NC/Sterile Group Meeting

March 31, 2019

NC Selection Associated Tree-based Algorithms Study I

Shaokai Yang
University of Cincinnati

Department of Physics

Outline

Analysis Motivation

- Making NC Interactions selection associated Tree-based black box models explainable;

- Better understanding of Tree-based algorithm classification capability for various NOvA detectable interactions with the reconstructed event-variables at hand;

Analysis Strategy

1. To limit input discriminating event-variables for decreasing model training complexity;

2. To tune a series of Algorithm Hyper-Parameters (AHP) to produce corresponding trained models;

3. To contrast with the series of model responses to understand the relative influence caused by AHP setting.

Analysis Setting I

- Employed Framework:
  1. Training Phase: *TMVA*[1];
  2. Application Phase: *CAFAna*[2];

- Employed Tagged Release:
  1. Training input files producing: *S17-10-30*;
  2. Application phase performing: *R17-08-22-prod3nus17.c*;

- Employed Machine Learning Algorithms :
  1. Decision Tree;
  2. Adaptive Boosting;
  3. Boosted Decision Trees (BDT);

Analysis Setting II

- Employed Input File:

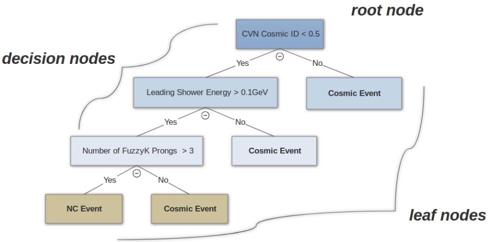
  1. NC Signal: Third Production FD Nonswap (DeCAF) Files;
  2. CC Backgrounds: Third Production FD Nonswap (DeCAF) Files;
  3. Cosmic Backgrounds*: Third Production Cosmic Trigger Files;

Type†	NC Signal	CC Background	Cosmic Background*
Number††	485024	2018536	242512

* Cosmic Backgrounds are the so-called ▸ Outlier in our study which are not used in the training phase.
† Type of Particle Interactions (Events) who can be detected by NOvA detectors.
†† Number of Events who passed the pre-selection cuts, and used for the model training and testing (half-and-half).

Decision Tree

- A decision tree is a binary structured classifier which was formed by a series of if-then-else decision rules, the so-called decision nodes;
- The sorting process will terminate at a leaf node, which labels the input event as signal or background;
- It is easy for humans to understand, but susceptible to statistical fluctuations of the input training events.



NC and Cosmic Separation Decision Tree Sample

Adaptive Boosting (AdaBoost)

- Overview – Pros and Cons
- Model (Tree) Building Procedure – Training & Weighting
- Derivation Process – Mathematics Point of View
- Variant – Confidence-Rated Predictions

Overview

- Boosting is a method of improving the classification power and enhancing the stability concerning statistical fluctuations in the training events of the machine learning algorithm(s);
- It performs the above advantage by sequentially applying one algorithm to reweighted (boosted) versions of the training events and then taking a weighted majority vote of the series of trained models;
- It was widely used in conjunction with various types of machine learning algorithms (weak learners†) to converge to a strong learner;
- AdaBoost is adaptive in the sense that the next model is tweaked in favor of those events misclassified by the previously trained model;
- *It is extremely sensitive to outlier and noisy data.*

† Weak Learner : its performance is slightly better than random guessing.

Model (Tree) Building Procedure I

- AdaBoost produces a stronger classifier by training one or more algorithm(s) (decision tree in our study) sequentially. Each training procedure is also called one training iteration;

- In the first iteration, the first tree $h_1(x)$, is trained with the original event weights. $h_1(x)$ takes an event x (with whose event-variables) as input and returns a value (prediction) representing the class (interaction type) of the event;

- The second tree is trained using a modified training events where the previously misclassified events are multiplied by a common boost weight (also called event weight, β_i);

Model (Tree) Building Procedure II

- The boost weight is derived from the misclassification rate of the previous tree, i.e. $\beta_2 = \frac{1-e_1}{e_1}$, where e_1 is the misclassification rate of the first tree and β_2 is the boost weight for the second tree training;

- The event weights of the entire training events are then renormalized so that the sum of the event weights remains constant;

- This procedure is repeated to built the forest. The training phase stops once it meets one of the stop criteria which is specified in the BDT configuration, for example, the tree number setting;

