Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

Classifying photons with machine learning in ALICE

Bachelor Thesis

Rick Mijsbergh

Supervisors:

Prof. Dr. Thomas Peitzmann Supervisor
Institute for Subatomic Physics

Mike Sas Supervisor
Institute for Subatomic Physics

June 11, 2019

Abstract

The ALICE detector at CERN is used to study collisions between heavy ions, which can create a high-energy quark-gluon plasma as they collide inside the detector. In this research the Boosted Decision Tree algorithm is applied to distinguish electron-positron pairs created by the conversion of photons emitted by this plasma, from background consisting of falsely identified "pairs" of electrons and positrons which do not originate from a photon. The algorithm is trained on over 1.5 million photon candidates generated by a Monte Carlo simulation. Suitable variables for training are determined, data separated into bins to ensure consistency and a K-S test is performed to confirm that the algorithm is not subject to overtraining. Comparison with traditional cuts on the same data show that this BDT method provides a 30% purity increase at maximum significance.

1 Variables and abbreviations

photonPt: p_T. The transverse (in the direction perpendicular to the beam axis) momentum of the photon.
photonR: Distance from the primary vertex at which the photon decays into an electron-positron pair.
photonQt: The relative momentum of the electron/positron pair with respect to the photon.
photonInvMass: The rest mass of the photon as defined in its own frame of reference.
photonEta: The pseudorapidity (parallel-ness to the lead atom beams) of the photon.
photonPsiPair: The angle between the plane spanned by the opening angle of the electron/positron pair, and the plane orthogonal to the magnetic field in the detector.
photonAlpha: The longitudinal momentum asymmetry of the electron/positron pair.
photonChi2: The chi-square test is a statistical test, applied here to determine the probability that the electron/positron pair have a photon as mother particle.
dEdxElectronITS: The energy loss of the electron as measured by the various sensors of the ITS detector. If there is no data for that electron (high-p_T photons won't decay inside ITS), it is assigned a value of 1000.
photonCosPoint: The pointing angle, which is the angle between the vector that points from the primary vertex to the location where the photon decays, and the momentum vector of the photon at the time of decay.
nSigmaTPCElectron: The number of standard deviations in which the energy loss of the electron as it travels through the TPC detector, differs from the mean.
clsITSElectron: The number of sensors in the ITS detector that are triggered by the electron.
fracClsTPCElectron: A measure of the fraction of sensors in the TPC detector that are triggered by the electron.
CERN: Conseil Europen pour la Recherche Nuclaire, or the European organisation for nuclear research.
LHC: Large Hadron Collider, CERNs particle accelerator near Geneva.
ALICE: A Large Ion Collider Experiment, one of the detector locations at the LHC.
BDT: Boosted Decision Tree, a type of machine learning algorithm.
ROOT: A C++ framework built to analyse large amounts of data easily.
TMVA: A package within ROOT which includes a generic BDT script.
AliROOT: The implementation of ROOT within the ALICE experiment.
QGP: Quark-gluon plasma, a high-energy state of matter in which protons and neutrons melt into their constituent parts.

Contents

2 Introduction

The Department for Subatomic Physics at Utrecht researches high-energy collisions of lead atoms in the ALICE detector at the LHC. One of the specific fields of interest in this research is to analyse electron-positron pairs in order to determine whether they were created by the conversion of a photon. At the moment, this process is done by removing those pairs which are unlikely to come from a photon because their properties exceed certain thresholds. All other pairs are accepted as coming from a photon, but in this selection of data there is still a large fraction of unwanted background.

One way to improve on this method, is to use a machine learning algorithm, called a Boosted Decision Tree (abbreviated as BDT), and teach it to recognise the distinctive properties of both signal and background. This teaching process can be carried out by feeding the BDT a set of data from pre-existing Monte Carlo simulations, which model the results of an actual experiment. Using these simulations as training data allows for a test of the algorithms output, since the desired output (signal or background) is known.

The goal now is to configure a Boosted Decision Tree to analyse data from ALICE experiments in such a way that it can assign a BDT output value to the detected photon candidates, corresponding to the probability that it is a signal photon, so that a single cut can be made by the researcher to separate background from signal. This method should produce reliable results for varying ranges of photon p_T and centrality of the primary collision event, and it should be demonstrably more efficient than the by hand cutting method currently used to separate background photons from the signal.

This text will now explain briefly the theoretical background of the ALICE experiment, before expanding on the methodology of the BDT algorithm in the third chapter. The fourth chapter guides the reader through the various building blocks necessary to successfully apply the algorithm to the context of this experiment, and provides analysis to support the choices made. The results of testing these developments are presented in the fifth chapter and its potential shortcomings discussed in the sixth. The final chapter provides a concluding summary.

3 Theoretical background

A good understanding of the context of the experiment is crucial to the understanding of the experiment itself, and therefore this chapter aims to explain the fundamentals of the photon experiments at ALICE which lie at the core of this thesis. The first section treats the basics of the Standard Model, the theoretical model behind photons and other subatomic particles. Section 3.2 elaborates on the technical aspect and the detectors involved. Finally, section 3.3 explains what happens during a collision event, and explains the creation of the particles that we detect.

3.1 The Standard Model

Most of our current understanding of the realm of subatomic physics is contained in what is called the standard model[2]. The elements of this model are shown in Figure 1. The standard model describes the elementary particles, which can be subdivided into two groups: fermions and bosons. The difference between these groups of particles is a property called spin, which is a quantum-mechanical number that has no analogue in classical physics. Fermions have half-integer spin, and bosons have integer spin.

In fact, gauge bosons, like the photon and the gluon, have a spin of exactly 1. These bosons are called gauge bosons, or force carriers, since they govern the interactions between fermions. Some of them have mass or charge, but the photon and the gluon are both massless and electrically neutral. The former mediates the electromagnetic force; the latter carries a force known as the strong interaction.

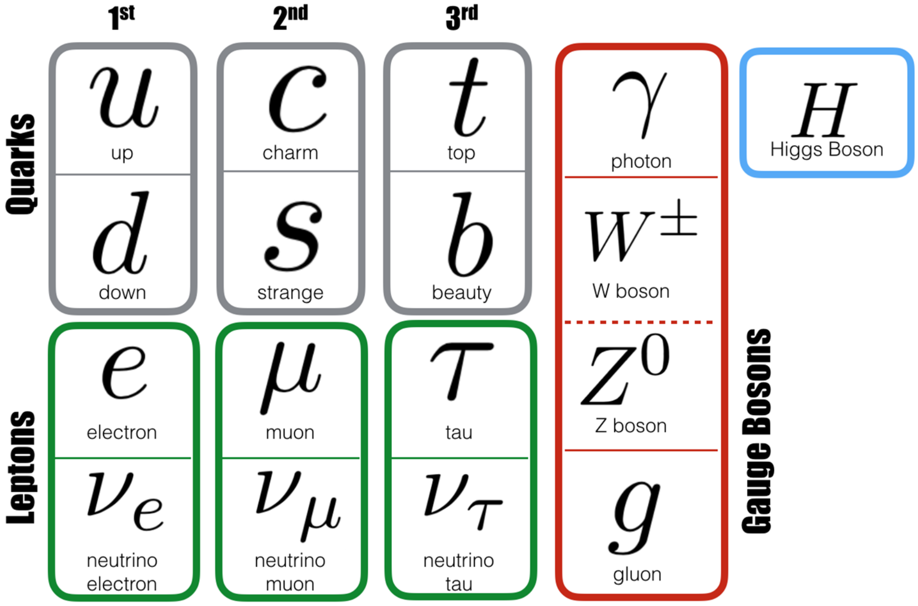


Figure 1: Schematic representation of the particles in the standard model.[1]

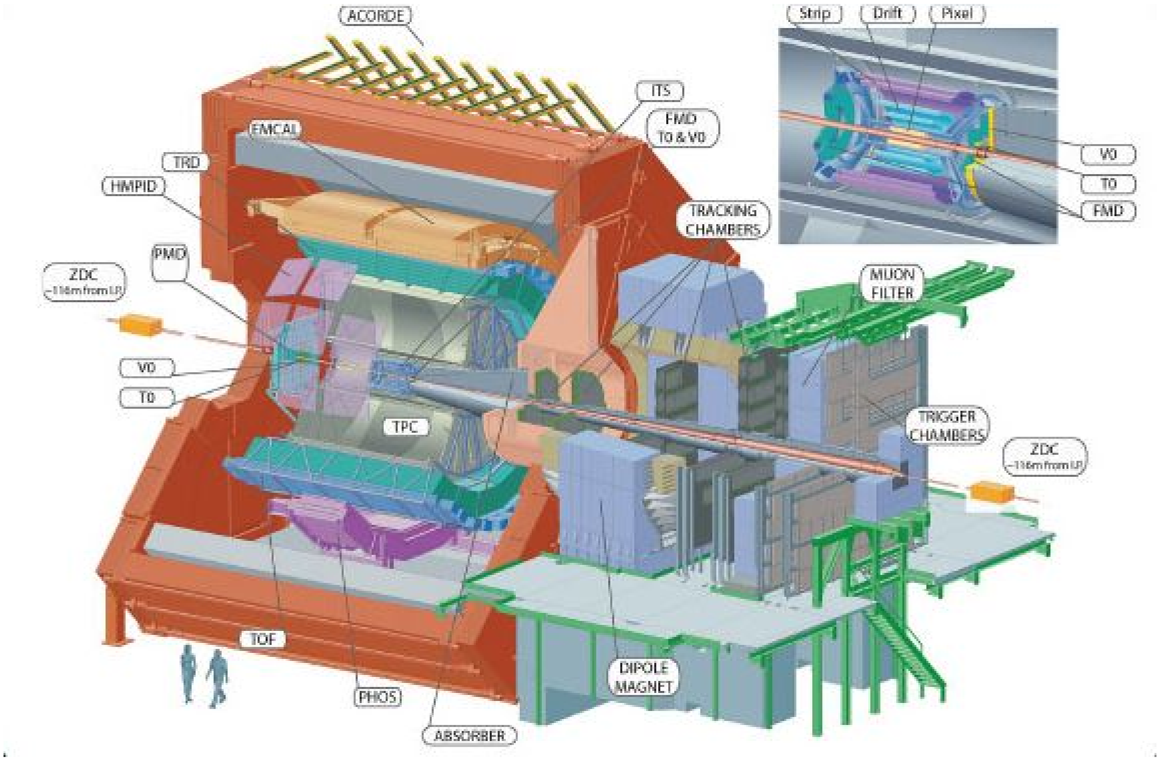


Figure 2: Overview of ALICE and its subdetectors.

