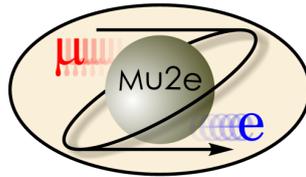


# Antiproton background rejection in the Mu2e experiment

Francesco Lucarelli



## Abstract

The 8 GeV proton beam hitting a fixed target in the Mu2e experiment has an energy slightly above the antiproton production threshold; therefore we cannot exclude that some antiprotons, arising from the interactions between the proton beam and the production target, propagate through the apparatus, eventually reaching the stopping target and annihilating. The annihilation process may turn out to produce a 105 MeV electron, which can be mistaken for a signal electron coming from a muon-to-electron conversion. Although possible, the total energy released in an antiproton annihilation is about 2 GeV, whereas in a conversion event it is only 105 MeV; many differences can be so found in the two processes, and the goal of this work is to take advantage of them by developing a neural network to perform pattern recognition of signal and background events in order to discriminate between them.

In this work, performed during the stage offered by Fermilab to Italian students recruited by the Cultural Association of Italians at Fermilab, I personally analyzed the data from the simulations of signal and background events in order to eliminate all the background ones which cannot be mistaken for signals and to choose the best variables to train the neural network with. The network was then trained through a machine learning algorithm that I coded in the ROOT environment.



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Overview of the experiment . . . . .	5
1.2	Apparatus . . . . .	6
1.3	Antiproton background . . . . .	7
1.4	Neural network and machine learning . . . . .	7
1.4.1	Overtraining . . . . .	8
<b>2</b>	<b>Selection of the events</b>	<b>9</b>
2.1	Number of tracks . . . . .	9
2.2	Track momentum . . . . .	9
2.3	Gate time . . . . .	9
2.4	Calorimeter's first disk . . . . .	10
2.5	Clusters and hits related to the event . . . . .	11
2.6	Particle identification . . . . .	12
<b>3</b>	<b>Training variables</b>	<b>13</b>
3.1	Number of clusters and hits . . . . .	13
3.2	Clusters and hits time . . . . .	14
3.3	Clusters and hits energy . . . . .	14
3.4	Highest cluster energy . . . . .	15
3.5	Clusters center of energy . . . . .	15
<b>4</b>	<b>Neural network training</b>	<b>17</b>
4.1	Output distributions . . . . .	17
4.2	Cut efficiencies . . . . .	17
4.3	Background rejection and signal efficiency . . . . .	20
4.4	Importance of variables . . . . .	20
<b>5</b>	<b>conclusions</b>	<b>22</b>

## Premise

The reported work was performed during the two-month training stage offered in summer 2017 by Fermilab to Italian students recruited by the Cultural Association of Italians at Fermilab.

As will be described in this report, my supervisor simulated events of conversion electrons (signal) and antiproton annihilations (background), which I then analyzed for two reasons:

- First, I want to get rid of the trivial background events which cannot be mistaken for signal events, by applying some cuts to the simulated variables;
- Then, I want to choose the best variables to train the neural network with, in order to discriminate between signal and background events.

These two issues will be discussed respectively in section 2 and section 3, where I will describe the problems I faced and the solutions I found. I will also show the plots that I made starting from the simulated data at disposal.

I then trained a neural network through a machine learning algorithm that I developed in the ROOT environment of multivariate analysis: all the features of this training and the results from it will be widely discussed in section 4 and some plots that I realized will make some assertions clearer.

Finally, in section 5 I will describe my observations about the work performed and give some suggestions to improve it.

# 1 Introduction

## 1.1 Overview of the experiment

After the discovery of the neutrino oscillations it was clear that the lepton flavor is not strictly conserved in the elementary processes: the possibility for a muon neutrino to oscillate into an electron one results in the direct lepton flavor violation (LFV). Although the violation has been observed only in the neutral leptons so far (i.e. neutrinos), according to Standard Model (SM) Charged Lepton Flavor Violation (CLFV) must also occur at some level, as shown for example in the process in figure 1 [5].

CLFV may also arise in other processes, mostly involving the muon decay, such as  $\mu \rightarrow e\gamma$ ,  $\mu \rightarrow eee$  or  $\mu N \rightarrow eN$  (the last one being the conversion of a muon into an electron in the field of a nucleus  $N$ ). In any case, the Standard Model (SM) predicts the charged lepton flavor violating processes to be extremely rare: the branching ratio for the process  $\mu \rightarrow e\gamma$  (figure 1), as an example, is about  $10^{-52}$  [1]. This is why, although several experiments on CLFV, only a limit on the branching ratio of these processes has been set so far.

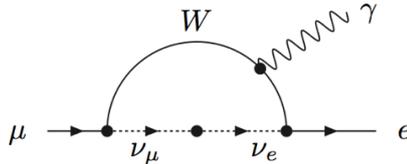


Figure 1: Simple diagram for CLFV in the process  $\mu \rightarrow e\gamma$  according to SM. The neutrino oscillation is necessary for it to occur.

The Mu2e experiment will look for the neutrinoless conversions of muons into electrons in the field of an Aluminium nucleus:  $\mu N \rightarrow eN$ . Since it is a two bodies process, the electrons resulting from the conversions are expected to be monochromatic with an energy [2]

$$E_e = m_\mu c^2 - E_{bind} - E_{recoil} = 104.96 \text{ MeV}$$

being  $m_\mu$  the muon mass,  $E_{bind}$  its binding energy and  $E_{recoil}$  the nucleus recoil energy. The purpose of the experiment is to improve the sensitivity for the CLFV by four orders of magnitude respect to the previous experiments (figure 2), by measuring the ratio between the rate of neutrinoless conversions and the rate of the muonic captures [10]:

$$R_{\mu e} = \frac{R(\mu^- N \rightarrow e^- N)}{R(\mu^- N \rightarrow \text{all muonic nuclear captures})}$$

Mu2e aims to reach a sensitivity of  $R_{\mu e} = 2.5 \times 10^{-17}$ : since many Beyond Standard Model (BSM) theories predict a rate  $R_{\mu e} = 10^{-15}$  [8], this experiment would provide the sufficient sensitivity to test many of them, and if some conversion electrons are observed, we will be provided for sure with proof of New Physics [6].

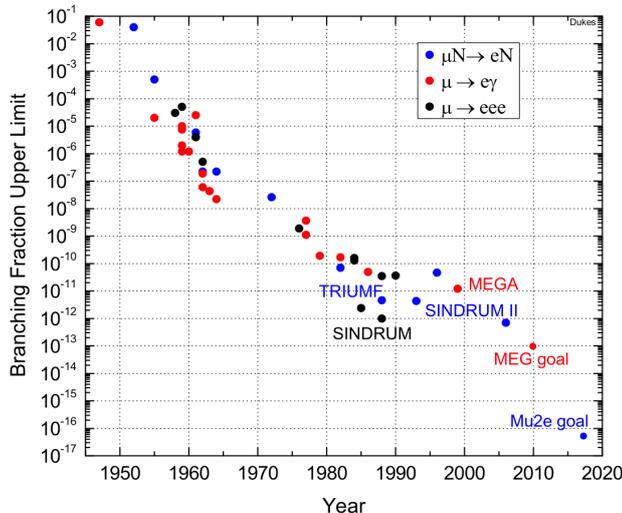


Figure 2: Experiments about CLFV and their sensitivity. Mu2e is expected to improve the sensitivity by four orders of magnitude compared to the previous experiments [5].

## 1.2 Apparatus

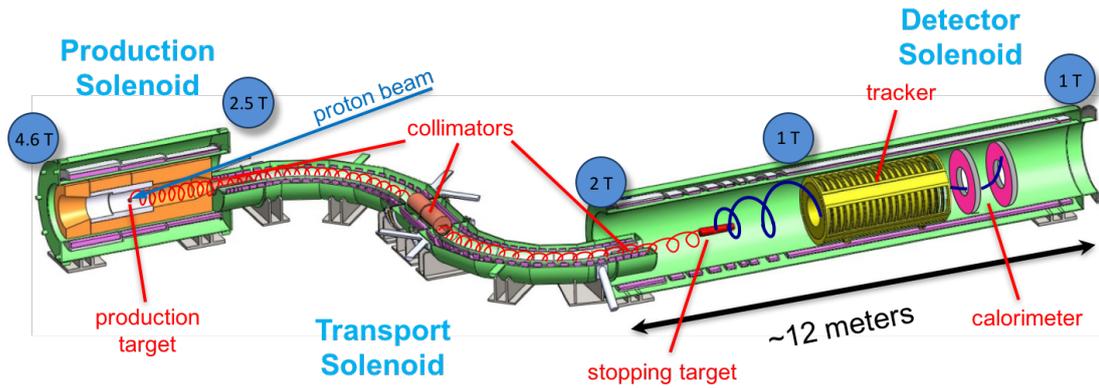


Figure 3: Apparatus of the Mu2e experiment.