Model (Tree) Building Procedure III

- The boosted classifier then can be expressed as :

$$H_i(x) = \frac{1}{I} \cdot \sum_{i=1}^{I} \alpha_i \cdot h_i(x) \tag{1}$$

$$\alpha_i = ln(\beta_i)$$

- The output of $h_i(x)$ is encoded for signal and background as $+1$ and -1 respectively. Therefore the sign of output of $H_i(x)$ identifies the predicted event interaction type and the absolute value gives the confidence in that classification;

- The performance may be further improved by forcing a slow learning and allowing a larger number of boost steps instead. It performs by a parameter, learning rate, giving as an exponent to the boost weight, $\beta \rightarrow \beta^l$.

Derivation Process I

This derivation follows Rojas[3]

- Binary DataSet: $(x_1, y_1), \cdots, (x_T, y_T)$

  where x_t is the input training event, and y_t is the associated label ($y_t \in \{-1, 1\}$). $y_t = -1$(background), $y_t = 1$(signal);

- Weak Classifiers: $(h_1, \cdots, h_j)$

- (i-1)-stage Boosted Classifiers:

$$H_{i-1}(x_t) = \alpha_1 h_1(x_t) + \cdots + \alpha_{i-1} h_{i-1}(x_t) \tag{2}$$

- i-th iteration: extending to a better classifier by adding a $\alpha_i h_i$.

$$H_i(x_t) = H_{i-1}(x_t) + \alpha_i h_i(x_t) \tag{3}$$

Derivation Process II

FINDING $\alpha_i h_i$

1. The boosting procedure is now employed to adjust the parameters therefore the deviation between the final model response and the true value obtained from the training events is minimised;

2. The deviation is measured by loss-function, which is also called sum err. Adaboost is based on exponential loss::

$$L(H, y) = \sum_{t=1}^{T} e^{-y_t H_i(x_t)} \tag{4}$$

3. However, exponential loss has the shortcoming that it lacks robustness in presence of outliers or mislabelled events.

Derivation Process III

4. Letting $\omega_t^1 = 1$ and $\omega_t^i = e^{-y_t H_{i-1}(x_t)}$ when $i > 1$, then:

$$L(H, y) = \sum_{t=1}^{T} \omega_t^i e^{-y_t \alpha_i h_i(x_t)} \tag{5}$$

5. The input events can be split into two groups based on the last classifier, h_i, prediction: 1) correctly classified events ($y_t h_i(x_t) = 1$), and 2) misclassified events ($y_t h_i(x_t) =$ -1).

$$\begin{aligned} L(H, y) &= \sum_{y_t = h_i(x_t)} \omega_t^i e^{-\alpha_i} + \sum_{y_t \neq h_i(x_t)} \omega_t^i e^{\alpha_i} \\ &= \sum_{t=1}^{T} \omega_t^i e^{-\alpha_i} + \sum_{y_t \neq h_i(x_t)} \omega_t^i (e^{\alpha_i} - e^{-\alpha_i}) \end{aligned} \tag{6}$$

Derivation Process IV

6. To minimize L, the selected h_i need to minimize $\sum_{y_t \neq h_i(x_t)} \omega_t^i$.

$$\frac{dL}{d\alpha_i} = \frac{d(\sum_{y_t = h_i(x_t)} \omega_t^i e^{-\alpha_i} + \sum_{y_t \neq h_i(x_t)} \omega_t^i e^{\alpha_i})}{d\alpha_i} \tag{7}$$

7. After set the equation 7 to zero, we get:

$$\alpha_i = \frac{1}{2} ln \left(\frac{\sum_{y_t = h_i(x_t)} \omega_t^i}{\sum_{y_t \neq h_i(x_t)} \omega_t^i} \right) \tag{8}$$

Real Adaptive Boosting

- A widely used modification of Eq.1 for the the result of the combined classifier from the forest is to use the training purity in the leaf node as respective signal or background weights rather than relying on the binary decision. This is then called Real AdaBoost.
- We describe several improvements to Freund and Schapires AdaBoost boosting algorithm, particularly in a setting in which hypotheses may assign confidences to each of their predictions
- boosting in an extended framework in which each weak hypothesis generates not only predicted classifications, but also self-rated confidence scores which estimate the reliability of each of its predictions.

