GAN-AE and BumpHunter

Author(s): Louis VASLIN
Laboratoire de Physique de Clermont, France

0.1 Method

The methods presented in this paper combine two independent anomaly detection algorthm. The objective is to have a fulle analysis workflow that can give a global p-value and evaluate the number of signal events in any black box dataset.

0.1.1 GAN-AE

The GAN-AE method is an attempt at assiocating and Auto-Encoder architechture to a discriminant neural network in a GAN-like fashion. The reason for this particular setting is to use information that don't come only from the "reconstruction error" usualy used to train AEs. This could be seen as a alternative way to constrain the training of an AE. As discriminant network, a simple feed-forward Multi-Layer Perceptron (MLP) is used.

This method been inspired by the GAN algorithm, the two participants (AE and MLP) are trained alternatively with opposite objectives :

- The MLP is trained for a few epochs using the binary crossentropy (BC) loss function on a labeled mixture of original and reconstruted events, the objective being to expose the weeknesses of the AE.
- The AE is trained for a few epochs using a loss function combining the recnstruction error (here, the Mean Euclidean Distance between the input and output, or MED for short) and the BC loss of the MLP. In order to decorelate as much as possible the reconstruction error and the invariant mass, the distance correlation (DisCo) term is used (ref). The loss is then given by :

  $$loss_{AE} = BC + \varepsilon \times MED + \alpha \times DisCo$$

  With ε and α two hyperparameters used to balance the wights of each terms. In this case, the BC term is evaluated by giving reconstructed events to the MLP, but this time with the "wrong label", the objective being to misslead the MLP.
- Then the AE is evaluated on a validation set using a Figure of Merit that also combines the reconstruction error and some information from the MLP. The FoM used is given by :

  $$FoM = MED + (1 - Mean\ MLP\ output)$$

  This second term is preffered over the binary crossentropy because it seems to be more stable, which makes it more suitable to set a early stopping condition. As for the reconstruction error, $1 - Mean\ MLP\ output$ must be minimized. In fact, the closer to zero is this term, the better the AE is at misleading the MLP.

These three steps are repeated in a loop until the FoM fails to improve for five cycles. Once the AE has been trained, the MLP can be discarded seens it is not needed anymore. Then, the AE can be used as usual using only the reconstruction error (Euclidean distance) as discriminantive feature.

The GAN-AE hyperparameter used for the LHC Olympics are shown in Tab. **??**

	AE	MLP
Neurons per hidden layer	30/20/10/20/30	150/100/50
Number of epochs per cycle	4	10
Activation function	ReLU (sigmoid for output)	LeakyReLU (sigmoid for output)
Dropout	0.2 (hidden layers only)	
Early-stopping condition	5 cycles without improvment	

Table 1. Hyperparameters used for the GAN-AE algorithm.

0.1.2 BumpHunter

The BumpHunter algorithm is a hypertest that compares a data distribution with a reference and evaluates the p-value and significance of any deviation. To do so, BumpHunter will scan the two distributions with a sliding window of variable width. For each position and width of the scan window, the local p-value is evaluted. The window coresponding to the most significant deviations is then defined as the one with the smallest local p-value.

In order to deal with the look elsewhere effect and evaluate a global p-value, BupHunter generates pseudo-experiment by sampling from the reference histogram. The scan is the repeated for each pesudo-data histogram by comparing them with the original reference. This gives a local p-value distribution that can be compared with the local p-value obtained for the real data. Thus, a global p-avlue and significance is obatianed. In order to evaluate the number of signal events, the difference of event count in the bump window for the data and the reference can be used.
THe BumpHunter hyperparameters used for the LHC Olympics are shown in Tab. **??**

0.1.3 Full analysis workflow

The objective of this work is to use the Auto-Encoder trained withe the GAN-AE algorithm to reduce the background and then use the BumpHunter algorithm to evaluate the p-value of a potential signal. However, the use of this second algorithm requires the use of a ”reference background” to be expected in the data. Unfortunately, such reference is not always available, as it is the case for the LHC Olympics black box dataset. Thus, in order to use BumpHunter,

min/max window width	2/7 bins
width step	1 bins
scan step	1 bin
number of bins	40
number of pseudo-experiments	10000

Table 2. Hyperparameters used for the BumpHunter algorithm.

ones must first extract a background model for the data.
Another point that has to be taken into acount is the fact that, despite the use of the DisCo term, the dijet mass spectum is not totally independent from the reconstuction error. Thus, simply rescaling the full dataset precut to fit the mass spectrum postcut will not work.

