Master Thesis

Using linear decorrelated machinelearning models for particle identification of high energy electrons with energy more than 80GeV in real data from the Large Hadron Collider in CERN

Lau Dam Mortensen, pdm882

Academic Advisors:
Associate Prof. Troels C. Petersen
Niels Bohr Institute
Experimental Subatomar Physics
University of Copenhagen

PhD Daniel S. Nielsen
Experimental Subatomar Physics
Niels Bohr Institute
University of Copenhagen

Submitted: May 31st, 2020

THANKS

I would like to thank my advisor Troels C. Petersen for his great patience and helpfullness with this project. Thanks to Daniel S. Nielsen for his help and advice. Thanks to Helle Leerberg, Rasmus F. Ørsøe, Sara D. Pinhold, Malte Algren, Bjørn Mølvig, Benjamin K. Henckel, Lukas Ehrke and others for talks and illuminating discussions.

Abstract

This thesis aims to find high energy electrons in data from the ATLAS Detector in CERN using machine learning models based on boosted decision trees. The main challenge in data is labeling the data correctly. With labels made by a model trained with isolation variables on simulated data from $W \rightarrow e\nu$, and predictions in $W \rightarrow e\nu$ data from ATLAS by a model trained with particle identification variables, results showed good separation, but were ambiguous due to a discrepancy in ATLAS Likelihood predictions and model predictions. Trying the same settings in $Z \rightarrow ee$ data from the ATLAS detector, showed no discrepancy between ATLAS Likelihood and model predictions. Using particle identification variables in label-making and isolation variables for training with data from $W \rightarrow e\nu$, showed good separation and no discrepancy between model predictions and ATLAS Likelihood. Linear decorrelation was tried but showed ambiguous results. Evaluation of models using the Z-peaks showed, that models trained on simulated data outperform models trained on data from ATLAS detector, but all models improve over ATLAS Likelihood Loose workingpoint.

CONTENTS

1 BACKGROUND 5
- 1.1 The Standard Model and high energy electrons 6
- 1.2 Electrons and photons in the ATLAS physics programme 9
- 1.3 LHC and the Atlas Detector 9
  - 1.3.1 The Inner Detector 11
  - 1.3.2 Calorimetry 13
  - 1.3.3 Other Systems 17
  - 1.3.4 The trigger system 18
- 1.4 Electron reconstruction 18
  - 1.4.1 The Atlas Likelihood 20
- 1.5 Machine Learning 22
  - 1.5.1 ML Datasets 23
  - 1.5.2 Decision trees 23
  - 1.5.3 Hyperparameters 24
  - 1.5.4 Gradient boosting 25
  - 1.5.5 Loss function 25
  - 1.5.6 Predictions 26
  - 1.5.7 Evaluation (ROC and AUC) 26
  - 1.5.8 LightGBM 26

2 TRAINING AND RESULTS IN SIMULATED DATA 28
- 2.1 Introduction 28
- 2.2 Selection of data 28
  - 2.2.1 Train, validation and test sets 28
- 2.3 MC Data selection 29
  - 2.3.1 MC Signal 29
  - 2.3.2 MC Background 29
  - 2.3.3 Energy distribution of data 29
- 2.4 Variables 30
- 2.5 ISO vs PID 30
  - 2.5.1 Isolation 30
  - 2.5.2 PID 31
- 2.6 Reweighing 35
- 2.7 Training 36
  - 2.7.1 Hyper parameter optimization 37
  - 2.7.2 Predictions 37
- 2.8 ROC curve 38

2.8.1 Correlations 38
3 REAL DATA 41
3.1 Introduction 41
3.2 Data selection 41
3.2.1 W data 41
3.2.2 Z data 42
3.3 Making labels in RD 43
3.3.1 Problems with Tag-and-Probe for W data 43
3.4 Initial labels in W - data 44
3.5 Initial labels in Z - data 46
3.6 RD variable distributions 47
3.7 Reweighing in RD 47
3.8 Decorrelation 51
3.9 Cleaning data 52
3.10 ROC curves in RD 54
3.11 Decorrelation results 57
3.11.1 Data from $W \to ev$ 57
3.11.2 Decorrelation In Data From $Z \to ee$ 60
3.12 Evaluation with Z-peaks 61
3.13 Discussion 64
3.14 Conclusion 64
Appendices 75
.1 Z-data initial predictions 76
.2 W-data MC(ISO)W75 initial predictions 76
.3 Variables distribution in MC PID 77
.4 Variables distributions in RD 79
.4.1 W-data 79
.4.2 Z-data 79

1

BACKGROUND

This project looks at data from the ATLAS detector at the Large Hadron Collider (LHC) in Cern. It's objective is to test the efficiency of the Machine Learning (ML) algorithm called Boosted Decision Trees (BDT) compared to the efficiency of the ATLAS Likelihood (LH) at electrons with energy higher than 80 GeV.

Electrons play a central role in the ATLAS physics program. Since there are no electrons inside protons, the emergence of an electron with high transverse energy signifies that something interesting has happened, which involves new and heavy particles (such as e.g. a Higgs particle) which plays the main role in the ATLAS physics program. Furthermore, in searches for new particles, high energy electrons are also very important, for example in the $Z' \rightarrow e^+e^-$ search [5].

Electrons produce a distinct signal in material, and the ATLAS detector is in many respects designed to identify electrons. And the higher the energy, the more distinct the signal becomes. [4]

However, producing an algorithm for selecting high energy electrons with very high efficiency and hardly any background turns out to be hard for several reasons:

- There are very few high energy electrons in the ATLAS data. Above the $Z \rightarrow ee$ peak, the spectrum falls very rapidly, and thus there is not much data to use for tuning the simulation to match this data.

- Even with a source of high energy electrons in real data, the challenge is, that the signal efficiencies and background rejections wanted are high, and this is hard to "prove", given already low statistics.

- At high energies, there are almost no electrons from the Z-peak to measure performance on or to make labels from.

This thesis use a mix of simulated data (MC) and data to gain experience with algorithmic performance. It uses isolation (ISO) as an orthogonal way of identifying electrons. The challenge lies in making the ML-models based on particle identifaction (PID) and ISO uncorrelated, and this has been attempted. The (small) $Z \to ee$ peak has been used as a cross check of performance.

The thesis is divided into two parts:

- A back ground chapter with theory
- Two results chapters with discussions of results for simulated data and real collider data.

This thesis will mainly focus on the machine learning part.
It is unknown whether the successful machine learning (ML) models that we use in this group at the Niels Bohr Institute are succesful at higher energies than 80 GeV. The Machine learning models from earlier work was not trained on specifically high energy electrons, but the models in this project will only be trained on data with energy higher than 80 GeV.

Note: With " ML models" or sometimes just model, it refers to a ML algorithm trained on a specific set of variables

1.1 THE STANDARD MODEL AND HIGH ENERGY ELECTRONS

The Standard Model (SM) is the current best picture, physicists can give, of the fundamental rules of physics that govern how matter behave. In the SM there are fundamental subatomic particles that all other particles and thereby matter consists of. They have been experimentally verified to a large precision. [1]
Among the next frontiers of high energy physics is new theories that are set to fix weaknesses of the SM. These theories predict new particles like the heavy Z′ which decays to two leptons. An overview of the different new search projects and the particles they look for are shown in fig. 3. Many of them include high energy electrons.
Electrons are very important in experimental high energy physics because the are easy to detect. An overview of all the important particles cross-sections that have been measured at the LHC is shown in figure . Except for the proton-proton and jets, all of the particles can convert or decay to electrons.
In many searches high energy electrons are important. And this project seeks to find out how good the current best methods in machine learning are to find high energy electrons.

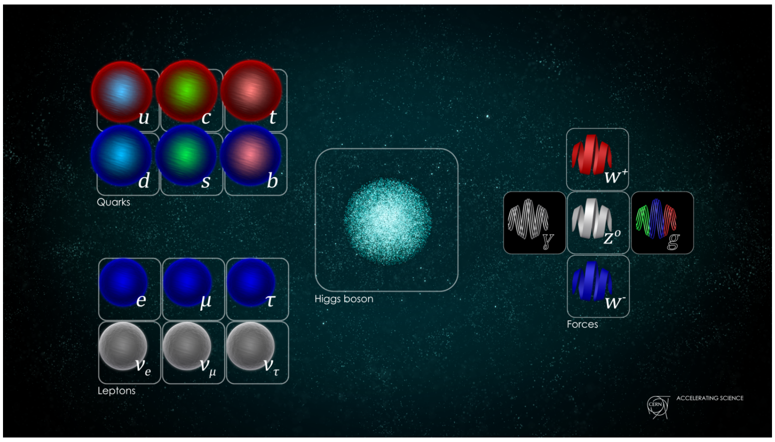


Figure 1: *Fundamental particles in the Standard Model*

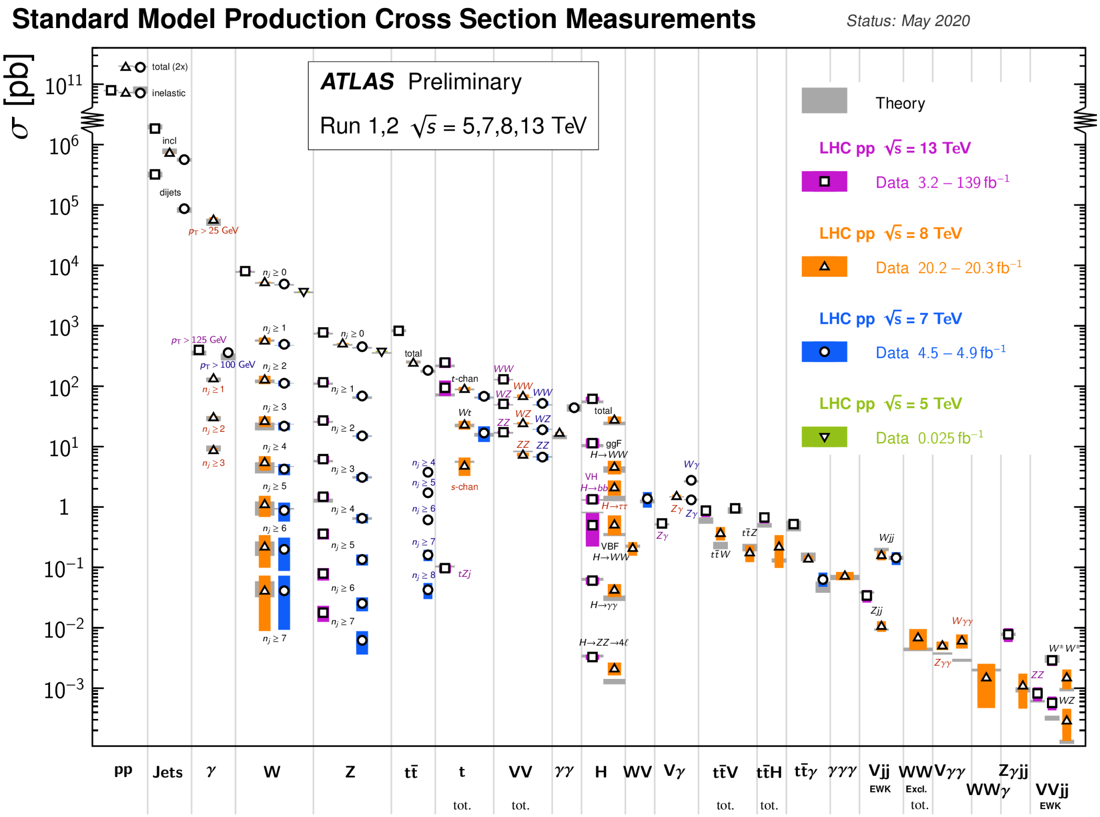


Figure 2: *Overview of cross-section measurements. Except for the Jets and proton-proton they all include possible decays to electrons*

