AdaBoost and GradBoost

Giulia Maineri

Università degli studi di Milano

April 2022

Table of Contents

BookMethod options

Training-BookMethod

The following options where kept unchanged in each training:

- !H; in order not to print a method-specific help message
- !V; necessary to deactivate "verbose mode", which prints explanations of what's going on
- NTrees=850; number of trees in the forest
- MinNodeSize=5%; minimum percentage of training events required in a leaf node
- MaxDepth=3; maximum depth of the decision tree allowed
- SeparationType=GiniIndex; separation criterion for node splitting
- nCuts=20;number of grid points in variable range used in finding optimal cut in node splitting

Training-BookMethod

The following options were changed in order to find the optimal algorithm:

- BoostType: AdaBoost or GradBoost
- AdaBoostBeta or Shrinkage
- Bagging or no Bagging

When bagging was on, a Sample Fraction of 0.6 was used.

Input Variables

Input Variables

Many different variables were tested for BDT analysis.

- E_T^{miss}, computed as:

$$-\left[\sum_e \vec{p}_T^{(e)} + \sum_\mu \vec{p}_T^{(\mu)} + \sum_\gamma \vec{p}_T^{(\gamma)} + \sum_\tau \vec{p}_T^{(\tau)} + \sum_{jet} \vec{p}_T^{(jet)} + \sum_x \vec{p}_T^{(x)}\right]$$

  where x indicates tracks or energy deposits not identified with any physics object.

- E_T^{miss} significance S, defined as

$$S = \frac{E_T^{miss}}{\sigma_{E_T^{miss}}} \qquad (1)$$

- m_{ll}, invariant mass of the two leptons, defined as:

$$m_{ll} = \sqrt{2p_T^{l1}p_T^{l2}[\cosh(\eta^{l1} - \eta^{l2}) - cos(\Phi^{l1} - \Phi^{l2})]} \qquad (2)$$

Input Variables

- m_T, transverse mass of the $\gamma - E_T^{miss}$ system, defined as

$$m_T = \sqrt{2p_T^{\gamma}E_T^{miss}(1 - \cos(\Phi^{\gamma} - \Phi^{E_T^{miss}}))} \tag{3}$$

- $\Delta\Phi(\vec{E}_T^{miss}, \vec{p}_T^{\gamma ll})$, difference in azimuthal angle between E_T^{miss} and $ll\gamma$ system.
- $\Delta\Phi(\vec{E}_T^{miss}, \vec{p}_T^{closest})$, difference in azimuthal angle between E_T^{miss} and the closest identified object to it.

Input Variables

- $\Delta\Phi(\vec{E}_T^{miss}, \vec{p}_T^{closestjet})$, difference in azimuthal angle between E_T^{miss} and the closest jet to it.
- p_T^{γ}, transverse momentum of the leading photon.
- $p_T^{balance}$, defined as

$$p_T^{balance} = \frac{p_T^{\gamma + E_T^{miss}}}{p_T^{ll}} \tag{4}$$

Linear correlation coefficients

Figure 1: Signal

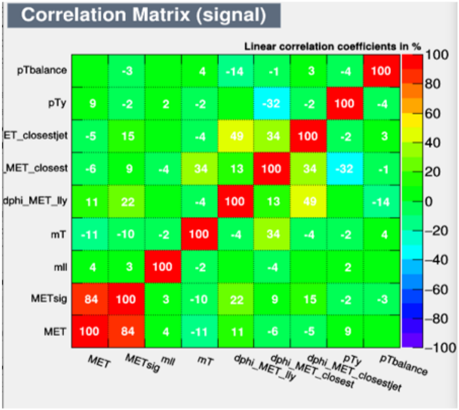
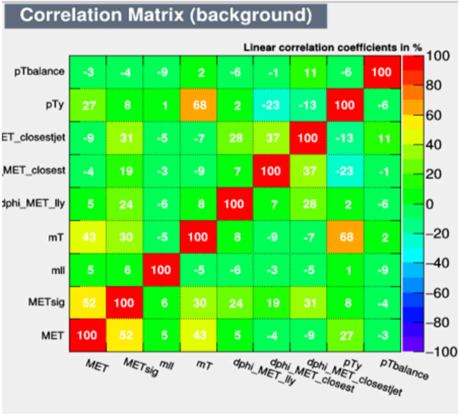


Figure 2: Background



GradBoost - no bagging

Shrinkage $\beta = 1$

A first training was done using $\beta = 1$, but returned unacceptable results, with BDT score polarized to $-1, 1$ and a very low ROC curve.
Results of this training were rejected.