NC Selection Classifiers Study

- Algorithm Input Event-Variables
- Employed Algorithm Hyper-Parameters Overview
- Tuning AHP and Response Comparison
- Conclusion

Algorithm Input Variable I – CVN NC ID

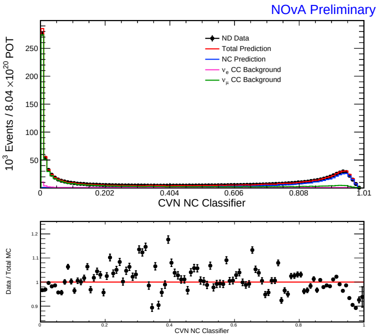


Figure: ND Data Vs MC Agreement

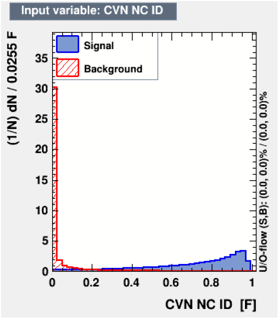


Figure: FD NC Vs CC Distribution Plot (MC)

Algorithm Input Variable II – Leading Prong Length

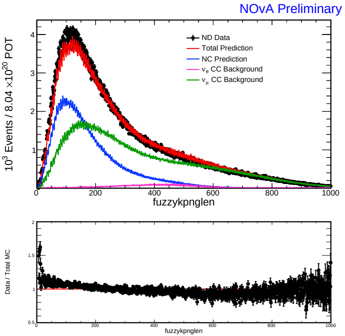


Figure: ND Data Vs MC Agreement

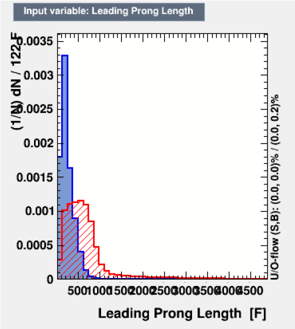


Figure: FD NC Vs CC Distribution Plot (MC)

Employed Hyper-Parameters

- NTrees – Number of trees in the forest
- Max Depth – Max depth of the decision tree allowed
- MinNodeSize – Minimum percentage of training events required in a leaf node
- NodePurityLimit – nodes with signal purity
- AdaBoostBeta – Learning rate for AdaBoost algorithm
- nCuts – Number of splitting point in each variable
- UseYesNoLeaf – Use the purity=S/(S+B) as classification of the leaf node
- SeparationType – Separation criterion for node splitting
- PruneMethod – Method used for pruning (removal) of statistically insignificant branches

Boosting Type Comparison – Model Response

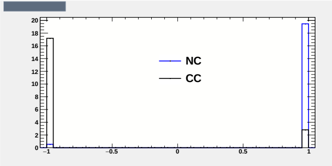


Figure: AdaBoost

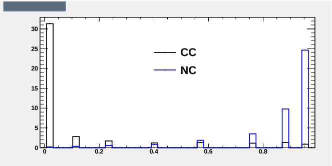


Figure: Real AdaBoost

AHP	Setting
NTrees	1
Max Depth	3
MinNodeSize	2%
NodePurityLimit	0.3
AdaBoostBeta	1
nCuts	100

Boosting Type Comparison – Tree Structure

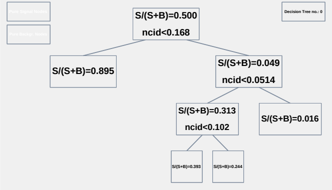


Figure: AdaBoost

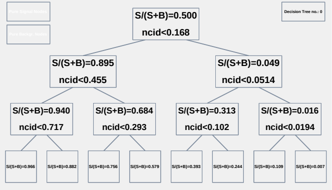
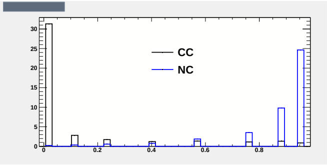
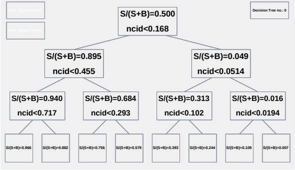


Figure: Real AdaBoost

AHP	Setting
NTrees	1
Max Depth	3
MinNodeSize	2%
NodePurityLimit	0.3
AdaBoostBeta	1
nCuts	100