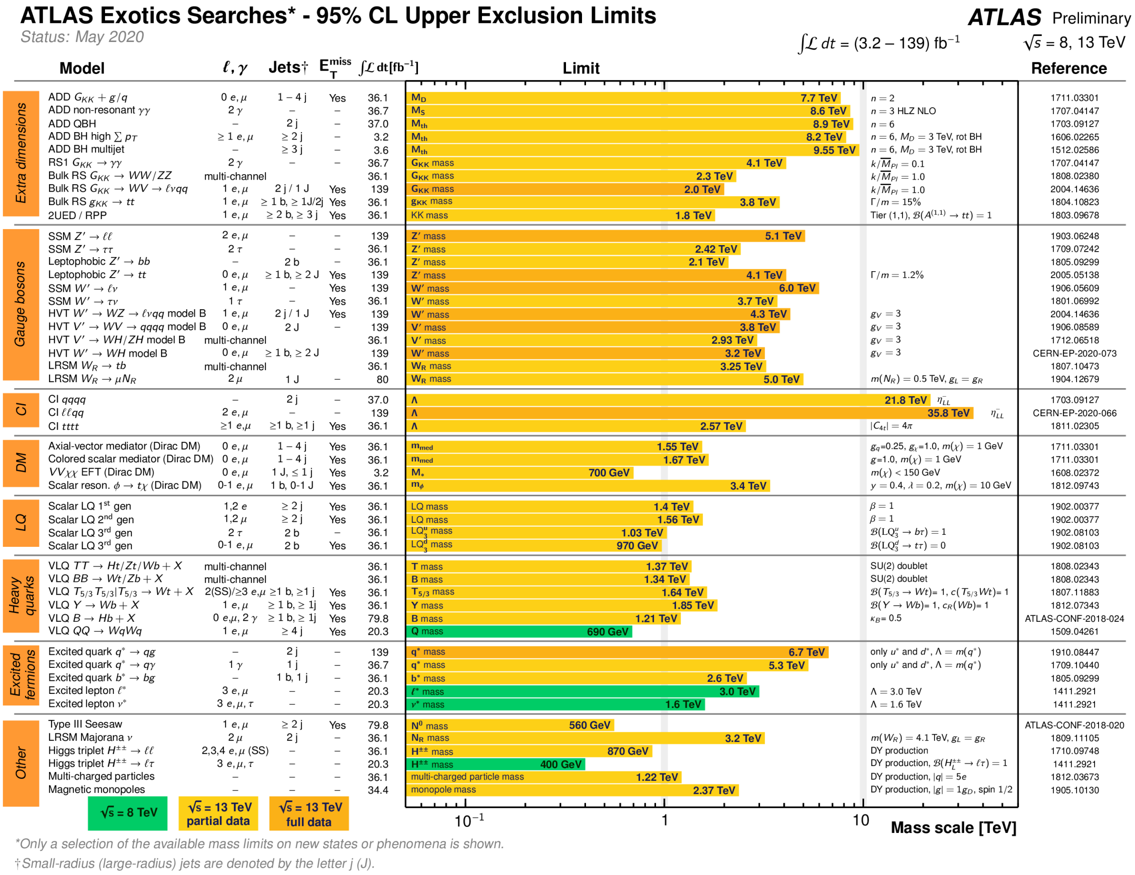


Figure 3: *Exotic searches that include the possibility of new particles. Many of them include high energy electrons. The yellow bars show the energy range in which the models are excluded at 95% confidence level.*

1.2 ELECTRONS AND PHOTONS IN THE ATLAS PHYSICS PROGRAMME

One of the main goals of the LHC was to search for the Higgs boson (from now on referred to as the Higgs). In 2012 the Higgs was discovered [2][3] from the final states

- H$\rightarrow$ZZ*$\rightarrow 4l$
- H$\rightarrow \gamma\gamma$
- H$\rightarrow$WW*$\rightarrow l\nu l\nu$

Asterisk (*) means virtual particle and ν is a neutrino. The neutrino is not possible to detect in the ATLAS detector, but is assumed to be the missing transverse energy (E_T^{miss}), which is the energy in-balance that isn't accounted for in the reconstruction of events.
The mass of the Higgs was found to be approximately 125 GeV. Now that the Higgs has been discovered, the process of making the measurement more accurate and searching for it at higher energies is going on and is what this project seeks to contribute to.

1.3 LHC AND THE ATLAS DETECTOR

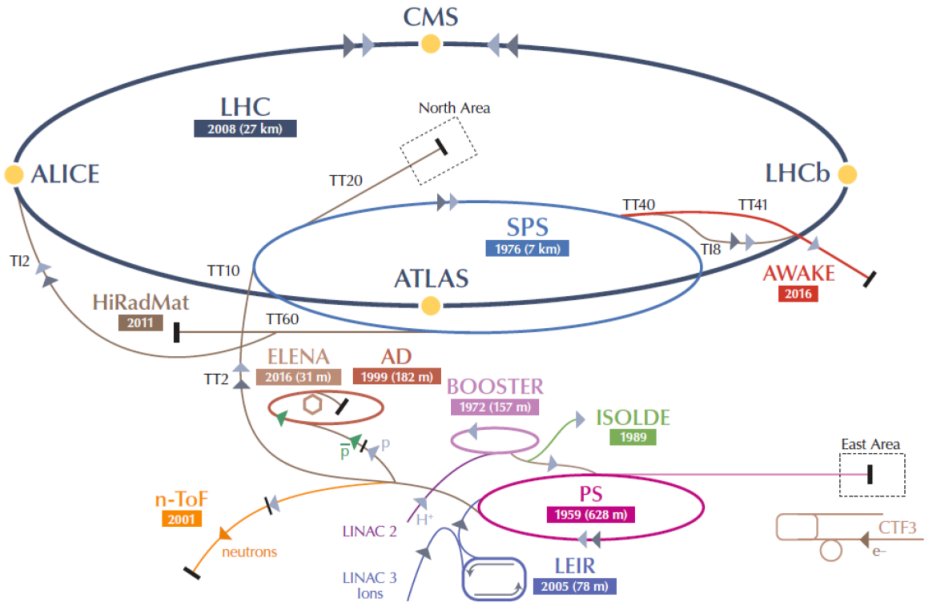


Figure 4: *The Large Hadron Collider [28]*

Physicists smash particles into each other to see what comes out. The Large Hadron Collider (LHC) in Cern, Switzerland, was built to do exactly this. The collider is a complex of wires and magnets build to accelerate protons close to the speed of light. To detect information 4 detectors has been set up. They are named ATLAS, Alice, CMS and LHCb. This project is part of the experiment that goes on in the ATLAS detector.

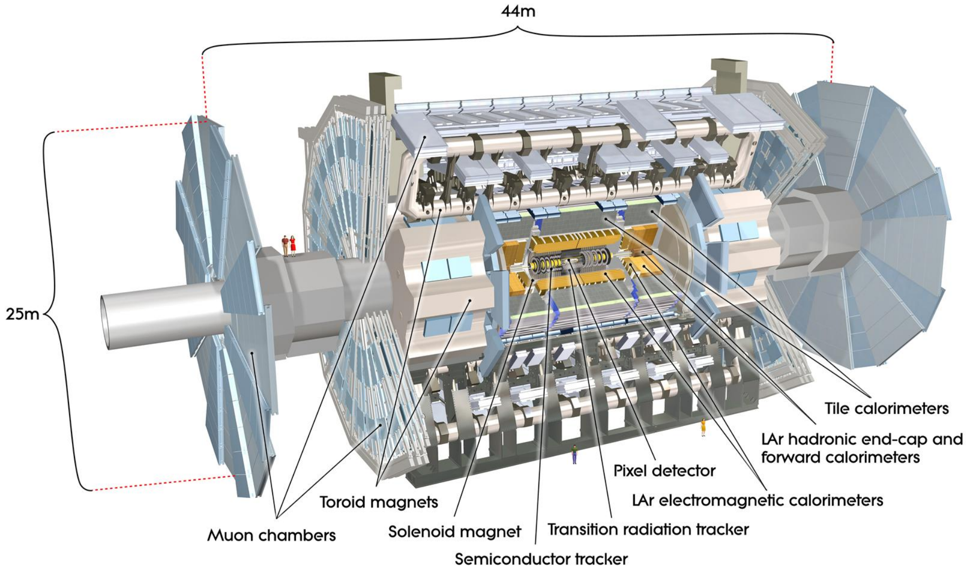

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

Figure 5: *The ATLAS detector [28]*

The ATLAS (A Toroidal LHC ApparatuS) detector is the largest of the detectors in the LHC. It is nominally forward-backward symmetric [9]. Inside collides bunches of up to 10^{11} protons (p) 40 mio. times pr. second.
To describe the detector some coordinates are defined:

- The z direction is the beam direction and the xy plane is transverse to the beam.
- x is vertical and y is horizontal.
- ϕ is the azimuthal angle and is measured around the beam-axis (z).
- θ is the polar angle away from the beam-axis. The (pseudo) rapidity (η) is defined as $\eta = -ln\,tan(\theta/2)$ [9] and is used in-stead of θ. One of the advantages is that differences in rapidity is Lorentz invariant for boosted particles.

- R is the radius with center at the beampipe

The four major parts of the ATLAS detector are the Inner Detector (ID), the Calorimeter, the Muon Spectrometer and the Magnet System. [7] The section most relevant to this thesis, is the Calorimeter, more specifically the Electromagnetic Calorimeter (ECAL), and this is where I will go into more depth than other parts of the detector.