Fermions have spin-1/2 and are, themselves, subdivided into two groups: leptons and quarks. There are six different flavours of quarks; usually bound together to form hadrons, which is the name for well-known larger particles such as the proton. They are not usually found as free particles, and their lifetime as such is extremely short. Leptons, on the other hand, can be found as free particles. Examples include the electron, the positron and the neutrino. These particles are part of the lightest group of leptons, with two other groups containing similar but heavier particles.

3.2 The ALICE Detector

A Large Ion Collider Experiment (ALICE) is one of the experiments at the LHC site.

The ALICE array consists of a number of detectors with varying specifications, each of which measures a different aspect of the particle shower created after the collision. The most important ones of these detectors for the research at hand will be described briefly in the next section, using information taken from the documentation on ALICEs website.[3]

The **Inner Tracking System** (ITS) and the **Time Projection Chamber** (TPC) are two of the main detectors in the array, and their data is of key importance in this research. Their function is to detect, track and identify charged particles as they fly through the detector. Since photons can produce an electron-positron pair when they interact with matter, detecting these particles can give valuable information about the nature of the photon that produced them. The ITS is the smaller detector of the two, and detects photons only when they decay close to the primary vertex. It consists of six layers of silicon detectors, surrounding this vertex, and made to be as lightweight as possible. The TPC is a larger detector behind the ITS which contains a volume of gas, which shows charged particles as they ionise the gas in their path. Due to its size, it can detect more photons as they have the time to decay within the limits of the detector.

The **Time of Flight Detector** (TOF) consists of 1593 detector strips which measure the time it takes for a particle to reach it. This time can be used to calculate the velocity of the particle, which also helps calculate the particle mass.

Whereas the first three detectors can only measure photons by their decay products, the **Photon Spectrometer** (PHOS) detects the photons themselves when they strike the detectors lead tungstate crystals and produce scintillation light. While the PHOS measures the photons very accurately on a limited domain, the **Electromagnetic Calorimeter** (EMCal) and **Photon Multiplicity Detector** (PMD) extend that reach over a wide area, albeit at a lower precision.

The **T0 detector** and **V0 detector** are used to determine whether, and when, a collision has taken place. The latter also provides one of the crucial data points for this research, namely the centrality of the collision event. It determines this by measuring the total energy deposited into the two parts of its detector (one on each side parallel to the beam), which scales with the number of particles generated in the collision, in turn a measure for the centrality. An estimate for the same is also produced by the **Zero-Degree Calorimeters**.

3.3 Collisions

When particles collide with each other in the centre of the detector array, there usually isnt a perfect head-on collision there is a certain centrality to the event, which describes how much overlap there is between the particles when they collide, or in other words, how well they are aligned. To illustrate this, Figure 3 shows examples of varying centrality, which is expressed in a percentage value. This percentage is the amount of collisions in the data that are more central; so, a centrality of 10% means that out of all events in the data, 10% are more central, and 90% are less central.

Even for a higher centrality percentage, the collisions in the LHC release an incredibly large amount of energy, so large in fact that it can temporarily create a special state of matter; a quark-gluon plasma (QGP)[4][5]. This soup of fundamental particles is in the same sort of condition as one could find in the beginning of the universe, just after the Big Bang, and the temperatures involved are 100,000 times hotter than the centre of the sun. Although the quarks are usually bound together by the strong force provided by gluons, in this extreme state the energies involved are so large that they overcome the forces between the elementary particles, the bonds are broken and both quarks and gluons float freely through the plasma.

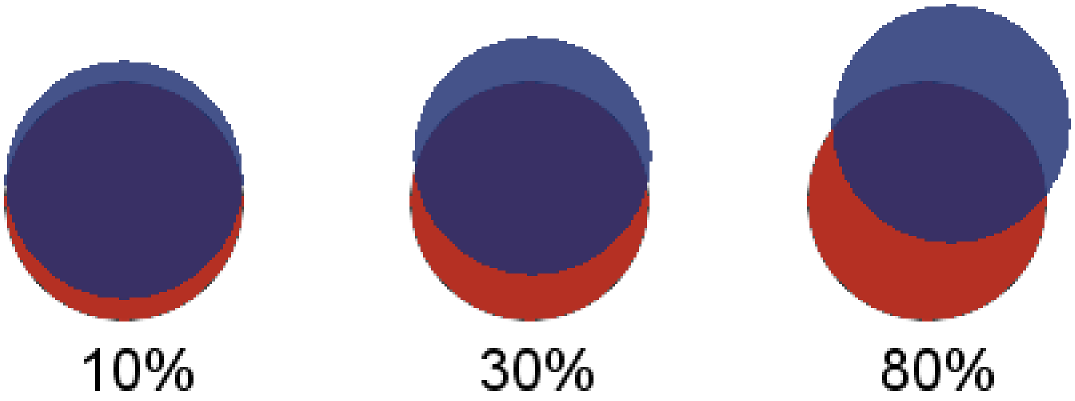


Figure 3: Rough illustration of a collision of a red particle (travelling out of the paper) with a blue particle (travelling into the paper) at varying centrality.

For a moment, the quark-gluon plasma takes the properties of an almost perfect fluid, with small viscosity. Then, inevitably and very quickly after it was formed the plasma cools and expands, and the quarks and gluons combine again. Once again they form ordinary matter, which shoots away from the primary vertex, along with many direct photons formed in the QGP fireball.

These and other photons can undergo a particularly interesting process called ”photon conversion”[6]. It is this transformation of a photon into an electron-positron pair that our research relies on. As a photon passes through matter (such as the detectors in ALICE), it has a chance to interact with the nuclei of the matter and decay into an electron-positron pair. These electrons and positrons are far easier to detect than the elusive photons, and from this conversion extra information about the photons can be gathered, through their daughter particles. The challenge is to identify which electrons and positrons are created through photon conversion, and which are not. To this end, this research will employ machine learning on the various properties of these leptons.

4 The Boosted Decision Tree

The machine learning method chosen for this research is the Boosted Decision Tree algorithm. It is a relatively simple and well-integrated method, with proven potential[7].
Section 4.1 of this chapter explains the basic concept of decision trees, and section 4.2 the improvement on this concept called *boosting*. The third section details the potential pitfall of overtraining, and some usual measurements that are used to determine the performance of the BDT. Finally, section 4.4 details how the BDT is implemented within the framework of the ALICE experiment.
The inner workings of a boosted decision tree algorithm have been described clearly by B.P. Roe et al (2005)[8] and M. Sas (2014)[9] , so the following chapter will liberally use information from their texts.

4.1 Decision Trees

The basic concept of a BDT revolves around a decision tree; a form of instruction for the algorithm to determine which cuts to make in which variables. When using a decision tree, it checks all data for the first condition. Data that matches the condition goes down one branch of the tree, data that does not travels through the other. The combination of a condition and the data it checks is called a node. When data is checked in a node, it travels onto another node where it is checked again, until it reaches the final node in its branch. This creates a chain of conditions that the data is checked with, depending on which checks it passed or failed previously.

Figure 4 illustrates how background and signal photons are separated in a typical decision tree. A cut in the photonR variable separates a node with relatively pure data (90% purity) from a node with relatively impure data (32% purity). Each node is then split three more times, as long as the total amount of data in a node exceeds the minimum size of a node for splitting. Each time, the node is split according to the variable which would create the best separation between signal and background.

The key to building a decision tree, is finding out which cuts in which variables create the best separation. To determine this ideal split, we define the purity of a node;

$$P = \frac{\sum_s W_s}{\sum_s W_s + \sum_b W_b},$$

where W is the weight of a photon in the calculation, $\sum_s$ the sum over signal photons and $\sum_b$ the sum over background photons. We also introduce the Gini index;

$$Gini = \sum_{i=1}^{n} (W_i)P(1-P),$$

where N is the number of events in that node. Now, the ideal split is that which decreases the Gini index of the daughter nodes as much as possible. A big difference between the parents and its daughters Gini index means that there is a high separation, and therefore that the cut has been very effective. The algorithm searches through all variables for the cut that maximises the separation gain, for each node until all nodes in the tree are filled.

The result of a single tree is a simple binary result; if a photon ends up in a node with a majority of signal photons, it is classified as signal (1). Similarly, all photons in a majority background node are classified as such (-1). Inevitably, since the end nodes are rarely 100% pure, this will result in photons receiving the wrong classification.