Real Adaptive Boosting I

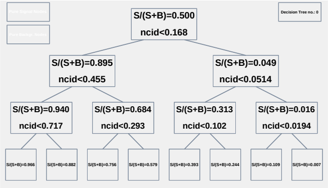




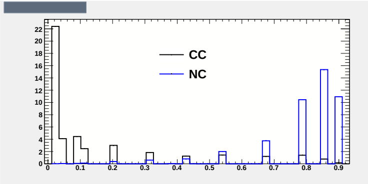
NC	CVN NC ID	Model Response
1	0.782835	0.965531
2	0.629479	0.882409
3	0.299279	0.7559
4	0.19299	0.578752
5	0.109766	0.393449
6	0.0693809	0.243708
7	0.0488229	0.10933
8	3.31009e-08	0.00694278

CC	CVN NC ID	Model Response
1	0.742959	0.965531
2	0.644474	0.882409
3	0.32087	0.7559
4	0.22879	0.578752
5	0.149629	0.393449
6	0.094832	0.243708
7	0.0416455	0.10933
8	0.000210658	0.00694278

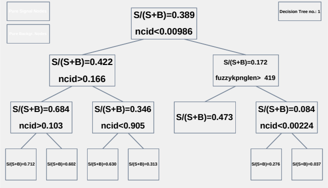
Tree Number Model Response



(a) Tree One



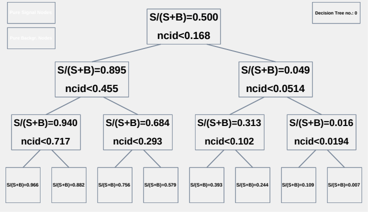
(b) Model Response



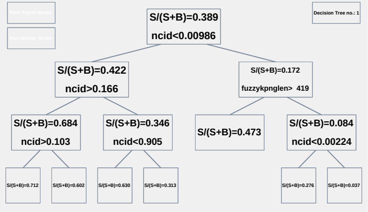
(c) Tree Two

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
MinNodeSize	0.1%
NodePurityLimit	0.9
AdaBoostBeta	0.1
nCuts	100
NTrees	2

Tree Number Model Response II



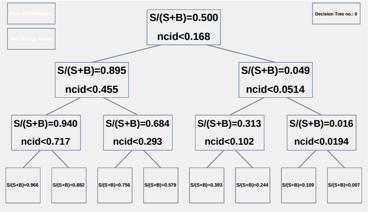
(d) Tree One



(e) Tree Two

Event	Prong Length	CVN NC ID	Model Response
1	176.91	0.966517	0.910576
2	146.291	0.782658	0.858628
3	160.882	0.577653	0.789132
4	480.924	0.299279	0.683361
5	220.044	0.211275	0.535251
6	315.776	0.133076	0.427621
7	103.638	0.0796748	0.320494
8	268.157	0.0210519	0.208144
9	153.164	0.0139447	0.12254
10	229.243	0.00255922	0.0833009
11	1165.73	0.00711549	0.0509888
12	1079.07	3.31009e-08	0.0118476

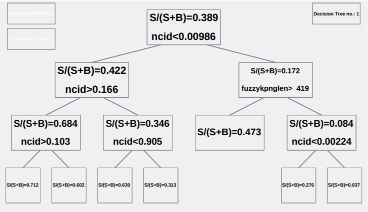
First Tree Number Model Response II



(f) First Tree

Event	Prong Length	CVN NC ID	Tree Response
1	176.91	0.966517	0.966
2	146.291	0.782658	0.966
3	160.882	0.577653	0.882
4	480.924	0.299279	0.756
5	220.044	0.211275	0.579
6	315.776	0.133076	0.393
7	103.638	0.0796748	0.244
8	268.157	0.0210519	0.244
9	153.164	0.0139447	0.109
10	229.243	0.00255922	0.007
11	1165.73	0.00711549	0.007
12	1079.07	3.31009e-08	0.007