One way to do this is to use a small subset of the data to compute a shaping function. The objective of this function is to capture how the mass spectum behaves when a cut on the reconstruction error is applied. This function is computed bin per bin on the dijet mass histogram by doing the ratio of the bin yeilds postcut and precut.
Of course, the presence of signal in the subset used for this calculation might impact this shaping function. In order to mtigate this efect, the shaping function can be fitted using the tools available in the scikit-learn toolkit. This will minimize the effect of the signal on the shaping function.
Once the shaping function is defined, it can be used to reshape the mass spectum precut in order to reproduce the behaviour of the background postcut.

With this final step, the full analysis workflow is the following :

- Data preprocessing (anti-Kt clusturing, precut on dijet mass)
- Training of GAN-AE on the RnD background
- Application of the trained AE on the black box dataset
- Use 100k events from the black box to compute a shaping function
- Use the shaping function to build a reference to use the BumpHunter algorithm

0.2 Results on LHC Olympics

The results shown were obtained with a AE trained with the GAN-AE algorithm on 100k events from the RnD bakground. Note that before the training and application, cuts were applied on the dijet mass at 2700 GeV and 7000 GeV.

0.2.1 R&D dataset

Here we discuss the result obtained on the R&D dataset. The trained AE have been tested on 100k background events (not used during the training), as well as on the two signals provided.

On Fig. **??** is shown the Euclidean distance distributions (left) and the corresponding ROC curves (right).

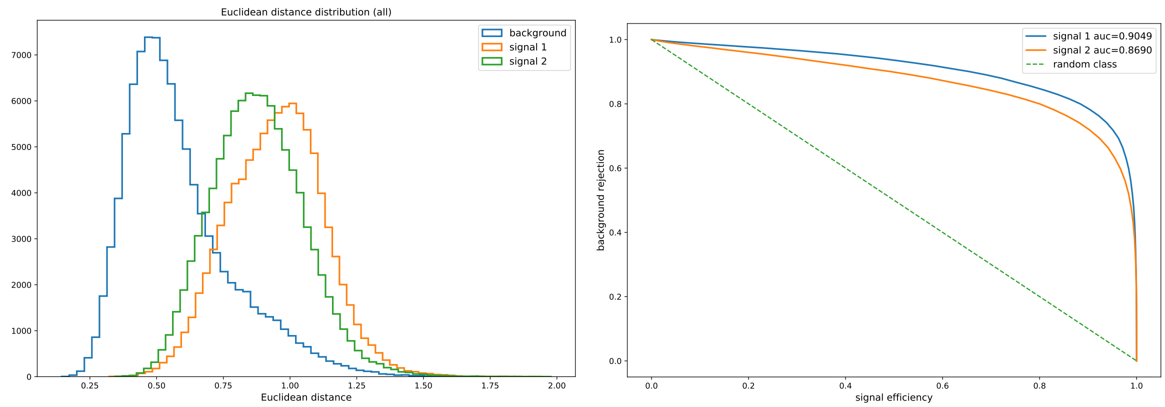

Figure 1. Euclidean distance distributions and ROC curves obtained for the R&D dataset.

This result shows the potential of the GAN-AE algorithm to obtain a good discrimination between the background and signals, event though only the background was used during the training. However, if the obtained AUC is good, it also appears that the Euclidean distance is still very correlated with the dijet mass. This might have a negative impact on the bump hunting algorithm performance.

0.2.2 Black Boxes dataset

Here we discuss the results obatained for the black box dataset provided for the LHC Olympics challenge.

On Fig. **??** is shown the Euclidean distance distribution obained for each black box. Compare to what was obtained with the R&D background, the distributions seem larger and globally shifted to the right. This is most likely due to the diference between the R&D background and the background generated in the black boxes. This fact show that the method used in quite sensible to the small defferencies in the background.

On Fig. **??** is shown the shaping function obtained using 100k events from each black box. The fit was made using scikit-learn. Unfortunately, the fit is very suboptimal. None of the tested functions seems to fit well. This might lead to the aprearance of fake bump or fake deficit during the BumpHunter scan.