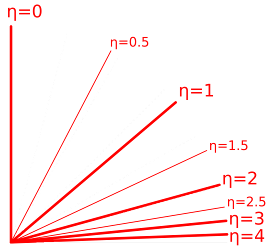


Figure 6: *Pseudo-rapid-*

1.3.1 *The Inner Detector*

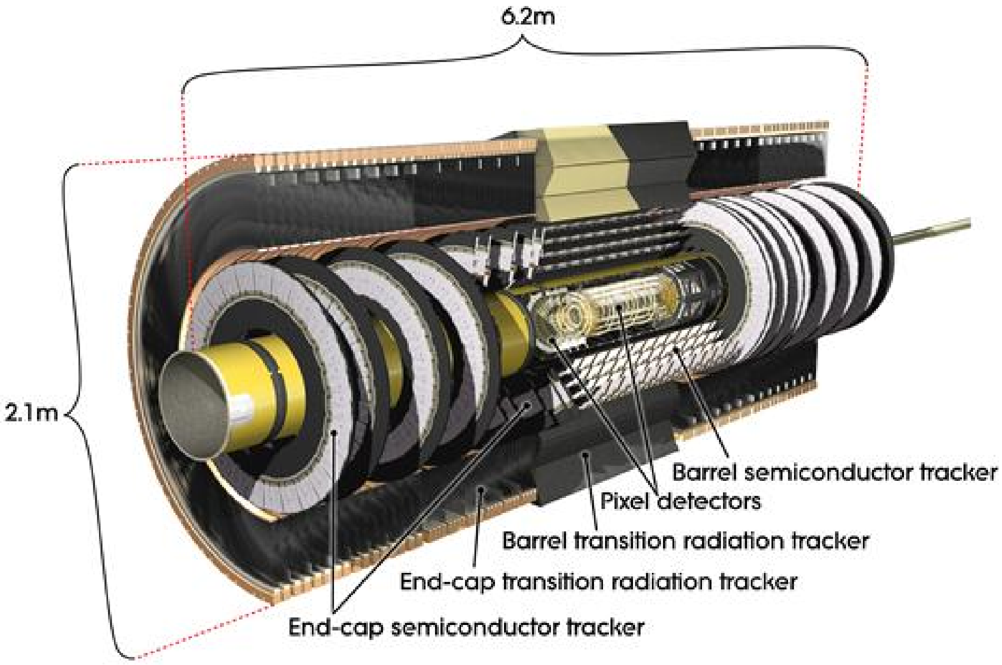

6.2m
2.1m
Barrel semiconductor tracker
Pixel detectors
Barrel transition radiation tracker
End-cap transition radiation tracker
End-cap semiconductor tracker

Figure 7: *Overview of the Inner Detector [9]*

The ID consists of 3 sub-detectors: The Pixel layers, the Semiconductor Tracker (SCT) and the Transition Radiation Tracker (TRT).
It provides pattern recognition and momentum measurements for charged tracks for a given p_T over a threshold of nominally 0.5 GeV. It also provides electron identification for particles with energy $<$ 150 GeV and $|\eta| <$ 2.0. It is contained within an solenoid and a magnetic field of 2T.
The high resolution pattern recognition is at inner radii achieved using discrete space points from silicon pixel layers and stereo silicon microstrip layers in the SCT. At larger radii the TRT

Detector	***Pixel***	***SCT***	***TRT***
Total number of channels	80×10^6	6×10^6	0.35×10^6
Total area or vol	$1.7m^2$	$60m^2$	$12m^3$
Resolution	$14 \times 115\mu m^2$	$17\mu m^2$ accuracy	$0.17mm$ precision

Table 1: *The values are for all modules of SCT, Pixel and TRT detectors in the ATLAS detector. Source [8]*

comprises of many layers of gaseous straw tubes. These tubes make it possible to have continous tracking of particles to enhance pattern recognition and improve the momentum resolution for $|\eta| < 2.0$ and electron identification complementary to the calorimeter over a wide range of energies. [9]
There are 4 pixel layers. The one closest to the beam-pipe is called the Insertable B-Layer (IBL). It was inserted in 2013/14 to improve the robustness and performance of the tracking system [13].

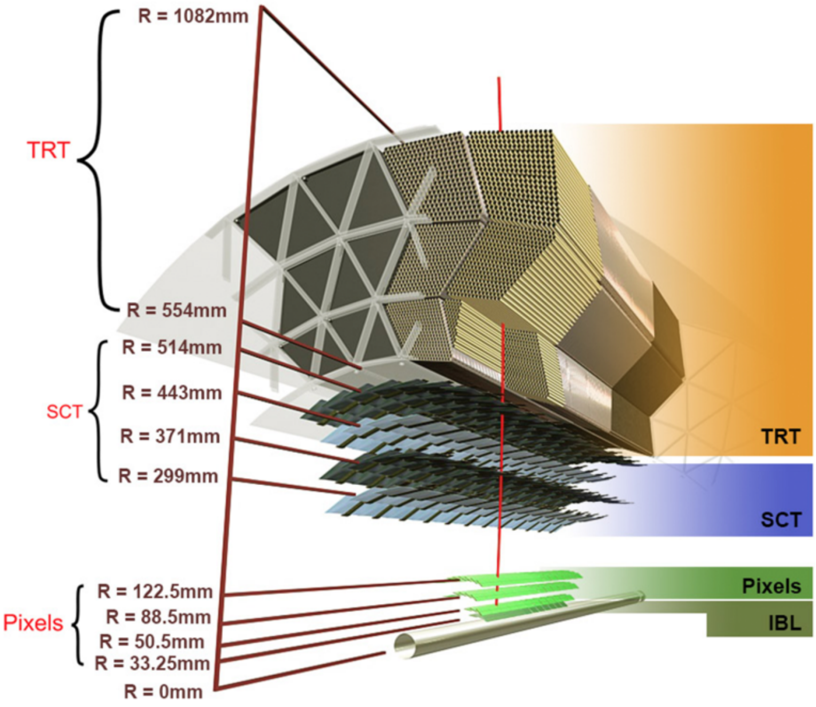


Figure 8: *Drawing showing the path a 10 GeV particle traverses in the barrel part of the ID[9]*

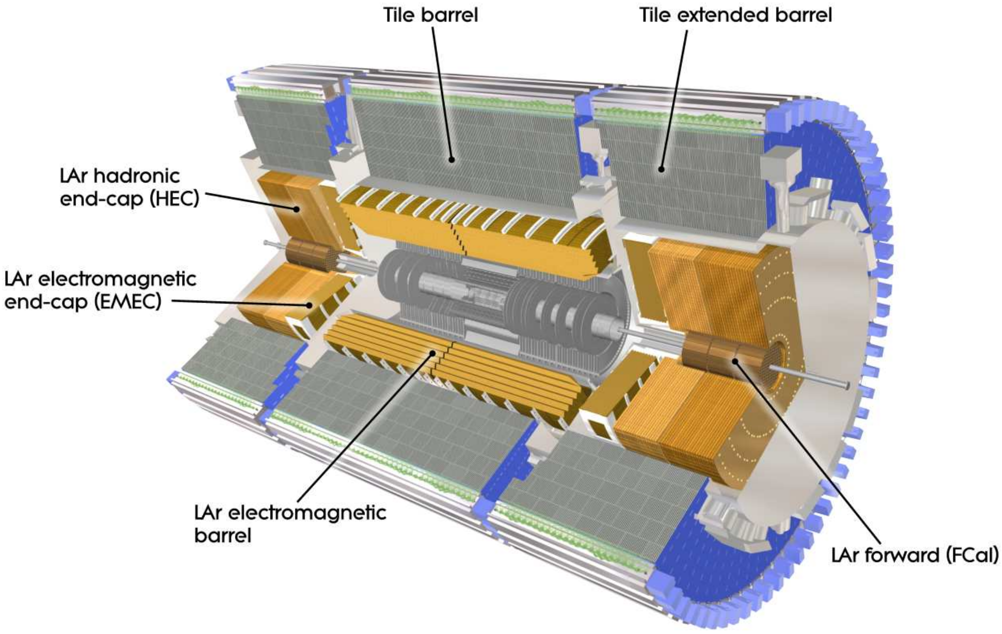

Tile barrel
Tile extended barrel
LAr hadronic end-cap (HEC)
LAr electromagnetic end-cap (EMEC)
LAr electromagnetic barrel
LAr forward (FCal)

Figure 9: *ATLAS calorimeters [9]*

1.3.2 *Calorimetry*

Calorimeters are blocks of instrumented material. When particles enter, they are fully absorbed and their energy transformed into a measurable quantity. The absorption results in the particle turning into a shower of secondary particles with progressively smaller energy. Sampling calorimeters consist of alternating layers of an absorber, a dense material used to stop the energy of the incident particle, and an active medium that provides the detectable signal.
Calorimeters have the advantage compared to magnetic spectrometers that the relative energy resolution improves with $1/\sqrt{E}$. It can also distinguish between different particles and provide fast signals that are easy to process and interpret for the trigger system. [12]

The Electromagnetic calorimeter
The ECAL consists of sampling calorimeters of lead and liquid argon (LAr). It is divided into a barrel part ($|\eta| < 1.475$) end two endcap components ($1.375| < |\eta| < 3.2$). The barrel part of the ECAL is divided at z=0 into two separate calorimeters 4 mm apart.

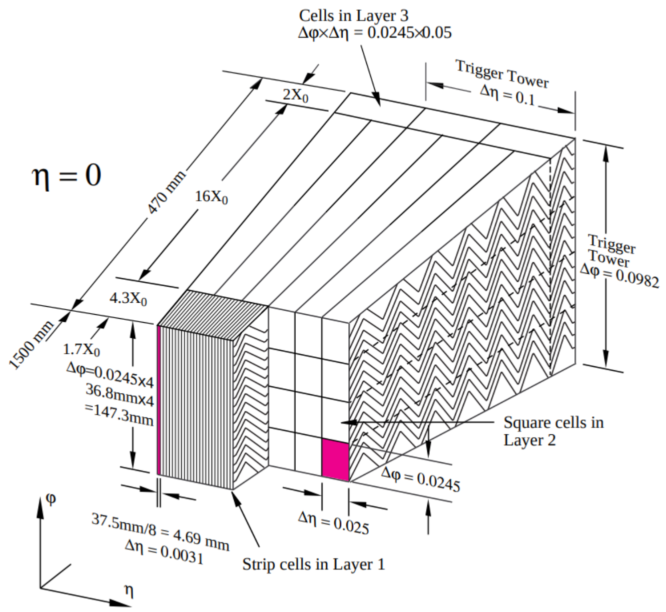


Figure 10: *ECAL barrel module*[9]

It leads to cracks when the lead and the LAr gap layers are perpendicular to the particle direction. This is avoided with accordion geometry of the absorber and the electrodes. See figure 10 [11]. The electrodes that collect the signal from the absorbers are placed as seen in Fig. 11. The ECAL is symmetric in ϕ. A module and its cells in the ECAL looks like in Fig 10 and it has 3 layers and a presampler. The presampler is a thin LAr layer with no absorbing material, whose purpose is to measure the amount of energy that is lost in the material before the calorimeter [23]. The first layer has a fine segmentation in η, the second layer is where most of the energy is deposited, and the third and last layer is to detect particles that isn't depleted in the ECAL but continues through it. The granularity and the η coverage of the modules in the ECAL can be seen in table 2. [9]

Each cell in the ECAL measure signal size and thus energy, and through these the cluster can be defined and the energy measured.

For electrons with energy higher than 10 MeV the main source of energy loss is bremstrahlung. When electrons and photons interact with material, they produce secondary particles which are either photons or electron and positron pairs from pairproduction. The new particles will again turn in to other particles with the same process and create a shower of particles with

progessively lower energy, until the energy reaches a threshold. [12]
π^0 particles are plentiful. They can decay to two photons and leave a trace in the ECAL and be a source of background. This is dealt with by noticing that π^0 doesn't leave a track and combined with the detection of two photons in the first layer of the ECAL, they can be identified as π^0.