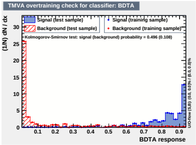
Second Tree Number Model Response II



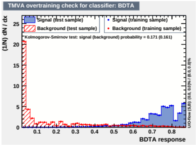
(g) Second Tree

Event	Prong Length	CVN NC ID	Tree Response
1	176.91	0.966517	0.630
2	146.291	0.782658	
3	160.882	0.577653	
4	480.924	0.299279	
5	220.044	0.211275	
6	315.776	0.133076	
7	103.638	0.0796748	
8	268.157	0.0210519	
9	153.164	0.0139447	
10	229.243	0.00255922	
11	1165.73	0.00711549	
12	1079.07	3.31009e-08	

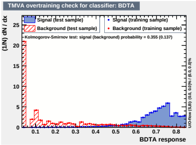
Mutative Tree Number Model Response I



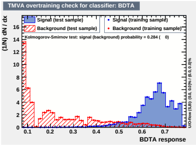
(h) 10



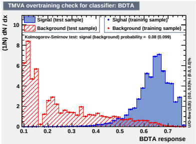
(i) 50



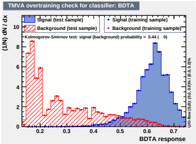
(j) 100



(k) 500



(l) 1000



(m) 3000

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
MinNodeSize	0.1%
NodePurityLimit	0.9
AdaBoostBeta	0.1
nCuts	100

Mutative Tree Number Model Response II

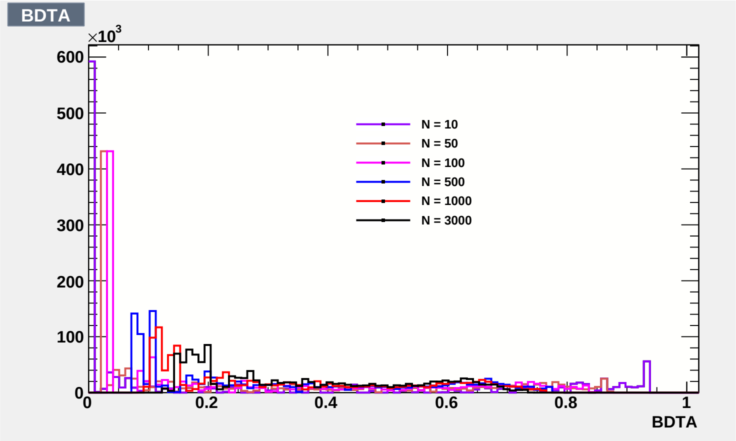


Figure: Background (1009268 CC Events) Distributions from Six trained models

Mutative Tree Number Model Response III

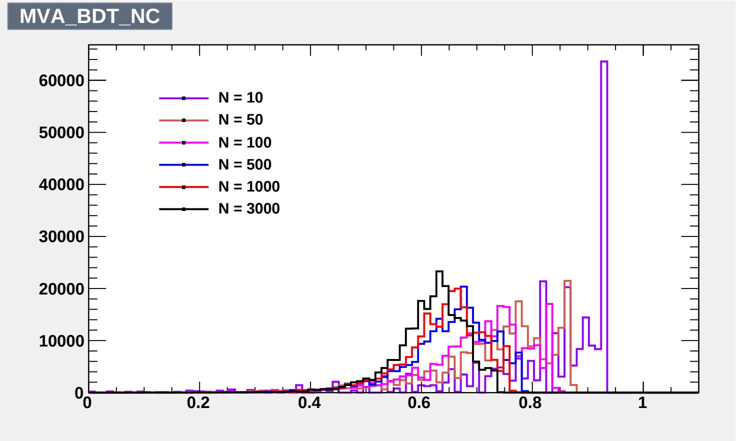


Figure: Signal (242512 NC Events) Distributions from Six trained models

Mutative Tree Number Model Response IV

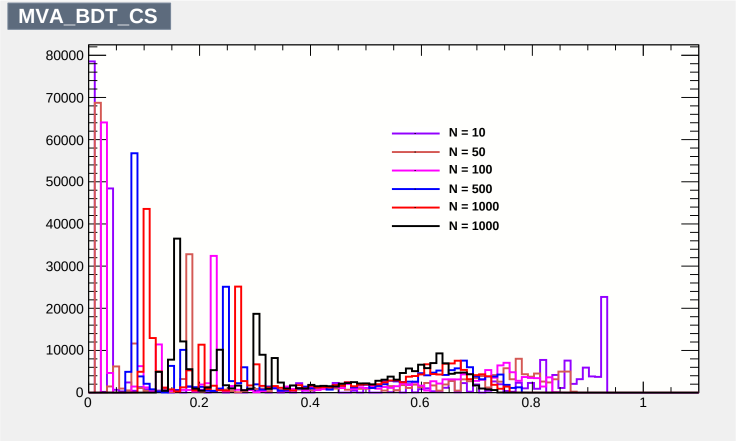
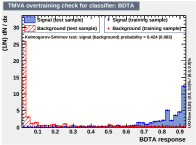
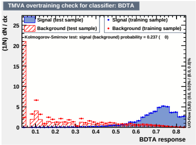


Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

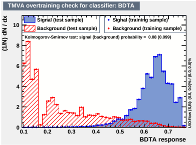
Learning Rate I



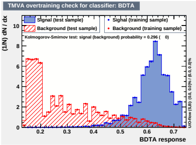
(a) 0.001



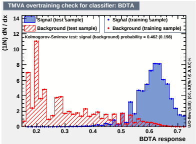
(b) 0.01



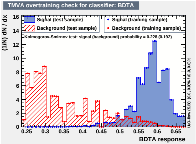
(c) 0.1



(d) 0.3



(e) 0.5



(f) 1.0

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
MinNodeSize	0.1%
NodePurityLimit	0.9
NTrees	1000
nCuts	100

Mutative Learning Rate Model Response II

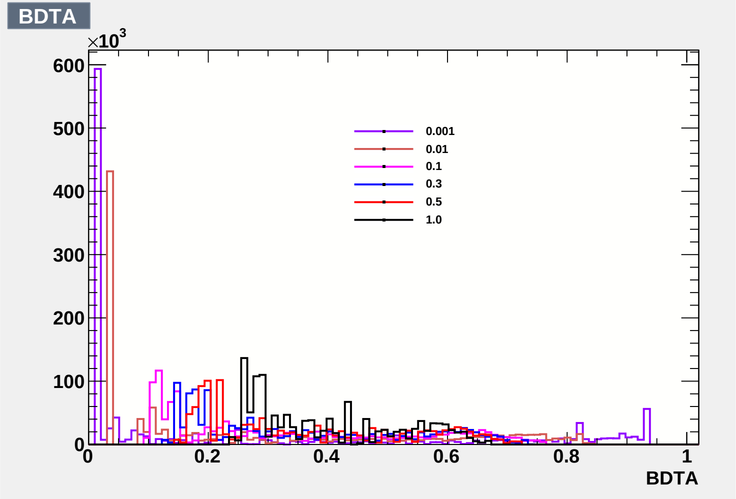


Figure: B... (1000268 CC E...) Distribution f... Si...