Classifier output distribution-Test and training sample - $\beta = 0.05$

Figure 3: Output distribution

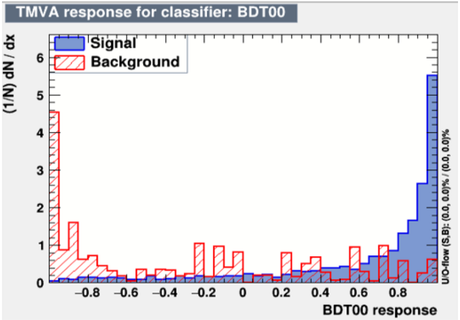
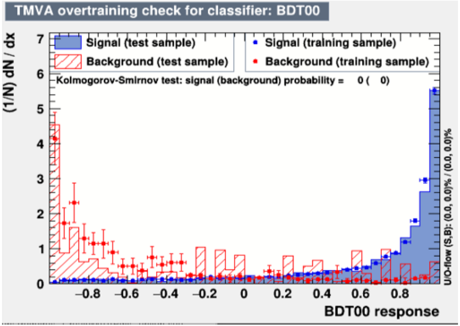


Figure 4: Overtraining control



Classifier output distribution-Test and training sample - $\beta = 0.1$

Figure 5: Output distribution

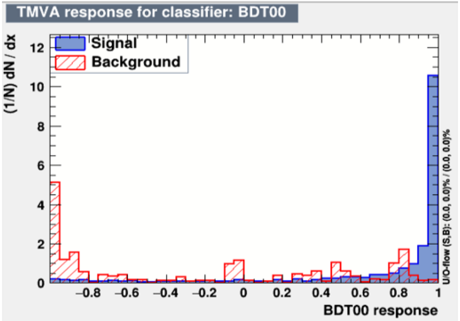
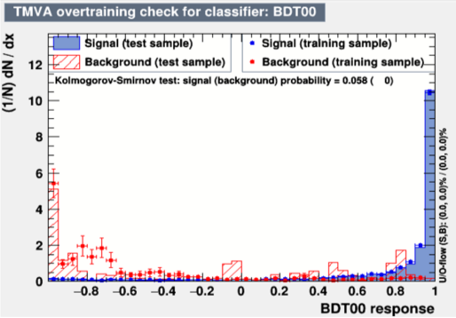


Figure 6: Overtraining control



Classifier output distribution-Test and training sample - $\beta = 0.2$

Figure 7: Output distribution

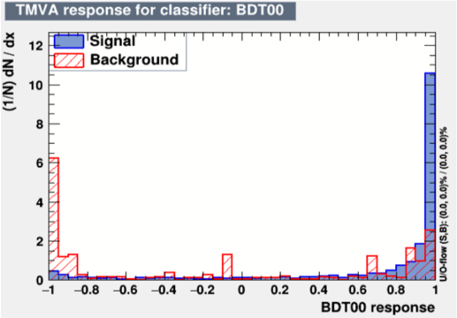
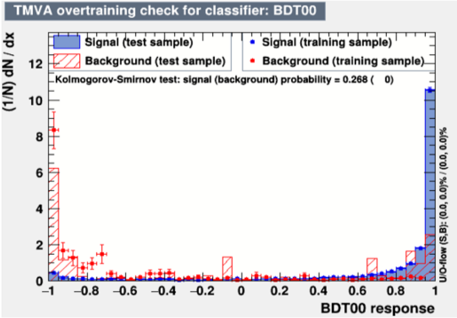