Figure 3 shows the apparatus of the experiment. A 8 GeV proton beam impinges on a fixed target (production target) and produces many secondary particles (mostly pions) as a consequence of the interactions arising in the target. Because of the relative orientation between the beam direction and the apparatus, designed to reduce the background in the detectors, the particle momenta are mainly produced in the backward direction: for this reason, a high magnetic field is used in the production solenoid to collect and reflect the pions towards the transport solenoid in helical trajectories.

In the transport solenoid, the gradient of the magnetic field is responsible for the curvature of the particle beam and during the flight almost every pion is expected to decay into a muon. Three collimators permit to select those particles with the desired charge and momentum. A layer of low  $Z$  material is placed in these collimators to stop some of the background source particles (such as antiprotons) [3], thin enough to produce no effects on the muons' motion.

Out of the transport solenoid the muons impinge on an Aluminium target (stopping target) where they slow down and eventually interact with the Aluminium nuclei producing muonic atoms, i.e. atoms where an electron is replaced by a muon. These muons are expected to be in the ground state, or to reach it very quickly from an excited state.

In the nuclei fields, muons can undergo at least three processes : they can decay in orbit ( $\mu^- \rightarrow e^- \nu_\mu \nu_e$ ), which is a background source, they can be captured ( $\mu^- N \rightarrow \nu_\mu N'$ ) or they can be neutrinoless converted into electrons ( $\mu^- N \rightarrow e^- N$ ), which is the signal. Particles emerging from the stopping target are then detected by a tracker and a calorimeter, which provide energy, momentum and time measurements.

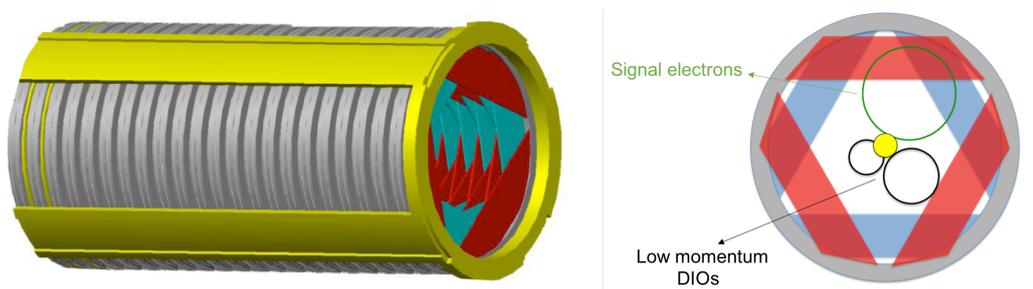


Figure 4: The Mu2e tracker (left) and its frontal section (right). In the right picture are also shown the projections of the helical trajectories for the signal electrons (green circle) and the background ones from the decay-in-orbit (DIO) process (black circles).

The tracker is shown in figure 4 [3] and is designed to get rid of most of the background electrons by detecting only those with energies greater than about 53 MeV [2]: infact a signal

electron is expected to have  $\sim 105$  MeV of energy, whereas most of the electrons, coming from a muon decay-in-orbit (DIO) process, have a spectrum peaked at about 50 MeV (Michel spectrum) with a long and low tail up to 105 MeV caused by the nucleus recoil [7]. Because of the overlap of these energy distributions, the tracker is required to have a high resolution: in the Mu2e experiment, it is about 180 keV [4].

The calorimeter consists of two disks and is designed, like the tracker, to optimize the detection of only signal electrons.

### 1.3 Antiproton background

The threshold of the antiproton production energy  $E_p$  for the process  $pp \rightarrow ppp\bar{p}$  in the laboratory frame of reference, where one of the initial protons is at rest, is ( $c = 1$  for simplicity)

$$s = 2m_p(m_p + E_p) = 16m_p^2 \quad \Rightarrow \quad E_p = 7m_p = 6.56 \text{ GeV}$$

where  $s$  is the square of the center-of-mass energy and  $m_p$  is the proton (and antiproton) rest mass. This value is eventually shifted downward to 5.4 GeV by the Fermi motion, but it is clear that with a 8 GeV proton beam the antiproton production cannot be avoided.

The antiprotons produced in such a way don't decay and propagate, with very low momenta, through the transport solenoid. If they are not stopped in the collimators, the combination of their charge and momentum may allow them to reach the stopping target and annihilate, producing eventually 105 MeV electrons as secondary particles that can fake signal events. Although the electron energy may be the same for the signal and the antiproton annihilation, many differences can be found in these two events: an antiproton annihilation releases about 2 GeV of energy, whereas a muon-to-electron conversion only 105 MeV. We may expect to find differences in the calorimeter clusters and hits in the tracker, in their energy and time, and so on.

To take advantage of these differences we want to develop a machine learning algorithm to train a neural network to recognize, through some selected variables, if a given event is a signal or a background one. In order to do it we have simulated events for the signal ( $\sim 10$  million) and the antiproton background ( $\sim 40$  million), analyzed the information to find the best variables to discriminate between them and provided these variables to a neural network. All the work has been performed with the Multivariate Analysis TMVA package of ROOT.

### 1.4 Neural network and machine learning

Human brains are very efficient at recognizing patterns, but they are restricted to three dimensions and are too slow to analyze the amount of information required in particle physics. This is the reason why the artificial neural networks (ANN) have been created: they are an artificial tool developed in a machine learning framework to perform pattern recognition of a huge amount of data in a short time and in high dimensionality.

The ANNs are inspired by the structure of the human brains, so they can be thought as a group of elementary units, called nodes (analogous to the “axons”), and connections among them (the “synapses”). Each node has a status, typically a real number between 0 and 1, and can send this information to the connected nodes. The receiving nodes elaborate the signals and send their outputs to the following nodes: the whole computing process of an ANN is so a linear combination of non-linear functions of a weighted sum of inputs, whose final outputs will be used, for our purpose, to discriminate between signal and background.

The nodes are generally organized in layers: the first layer provides the network with the input variables, then the nodes in the so

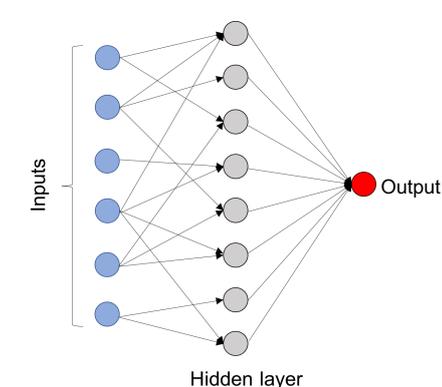


Figure 5: A scheme of an ANN. An ANN generally has more than one hidden layer and may have more than one output.

called hidden layers elaborate the information and finally the last layer produces the outputs. Figure 5 shows a scheme of a simple ANN with one hidden layer and one output response. Each node and connection may also have a weight depending on how important the information it provides to the network is.

Building an ANN involves two steps: first, it has to be trained on a training sample to let it recognize the main features of the input data and their correlations, both for signal and background; then, it can be tested on a test sample to verify how successful the training has been. The TMVA package of ROOT provides different training methods: the ones that we have used are the Boosted Decision Tree (BDT), the Support Vector Machine (SVM) and the MultiLayer Perceptron (MLP). Figure 6 shows the training methods provided by ROOT and their main characteristics [9].

CRITERIA		MVA METHOD									
		Cuts	Likeli- hood	PDE- RS / k-NN	PDE- Foam	H- Matrix	Fisher / LD	MLP	BDT	Rule- Fit	SVM
Perfor- mance	No or linear correlations	★	★★	★	★	★	★★	★★	★	★★	★
	Nonlinear correlations	○	○	★★	★★	○	○	★★	★★	★★	★★
Speed	Training	○	★★	★★	★★	★★	★★	★	○	★	○
	Response	★★	★★	○	★	★★	★★	★★	★	★★	★
Robust- ness	Overtraining	★★	★	★	★	★★	★★	★	○	★	★★
	Weak variables	★★	★	○	○	★★	★★	★	★★	★	★
Curse of dimensionality		○	★★	○	○	★★	★★	★	★	★	
Transparency		★★	★★	★	★	★★	★★	○	○	○	○

Figure 6: The ANN training methods provided by ROOT and their mainly characteristics as described in the TMVA Users Guide. The symbols stand for “good” (★★), “fair” (★) and “bad” (○). We are interested in the BDT, SVM and MLP.