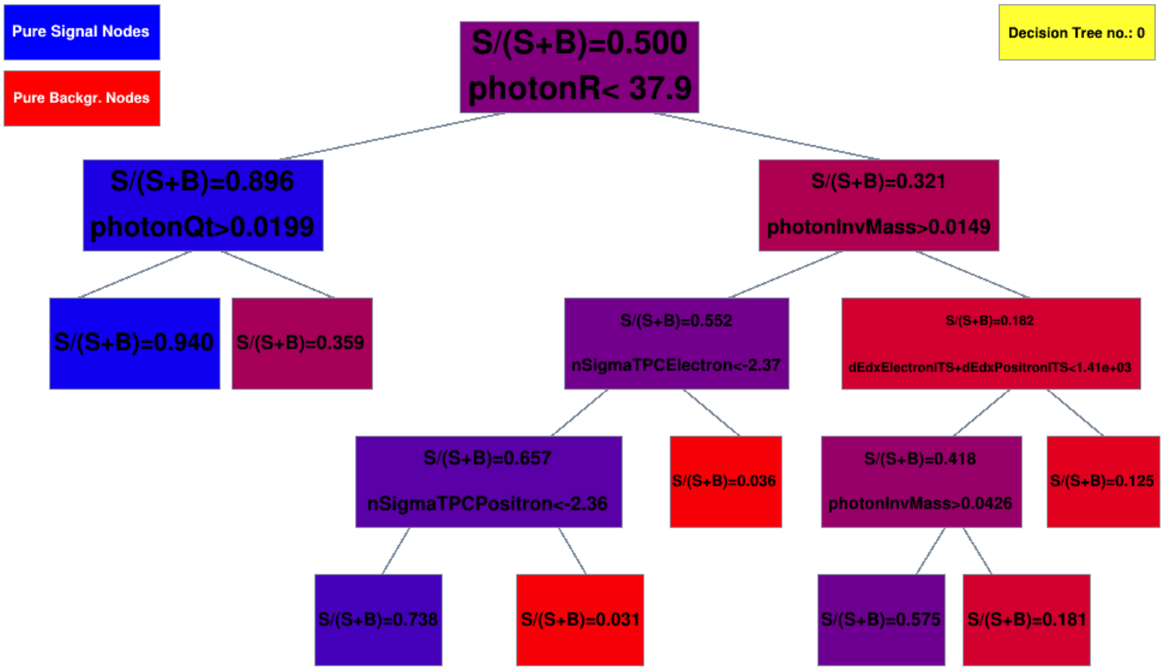


Figure 4: The first decision tree in a BDT forest (trained on data with a centrality of 20-40% and a photon p_T between 0.5 and 1.0). A more blue colour indicates that more signal data takes that path, a red colour indicates more background data.

4.2 Boosting

For the first tree generated by a boosted decision tree, the weights W of all photons are the same. However, the strength of this method lies in its boosting that is to say, the algorithm increases the weight of all photons that received the wrong classification and then runs a new training round with the new weights. In this way, many trees and outputs are created, forming a so-called forest.

The final score the BDT output of the photon is determined as an average of the result of all the trees used in the training. A photon which lands into a final signal node in most of the trees in the forest, will end up with an output value close to 1. Conversely, a photon which often lands in background nodes, will have a value closer to -1.

4.3 Testing

A trained BDT can be tested by running the finished algorithm on a test sample of similar data. From the resulting output graph, three important measures of the quality of the boosted decision tree can be determined, for a certain cut in the output. The purity of the accepted data;

$$P = S/(S + B),$$

where S is the amount of signal photons accepted, and B the amount of background accepted. The efficiency of the cut;

$$E = S/S_{tot},$$

where S_tot is the total amount of signal photons in the data. If we accept more data, the efficiency will rise and purity will fall. If we accept less, vice versa. In order to determine an optimal balance between the two, we measure the significance of the accepted data;

$$Sig = S/\sqrt{(S + B)}.$$

One of the dangers that can occur is that a BDT is over-trained. This means that the algorithm judges photons by very specific properties which are only a good measure in the training data that it was given, and not in all data that it might be applied to. The training data should be a good representation of a typical data set, but even then overtraining can occur. For one, the data set must be large enough that statistical fluctuations are not relevant to the training. Furthermore, if the settings of the BDT allow it to pick out very specific properties, the chance that these properties cannot be generalised increases and so the result has an increased risk of overtraining.

Applying a Kolmogorov-Smirnov test[10] is a good way to ensure that the training of the boosted decision tree has not actually caused overtraining of the algorithm. The test is used on BDT output from a training sample and a test sample, with the assumption that they should produce the same result. If the chance that their difference is due to statistical error is small (usually: more than two standard deviations away from the mean, or <5% for a normal distribution), then there is an indication that the BDT is over-trained to some degree.

4.4 Framework

The BDT used in this research builds on elements present within the Toolkit for Multivariate Analysis (TMVA). TMVA is a package within ROOT[11], a C++ framework built to analyse large amounts of data easily. Critically, it provides an easy and quick way to access the millions of variable data points produced by the Monte Carlo simulation, and convenient plotting features for histograms. The TMVA package comes with several generic machine learning scripts, including a BDT option which is used as the base for this research.

In ROOT, the forest of decision trees can be condensed into a weight file which produces the same output. These weight files can then be applied to new data using an application macro which creates an output histogram, showing the BDT output for all included photons.

The version of ROOT that is used within the ALICE experiment is called AliROOT; this research has aimed to integrate its results into the AliROOT framework in order to make it available for all future research there into photon physics.

For this experiment, the input data will consist of a Monte Carlo simulation of a lead-lead collision event. These simulations are carried out within AliROOT, with the help of software specially tailored for ALICE[12]. The software simulates a collision, the particles that it produces, and then separately simulates how the detectors respond to these particles. This method is widely used as the standard for simulating particles in the ALICE experiment.

5 Implementation

The following chapter details the steps taken to implement the BDT algorithm to separate signal photons from background in simulated lead-lead collisions. Section 5.1 details the input data generated in these simulations. Section 5.2 and 5.3 build on previous research in this field to determine the best settings and variables for the training process, improving them as needed. The final two sections detail a crucial concept for the reliability of the algorithm, which is data binning. These sections contain an in-depth analysis of the input variables to support the specific bin choices and their efficiency.

5.1 Input data

The input data consists of just over 1.5 million events, taken from 7 separate runs of the Monte Carlo simulation. Table 1 shows how these data points are distributed across varying centrality and photon p_T intervals. In general, the higher the event centrality percentage or photon transverse momentum, the more background there is to contend with.

Photon p_T (GeV/c) *Centrality (%)*	**0.2-0.5**	**0.5-1.0**	**1.0-2.0**	**>2.0**
0-20%	S: 193966 B: 218003	S: 146891 B: 75971	S: 48027 B: 27376	S: 9833 B: 5244
20-40%	S: 213017 B: 97205	S: 146621 B: 28951	S: 54532 B: 11714	
40-90%	S: 124461 B: 18802	Signal: 109636 events Background: 7281 events		

Table 1: Centrality and photon p_T bins used in this research, and the number of data points within them.

As shown in Table 1, the ratio between background and signal photons does not remain constant throughout the data. However, for the BDT to work optimally, the amount of signal and background fed into the algorithm will be taken as 50% each. Any excess data will just be used for testing.

5.2 BDT Settings

Outside the ALICE photon research group, BDTs have been used previously, and the paper *Photon Conversion Classication by Boosting Decision Trees* (Schaapherder, 2018)[7] has studied the feasibility of using machine learning to classify photons. In his paper, Schaapherder determined that the standard settings of the BDT as provided in the TMVA package remain valid for photon classification.

Indeed, as a part of this research it could be verified that increasing the number of trees above the standard settings has a marginal effect on the significance of the resulting cut.

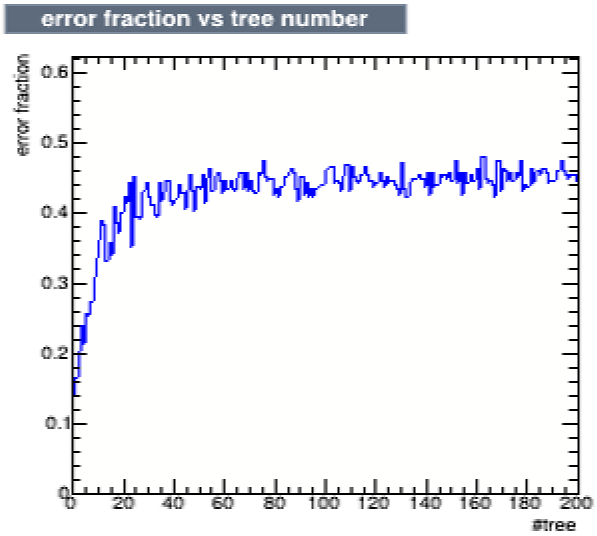


Figure 5: Error fraction for the BDT after training up to 200 trees in the forest.

Option	Value
NTrees	200
MinNodeSize	2.0%
MaxDepth	4
BoostType	AdaBoost
AdaBoostBeta	0.5
BaggedSampleFraction	0.5
SeparationType	GiniIndex
nCuts	20

Table 2: Settings used within the standard implementation of the BDT.

Different settings above and below the standard yield a very similar result, and Figure 5 shows that the error fraction as a function of the decision tree number levels off long before n=200. To reduce the risk of overtraining, then, rather low settings will be used. These settings are shown in Table 2.

5.3 Variables used for training

The Monte Carlo simulations and ALICE experiments yield a whole host of variable measurements coming from its many detectors. The first step in configuring the BDT algorithm should be to critically assess these variables and their usability in the training and testing phase. Since the training of a BDT is a relatively fast process (i.e. computing time is not a major issue), in principle, more variables for training will yield a better result.