Figure 8: Overtraining control



GradBoost - with bagging

Classifier output distribution-Test and training sample - $\beta = 0.05$

Figure 9: Output distribution

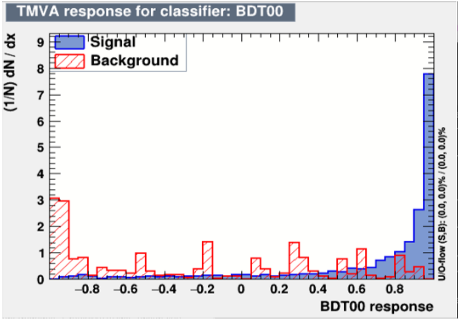
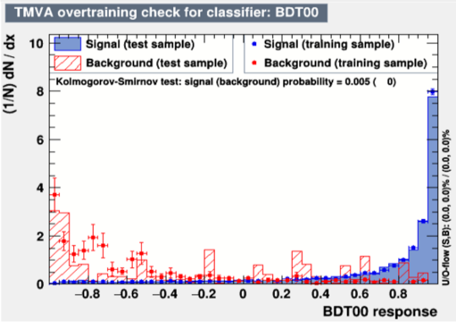


Figure 10: Overtraining control



Classifier output distribution-Test and training sample - $\beta = 0.1$

Figure 11: Output distribution

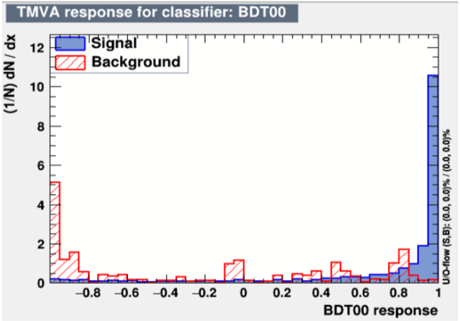
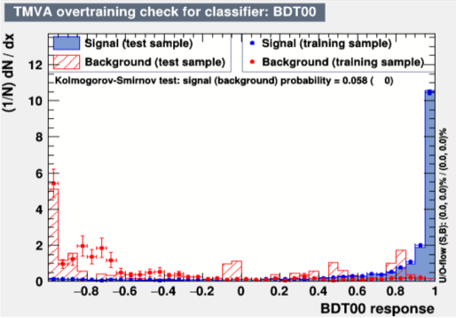


Figure 12: Overtraining control



Classifier output distribution-Test and training sample - $\beta = 0.2$

Figure 13: Output distribution

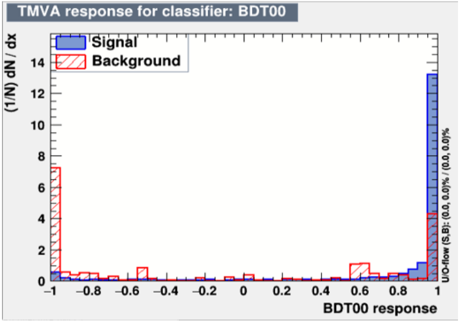
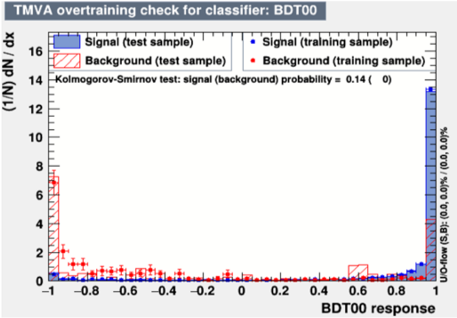


Figure 14: Overtraining control



AdaBoost - no bagging

Classifier output distribution-Test and training sample - ($\beta = 0.5$)

Figure 15: Output distribution

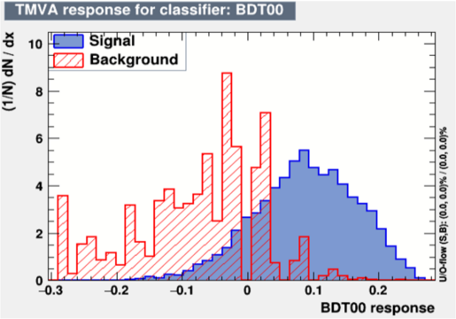
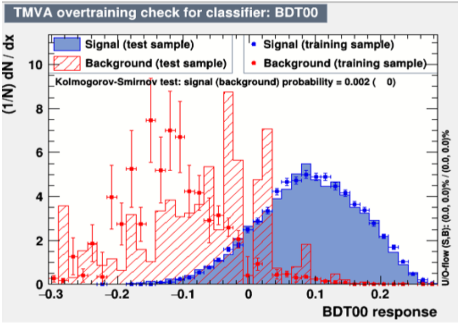


Figure 16: Overtraining control



Classifier output distribution-Test and training sample - ($\beta = 0.2$)