#### 1.4.1 Overtraining

One of the main problems that can arise in training an ANN is overtraining. It may occur for two reasons:

- A machine learning problem has too many parameters compared to the data in the training sample and the resulting degrees of freedom are not enough. In this the case, some statistical fluctuations in the training distributions may be seen by the ANN as features of those distributions, leading to worse performances during the testing phase.
- The training data contain so many information that the network is able to find correlations among the input variables which are not characteristic of the machine learning problem.

Overtraining must always be checked and, if possible, avoided: when it occurs, the ANN’s response is not trustworthy. The sensitivity to overtraining depends on the training methods, as shown in figure 6.

## 2 Selection of the events

To build a neural network we first need to simulate the signal and background events to obtain the necessary information to use in the training and testing phases. In this work  $\sim 10$  million signal events and  $\sim 40$  million background events are simulated, and all the information (calorimeter clusters, hits in the tracker, energy and momentum, position and time of clusters and hits, etc.) is stored in a ROOT Tree.

Since not all the background events contain a fake signal electron, we first need to get rid of the trivial background events by applying specific cuts on the variables from the simulations. The same cuts will be applied to the signal and background samples.

### 2.1 Number of tracks

One of the information we have from the simulations is the number of tracks which hits the tracker and may be related to an electron. Typically this number is one for a signal event and zero for a background event, but it may also happen that more than one track is reconstructed by the detector because of the noise, the coincidence with random processes, etc. In our analysis, we want to be sure that the events we provide the ANN with consist of only one track.

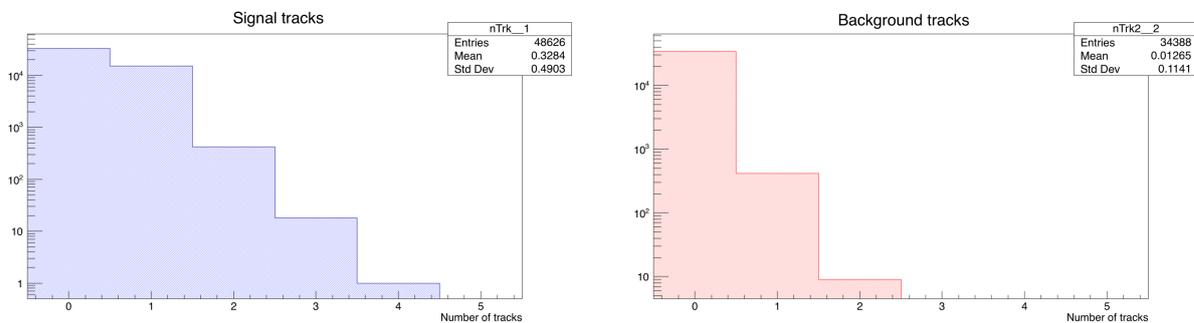


Figure 7: The number of tracks of some simulated events for the signal (left) and background (right). Note the logarithmic scale on the vertical axis.

Figure 7 shows the number of tracks for some of the simulated signal and background events: as expected, in the background case most of the events does not have a track. Our requirement is to exclude all the events which do not have exactly one track.

### 2.2 Track momentum

Another cut we want to apply concerns the track momentum: once we are sure to deal with only one-track events, we also require that its momentum is restricted to a certain range, otherwise it couldn't be a conversion electron.

As we can see in figure 8, the track momentum is peaked at about 104 MeV for the signal, as it should, but this is not the case for the background. Our choice is to reject all the events whose track momentum is less than 90 MeV/c and greater than 115 MeV/c.

### 2.3 Gate time

The muon beam which reaches the stopping target may have a significant contamination of other particles. The main sources for this contamination are [4]:

- muons which decay in flight;
- pions which decay into electrons in flight;
- radiative pion captures (RPC).

In a RPC, i.e. the process  $\pi^- N(A, Z) \rightarrow \gamma N'(A, Z - 1)$ , a pion is captured by a nucleus and produces a high energy photon: if the photon converts, a 105 MeV electron may be detected by the tracker and the calorimeter.

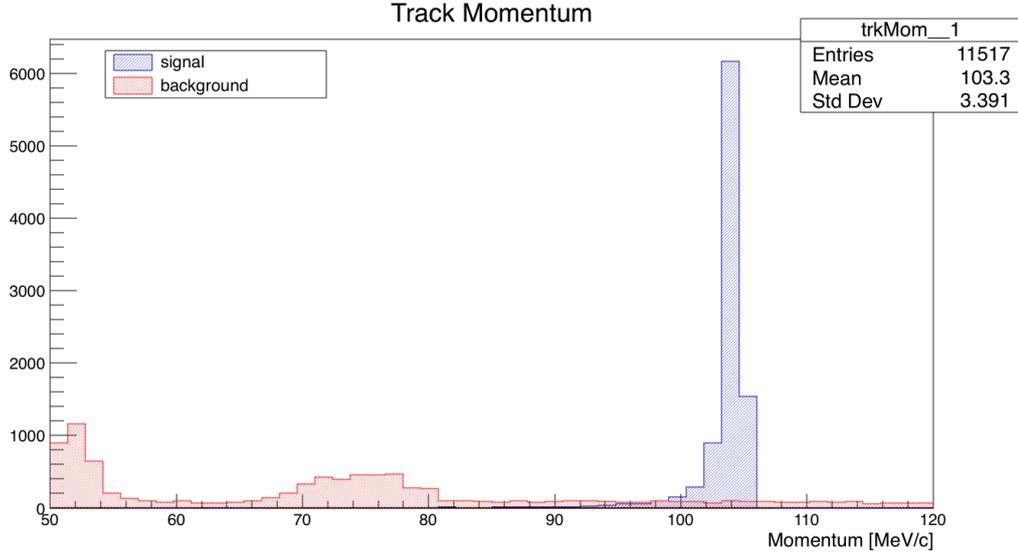


Figure 8: The track momentum for the signal (blue) and background (red) events having only one track. The signal histogram is normalized to the area of the background histogram.

All the events listed above may fake a signal electron, but all of them give rise to a prompt background which can be suppressed by avoiding taking data during the first nanoseconds after the peak of the proton pulse. Since the next proton pulse arrives about 1700 ns after the first one, the Mu2e experiment decided to extend the data acquisition from about 700 ns to 1700 ns after the proton pulse [3]. Figure 9 shows a schematic depiction of the timing of the main events.

In our simulations we want to apply the same cut: we select only those events which occur from 700 ns to 1695 ns after the proton pulse.

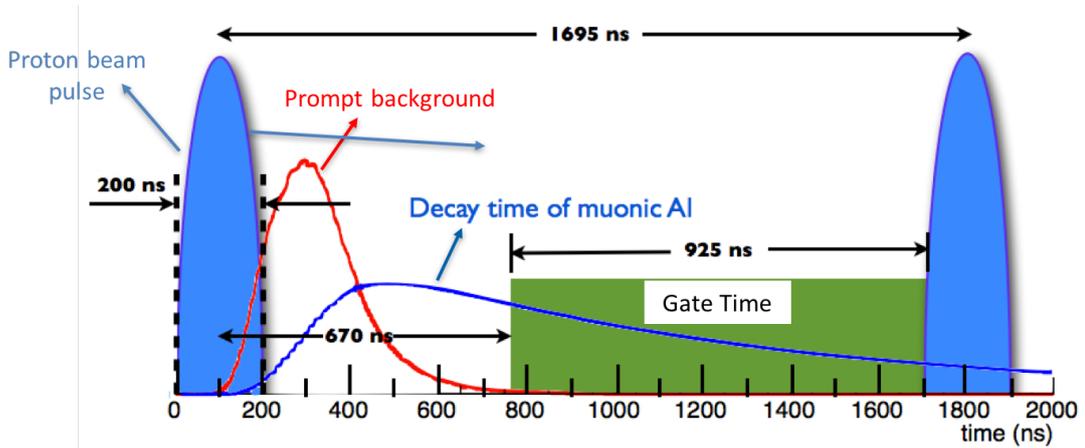


Figure 9: Timing of the main events between two proton pulses. The prompt background (in red) is suppressed by taking data only 700 ns after the proton pulse. The distributions are not in scale.

## 2.4 Calorimeter's first disk

As we have already mentioned, the calorimeter consists of two disks but in our analysis we want to select only those particles impinging on the first one, to be sure they are not produced in other processes after the stopping target. Because of a bug in the software, anyway, the disk

has not been recorded after the simulations and to fix this problem we have to use a different way.