However, there remain two good reasons not to use a certain variable for training. The first of these is a strong, direct correlation with photon p_T. The data for lower p_T contain a higher fraction of signal to background. If the BDT is allowed to take this into consideration

Rank	Variable	Separation
1	dEdxElectronITS +dEdxPositronITS	3.353 e-01
2	photonR	3.332 e-01
3	photonCosPoint	3.148 e-01
4	nSigmaTPCPositron	2.434 e-01
5	nSigmaTPCElectron	2.346 e-01
6	photonInvMass	2.112 e-01
7	photonQt	1.964 e-01
8	clsITSPositron	1.841 e-01
9	clsITSElectron	1.765 e-01
10	photonPsiPair	1.555 e-01
11	photonAlpha	7.049 e-02
12	fracClsTPCPositron	5.661 e-02
13	fracClsTPCElectron	5.584 e-02

Table 3: All variables used to train the BDT. The second column shows their variable separation ranking for a training run of the BDT on data with a centrality of 20-40% and a photon p_T cut from 0.5 to 1.0.

when training, it will be more inclined to cut away data at lower p_T, which decreases the reliability of the BDT output when asked for data points at varying transverse momenta. The second reason is when a variable is simply independent of the fact that a photon candidate is background or signal. This can be determined visually by looking at the input variable plots generated by the BDT, but the training macro in TMVA also provides a numerical measure for the importance of a variable in the training process, the variable separation ranking. Variables that performed exceptionally poorly in this test were removed.

According to these requirements, Table 3 shows the variables that were deemed suitable for the training process. Histograms for these variables are shown in Appendix A, for signal and background. The bigger the difference between the signal and background distributions, the better; this allows the BDT to make effective cuts, and this is reflected in the separation score. A similar list of variables was compiled in the feasibility study for applying BDTs to photon classification (Schaapherder, 2018), which differs from this list on a few points. Notably, the ptElectron/ptPositron variables were removed due to their high linear correlation with photon p_T, and the clsITSElectron/clsITSPositron variables were retained.

5.4 Photon p_T binning

The variable photonAlpha also shows a high correlation with the transverse momentum of the photon. However, as Figure 6 shows, this correlation is certainly not linear and due to the exceptional shape of the correlation we can account for this by separating the data into several bins, for different values of p_T. Even though this decreases the amount of data available for training each BDT, separating the data into bins also helps increase the reliability

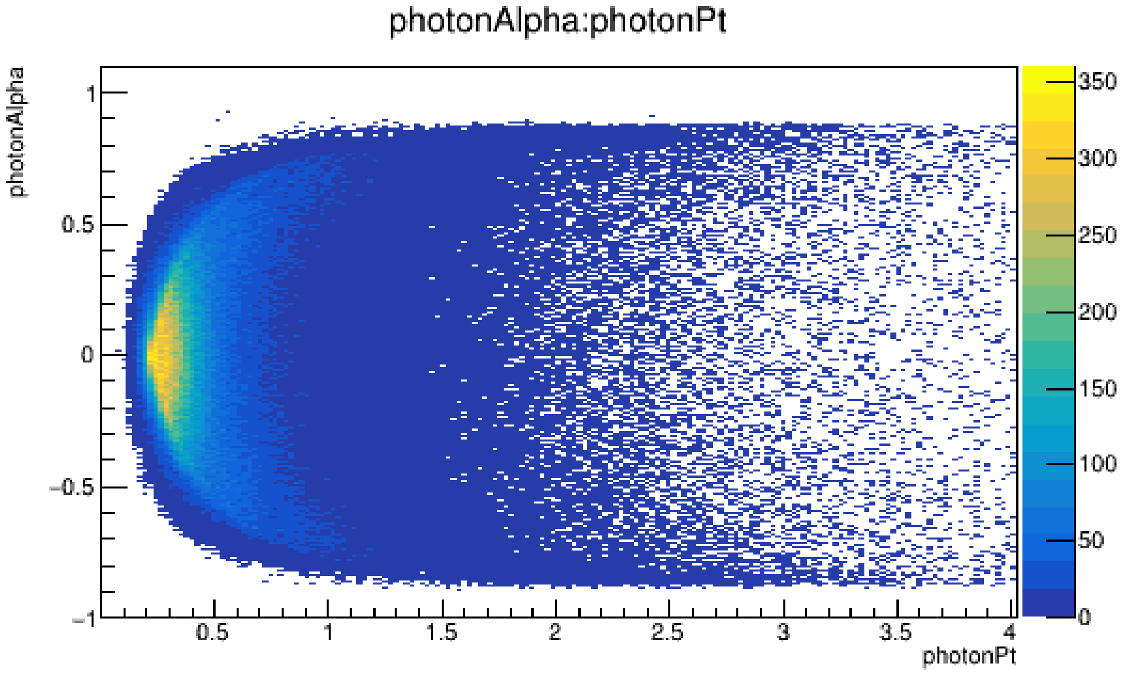


Figure 6: Scatter plot of photon alpha / p_T combinations in the background input data.

of the output. Most variables have some kind of small correlation with respect to p_T, which is often a variable of interest when researching photons (or, in fact, many other particles).

The correlation between alpha and transverse momentum of the photon shows a radical change around $p_T = 0.5$, so it makes sense to put the border of the first bin at this point. Further borders for the bins are chosen with two other considerations in mind. Firstly, that some of the variables distributions change slowly as p_T values increase above 0.5, which affects the ideal cut locations only subtly. Secondly, that there is a rapidly decreasing amount of data available as p_T increases. $p_T = \{1.0; 2.0\}$ therefore work well as the other borders.

Figure 7 shows the distribution of the photons alpha values, for each of the transverse momenta bins. It is clear that the shape of the graph changes radically around $p_T = 0.5$, as the single peak around alpha = 0 disappears. This affects the background photons in particular. The gradual change after $p_T = 0.5$ is also visible as the standard deviation of the background photon alpha value increases.

5.5 Centrality binning

The data received from the Monte Carlo simulation is grouped in 7 bins according to the centrality of the lead-lead collision. It would be convenient to merge as many of these bins as possible to reduce the number of different BDTs that have to be trained. However, a different centrality gives rise to different physical events for example, the signal to background ratio correlates with the centrality.

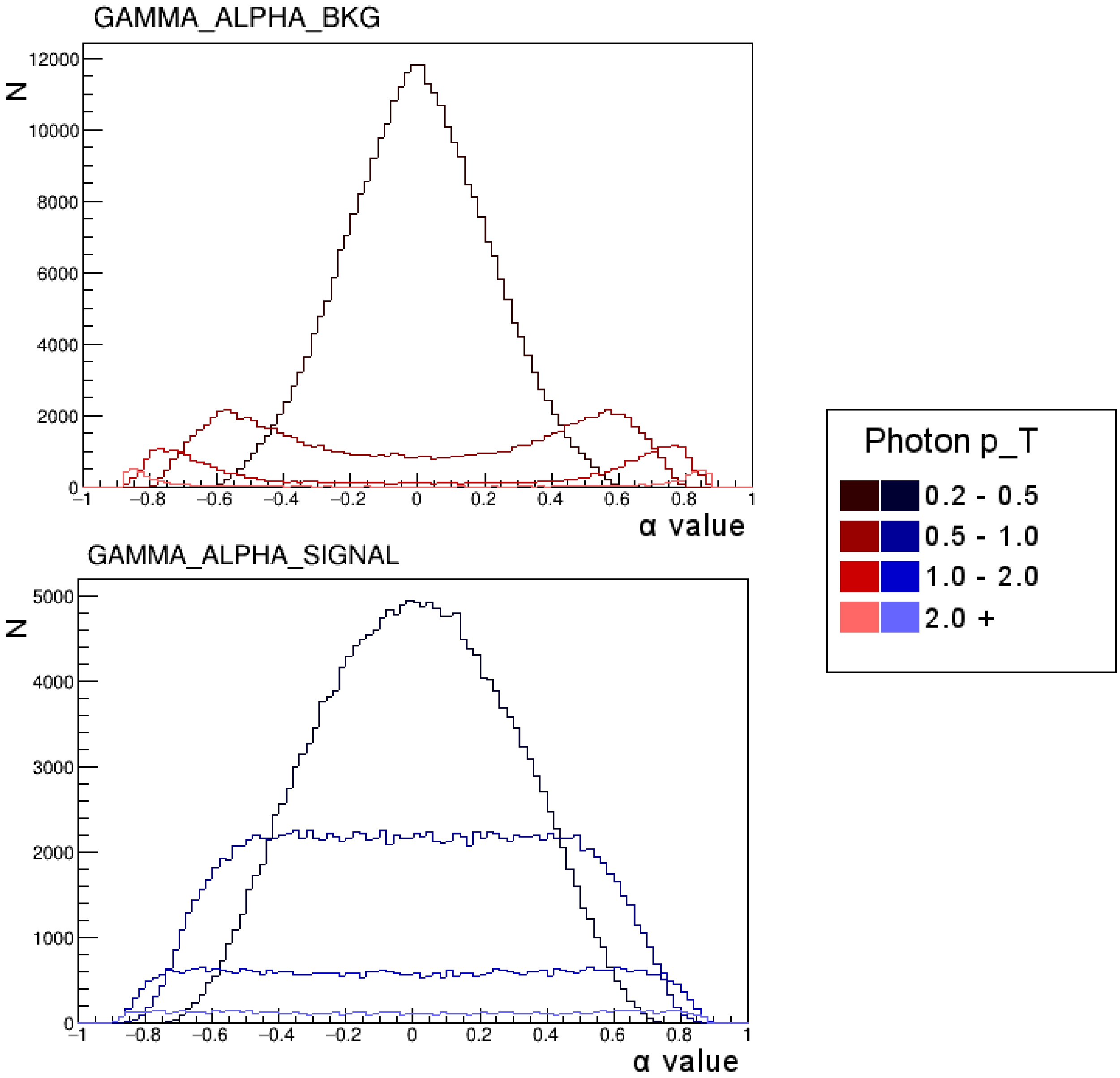


Figure 7: Histogram of photon alpha value occurrence for varying photon p_T. The top graph shows only background photons, the bottom graph only signal photons. Data pictured for a centrality of 0-20%. The graphs for lower photon p_T have more data and are therefore higher overall.