Figure 17: Output distribution

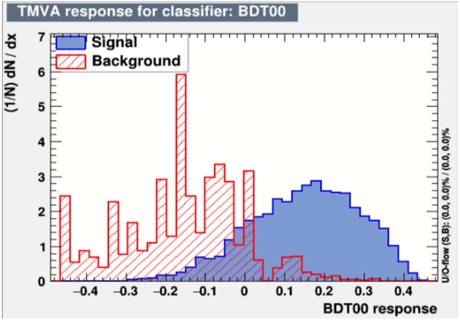
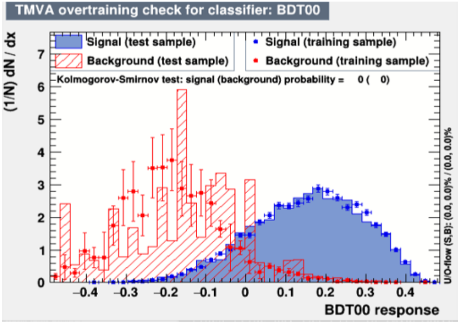


Figure 18: Overtraining control



Classifier output distribution-Test and training sample - ($\beta = 0.05$)

Figure 19: Output distribution

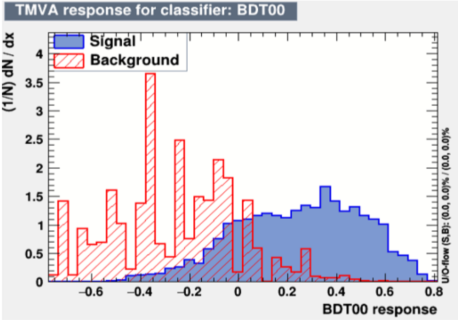
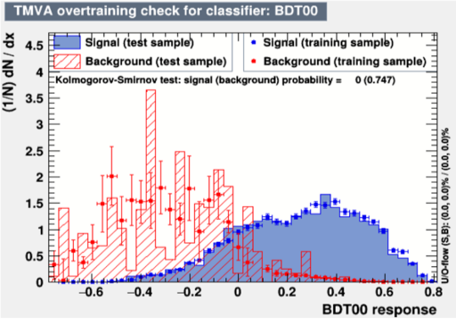


Figure 20: Overtraining control



Classifier output distribution-Test and training sample - ($\beta = 1$)

Figure 21: Output distribution

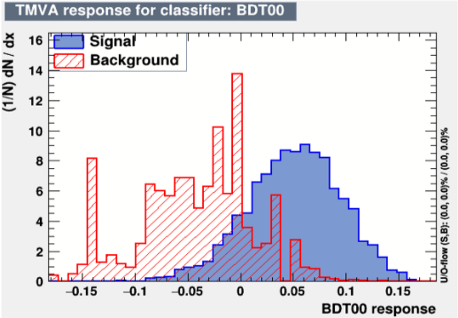
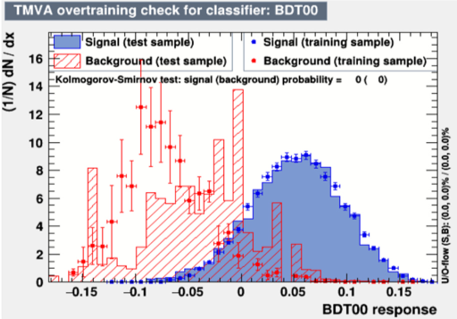


Figure 22: Overtraining control



AdaBoost - with bagging

Classifier output distribution-Test and training sample - ($\beta = 0.5$)

Figure 23: Output distribution

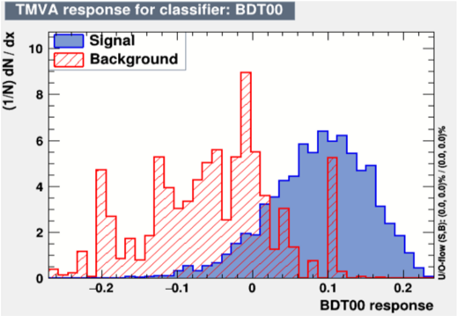
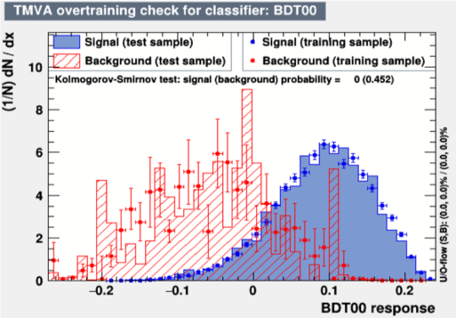