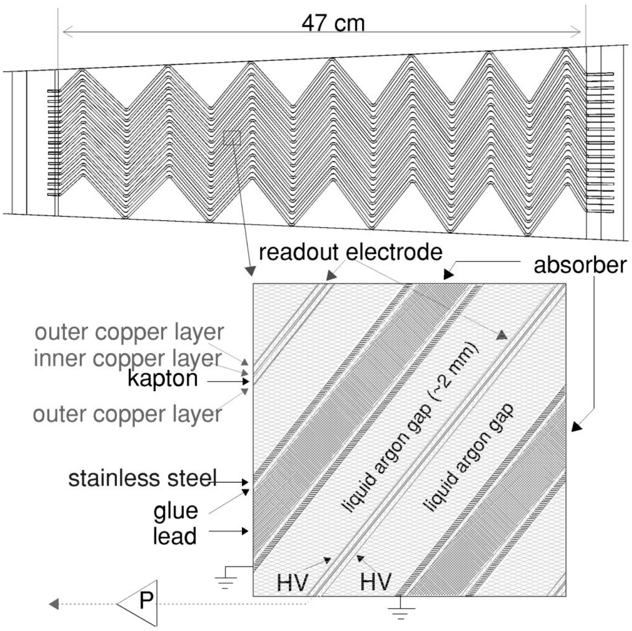


Figure 11: *Electrode structure of the ATLAS electromagnetic calorimeter [12]*

The Hadronic calorimeters (HCAL)
The most abundant particle species created in a hadron collision are pions. They will typically only deposit a fraction of their energy in the ECAL. What is needed, is much more material to stop the hadrons. This is some of the reasons for having the HCAL.[24] The HCAL are the tile calorimeter (TileCal), the LAr hadronic endcap calorimeter (HEC) and the LAr forward calorimeter (FCal).
The TileCal consists of scintillator tiles and the absorber medium is steel. The light reemmited by the scintillating medium is collected at the edge of each tile. The tilecal is placed directly outside the ECAL envelope. It's barrel covers the region $|\eta| < 1.0$, and its two extended barrels covers the region $0.8 < |\eta| < 1.7$. It is a sampling calorimeter with steel as its absorber and scintilating tiles as the active material.

Layer	**Granularity** $\Delta\eta \times \Delta\phi$	$\lvert\eta\rvert$ **coverage**
ECAL Barrel:		
Presampler	0.025×0.1	$\lvert\eta\rvert < 1.52$
Layer 1	$0.025/8 \times 0.1$	$\lvert\eta\rvert < 1.40$
	0.025×0.025	$1.40 < \lvert\eta\rvert < 1.475$
Layer 2	0.025×0.025	$\lvert\eta\rvert < 1.40$
	0.075×0.025	$1.40 < \lvert\eta\rvert < 1.475$
Layer 3	0.050×0.025	$\lvert\eta\rvert < 1.35$
ECAL Endcap:		
Presampler	0.025×0.1	$1.5 < \lvert\eta\rvert < 1.8$
Layer 1	0.050×0.1	$1.375 < \lvert\eta\rvert < 1.425$
	0.025×0.1	$1.425 < \lvert\eta\rvert < 1.5$
	$0.025/8 \times 0.1$	$1.5 < \lvert\eta\rvert < 1.8$
	$0.025/6 \times 0.1$	$1.8 < \lvert\eta\rvert < 2.0$
	$0.025/4 \times 0.1$	$2.0 < \lvert\eta\rvert < 2.4$
	0.025×0.1	$2.4 < \lvert\eta\rvert < 2.5$
	0.1×0.1	$2.5 < \lvert\eta\rvert < 3.2$
Layer 2	0.050×0.025	$1.375 < \lvert\eta\rvert < 1.425$
	0.025×0.025	$1.425 < \lvert\eta\rvert < 2.5$
	0.1×0.1	$2.5 < \lvert\eta\rvert < 3.2$
Layer 3	0.050×0.025	$1.5 < \lvert\eta\rvert < 2.5$

Table 2: η coverage and granularity of the ECAL layers

For electron particle identification purposes it can be useful that there should be little to no energy in the HCAL to exclude the possibility of the particle being an electron.

1.3.3 *Other Systems*

The following two systems are not so relevant for the work done on this project, and is therefore only mentioned shortly here.

The Muon System

A muon is the only charged particle that can penetrate all the calorimeter material. It only leaves a minimum of ionizing signal, due to its lack of strong interactions and relatively large mass.[24]

Muons leave a track in the ID and 1-3 GeV in the calorimeters and are subsequently detected again in the muon system.

Detection of muons is based on magnetic deflection of tracks in large superconducting air-core toroid magnets, instrumented with separate trigger and high-precision tracking chambers. [9]

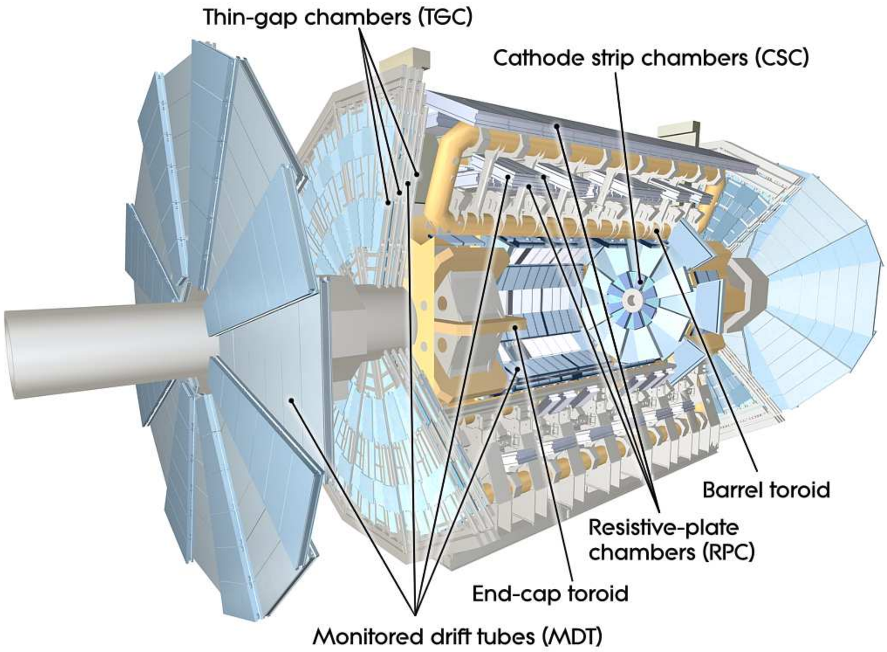

Thin-gap chambers (TGC)
Cathode strip chambers (CSC)
Barrel toroid
Resistive-plate chambers (RPC)
End-cap toroid
Monitored drift tubes (MDT)

Figure 12: *Overview of the Muon System in the ATLAS detector. Picture source [9]*

The Magnet system

The magnet system is essential to measure momentum and the tracks of particles. The main sections are the Central Solenoid Magnet, the Barrel Toroid and End-cap Toroids. The central solenoid encapsulates the ID and makes momentum measurement possible for charged particles that aren't muons. [9] The toroids are used for for muon detection.

1.3.4 *The trigger system*

The trigger system has tree levels: Level-1 (L1), Level-2 (L2) and the eventfilter. The eventfilter together with the L2 trigger is called the High Level Trigger (HLT) and is software based. L1 is based in custom made electronics and performs the first selection based on information from the calorimeters and the muon detectors and defines a region of interest (RoI) in η and ϕ coordinates. The proton-proton interaction rate is about 1 GHz (Average 25 protons in bunch with a 40 MHz bunch-collision rate) at the design luminosity of $10^{34}cm^{-2}s^{-1}$. After L1 the rate is reduced to 75 kHz.
The L2 makes further selection of events based on mainly the RoI from L1 and thereby reduces the rate of accepted events. The eventfilter considers the full event and uses offline procedures to make selections and reduces the event rate to 1000 Hz. [9]

1.4 ELECTRON RECONSTRUCTION

Incoming particles usually deposit their energy in many calorimeter cells. Clustering algorithms are designed to group these cells and to sum the total deposited energy within each cluster. These energies are then calibrated to account for the energy deposited outside the cluster and in dead material. The calibration depends on the incoming particle type.
Two types of clustering algorithms are used in the ATLAS detector:
A sliding window algorithm that sums up a cell in a fixed rectangular window. The position of the window is adjusted so that its contained transverse energy (E_T) is a local maximum.
The second is a topological cell clustering algorithm.
Three components are essential to reconstruct electrons in the precision region $|\eta| < 2.47$:

- Localized clusters of energy deposits in the ECAL
- Charged particle tracks in the ID
- close matching between the tracks and the clusters in $\eta \times \phi$ space.

Seed-cluster reconstruction
The $\eta \times \phi$ space in the ECAL is divided into a grid of 200×256 elements (called towers) of size $\Delta\eta \times \Delta\phi = 0.025 \times 0.025$ corresponding to the granularity of the second layer of the ECAL. The sliding-window algorithm with a window size of 3×7 cells in layer 2 is then used on the tower grid to search for seed-clusters. It seeds electron candidates if the summed E_T is higher than 2.5 GeV. [15]
The clustering algorithm follows cell signal-significance patterns from electromagnetic and hadronic showers, and removes cells that doesn't have a significant amount of signal. This reduces noise and results in a topological isolated cluster of cells that have position and energy information [20].

Track reconstruction
Charged particle reconstruction in the Pixel and SCT is based on hits in the ID tracking layers. From these hits clusters are being build. These clusters are being used to make 3 dimensional measurements referred to as spacepoints. In the Pixel detector one cluster is enough to make a spacepoint, but in the SCT clusters in both stereo views of a striplayer are required to make a spacepoint. Track seeds are formed from sets of three spacepoints in the silicon detector layers.
Track recognition then continues in three steps: Pattern recognition, algorithm for ambiguity resolution, and an TRT extension algorithm.
The pattern recognition uses the pion hypothesis for the model of energy-loss of the particle with the detector material.Track candidates with $p_T > 400 MeV$ are fit using the Global χ^2 trackfitter. If the fit fails, an electron hypothesis are used for a fit. A subsequent fitting procedure, Gaussian Sum Filter (GSF), is applied to clusters of raw measurements. [15]

Electron-candidate reconstruction
The final electron reconstruction procedure is the matching of the GSF-track candidate to the candidate calorimeter seed

cluster, and the determination of the final cluster size. Requirements to trackmatching in ϕ is $-0.10 < \Delta\phi < 0.05$ or $-0.10 < \Delta\phi_{res} < 0.05$, where $\Delta\phi$ and $\Delta\phi_{res}$ are calculated as $-q \times (\phi_{cluster} - \phi_{track})$, where q is the charge sign of the particle, and for $\Delta\phi_{res}$ the momentum of the track is rescaled to the energy of the cluster. If several tracks fulfill the criteria, the track chosen is selected by an algorithm. A candidate then with at least 4 hits in the silicon layers and no association with a vertex from a photon conversion is considered an electron candidate. [15]

Electron-isolation
At LHC experiments it is a challenge to differentiate between signal processes such as production of electrons, muons and photons from background processes like semileptonic decays of heavy quarks, hadrons misidentified as leptons and photons, and photons converting into electron-positron pairs. The signal processes are characteristic by little activity in the calorimeter and in the ID in an area of $\Delta\eta \times \Delta\phi$ surrounding the candidate object. Therefore variables are constructed to quantify the amount of activity in vicinity of the candidate referred to as isolation variables.