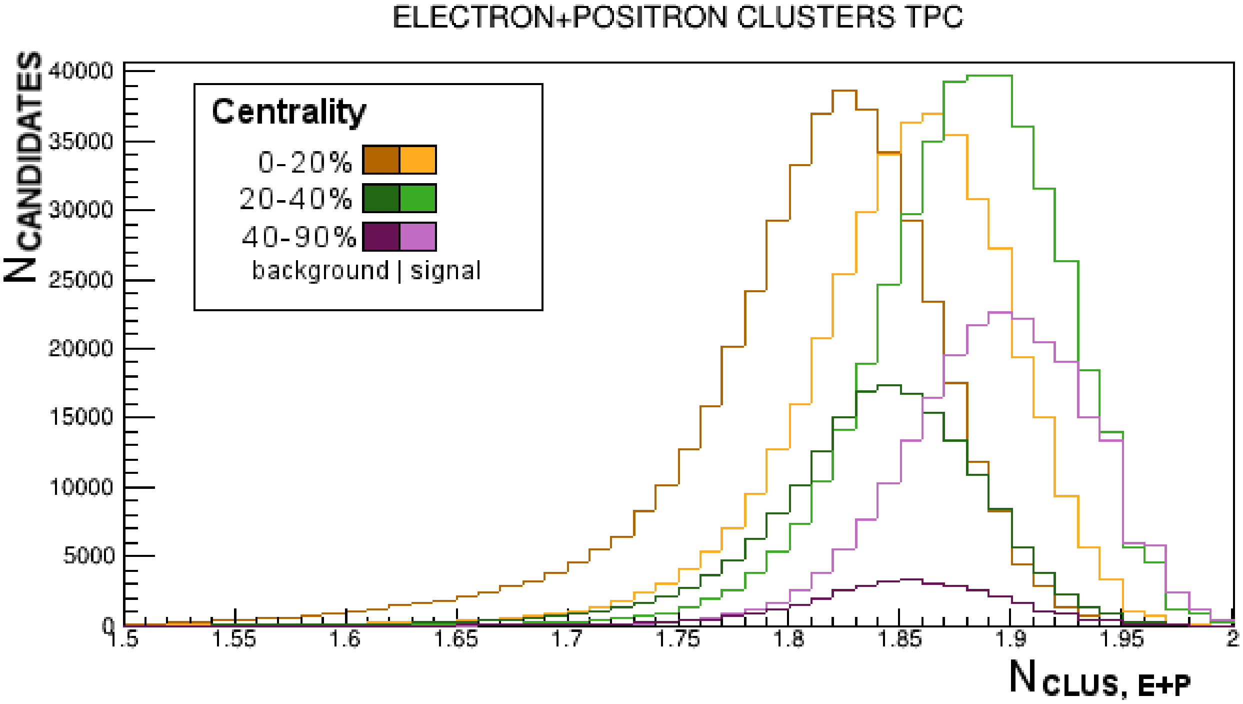


Figure 8: Histogram of the distribution of the combined amount of TPC clusters triggered by an electron-positron pair, coming from signal (light) and background (dark) photons, for varying centrality of the collision event creating those photons.

Furthermore, some of the input variables have a different distribution when the centrality of the event is changed. In particular, the dE/dx of the electron-positron pair in the ITS, and the number of clusters they trigger in the ITS and TPC are different. The latter of these variables is shown in Figure 8 for three centrality ranges. It is clear that the relatively straightforward instructions in a decision tree (i.e. make a cut when variable y is larger/smaller than x) arent ideal for this variable, because the locations of the signal and background peaks change, and therefore the ideal cut changes.

Therefore, instead of universally applying the same cuts to all variables, independent of the centrality of the event, separating the data into centrality bins will yield a better end result.

Unfortunately, the least central event bins (such as 80-90%) contain much less data than the more central bins (such as 0-10%). The least central event bins are also skewed to contain a much higher fraction of signal to background, which means that a lot of signal data cannot be used for training. After all, the data fed to the BDT should ideally have as many signal as background events. Therefore, in order to preserve a high enough amount of data to train the BDT accurately, some bins will have to be merged, particularly at higher centrality percentiles. The cut-off point for enough data is chosen to be at least 5000 background and signal events per bin. The resulting bins and the data they contain are shown in Table 1.

6 Results

According to the specifications determined in the previous sections, nine different BDTs were trained, one for each of the centrality and p_T bins. This chapter includes the results of various tests on the reliability and performance of the algorithms. Section 4.3 provides the theoretical context for these tests. Section 6.1 is focused on the reliability of the BDTs and describes the application of a Kolmogorov-Smirnov test and its implications. Then in section 6.2, the performance of the algorithms is tested by applying them to Monte Carlo test data in AliROOT.

6.1 Kolmogorov-Smirnov test

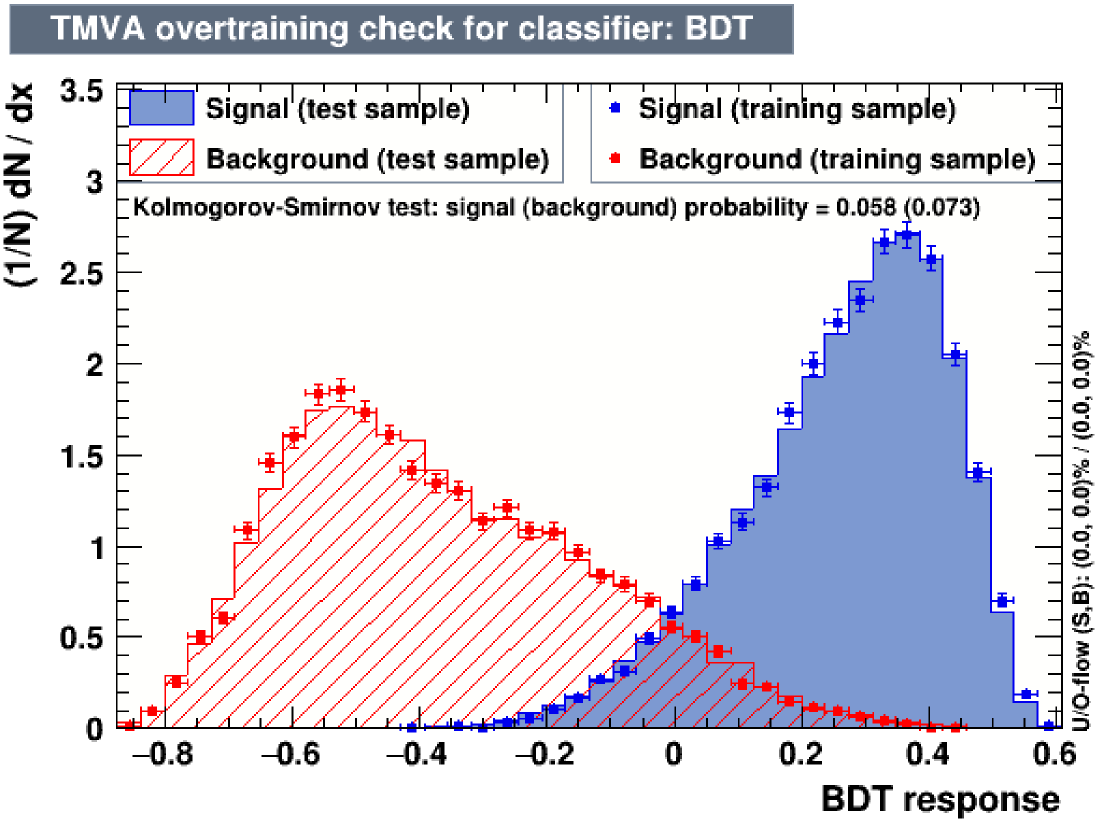


Figure 9: BDT response for test and training samples of data at a centrality of 20-40% and p_T of 0.5-1.0.

In order to test the reliability of the trained algorithms and ensure that they were not subjected to overtraining, a K-S test was applied to each of them. Table 4 shows the result of this test for all the BDTs used in this research. It can be seen that all results are within a reasonable margin of likelihood. Figure 9 shows the worst test result, and illustrates that even when the probability that the distributions are exactly the same is relatively small, they still have a very similar shape, which means that potential small differences due to overtraining or a difference in the training data do not have a big effect on the result. The ideal cut

Centrality (%)	p_T (GeV/c)	Probability (signal, in %)	Prob. (background, %)
0-20	0.2-0.5	17.5	53.1
0-20	0.5-1.0	59.0	26.8
0-20	1.0-2.0	13.4	42.9
0-20	2.0+	8.0	6.6
20-40	0.2-0.5	91.5	40.5
20-40	0.5-1.0	5.8	7.3
20-40	1.0+	16.6	8.0
40-90	0.2-0.5	90.8	11.3
40-90	0.5+	28.3	7.0

Table 4: K-S test results for all trained BDTs. The test is applied separately to background and signal distributions.

number and significance remain similar even in this case.