Our idea is to look at the time distribution of the clusters: in our simulations one variable is used to know which cluster is the best cluster for the electron. Since, after being detected by the tracker, it takes more time for a particle to reach the second disk rather than the first one, we may expect that the distribution of the difference of the best cluster times and the track times shows two peaks, each one related to a different disk. The two peaks are well recognizable in the distribution shown in figure 10 and we decide to put a cut at 10 ns in order to exclude most of the clusters arising in the second disk.

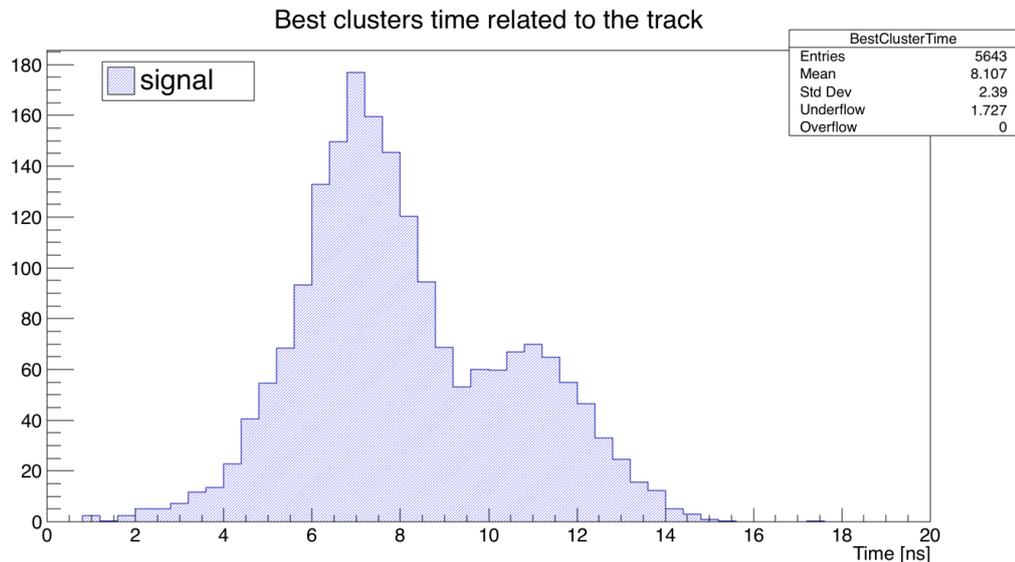


Figure 10: Difference of the best cluster times and the track times. The two peaks in the time distribution are related to the two calorimeter's disks. To select the first disk we will keep only those events whose best cluster time related to the track time is less than 10 ns.

## 2.5 Clusters and hits related to the event

Because of the noise and the random activity of the detectors, not all the clusters in the calorimeter and hits in the tracker are related to the events and we should get rid of those which are not. During a signal or a background event we may expect that detectors' activity is increased: figure 11 shows the distributions of the time difference between the clusters and track and between the hits and track.

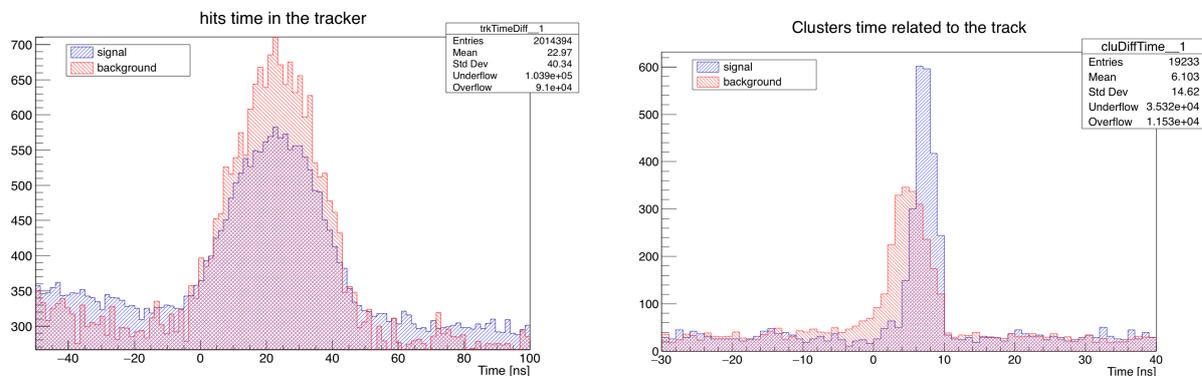


Figure 11: Hits (left) and clusters (right) time related to the track time.

As we can see, both distributions have a spike and we want to exclude those clusters and hits not related to it. For this reason we will keep

- Hits whose time related to the track time is no less than  $-10$  ns and no greater than  $50$  ns;
- Clusters whose time related to the track time is no less than  $-5$  ns and no greater than  $25$  ns.

## 2.6 Particle identification

Once we have selected the events with only one track, within the expected momentum range, in the correct time interval and we have got rid of the prompt background, we may still wonder what particle has produced the track. There is indeed one kind of particle which can be mistaken for a signal electron: a muon. Our simulations don't provide us with direct information on particle identification, but we can take advantage of the following idea: let's suppose we have a muon and an electron which release all their kinetic energy in the calorimeter. If they both have  $105$  MeV/c of momentum, a muon will release:

$$T_\mu = E_\mu - m_\mu c^2 = \sqrt{p^2 c^2 + m_\mu^2 c^4} - m_\mu c^2 \approx 43 \text{ MeV}$$

whereas an electron:

$$T_e = E_e - m_e c^2 \approx 105 \text{ MeV}$$

since its mass is negligible. We can then compute the ratio of the energy released to the momentum for both particles (for simplicity we put  $c = 1$ ):

$$\text{muon} : \frac{T_\mu}{p} \approx 0.4 \qquad \text{electron} : \frac{T_e}{p} \approx 1$$

Figure 12 shows the distributions of this ratio: in the background one we observe two peaks corresponding to the two kinds of particles, in the signal one, as we could expect, we observe only the electron peak.

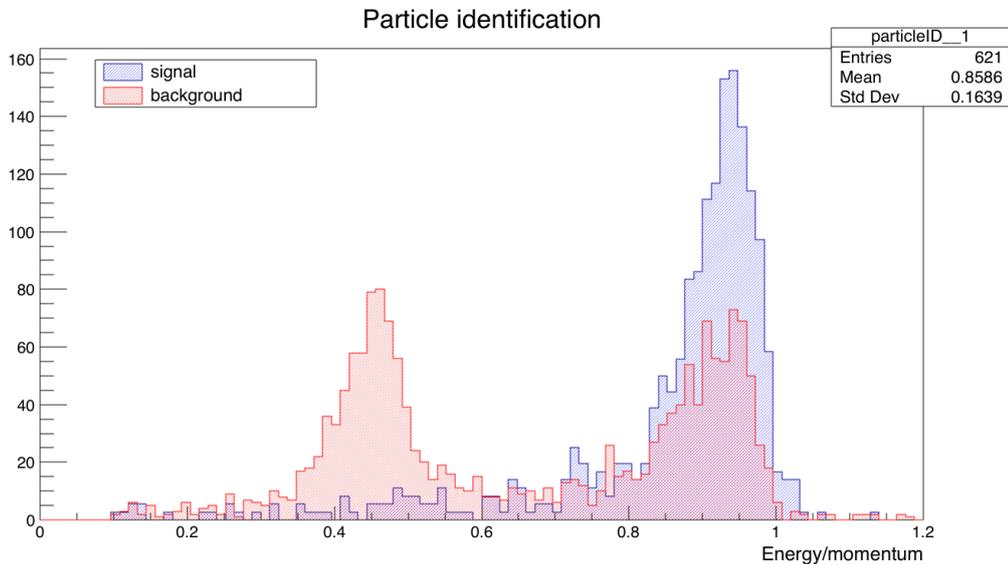


Figure 12: Histograms of the ratio of the energy released to the track momentum for the signal (blue) and background (red).

We can cut off most of the muons from our data by excluding all the events whose energy-momentum ratio is less than  $0.75$ .

### 3 Training variables

After applying all the cuts listed in the previous section to the background and signal samples we have

- about 2,000 left background events (40 million before the cuts)
- about 800,000 left signal events (10 million before the cuts)

We observe that the background data have been significantly reduced after the cuts.

We are now in a position to choose the best variables to train the neural network with. Since an antiproton annihilation releases much more energy (about 2 GeV) and particles in a shorter time than a muon conversion (about 105 MeV), we want to focus on variables related to this physical quantities. If we find significant differences in the distributions of those variables, it will be easier for the neural network to discriminate between signal and background, but even if it is not the case the network may recognize useful correlations among the inputs that are hidden from our eyes.