1.4.1 *The Atlas Likelihood*

ATLAS currently uses a likelihood (LH) based method to discriminate between reconstructed electron candidates (signal) and background (not signal). The likelihood method is based on a range of variables from the detector, such as calorimeter shower shapes, bremstrahlung effects, etc. An overview over the variables are shown in table 5. The LH uses the signal and background probability density functions (PDF) of the discriminating variables to calculate the probability of whether a particle is signal or background. The signal and background probabilities are then combined into an discriminant $d_{\mathcal{L}}$:

$$d_{\mathcal{L}} = \frac{\mathcal{L}_S}{\mathcal{L}_S + \mathcal{L}_B}, \quad \mathcal{L}_{S(B)}(\vec{x}) = \prod_{i=1}^{n} P_{s(b),i}(x_i)$$

where $\vec{x}$ is the vector of discriminating variable values and $P_{s,i}(x_i)$ is the value of the signal probability density function of the i_{th} variable evaluated at x_i.
Some variables are not based on PDF's but are based on rectangular cuts. Those are included in table 5.

Since the particle shower shapes in the detector depends on how much material the particle passes and how much energy the particle have, the likelihood is binned in η and E_T. The bin values for E_T are shown in table 3 and for η in table 4.
Three levels of operating points are provided for the ID likelihood algorithm. Loose, Medium and Tight. They express different levels of background rejection and the level increase from Loose to Medium to Tight. They are subsets of each other, meaning all electrons selected by Tight are also selected by Medium, and all electrons selected by Medium are also selected by Loose.

LHValue variable
In the data from ATLAS there is also a variable called LHValue. It is a Likelihood calculation only based on the pdf's in the ATLAS likelihood variables in table 5, not using the cuts.

Bin edges in E_T [GeV]												
7	10	15	20	25	30	35	40	45	50	60	80	150

Table 3: Note that for the high ET optimisation of the Tight operating point, the higher bins have been modified to be 80-125-200 GeV. Source: [16]

Bin edges in η																				
-2.47	-2.37	-2.01	-1.81	-1.52	-1.37	-1.15	-0.8	-0.6	-0.1	0	0.1	0.6	0.8	1.15	1.37	1.52	1.81	2.01	2.37	2.47

Table 4: Source: [16]

Type	Description	Name	Usage
Hadronic leakage	Ratio of ET in the first layer of the hadronic calorimeter to ET of the EM cluster Rhad1 (used over the range $\lvert\eta\rvert < 0.8$ or $\lvert\eta\rvert > 1.37$)	R_{had1}	LH
	Ratio of ET in the hadronic calorimeter to ET of the EM cluster (used over the range $0.8 < \lvert\eta\rvert < 1.37$)	R_{had}	LH
Back layer of EM calorimeter	Ratio of the energy in the back layer to the total energy in the EM accordion calorimeter. This variable is only used below 100 GeV because it is known to be inefficient at high energies.	f_3	LH
Middle layer of EM calorimeter	Lateral shower width, $\sqrt{(\Sigma E_i \eta_i^2)/(\Sigma E_i) - ((\Sigma E_i \eta_i)/(\Sigma E_i))^2}$, where E_i is the energy and η_i is the pseudorapidity of cell i and the sum is calculated within a window of 3×5 cells	$w_{\eta 2}$	LH
	Ratio of the energy in 3×3 cells over the energy in 3×7 cells centered at the electron cluster position	R_{ϕ}	LH
	Ratio of the energy in 3×7 cells over the energy in 7×7 cells centered at the electron cluster position	R_{η}	LH
Back layer of EM calorimeter	Shower width, $\sqrt{(\Sigma E_i (i - i_{max})^2)/(\Sigma E_i)}$, where i runs over all strips in a window of $\Delta\eta \times \Delta\phi \approx 0.0625 \times 0.2$, corresponding typically to 20 strips in η, and i_{max} is the index of the highest-energy strip	w_{stot}	CUT
	Ratio of the energy difference between the largest and second largest energy deposits in the cluster over the sum of these energies	E_{ratio}	LH
	Ratio of the energy in the strip layer to the total energy in the EM accordion calorimeter	f_1	LH
Track conditions	Number of hits in the innermost pixel layer; discriminates against photon conversions	η_{Blayer}	CUT
	Number of hits in the pixel detector	η_{pixel}	CUT
	Number of total hits in the pixel and SCT detectors	η_{Si}	CUT
	Transverse impact parameter with respect to the beam-line	d_0	LH
	Significance of transverse impact parameter defined as the ratio of do and its uncertainty	d_0/σ_{d_0}	LH
	Momentum lost by the track between the perigee and the last measurement point divided by the original momentum	$\Delta p/p$	LH
TRT	Likelihood probability based on transition radiation in the TRT	*eProbabilityHT*	LH
Track-cluster matching	$\Delta\eta$ between the cluster position in the strip layer and the extrapolated track	$\Delta\eta_1$	LH
	$\Delta\phi$ between the cluster position in the middle layer and the track extrapolated from the perigee	$\Delta\phi_2$	LH
	Defined as $\Delta\phi_2$, but the track momentum is rescaled to the cluster energy before extrapolating the track from the perigee to the middle layer of the calorimeter	$\Delta\phi_{res}$	LH
	Ratio of the cluster energy to the track momentum	E/p	CUT

Table 5: Variables used for the ATLAS likelihood. Source: [16]

1.5 MACHINE LEARNING

Machine learning (ML) is a statistical optimization method where the 'machine' or the model for data manipulation seeks to keep on updating itself and guess the right answer to a question and 'learn' from previous guesses and thereby becoming better and improve it-self. The question can be a binary yes or no question, is this an electron or not? It can also be multi classification, is this an electron a muon or a photon? Or it can be a value question, how much is the energy of this

electron? The latter is called regression. In this project I will only focus on binary classification.
Neural networks are a very hot subject due to their versatility, flexibility and success with regards to images. But they are slow to train and decision trees offer high performance, faster training and have already proved to be efficient with ATLAS data. Therefore decision trees are chosen for this project.

1.5.1 *ML Datasets*

The first thing you encounter in ML is the dataset and it consists of different variables (features in ML lingo). The variable values represent datapoints and can have a label for each point. This is called supervised learning. Some ML models can try to classify without labels, this is called unsupervised learning and will not be used here.
Data is separated into training, validation and test data. This is done to make the model more robust, so it doesn't predict on data that it already has used to learn from.
Validation data is used in the training process to evaluate when training is satisfied. If the predictions on the validation data doesn't improve for a chosen number of rounds, training is terminated and the model that has performed the best is chosen. This is called early stopping. Test data represents new unseen data and is used to evaluate the final model.

1.5.2 *Decision trees*

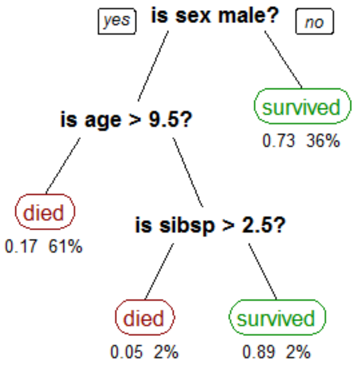


Figure 13: *Example of decision tree. Source: [30]*

A decision tree looks graphically like figure 13. To explain what is going on, some ML terminology is required. The questions (conditions) in bold in figure 13 are called nodes. After each node the tree splits into branches. If the node doesn't split into new branches, the node is called a leaf.

1.5.3 *Hyperparameters*

How to build the tree is decided by its parameters. For a ML algorithm they are called hyperparameters (HPs). Examples of HPs are:

- Number of leaves: The max number of leaves in a tree. This hyper parameter is important for avoiding overfitting.
- Learning rate: This constant is multiplied the calculated loss and thereby decides how big steps the algorith takes to correct its errors. Low learning rate makes a slower and more thorough 'learning', but also makes the tree prone to overfitting.
- Max depth: Determines the maximum depth of a tree. Low values can lead to less overfitting.
- Feature fraction: Randomly select a subset of features on each iteration (tree)
- Max bin: Max number of bins that feature values will be bucketed in.
- Minimum of data in a leaf: Minimal number of data in one leaf.
- Bagging fraction: Divide the data into subsets and build a tree for each subset of data and use the average of all predictions for each tree.

Boosting type can be gradient boosting or random forest [22]. In this project I worked with gradient boosting.
To find the optimal value for HPs usually a search is done using an algorithm which trains on the different values.

1.5.4 *Gradient boosting*

Boosting is a technique where an initial tree will be build to predict a result. Then a prediction on a dataset will be done. Each prediction of a datapoint will be given a weight. If the prediction is correct the weight will be given a small or no change to the weight, and a if the prediction is wrong the change will be large. A new tree will be built based on the new weights and different cuts will be made at each node. The method used to update the weights is called Gradient Descent. It is based on the gradient of the error function. It takes a step for every tree calculated and the hope is to find a minimum so the error in the multidimensional space is minimal. [21] [22]

1.5.5 *Loss function*

An essential part of ML models is the loss function. It is used to calculate the error or the loss. The guess that the ML model produce for each datapoint is between 0 and 1. The loss function used in this project is called cross-entropy:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$$

y_i is the label, p_i is the ML prediction and N is the number of datapoints. It punishes wrong and rewards precise classification.

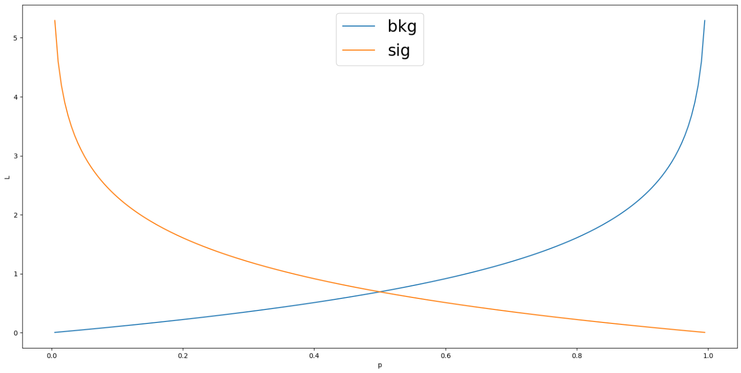


Figure 14: *Cross entropy of possible ML values to signal (y=1) and background (y=0)*

1.5.6 *Predictions*

At the end of the training, many decision trees will be built. To evaluate the ML model a new data set will be used for testing. Each point will be evaluated and be given a score (referred to in this project as *predictions*) between 0 and 1. Typically and in this project the closer to 1, the higher the chance of being signal.

1.5.7 *Evaluation (ROC and AUC)*

Evaluation of how good a ML model is can be found using the AUC score. The evaluation consists of choosing a prediction score threshold value (called cutting) and see how much signal and background is above the threshold and below given the labels of the predictions. True positives (TP) are then all scores above the threshold correctly labeled as signal. False positives (FP) are all scores above the threshold labeled as background. True negatives (TN) are below and labeled as background and false negatives (FN) are below and labeled as signal. The true positive rate (TPR) also called signal efficiency and the true negative rate (TNR) also called background efficiency are calculated as follows

$$TPR = \frac{TP}{TP + FN} \quad \text{and} \quad TNR = \frac{TN}{TN + FP}$$

The ROC (Receiver Operator Curve) is a 2 dimensinal plot of TNR on the x-axis and TPR on the y-axis for many different cuts in the 0 to 1 range of the predictions. Each cut is a different point on the curve. A single value for the ability of the ML model to separate signal and background is the AUC (Area Under the Curve). This is the integrated area under the ROC curve. The closer to 1 the better the separation between signal and background.