Mutative Learning Rate Model Response III

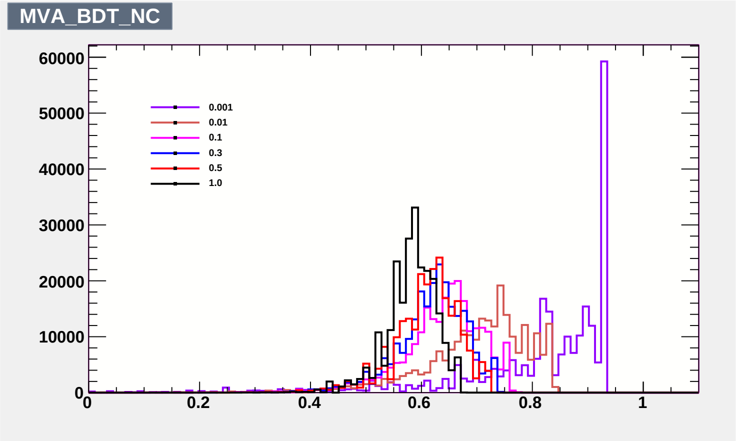


Figure: Signal (242512 NC Events) Distributions from Six trained models

Mutative Learning Rate Model Response IV

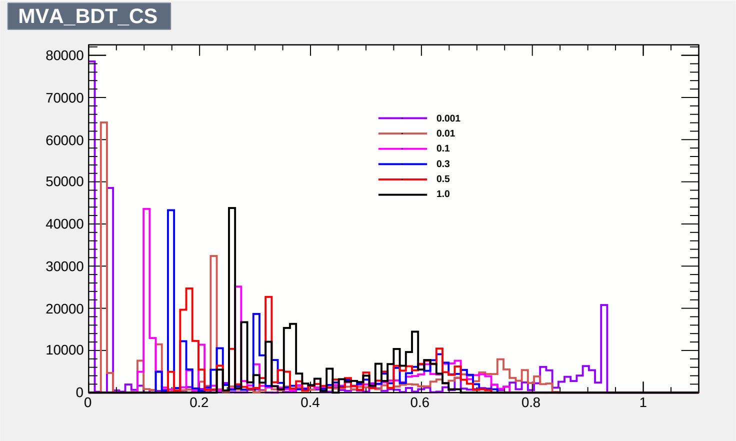
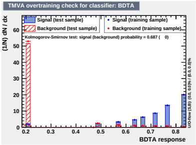
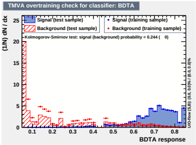


Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

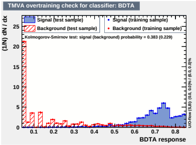
Mutative Splitting Points Model Response I



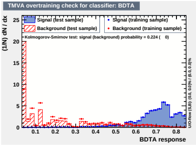
(a) 10



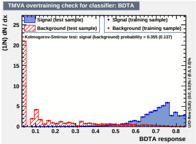
(b) 50



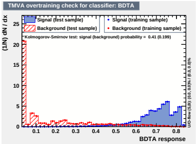
(c) 100



(d) 500



(e) 1000



(f) 3000

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
MinNodeSize	0.1%
NodePurityLimit	0.9
AdaBoostBeta	0.1
NTrees	100

Mutative Splitting Points Model Response II

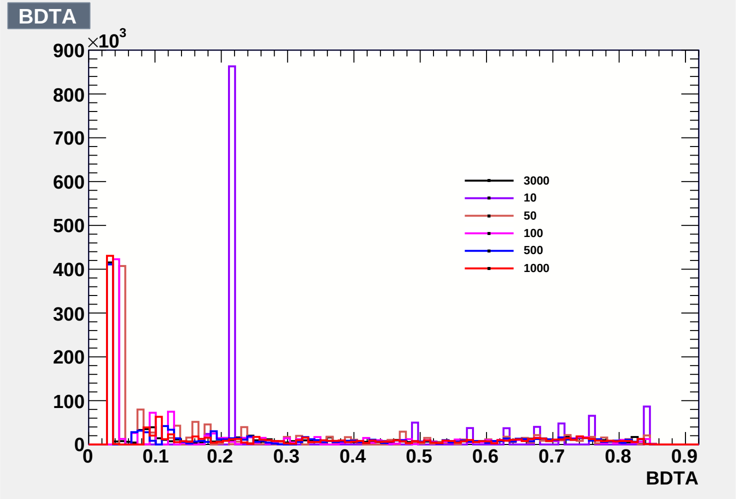


Figure: Background (1000268 CC Events) Distribution for a Six-trained model...

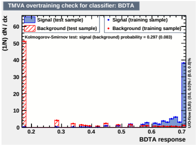
Mutative Splitting Points Model Response III

Figure: Signal (242512 NC Events) Distributions from Six trained models

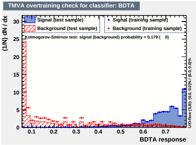
Mutative Splitting Points Model Response IV

Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

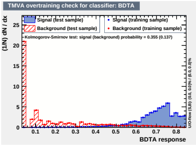
Mutative Tree Depth Model Response I



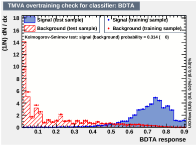
(a) 1



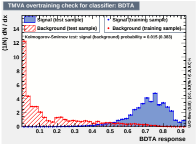
(b) 2



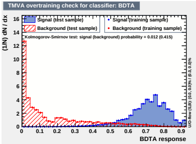
(c) 3



(d) 5



(e) 10



(f) 100

AHP	Setting
BoostType	Real AdaBoost
NTrees	100
MinNodeSize	0.1%
NodePurityLimit	0.9
AdaBoostBeta	0.1
nCuts	100