Figure 24: Overtraining control



Classifier output distribution-Test and training sample - ($\beta = 0.05$)

Figure 25: Output distribution

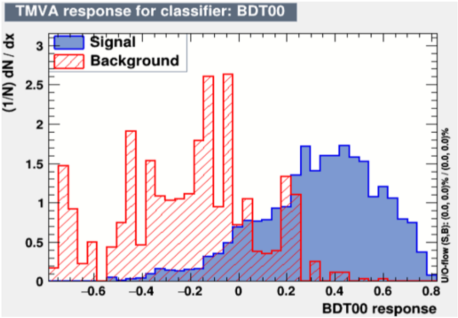
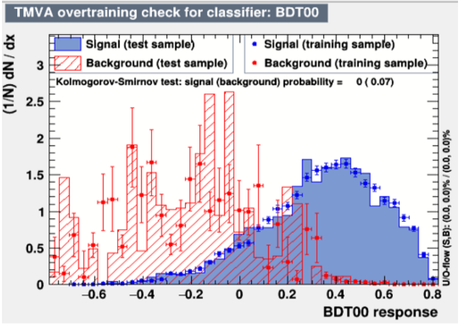


Figure 26: Overtraining control



Classifier output distribution-Test and training sample - $(\beta = 0.2)$

Figure 27: Output distribution

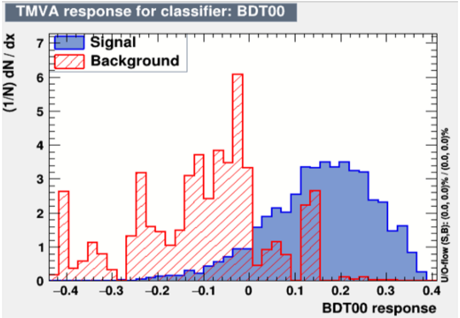
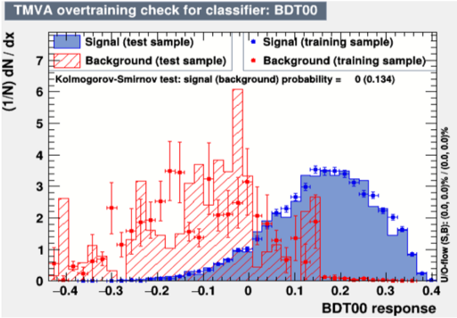


Figure 28: Overtraining control



Classifier output distribution-Test and training sample - ($\beta = 1$)

Figure 29: Output distribution

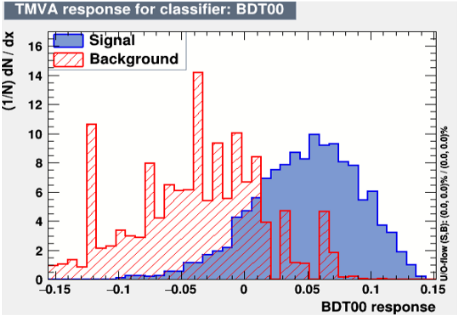
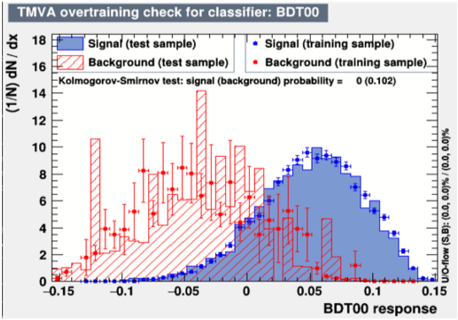