1.5.8 *LightGBM*

LightGBM is the algorithm for the gradient boosted decision trees that is used in this project. It differs from many other algorithms by having leafwise growth in stead of level-wise (see figure 15a and 15b)

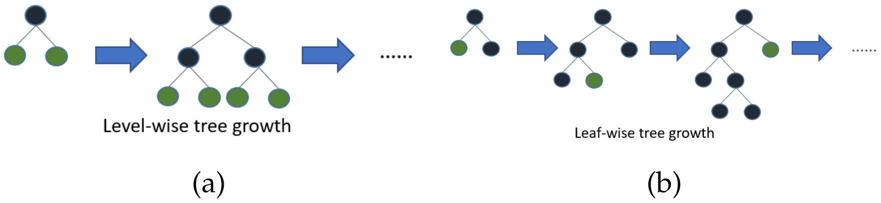


Figure 15: LighGBM compared to other tree algorithms

2

TRAINING AND RESULTS IN SIMULATED DATA

2.1 INTRODUCTION

In real data (RD) from collisions detected in the ATLAS detector, there is no labels. This makes for a more challenging case (see chap. 3). But in data generated by Monte Carlo-simulation (MC) "life is easy", meaning it is simulated data and everything is known. All particles and where they have decayed from are known with certainty. Therefore labels were made with the demands that it had to be an electron and it had originated from a W according to the ATLAS variables *particleType* and *particleOrigin* [17].

2.2 SELECTION OF DATA

To test a ML algorithm you need data, and the more the better. The W is the largest source of isolated electrons due to its larger cross-section, $\sigma_W \approx 3\sigma_Z$. Also the branching ratio of $W \rightarrow e\nu$ is 10%, compared to $Z \rightarrow ee$ that only has a branching ratio of 3%. The problem in earlier theses from this group, that used $Z \rightarrow ee$, was lack of data for electrons with high energy. Therefore samples with W was attempted.

2.2.1 *Train, validation and test sets*

In ML it is important to keep the training data and the testing data separate and the data was separated into 3 different sets: training (80%), validation (10%) and testing (10%). The validation-set is used during hyperparameter optimization and training. During training 'early stopping' is used to make sure that the loss function keeps improving in both training data and validation data. When the loss diverge, training is stopped. Then the trained model is applied to testdata for evaluation.

2.3 MC DATA SELECTION

The MC data is generated by ATLAS's EGamma group and further filtered to fit this project.

2.3.1 *MC Signal*

The Signal comes from EGAM5 files, which contain electrons from W particles via the process $W \rightarrow ev$ (W to electron and positron pair and a neutrino). The selections to Egam 5 files are (Source [19]):

1. a trigger-based selection: OR of a long list of EGamma triggers. Link to triggers are in the appendix
2. An offline-based selection:
   - $MET\_LocHadTopo > 25$ GeV
   - A central electron, tight or LLHtight, $p_T > 24.5 GeV$
   - $mT > 40$ GeV
3. A mixed trigger (see appendix)+offline selection:
   - one central electron with $p_T > 14.5$ GeV

2.3.2 *MC Background*

Background comes from a JET-sample from EGAM7 derivations. EGAM7 file derivations are specifically to generate fake electrons. They are described as [19]

- An event that passes at least one HLT e/gamma supporting trigger (e.g. etcut or lhvloose trigger) and there is at least one central reconstructed electron with $p_T > 4.5 GeV$

2.3.3 *Energy distribution of data*

The number of datapoints as a function of energy can be seen in table 6

GeV	# Bkg	# Sig
80<E_T <100	3.661.171	41.284
100<E_T <150	5.492.722	24.824
150<E_T <200	2.229.102	4.564
200<E_T	953.462	2.023
Total	12.336.457	72.695

Table 6: Signal and background in MC as a function of E_T.

2.4 VARIABLES

2.5 ISO VS PID

The reason for choosing isolation (ISO) and particle identification (PID) variables is to make labels in RD. Training in RD works if the labels for signal and background are unbiased. Therefore two different sets of variables are used for label-making and training. An ML-model trained on ISO in MC for labels, and an ML-model on PID variables for training and evaluation and vice versa.

2.5.1 *Isolation*

The isolation variables used to train the MC model are listed in table 7, and their distributions are shown in figure 17, and briefly described below.
The cone-variables are all based on the sum of energy in a cone around the particle in the (η, ϕ) plane. Fx. $cone20$ represents a cone with radius $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} < 0.20$.
When collissions take place, the protons within the two beams in the ATLAS detector are grouped in bunches. An event is the data resulting from a particular bunch-crossing.
Pile-up is defined as the average number of particle interactions per bunch-crossing. This is represented by the variable $averageInteractionsPerCrossing$, also abbrevated $\langle\mu\rangle$.
A primary vertex comes from the collision of two protons. In a typical collission event, several primary vertices are produced. The number of reconstructed vertices are represented by the variable $Nvtxreco$.
et is E_T of the reconstructed particle.

$ptcone20$	Sum of E_T or p_t in a cone around the particle
$ptcone30$	
$ptcone40$	
$etcone20$	
$etcone30$	
$etcone40$	
$NvtxReco$	The number of reconstructed vertices
et	E_T in the ECAL
$averageInteractionsPerCrossing$	Average number of particle interactions per bunch-crossing

Table 7: *Description of Isolation variables*

2.5.2 *PID*

The variable descriptions for PID are summarized in table 8.
The variable distributions are in figure 8

$Rhad$	LH. See table 5
$Reta$	LH. See table 5
$Rhad1$	LH. See table 5
$deltaEta1$	LH. See table 5
$Eratio$	LH. See table 5
$Rphi$	LH. See table 5
$weta2$	LH. See table 5
$dPOverP$	LH. See table 5
$f3$	LH. See table 5
$d0$	LH. See table 5
$f1$	LH. See table 5
$NvtxReco$	Number of reconstructed vertices for the event
$deltaPhiRescaled2$	Difference in ϕ between track and cluster
$TRTPID$	Likelihood probability based on transition radiation in the TRT
et	The energy as reconstructed by the calorimeter
eta	The pseudo rapidity of the particle
$wtots1$	Total width in em sampling 1 in 20 strips
$numberOfSCTHits$	Number of hits in the SCT
$numberOfInnermostPixelHits$	Number of hits in the inner most pixel layer
$EptRatio$	E_T/P_T
$numberOfPixelHits$	Number of hits in the pixel layers
$nTracks$	The number of tracks associated with the electron cluster
$E7x11Lr3$	Energy deposited in 7×11 cells centered around the cluster in layer 3
$core57cellsEnergyCorrection$	An energy correction, calculated from the 5×7 cells in the calorimeter, centered around the cluster