Mutative Tree Depth Model Response II

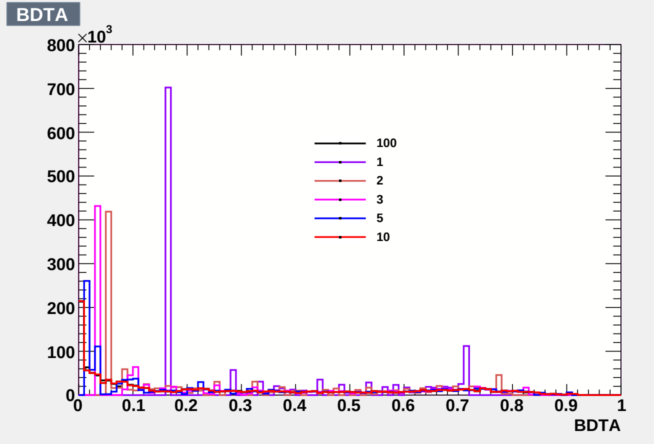


Figure: Background (1009268 CC Events) Distributions from Six trained models

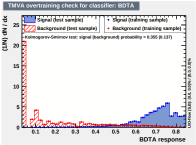
Mutative Tree Depth Model Response III

Figure: Signal (242512 NC Events) Distributions from Six trained models

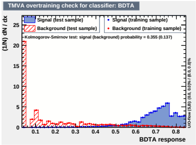
Mutative Tree Depth Model Response IV

Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

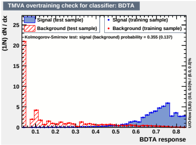
Mutative Node Purity Limit Model Response I



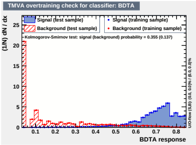
(a) 0.001%



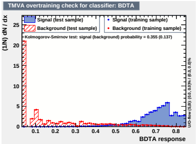
(b) 0.1



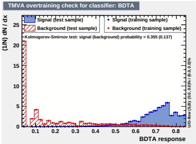
(c) 0.2



(d) 0.5



(e) 0.9



(f) 1

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
MinNodeSize	0.1%
NTrees	100
AdaBoostBeta	0.1
nCuts	100

Mutative Node Purity Limit Model Response II

Figure: Background (1009268 CC Events) Distributions from Six trained models

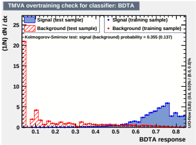
Mutative Node Purity Limit Model Response III

Figure: Signal (242512 NC Events) Distributions from Six trained models

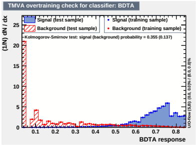
Mutative Node Purity Limit Model Response IV

Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

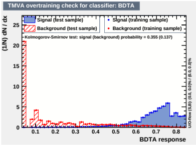
Mutative Minimum Node Size Model Response I



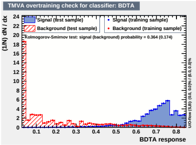
(a) 0.001%



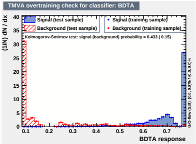
(b) 0.01%



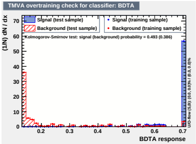
(c) 0.1%



(d) 1%



(e) 10%



(f) 20%

AHP	Setting
BoostType	Real AdaBoost
Max Depth	3
NodePurityLimit	0.9
NTrees	100
AdaBoostBeta	0.1
nCuts	100

Mutative Minimum Node Size Model Response II

Figure: Background (1009268 CC Events) Distributions from Six trained models

Mutative Minimum Node Size Model Response III

Figure: Signal (242512 NC Events) Distributions from Six trained models

Mutative Minimum Node Size Model Response IV

Figure: Outlier (242512 Cosmic Events) Distributions from Six trained models

References

- A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, and H. Voss, "TMVA: Toolkit for Multivariate Data Analysis," PoS A CAT 040 (2007) [physics/0703039].
- D. Rocco "CAF Introduction" NOvA Internal Document DocDB:11002
- Rojas, R. (2009) " AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting." Freie University, Berlin, Tech. Rep
- Robert E. Schapire, Yoram Singer, " Improved Boosting Algorithms Using Confidence-rated Predictions" Machine Learning , Vol.37, No. 3, (297-336) 1999.

Terms and definitions

- Outlier:
- dd
- dd