6.2 Application to Monte Carlo data

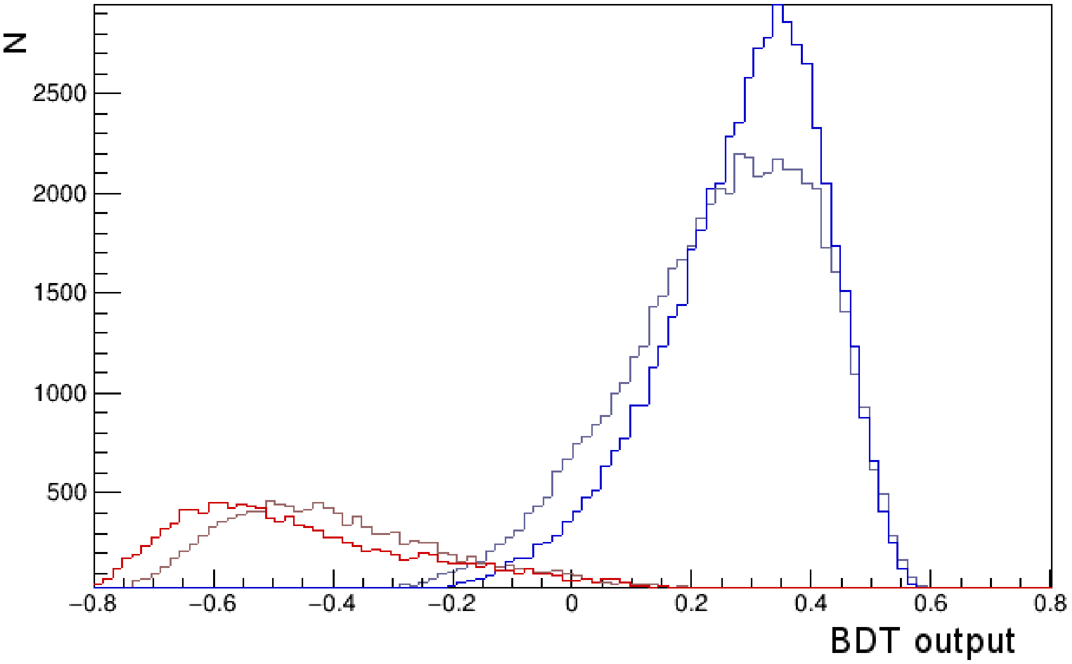


Figure 10: BDT output histograms, for photons of event centrality 20-40%, $p_T > 1$. The dark coloured lines are signal (blue) and background (red) output after applying the matching BDT weight files. The lighter coloured lines are similar, but using the weight files for centrality 40-90%, photon p_T 0.2-0.5.

The nine different weight files and their functionality were successfully loaded into the AliROOT framework which allows for further testing of the BDT performance. Specifically, the improvement gained by applying BDTs for several data bins, over applying one universal BDT, and over the by-hand method of applying cuts to variables which is currently in use.

Figure 10 illustrates the difference that centrality and photon p_T binning makes on the eventual purity and efficiency of the data after using the BDT. When using the correctly trained BDT (the one that was trained using data of the same centrality and p_T as the data it is applied to), the separation achieved between signal and background is better than using a BDT trained for one of the other event bins. Especially the signal output is more peaked around an output value of 0.3, at which point there is no more background output. This means that a cut in the data can achieve a higher purity, for the same efficiency, if the data is analysed using the BDT weight files with the appropriate centrality and photon p_T.

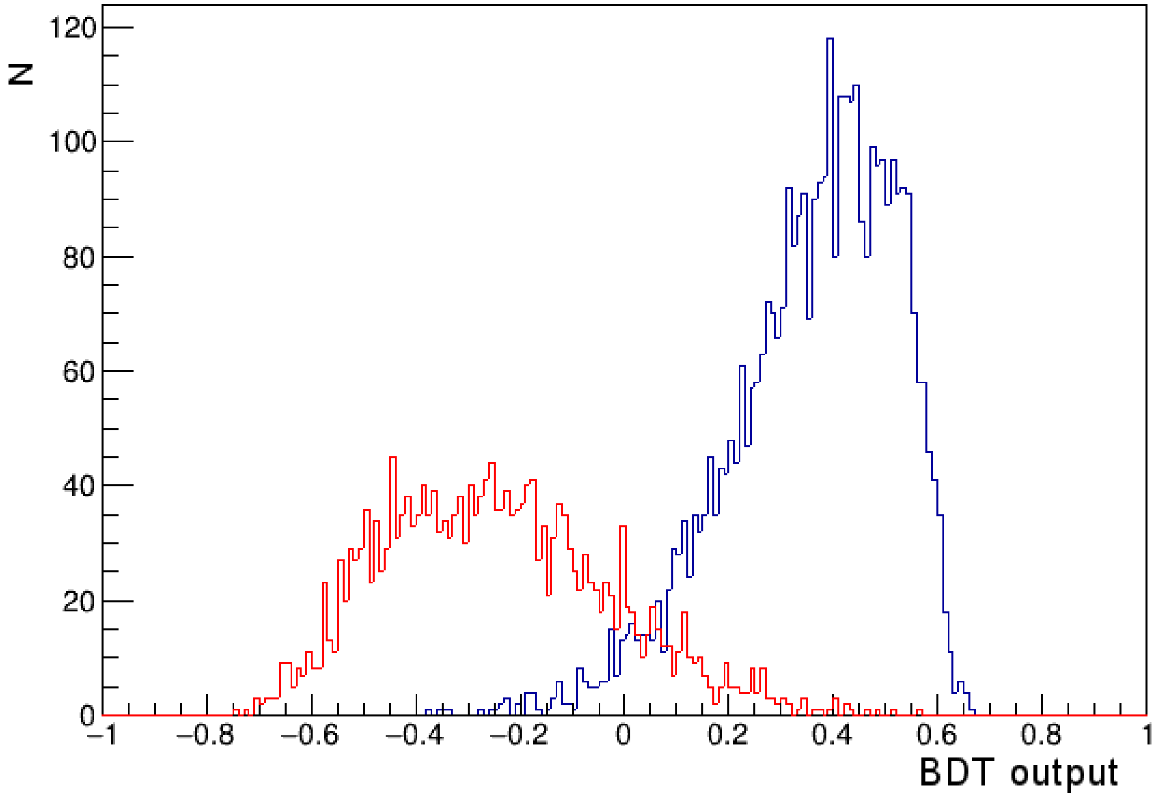


Figure 11: BDT output histogram of a Monte Carlo simulation run within the ALICE framework, on data with all usual cuts already pre-applied, displaying signal (blue) and background (red) as a function of the BDT output value.

Figure 11 shows that even after the by-hand cuts have been applied, there is ample background contaminating the data. The figure shows only data that would be classified as signal with the usual method, but the real signal and background are known since we can access that data in the Monte Carlo simulation. Before applying the boosted decision tree, the purity of the data equals 64.1%, for a total significance of 49.9. By applying the BDT algorithm to further separate out signal and background, a cut in its output can radically increase the purity of the accepted data at a low loss of efficiency.

For this data a cut on the BDT output value between -0.2 and +0.3 makes the most sense, depending on whether purity, efficiency or significance is required. By computing the integral of the signal and background for various cutting points, the ideal cut can be determined. Table 5 shows some of the possible cuts within the interval [-0.2;+0.3] and their attributes, and it can be seen that the highest significance of the resulting data can be achieved with a cut on the BDT output around 0.05. At this point, a cut would result in a purity of 94.1%, which is an improvement of 30.0% compared to the manual cut.

Figure 12 illustrates this point; the safest cut at a low BDT output of -0.2 will achieve a similar efficiency to the manual cuts, but with a much higher purity. If efficiency is not essential, then even higher purity values can easily be reached by cutting at a higher BDT output.

BDT output cut	#Background	#Signal	Efficiency	Purity	**Significance**
-0.2	891	3872	0.996	0.813	56.1
-0.1	553	3849	0.991	0.874	58.0
0	327	3788	0.975	0.921	59.1
0.05	233	3718	0.957	0.941	59.2
0.1	161	3638	0.937	0.958	59.0
0.2	77	3298	0.849	0.977	56.8
0.3	24	2736	0.704	0.991	52.1

Table 5: Some of the possible BDT output cuts within the interval [-0.2;+0.3] and their attributes.

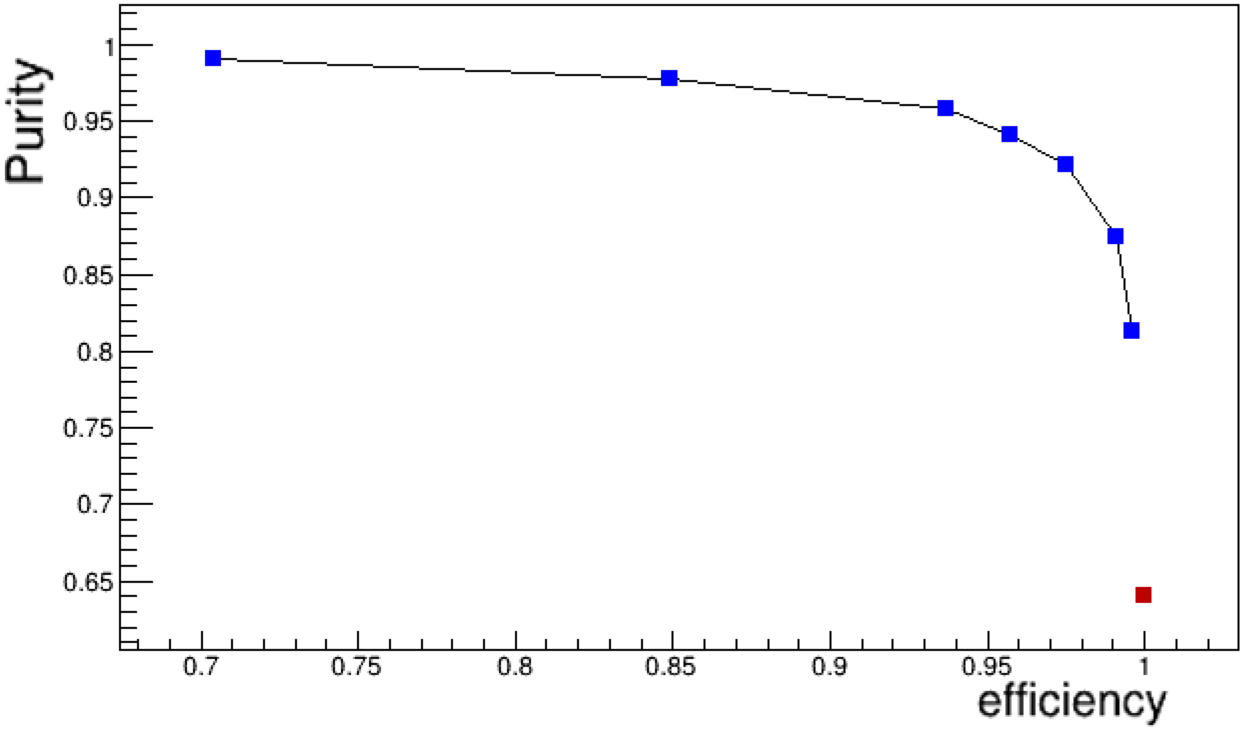


Figure 12: Purity as a function of efficiency (defined as 1 for manually cut data) for various BDT cuts (as in Table 5, blue) and the manual cut (red).