Table 8: *PID variables*

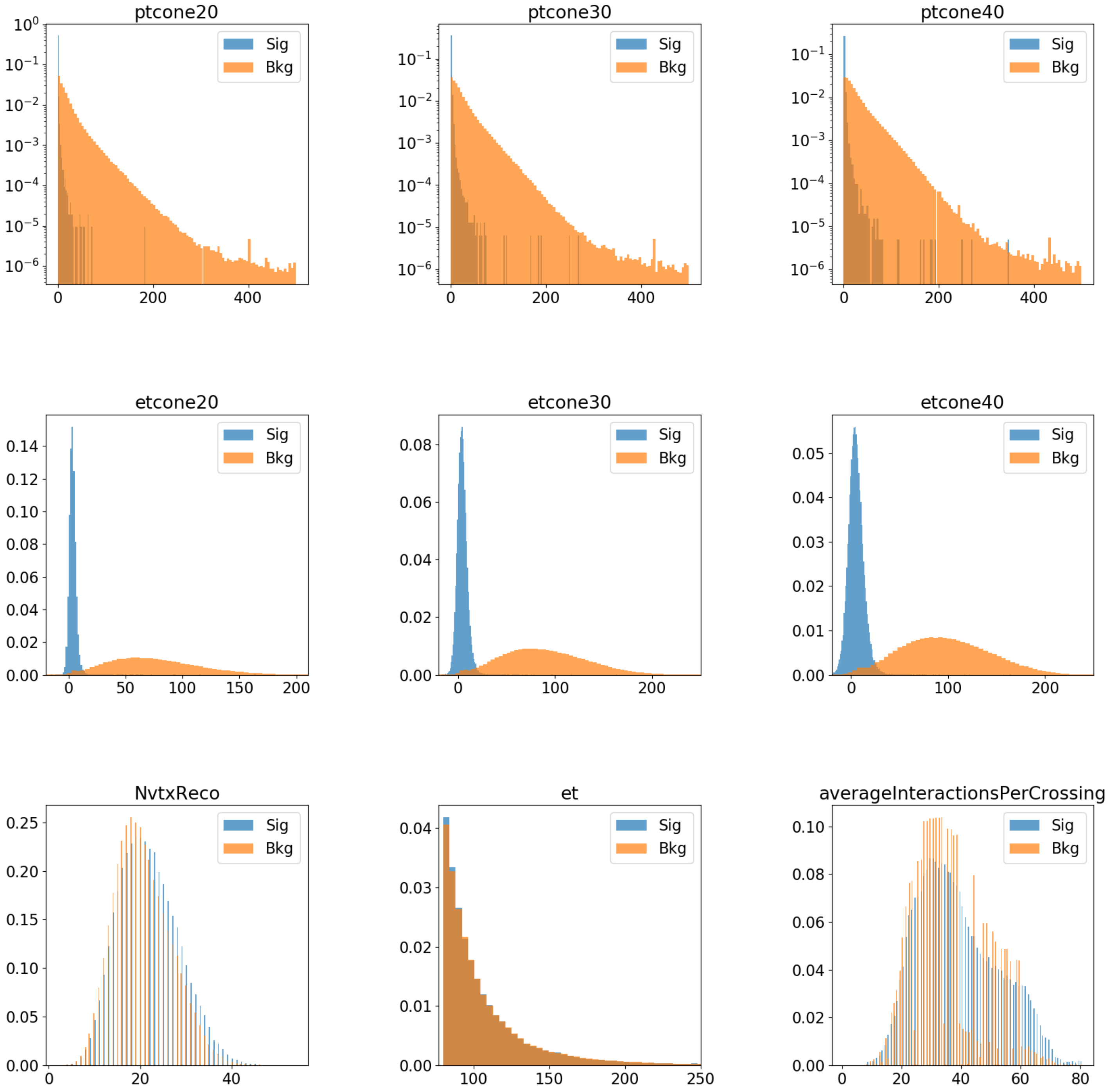


Figure 16: *The distributions of the isolation variables*

Figure 17

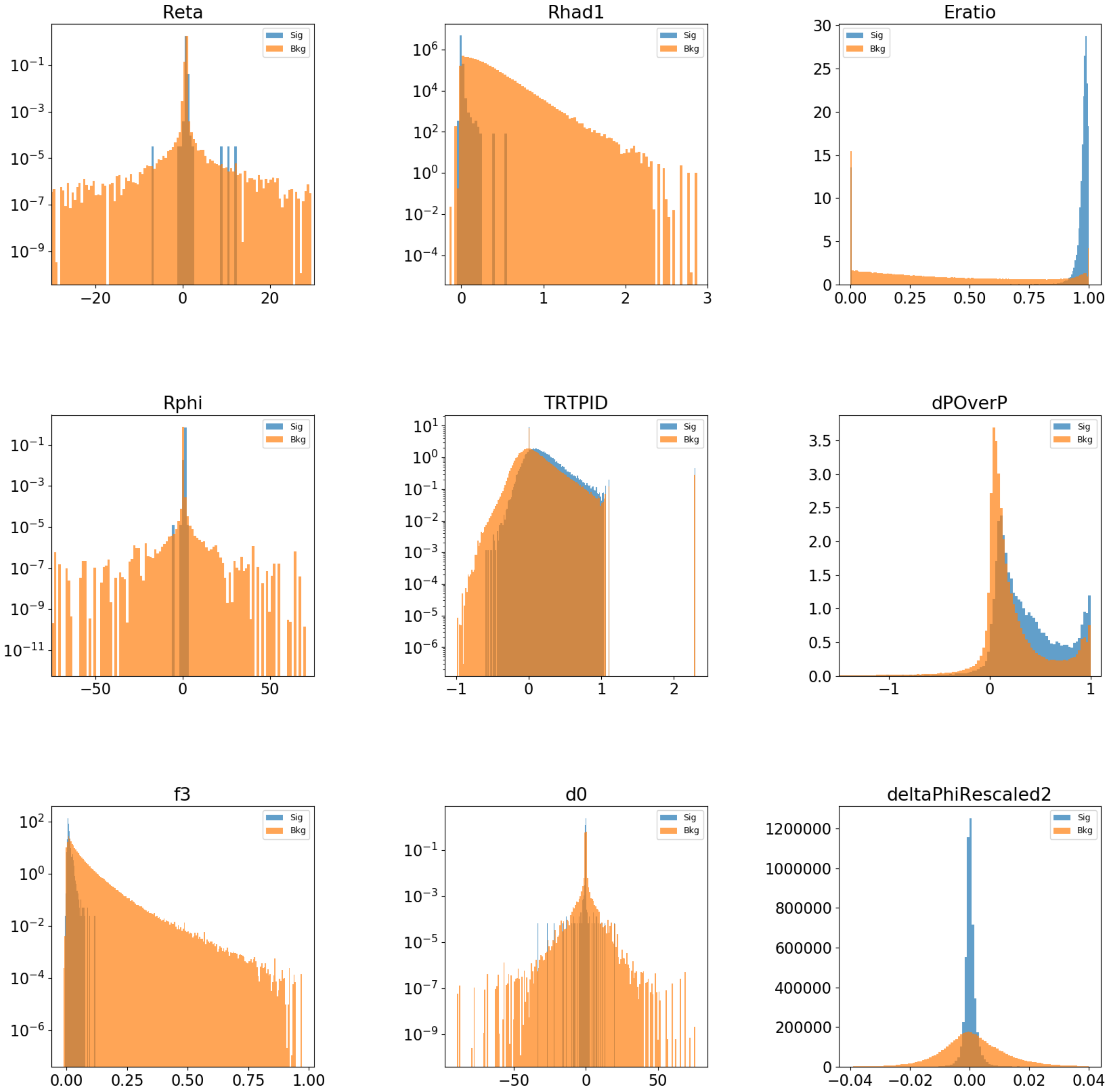


Figure 18: *The distributions of the PID variables, with weights from reweighing. The rest of the 24 variables can be found in the Appendix*

2.6 REWEIGHING

$\langle\mu\rangle$, E_T and η are all important variables for electron identification. But when training models on MC and then using them to predict on data, some modifications are necessary. Reweighing background to have the same distributions in signal in these three variables are done to mitigate the importance of these variables in the decision making of the decision tree.

$\langle\mu\rangle$can randomly vary for different periods of time. This is not optimal for ML algorithms to train on MC and predict on RD, because the variable can have random changes which affects the prediction.
E_T is an important variable, but it is not good if the signal is distributed so that some energy ranges are dominant compared to others. The whole purpose of this thesis was from the beginning to test the ML algorithms on different energy ranges.
η is the angle along the beam. There are parts of the detector where the signal is weak, due to fx. physical placements of the elements in the detector. This can affect the signal/background distribution in the η variable. In MC, where the particle is known, this is not a problem, but it affects data.
Of the three variables E_T is the most important. The others are minor corrections. Reweighing is done using a python package called GBReweighter (Gradient Boost Reweighter), which uses a decision tree to calculate the weights. Plot of the reweighing variables is shown in 19.
The optimal value of the hyperparameters for GBRewighter are in table 9. These were optimal for both MC and RD.

Name	**Value**	**Description**
n_estimators	300	*Number of trees*
learning_rate	0.1	*Learning rate*
max_depth	20	*Maximal depth of trees*
max_leaf_nodes	60	*Max number of nodes that ends with a leaf*
min_samples_leaf	100	*Minimal number of events in the leaf.*

Table 9: Hyper parameter values for GBRewighter

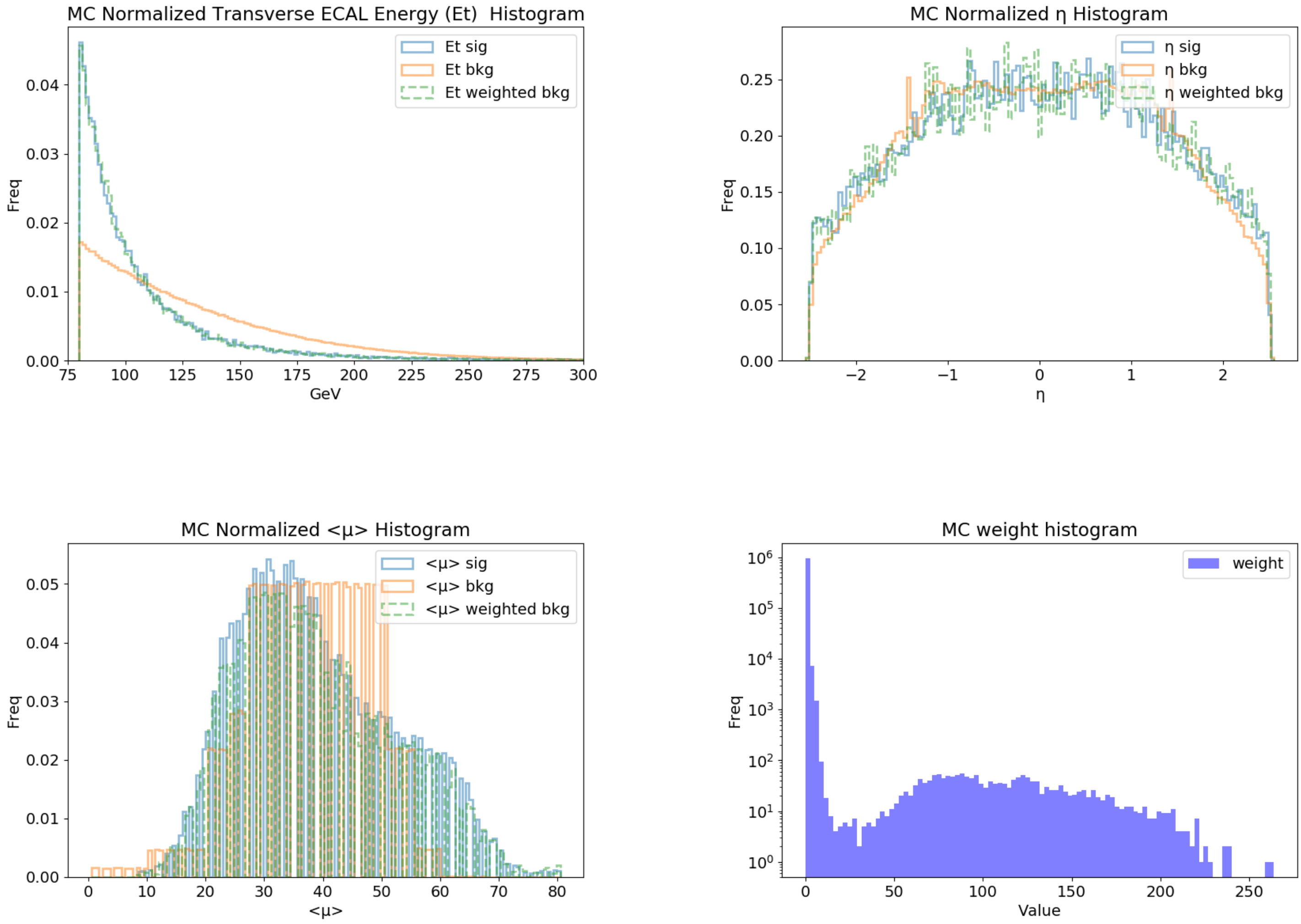


Figure 19: *Histograms of signal, background and weighted background of $\langle\mu\rangle$, E_T and η. The weight distribution is in the bottom right*

2.7 TRAINING

Two models were trained: One model based on isolation variables MC(ISO)$_W$ and a second model trained on PID variables MC(PID)$_W$. The W in the subscript refers to it is trained on data from $W \to e\nu$.

	Tuned LGBM HPs
MC(ISO)$_W$	learning_rate: 0.10 num_leaves : 5
MC(PID)$_W$	learning_rate: 0.05 num_leaves: 9

Table 10: *Overview of the different ML-models trained in MC and used on RD. The column "Tuned LGBM HPs" are the LightGBM hyper parameters that wasn't the default, but found through hyper parameter tuning.*

2.7.1 *Hyper parameter optimization*

The gradient boosted decision trees (GBDT) in LightGBM can take many parameters and the value of these can have significant importance for the performance of a decision tree. The list of parameters LightGBM use and the default values is listed in the appendix in table 17. A grid search of parameters can be slow. Therefore to find the optimal values a gaussian process approach was used.
The models hyper parameters, that wasn't set as default, can be seen in table 10. Especially *num_leaves* was important for avoiding overfitting.