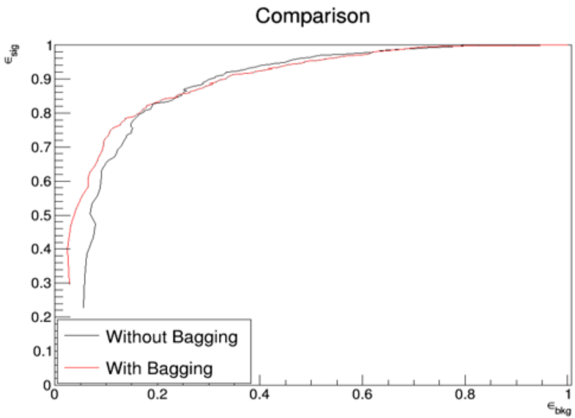
Figure 30: Overtraining control



Comparisons

Comparison

Figure 31: Comparison GradBoost with/without Bagging $\beta = 0.05$



Comparison

Figure 32: Comparison GradBoost

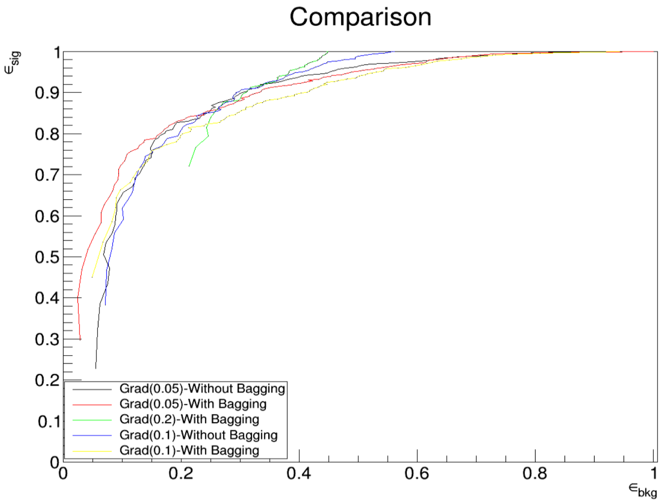


Figure 33: Comparison AdaBoost without Bagging

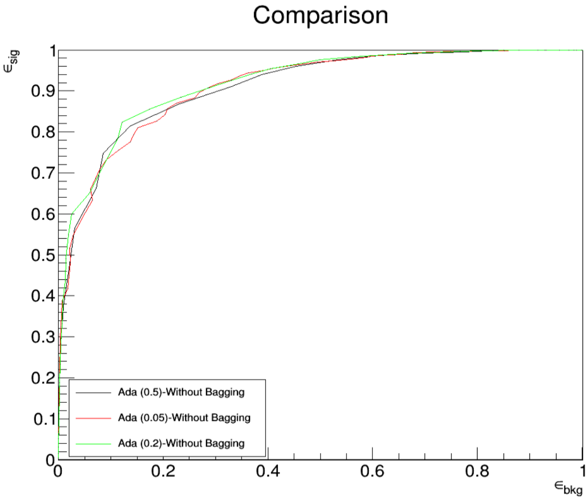


Figure 34: Comparison AdaBoost with/without Bagging ($\beta = 0.5$)

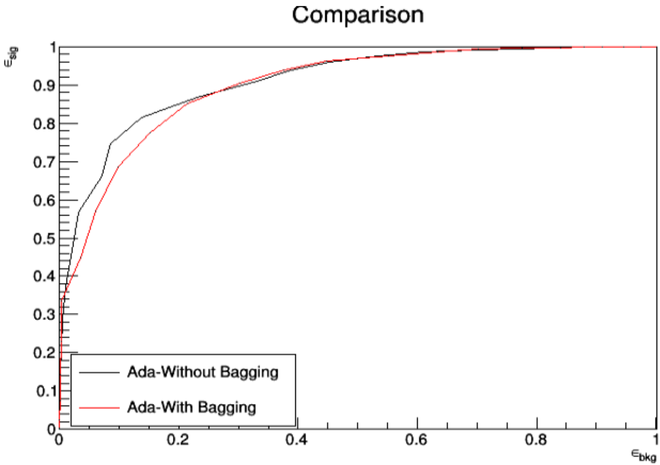


Figure 35: Comparison AdaBoost with/without Bagging ($\beta = 0.05$)