In the following plots of this section all the signal histograms are normalized to the area of the respective background histogram.

#### 3.1 Number of clusters and hits

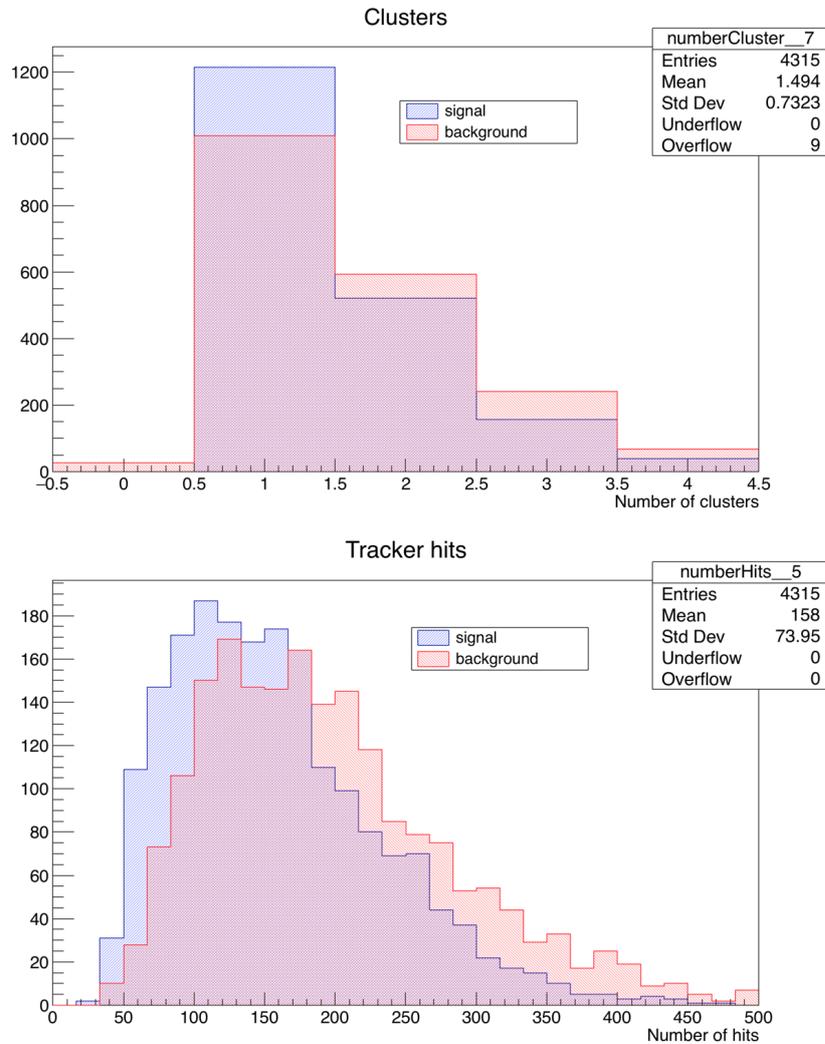


Figure 13: Number of clusters and hits for the background (red) and signal (blue).

Since more particles may be released in an antiproton annihilation, we want to check if more clusters in the calorimeter and hits in the tracker arise compared to the signal. Their number is shown in figure 13: the excess in the background case cannot be only a statistical fluctuation and as we expected an antiproton annihilation has on average more clusters and hits than a signal event.

### 3.2 Clusters and hits time

In an antiproton annihilation many particles are produced in a short time: we may expect that, in the background events, clusters and hits related to the electron track arise earlier than in the signal ones. Therefore we want to have a look at their average time distribution, which is shown in figure 14: what we observe is that the background distribution of the cluster time has a peak just some nanoseconds earlier than the signal distribution, as we could expect, whereas in the hits time distribution we can't see many differences. Despite it, we will keep both variables in our training because there may be some correlations between them that we are not able to recognize.

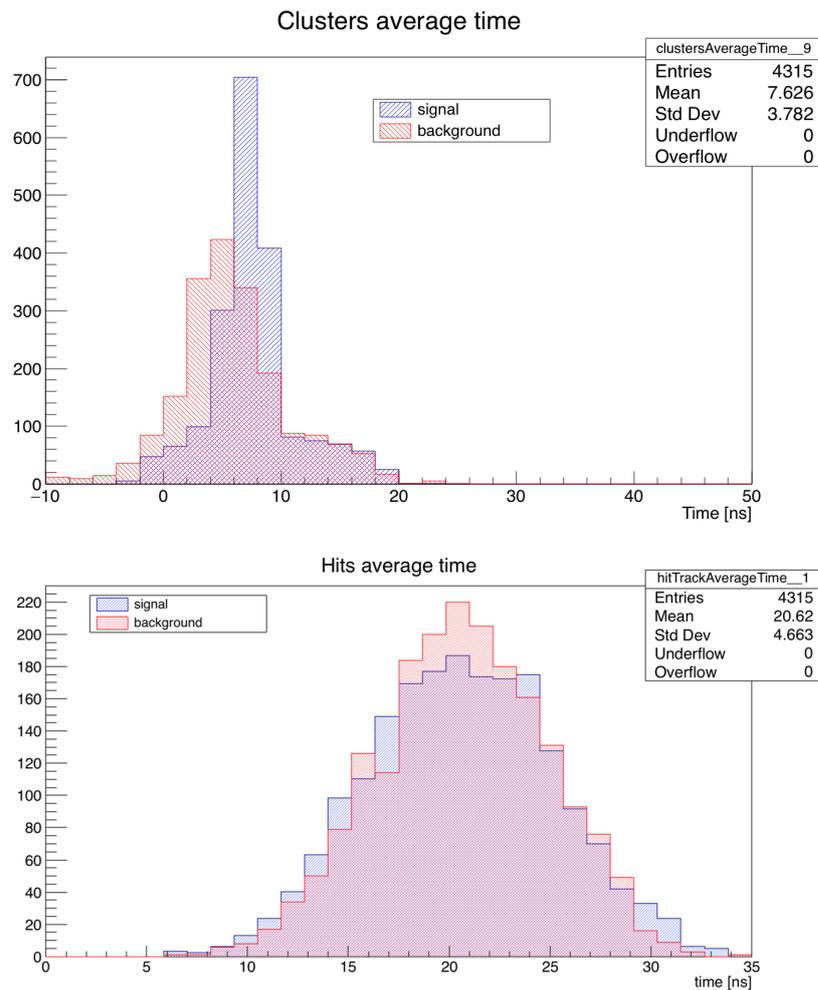


Figure 14: Average time of the clusters and hits related to the track.

### 3.3 Clusters and hits energy

As we have already said, we expect to find more energy in an antiproton annihilation rather than in a signal event. The total energy distributions for the clusters and hits is shown in figure 15 and we observe two interesting features of them:

- The energy is generally higher for the background distributions;

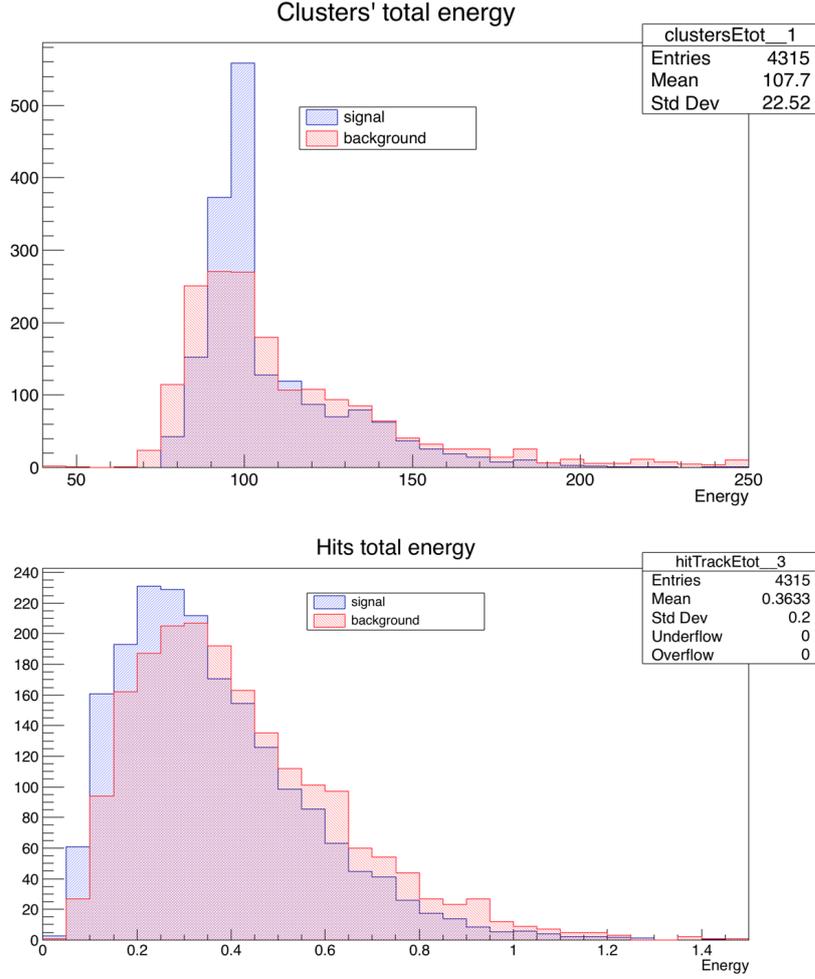


Figure 15: Total energy of the clusters and hits related to the track. Note the long tail at high energies in the background distributions.

