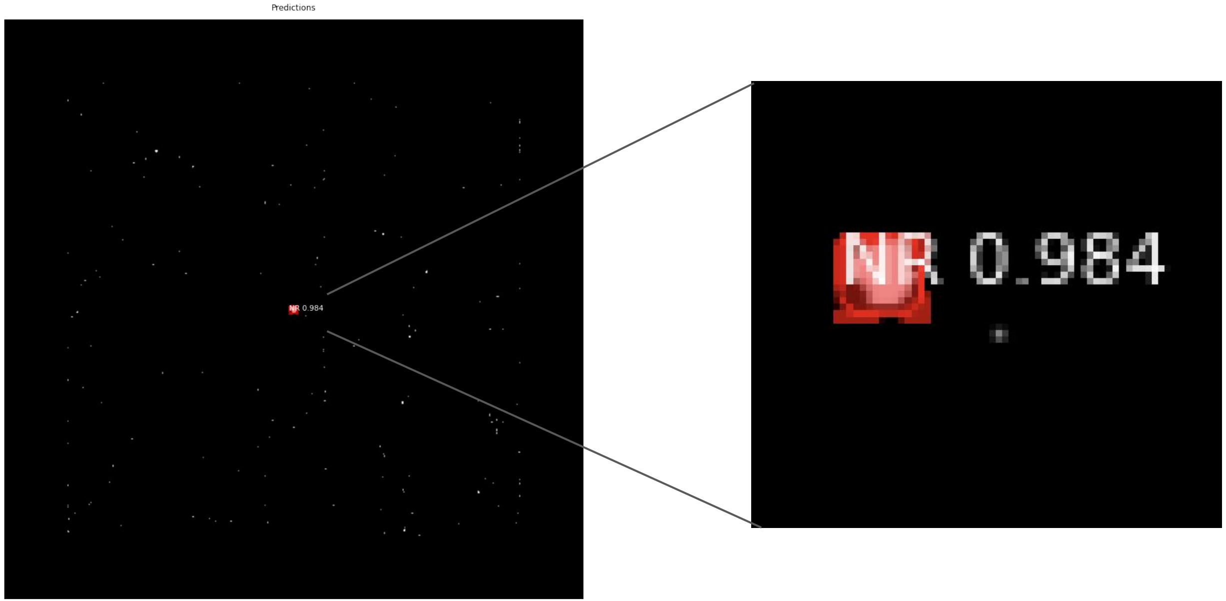


Prediction

60 keV NR track



Input Image



Prediction