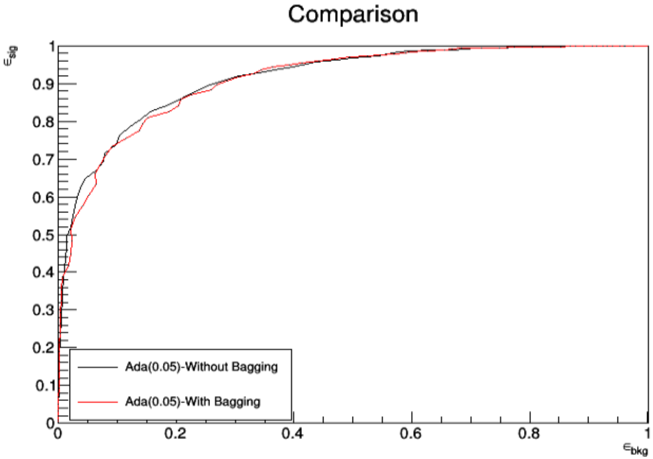
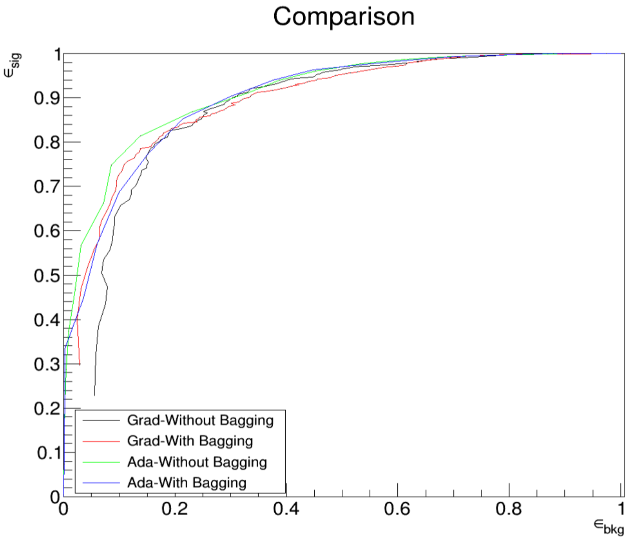
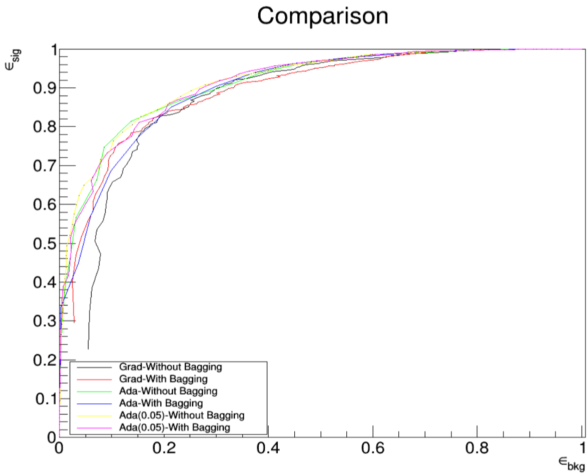


Figure 36: Comparison AdaBoost and Grad with/without Bagging ($\beta = 0.5$)



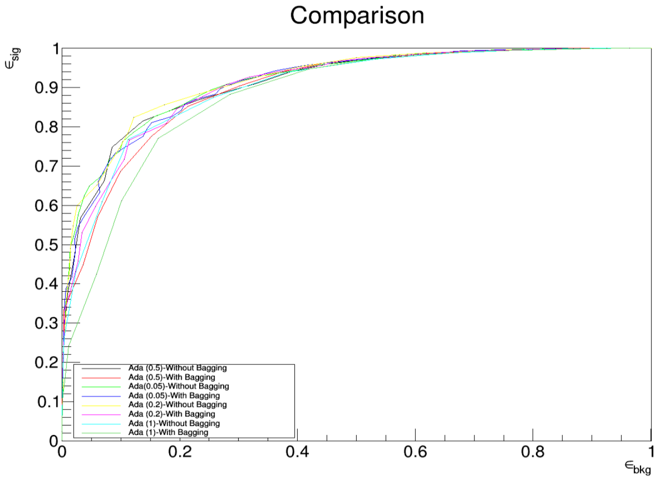
If not specified, $\beta = 0.5$ for both AdaBoost and GradBoost.

Figure 37: Comparison AdaBoost and Grad with/without Bagging



Comparison

Figure 38: Comparison AdaBoost with/without Bagging



Conclusions

Conclusions

With reference to ROC curves, AdaBoost without Bagging seems to perform better than Gradient Boost. Among the learning rates used in the different training trials, the best performance was given by $\beta = 0.2$.

ROC curves behaviour was observed especially in the first region of low background efficiency, as it is the region of interest.

If Gradient Boost increases its performance uses Bagging, this doesn't happen for AdaBoost.