- In the background case, the clusters and hits energy distributions show a long tail for high energies which is not shown in the signal case.

Both features are very interesting for our purpose and we can expect the energy variables to have a high discriminating power in the neural network.

### 3.4 Highest cluster energy

Since the background events release more energy, we may expect to find more energetic particles from them. We then want to find the highest cluster energy not related to the electron track for both the signal and background. This distribution is shown in figure 16 and we can recognize a long tail at high energy only in the background case. Since the differences for the signal and background distributions are clear, we may expect this variable to be very important in the neural network's training.

### 3.5 Clusters center of energy

The last variable we want to deal with is the square of the weighted average of the clusters distances to the electron cluster, where the weighting factor is the energy of the clusters itself:

$$r_{CoE}^2 = \frac{\sum_i r_i^2 E_i}{\sum_i E_i}$$

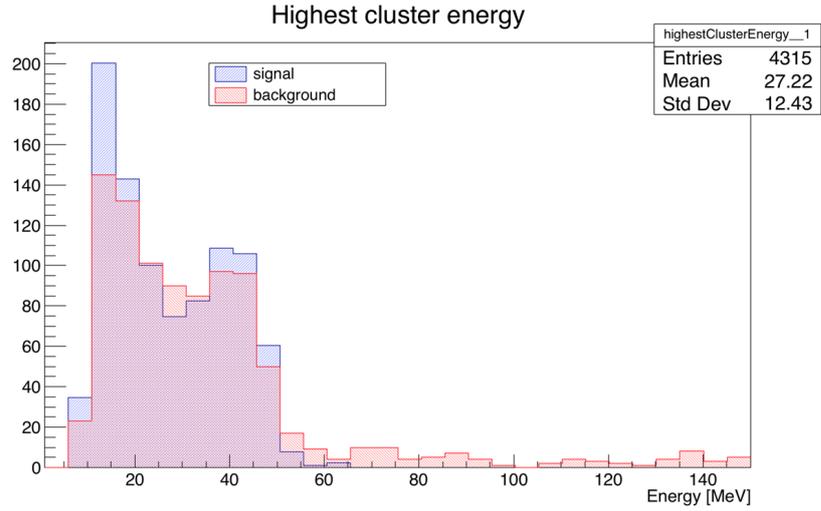


Figure 16: Highest cluster energy not related to the electron track. As we could expect, a long tail is shown by the background distribution.

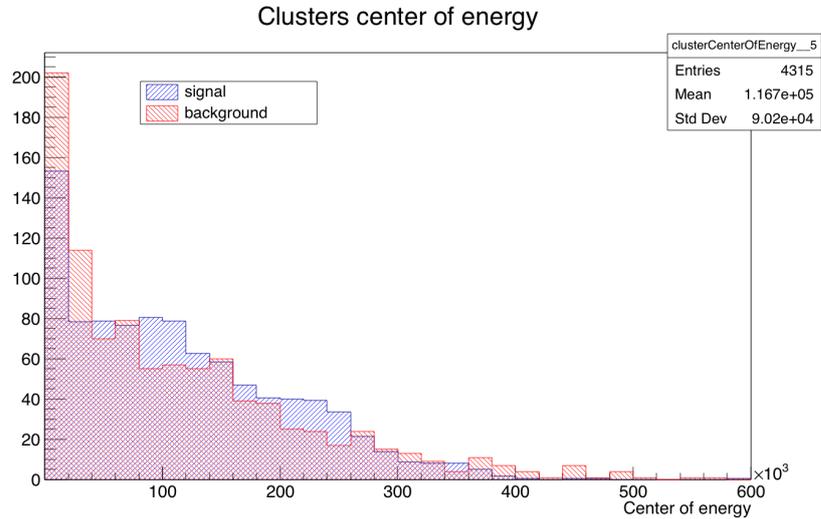


Figure 17: Distribution of the cluster center of energy.

The sum is over the clusters,  $r_i$  is the distance of the  $i$ -th cluster to the electron cluster and  $E_i$  is its energy. For simplicity, we will refer to this variable as the square of the center of energy of the clusters, and its distribution is shown in figure 17. Although some differences in the background and signal distributions, this variable does not seem to have such an important discriminating power.

## 4 Neural network training

After we have decided the variables to deal with, we are able to use them to build the neural network. The neural network training will be developed in the ROOT environment of Multivariate Analysis, which provides different training methods. As we have already mentioned in section 1.4, in our work we will use three of them: the Boosted Decision Tree (BDT), the Support Vector Machine (SVM) and the MultiLayer Perceptron (MLP). Each of these training methods has different features as we can read from figure 6, so we may expect that the neural network shows different behaviors depending on the chosen method.

The results of the neural network training are analyzed in the following subsections.

### 4.1 Output distributions

The aim of a neural network is to produce one or more outputs from the inputs provided, which are then used to perform pattern recognition of the input data. More specifically, like in our case, the outputs can be used to discriminate between two kinds of events: signal and background. These outputs, being the final products of a neural network, are therefore extremely important and can provide us with information about the network, like its behavior, its discriminating power, the overtraining and so on.

The output distributions for our three different training methods are shown in figure 18. They show at least two relevant aspects which are important to point out:

- Regarding the BDT, we can see that the background outputs for the training sample (red points) and the test sample (red histogram) do not overlap very well: it is a clear hint of overtraining (BDT is much sensitive to it, as we can realize from figure 6). The same problem is not shown in the signal outputs, whose training and test samples overlap: the different response of the network to the signal and background events let us suppose that the problem may arise because of too few data in the background case.
- A look at the SVM output distributions shows that overtraining is not present in this case, but we have another problem: the signal and background distributions overlap almost everywhere, meaning that the SVM is unable to discriminate between signal and background. For this reason we cannot expect a good discriminating power of our neural network if SVM is chosen to train it.

MLP instead does not show many problems, even if more events would probably have been useful.

### 4.2 Cut efficiencies

Once we have the output distributions, we can choose an output cut value and reject all those events whose output is less than the chosen value. By doing it we want to get rid of as much background as possible while keeping an acceptable amount of signal events.

For example, having a look at the MLP output distributions in figure 18, if we apply a cut at 0.2 and reject all the lower outputs, we can eliminate a significant part of background and a small part of signal; but we can also decide to apply the cut at a higher value, let's say 0.4, which let us get rid of more background events but also of a larger amount of signal ones. We therefore want to optimize the cut value.

Figure 19 shows the signal and background efficiency as a function of the cut value applied on the output distributions for the three training methods. It also shows, in the green curve, the ratio  $S/\sqrt{S+B}$ , where  $S$  is the number of signal events above the cut value and  $B$  the number of background ones. Every green curve has a maximum for a certain output value, and that value can be chosen as the optimal cut value.

If we neglect BDT, which is overtrained, we can say that:

- About SVM, the background and signal efficiency curve are both above 90% for most of the output values, showing a sharp vertical fall around almost the same value. This is what we could expect after the considerations in section 4.1, and confirms that SVM is not so good at discriminating between signal and background.