7 Discussion

Because the transverse momentum of the photon (p_T) is often studied and used as a variable in research into photons, it would be ideal if the BDT did not make any cuts that correlate with the transverse momentum of the photon. Unfortunately, the training and testing data has a much higher ratio of signal to background for higher centrality percentages. In particular, the bin for p_T between 0.2 and 0.5 has a significantly lower fraction of signal data, as shown in Table 1. Figure 13 shows that the BDT output indeed has the expected dip between 0.2 and 0.5 p_T. This makes it hard to assess whether the algorithm explicitly selects for high p_T. The variables used in this research were chosen to avoid this selection (section 5.3), but further research may be necessary to confirm that explicit p_T selection indeed doesn't occur. Of course, this is only a potential issue for the lowest p_T bin, as Figure 13 shows that the input/output ratios fluctuate a lot less for p_T higher than 0.55.

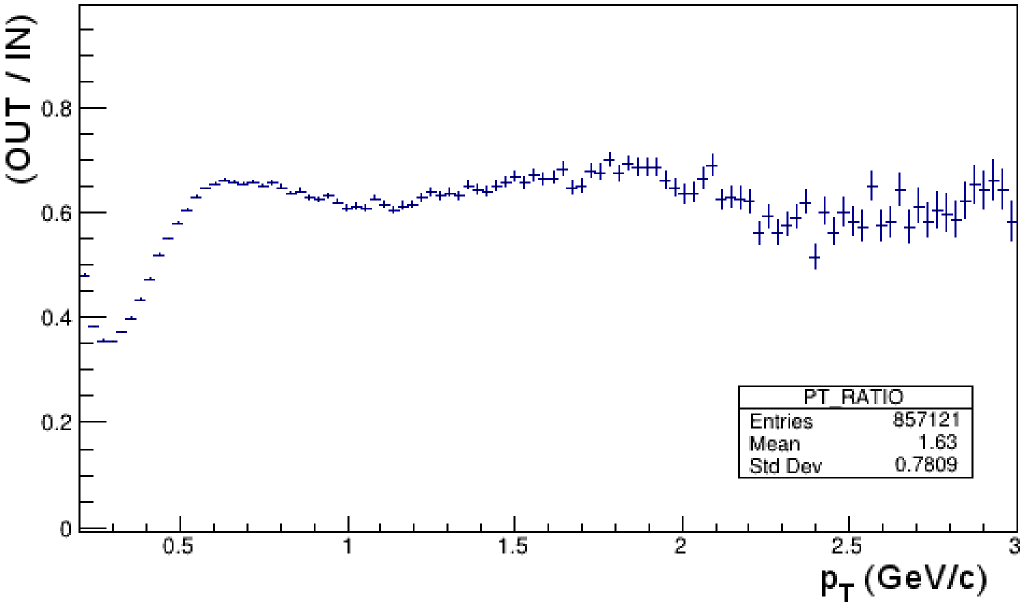


Figure 13: The ratio between the photon p_T spectrum after selecting for a BDT output higher than -0.1 and the p_T spectrum before applying the BDT, for 100 p_T bins. Data for centrality 0-20%.

It is possible that some uncertainties and inaccuracies arise from the fact that for the training of the BDT algorithm, Monte Carlo simulated data was used instead of real data collected from an experiment at ALICE. Using experimental data is very impractical since we dont have separate lists of signal and background photons which can be used to check and improve the algorithm, whereas the Monte Carlo simulation gives a theoretically infinite amount of this data to work with. Of course, if the simulation is subtly wrong, it could cause the BDT to train itself on an artefact of the simulation which does not occur in a real experiment. In this research, extra care has gone into avoiding overtraining of the BDT, which should also reduce this effect of small errors in the simulation.

8 Conclusion

The aim of this research was to develop a Boosted Decision Tree to analyse data from ALICE experiments in such a way that it can assign a BDT output value to the detected photons, corresponding to the probability that it is a signal photon, so that a single cut can be made by the researcher to separate background from signal. 7 separate Monte Carlo simulation runs produced a list of variables for each of just over 1.5 million events for the algorithm's training. Building on feasibility research into this subject, suitable variables were chosen from these, as shown in Table 3.

The first criterion was that the BDT method should produce reliable results for varying ranges of photon p_T and centrality of the primary collision event. As shown in Figures 6 and 8, these ranges give rise to different physical effects, so the input data was separated into 9 bins. They were chosen to have both sufficient data for reliable training, as well as negligible correlations between variables and the photon's transverse momentum. Table 1 shows these bins and the data they contain. There is less signal data for lower p_T, which shows as the BDT cuts away more photon candidates here. However, there is no indication that this is due to explicit selection of higher p_T by the algorithm.

The second criterion was that, by using the BDT and making a cut on the output values, the resulting data should be demonstrably better than the by hand cutting method currently used to separate background photons from signal. To this end a K-S test was performed to check for overtraining (Table 4) which yielded a good result. The BDT was then applied on data which was already improved in the usual method, and Figure 11 shows that significant improvement can be gained by applying the BDT. At maximum significance, signal purity was increased by 30%, from 64.1% to 94.1%. A sharper cut on the BDT output could increase this even further, but at the cost of data efficiency.

With the tools developed in this research, which have been added into the AliROOT framework, the ALICE photon research group will be able to improve the accuracy of their results. As shown, the significance of the data will increase by using the BDT method, and thus significant conclusions can be drawn even with less data to work on, and fewer corrections for contamination by background pairs. A next step to this research could, for example, be to determine which of the photons identified by the BDT are direct photons emitted from a quark-gluon plasma. Identifying these specific photons might shed further light on this interesting state of matter. Further improvement to the results themselves may be gained by looking into different kinds of machine learning techniques such as neural networks. Perhaps these algorithms can achieve an even higher purity and efficiency. Furthermore, any improvement to the Monte Carlo simulation methods will also prove a boon for the usefulness of the BDT, since it relies on the accuracy of the training data.

References

[1] *Standard model*, URL `https://www.physik.uzh.ch/en/researcharea/lhcb/outreach/StandardModel.html`.

[2] B. R. Martin, *Nuclear and Particle Physics* (Wiley, 2009), ISBN 9780470742754.

[3] *Alice experiment*, URL `http://alice.web.cern.ch/content/experiment`.

[4] *Heavy ions and quark-gluon plasma*, URL `https://home.cern/science/physics/heavy-ions-and-quark-gluon-plasma`.

[5] *Alice at cern*, URL `https://home.cern/science/experiments/alice`.

[6] M. Sas, Utrecht University (2016), master's thesis.

[7] T. Schaapherder, Utrecht University (2018), bachelor's thesis.

[8] B. P. R. et al, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **543**, 577 (2005), https://doi.org/10.1016/j.nima.2004.12.018.

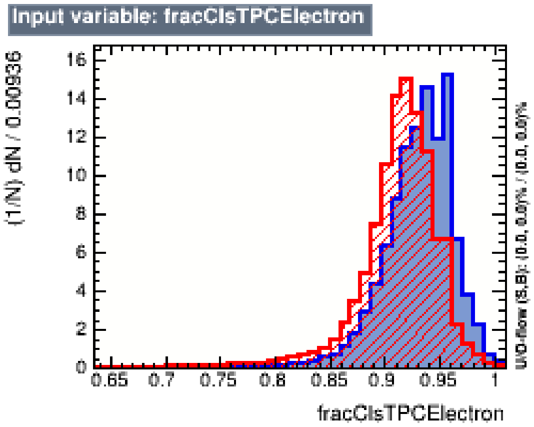
[9] M. Sas, Fontys Eindhoven (2014), bachelor's thesis.

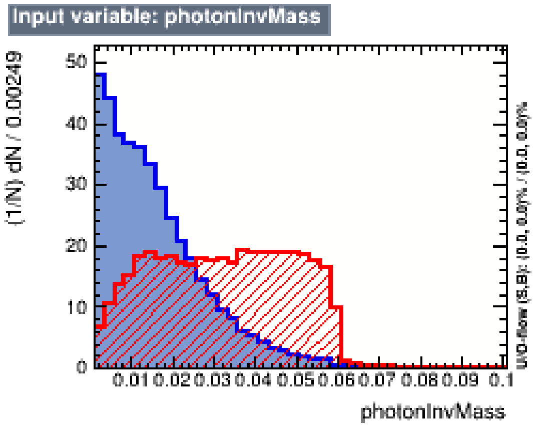
[10] N. J. Salkind, *Encyclopedia of research design* (SAGE Publications, 2010), ISBN 9781412961271.

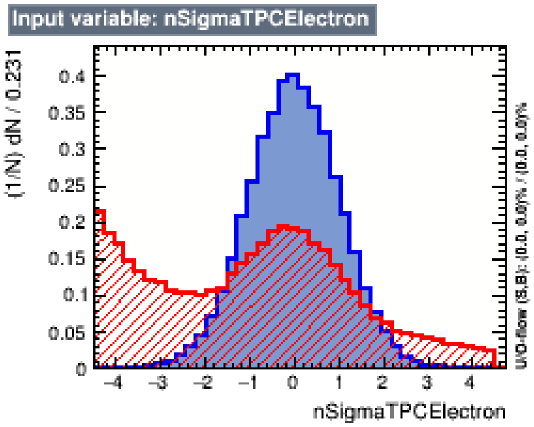
[11] *Root user guide*, URL `https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html#introduction`.