On Fig. **??** is shown the results obtained with BumpHunter for all black boxes. As foreseen with the poor fit of the shaping functions, the reference backgrounds built to serve as reference don't fit well the data after cut on the Euclidean distance. In this conditions we can't really evaluate a meanigfull p-value for a potential signal. If the results were good on the RnD

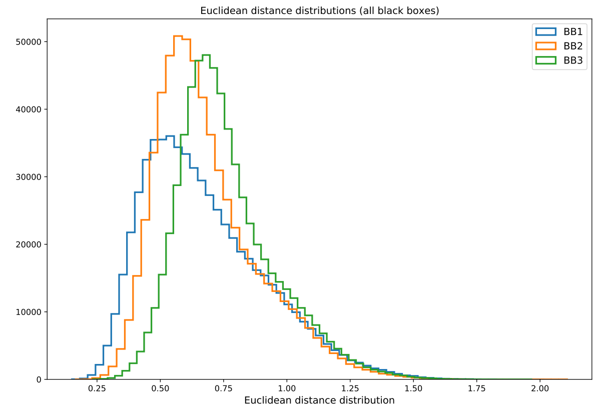

Figure 2. Euclidean distance distributions and ROC curves obtained for the R&D dataset.

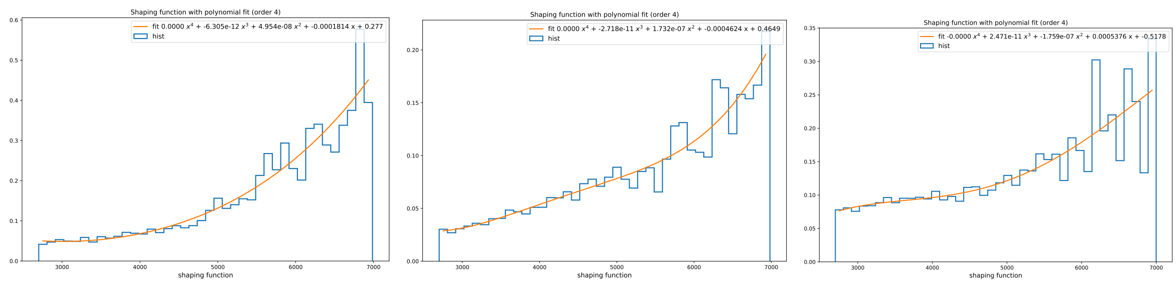

Figure 3. Shaping function obtained for each black box. From left to right, black box 1, 2 and 3.

dataset, it seems that the method is very dificult to apply when there are no good references for the background.

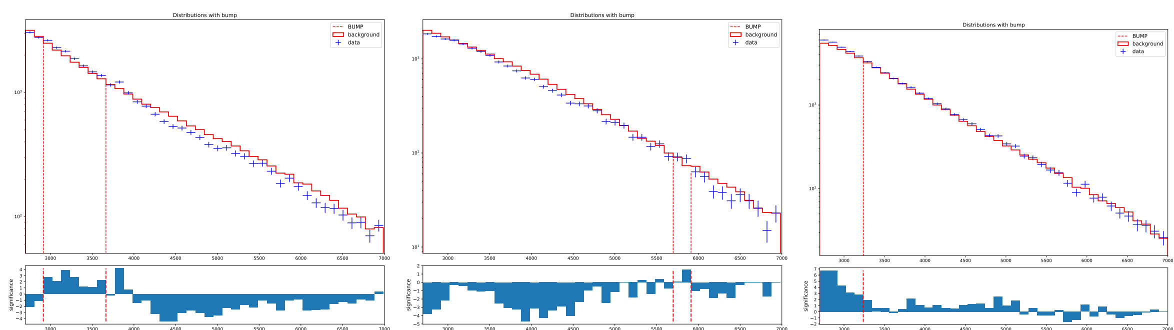

Figure 4. Result of the BumpHunter scan obtained for each black box. From left to right, black box 1, 2 and 3.

0.3 Lessons Learned

The LHC Olympic challenge have been a good opprtunity to test the potential of the GAN-AE algorithm that I have been developping. This shows the potential of this method with

the good results on the RnD dataset, but also its limits.
The results obtained revealed the sensibility of GAN-AE to the small defferencies in the background and also that the distance distribution is stilll very correlated to the dijet mass despite the use of the DisCo term. In addition, the fact that no background simulation that fits the black boxes data were available made the use of the BumpHunter algorithm difficult to apply. Unfortunately, I started working on the LHC Olympics data after the content of black box 1 was revealed.

0.4 Code Availability

All the scripts used to train and apply the GAN-AE algorithm are given at this link :
"https://github.com/lovaslin/GAN-AE$_{L}HCOlympics$"

The implementation of the BumpHunter algorithm used in this work can be found at this link :
https://github.com/lovaslin/pyBumpHunter
In near future, it is planed that this implementation of BumpHunter becames a official package that might be included in the scikit-HEP toolkit.

Acknowledgments

This work was supported by the U.S. Department of Energy, Office of Science under contract DE-AC02-05CH11231.