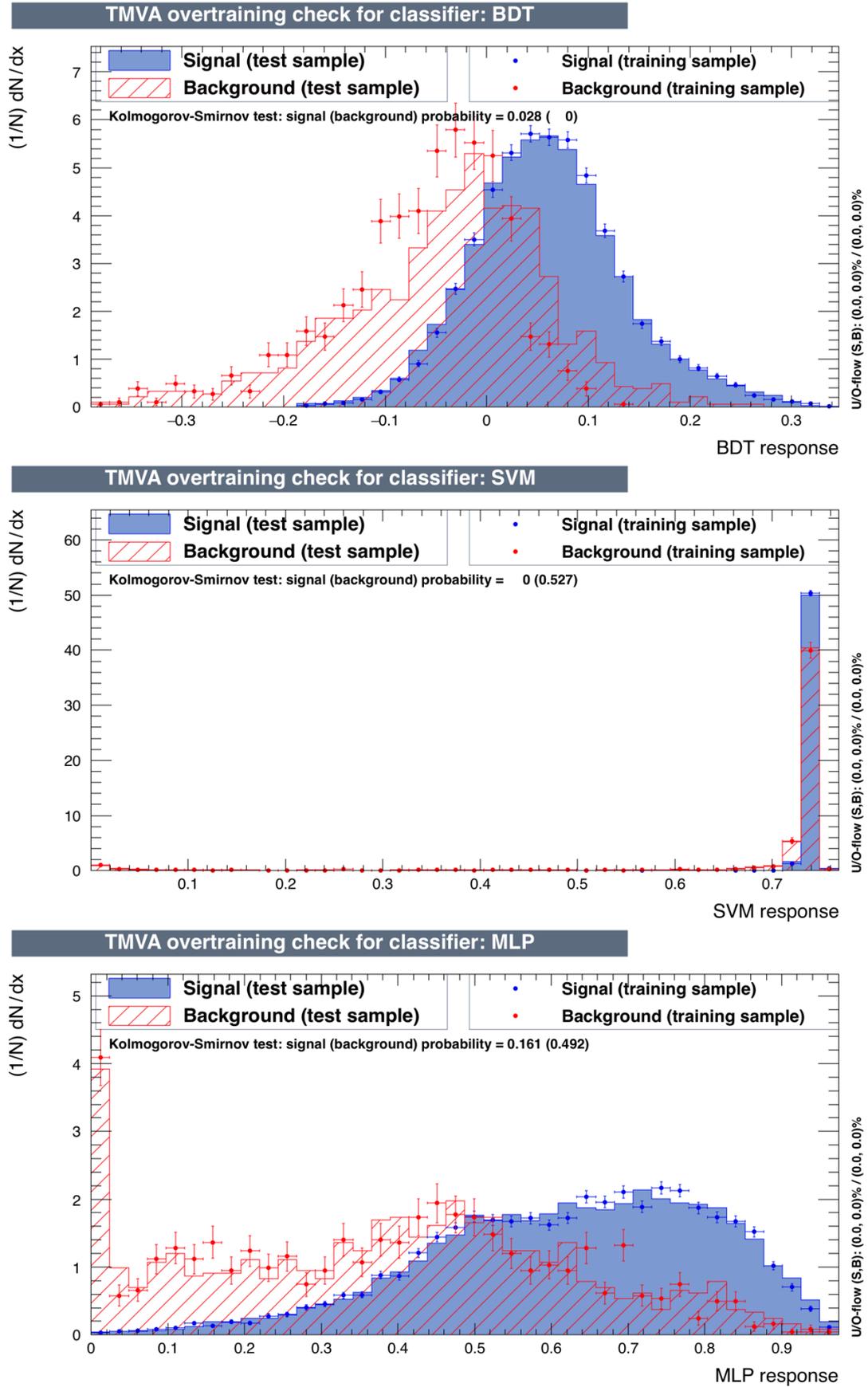


Figure 18: Output distributions for the training methods adopted in this work: BDT (first plot), SVM (second plot) and MLP (third plot). In these plots, the points represent the training sample whereas the histograms refer to the test sample. In both cases, the red color is used for the background samples and the blue for the signal ones.

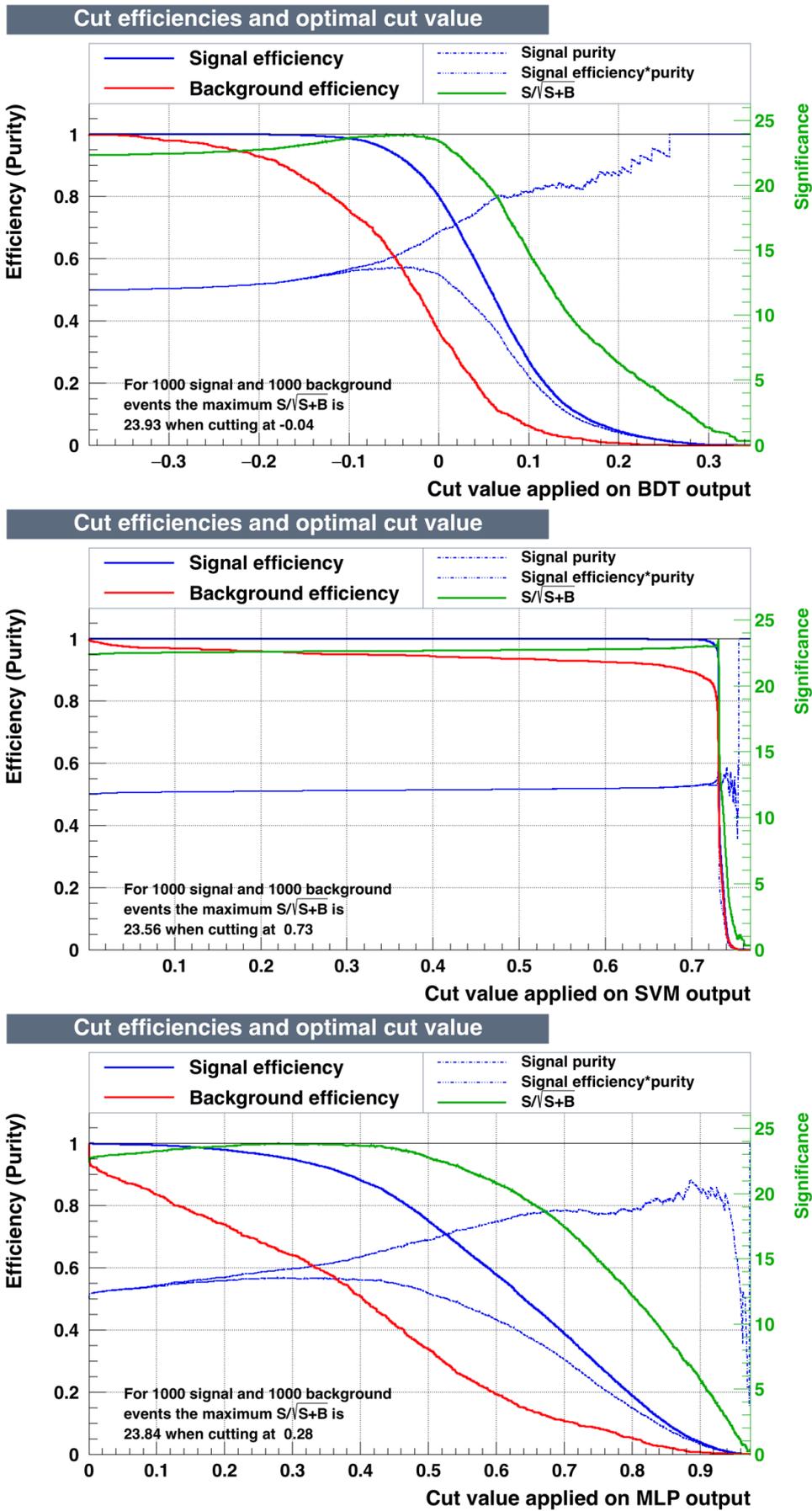


Figure 19: Signal (blue) and background (red) efficiency as a function of the cut value for the three training methods. The maximum of the green curve (the ratio  $S/\sqrt{S+B}$ ) can be used to choose the optimal cut value.

- About MLP, we can see that the background efficiency decreases earlier than the signal efficiency: we get rid of a reasonable part of the background yet at low output values, whereas the signal begins to decrease only after a higher value. This is what we want from a good neural network: a clear separation between signal and background.

### 4.3 Background rejection and signal efficiency

In the plots in figure 19 one value of background and signal efficiency is related to each output value: we can therefore construct the background rejection versus signal efficiency curves, plotted in figure 20, where each color refers to a different training method. The background rejection is defined as  $1 - \text{background efficiency}$ . From a good neural network we expect these curves to approach as much as possible the right upper corner, where we will be ideally able to reject the whole background and keep all the signal.

We can realize from figure 20 that, whereas SVM has not performed a good job as we had already expected, MLP has achieved a satisfying compromise, being able to reject (for example) 80% of background while keeping 60% of signal. The same is true for BDT, but since this method has been overtrained, its results are not trustworthy.

Even if these values are not excellent (neural networks can perform even a better job), we can be satisfied if we consider the low statistics in our data.