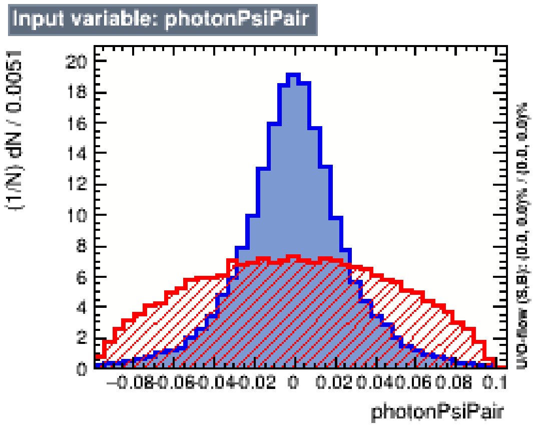
[12] *Particle transport and detector simulation*, URL `http://alice-offline.web.cern.ch/Activities/Simulation/ParticleTransport.html`.

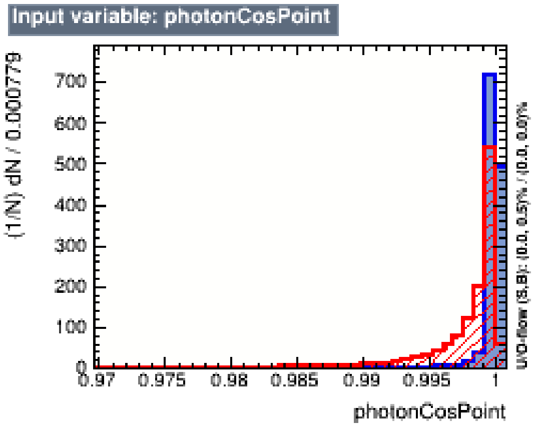
A Histograms for all input variables

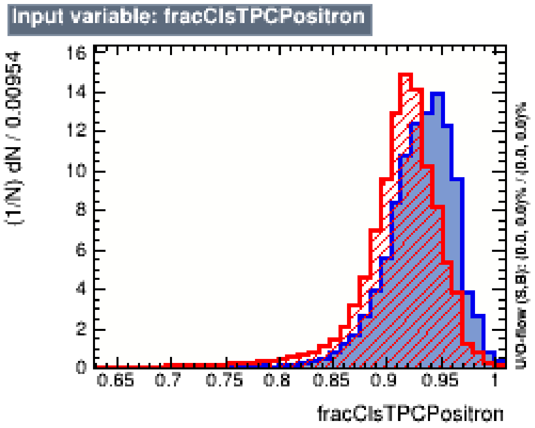


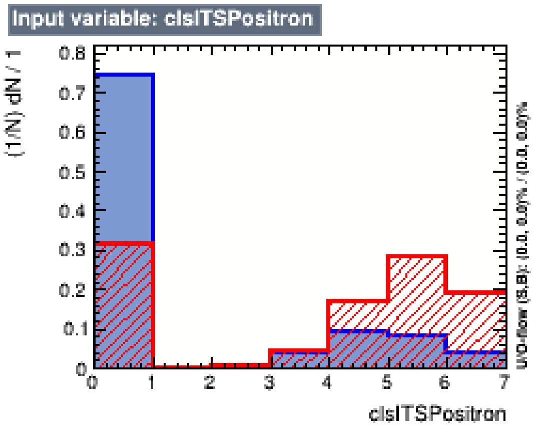


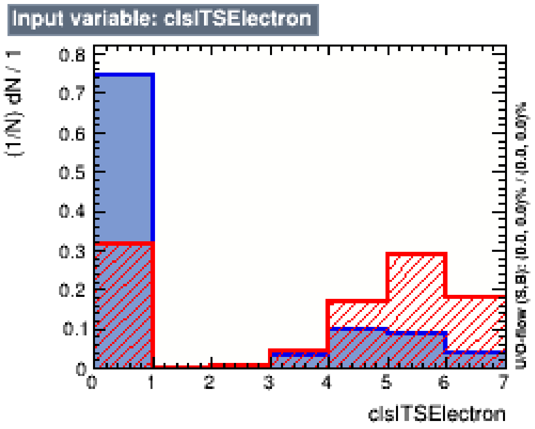


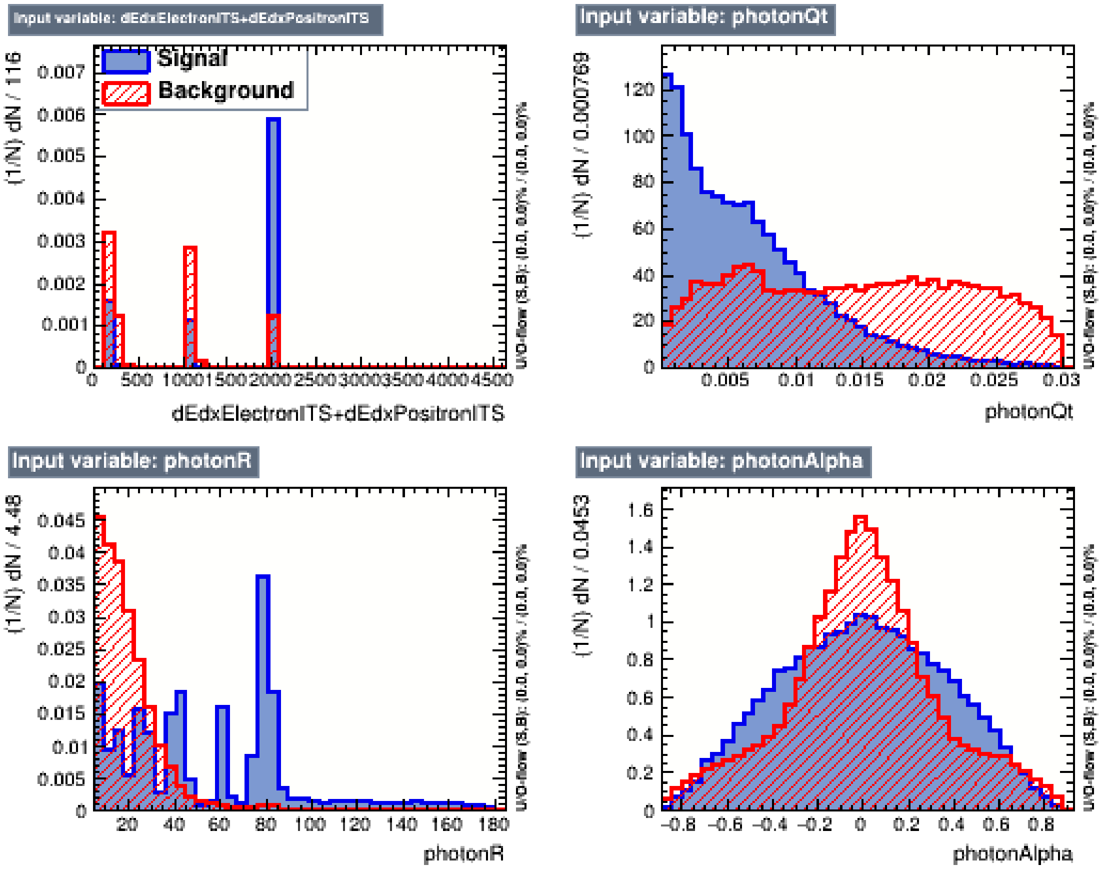


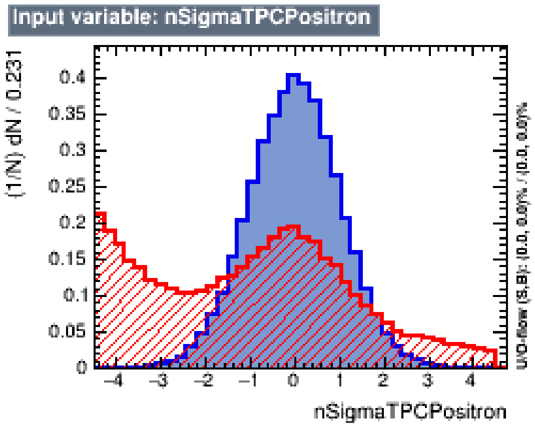

Input variable: photonCosPoint
(1/N) dN / 0.000779
0
100
200
300
400
500
600
700
0.97
0.975
0.98
0.985
0.99
0.995
1
photonCosPoint
U/O-flow (S,B): (0.0, 0.5)% / (0.0, 0.0)%


Input variable: fracClsTPCPositron
(1/N) dN / 0.00954
0
2
4
6
8
10
12
14
16
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
fracClsTPCPositron
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%


Input variable: clsITSPositron
(1/N) dN / 1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
6
7
clsITSPositron
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%


Input variable: clsITSElectron
(1/N) dN / 1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
6
7
clsITSElectron
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%


Input variable: dEdxElectronITS+dEdxPositronITS
Signal
Background
(1/N) dN / 116
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
dEdxElectronITS+dEdxPositronITS
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%
Input variable: photonQt
(1/N) dN / 0.000769
120
100
80
60
40
20
0
0.005
0.01
0.015
0.02
0.025
0.03
photonQt
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%
Input variable: photonR
(1/N) dN / 4.48
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
20
40
60
80
100
120
140
160
180
photonR
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%
Input variable: photonAlpha
(1/N) dN / 0.0453
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
photonAlpha
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%


Input variable: nSigmaTPCPositron
(1/N) dN / 0.231
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
−4
−3
−2
−1
0
1
2
3
4
nSigmaTPCPositron
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%