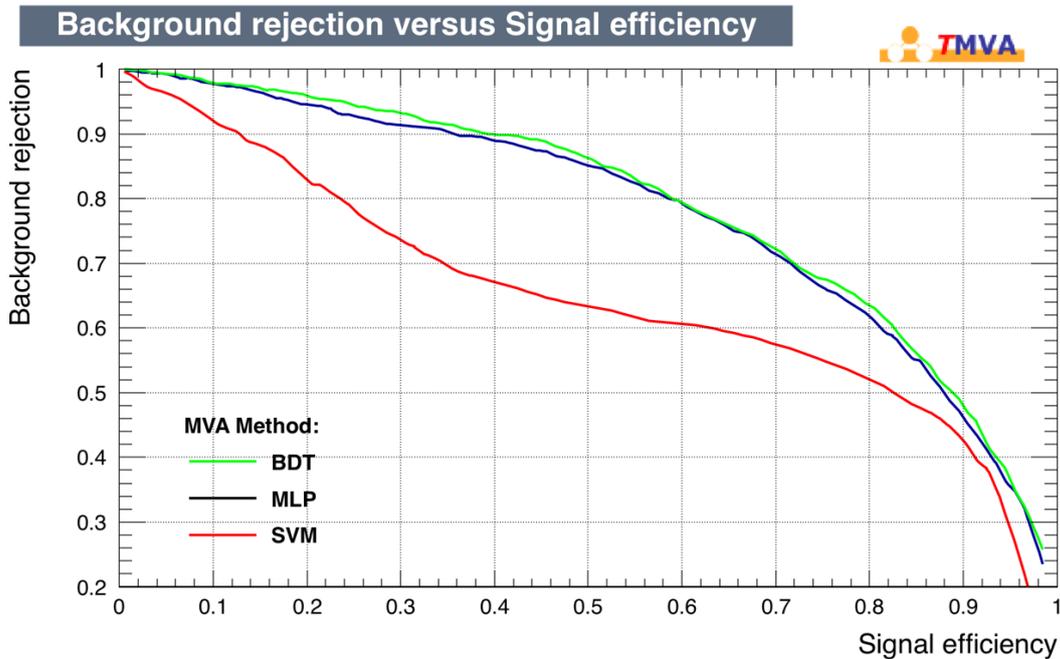


Figure 20: Background rejection versus signal efficiency curves for each training method.

### 4.4 Importance of variables

ROOT also provides us with the importance of each variable in the training phase; in this section we want to focus on MLP, which turned out to be the best training method.

The importance of the variables for MLP is shown in figure 21; the histogram has 8 columns, one for each variable, and the higher the column, the more important the related variable is. The importance  $I_i$  of the  $i$ -th variable is computed as follows [9]:

$$I_i = \bar{x}_i^2 \sum_{j=1}^{n_h} w_{ij}^2 \quad j = 1, \dots, n_{var}$$

where  $\bar{x}_i$  is the sample mean of the  $i$ -th variable,  $w_{ij}$  the weight between the input-layer neuron  $i$  and the hidden-layer neuron  $j$ ,  $n_h$  the number of neurons in the hidden layer and  $n_{var}$  the number of neurons in the input layer (i.e. the number of variables).

In this plot the variables have been ordered according to their importance, and we realize from it that the most discriminating variables are the ones related to the calorimeter. In particular, the number of clusters and the cluster total energy have the best discriminating power, as we could expect according to what we said in section 3.

This is true for MLP, but different training methods could find better discriminating features in different variables; a first analysis suggests that the tracker variables are the most useful in the BDT case, but we would need more statistics to examine this aspect.

Variable importance - MLP

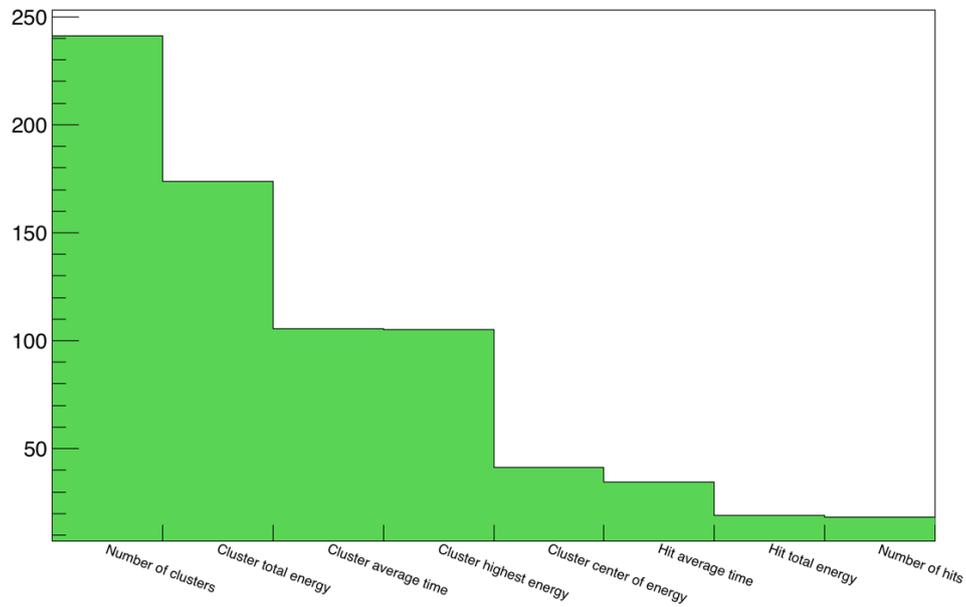


Figure 21: Importance of variables in the MLP training method.

## 5 conclusions

To summarize, in order to build a neural network capable of discriminating between a signal and a background event, we first simulated millions of conversion electrons (signal events) and antiproton annihilations (background events); then we got rid of the trivial background by applying some specific cuts on our data; finally, analyzing the features of both kinds of events we chose some variables to train our neural network with.

The results of the training were provided in section 1.4, and we may conclude by asserting that two of the three training methods adopted in this work did not result to produce a good job: SVM because of its low discriminating power, BDT because overtraining occurred during the building process of the network. Nonetheless, MLP performed a good job, achieving a satisfying background rejection and keeping more than half of the signals.

Although good, the MLP's results cannot be defined excellent, and we would like to improve its performances. One way to do it is to increase the statistics of our samples (especially in the background case), by simulating more and more events: that would probably avoid overtraining, as occurred in the BDT case, and would reduce the statistical fluctuations of the data, letting the training methods (in particular SVM) recognize the features of background and signal distributions more easily.

A second attempt to improve the MLP's performances is to look for more discriminating variables to add to the neural network. Whatever these variables are, we expect them to refer to the energy and time of the detectors' response.

Unfortunately, the time for the analysis was reduced by the large amount of time needed to run the simulations. We spent at least three weeks to obtain the samples used in this work because:

- The number of events simulated were huge (millions);
- Some jobs did not successfully end;
- The codes were affected by some bugs which had to be fixed: it happened that these bugs were found out only after the jobs were run, resulting in a waste of time.

Therefore we had no much time to improve the analysis in the ways we described above, but we believe that larger samples and more training variables will turn out to produce better performances of our neural network.

## References

- [1] *Fundamental Physics at the Intensity Frontier*, 2012.
- [2] R. J. Abrams et al. Mu2e Conceptual Design Report. 2012.
- [3] N. Atanov et al. The calorimeter of the Mu2e experiment at Fermilab. *JINST*, 12(01):C01061, 2017.
- [4] L. Bartoszek et al. Mu2e Technical Design Report. 2014.
- [5] Robert H. Bernstein and Peter S. Cooper. Charged Lepton Flavor Violation: An Experimenter's Guide. *Phys. Rept.*, 532:27–64, 2013.
- [6] R. M. Carey et al. Proposal to search for  $\mu^- N \rightarrow e^- N$  with a single event sensitivity below  $10^{-16}$ . 2008.
- [7] Andrzej Czarnecki, Xavier Garcia i Tormo, and William J. Marciano. Muon decay in orbit: spectrum of high-energy electrons. *Phys. Rev.*, D84:013006, 2011.
- [8] Andrei Gaponenko. The Mu2e Experiment: A New High-Sensitivity Muon to Electron Conversion Search at Fermilab. In *Proceedings, 11th Conference on the Intersections of Particle and Nuclear Physics (CIPANP 2012): St. Petersburg, Florida, USA, May 29-June 2, 2012*, 2012.
- [9] Andreas Hoecker, Peter Speckmayer, Joerg Stelzer, Jan Therhaag, Eckhard von Toerne, and Helge Voss. TMVA: Toolkit for Multivariate Data Analysis. *PoS*, ACAT:040, 2007.
- [10] Luca Morescalchi. The Mu2e Experiment at Fermilab. *PoS*, DIS2016:259, 2016.