2.7.2 *Predictions*

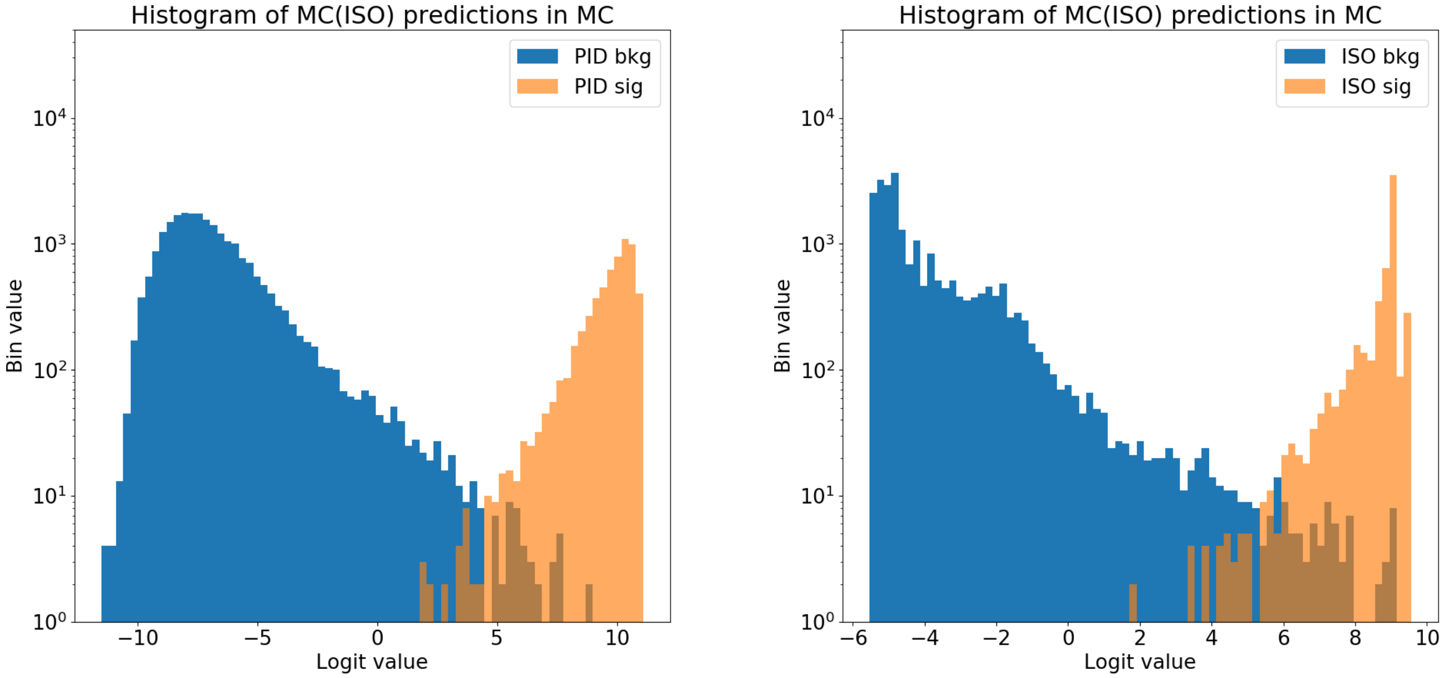
The predictions for MC(ISO)$_W$ and MC(PID)$_W$ is in picture 20a and picture 20b. Predictions from LightGbM are between 0 and 1 but are transformed to logits through the formula:

$$logits = \log\left(\left(\frac{1}{x}\right) - 1\right) \tag{1}$$

Advantages of logit transforming the scores are:

- More gaussian distributions, which makes correlation more obvious in plots
- Avoids numerical problems close to 0 and 1

Looking at the predictions, the MC(PID)$_W$ model has more round peaks. Both MC(PID)$_W$ and MC(ISO)$_W$ separate background from signal very well.



(a)

(b) 1b

2.8 ROC CURVE

From the predictions a calculation of a ROC curve is done in figure 21. It shows that MC(PID)$_W$ beats MC(ISO)$_W$ and the LH-points by a significant margin. MC(ISO)$_W$ also beats the Loose working point, but is still not as good as the other LH-points and MC(PID)$_W$, but with a low AUC it is still quite good at separating signal and background.

2.8.1 *Correlations*

If MC(PID)$_W$ and MC(ISO)$_W$-predictions are correlated in MC$_{bkg}(ISO)_W$ vs. MC$_{bkg}(PID)_W$ and MC$_{sig}(ISO)_W$ vs. MC$_{bkg}(PID)_W$ (bkg: background, sig: signal) they are not optimal to predict labels with in RD. If labels aren't independent to a large degree with the training model, the features in new data will be biased by the correlated sections.

There are several ways to measure correlations. Pearsons linear correlation coefficient is one way. Another is a relative new method called distance correlation, which was introduced in

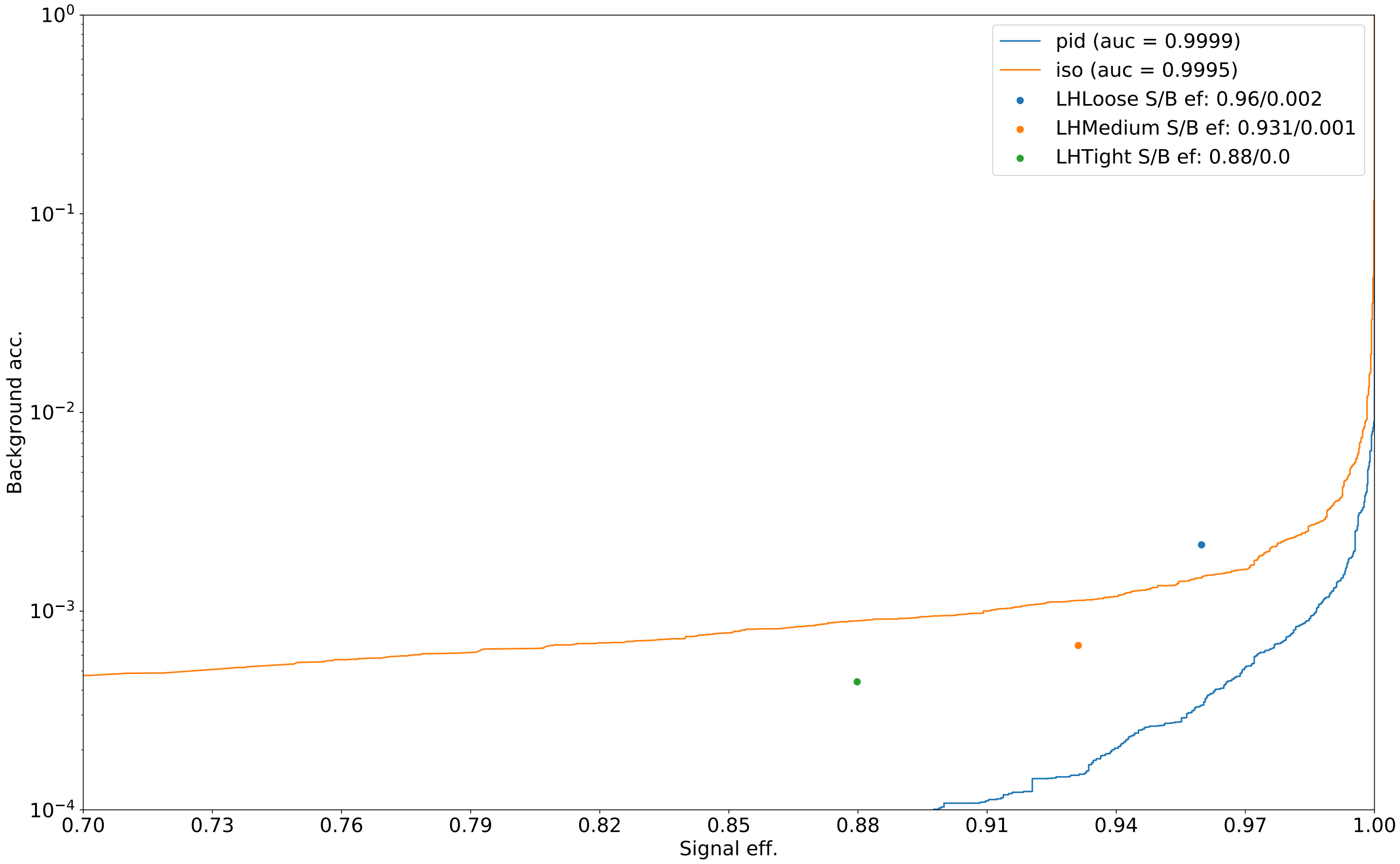


Figure 21: *ROC curve of ATLAS Likelihood value, pid (MC(PID)$_W$), iso (MC(ISO)$_W$), and the Tight, Medium and Loose ATLAS likelihood points*

2005 by Gábor J. Székely [18]. An advantage with distance correlation compared to Pearson is, it can measure non-linear correlations.
The signal and background correlation between MC(ISO)$_W$ and MC(PID)$_W$ is shown in table 11.
The 2D histogram of the MC(PID)$_W$- and MC(ISO)$_W$ predictions are shown in figure 22. The plot shows good separation between signal and background for both MC(PID)$_W$ and

	Background	Signal
Pearson corr. koeff.	0.42	0.05
Distance corr. koeff.	0.46	0.15

Table 11: Correlation coefficients between predictions by MC(PID)$_W$ and MC(ISO)$_W$

$\mathrm{MC(ISO)}_W$. With this good separation, decorrelation might not be necessary.

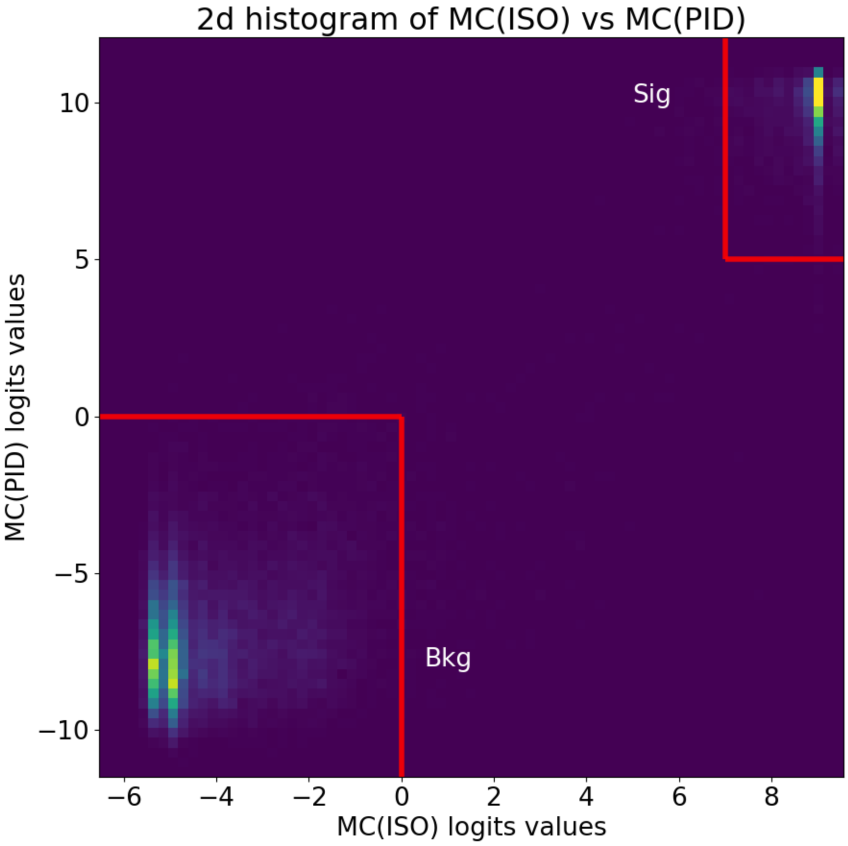


Figure 22: *2d histogram of logit-transformed predictions from* $\mathit{MC(PID)}_W$ *and* $\mathit{MC(ISO)}_W$

3

REAL DATA

3.1 INTRODUCTION

With ML-models trained in MC it is time to use them to make labels in RD.
Two sets of RD was used: Data from $W \rightarrow e\nu$ and $Z \rightarrow ee$ decay. For W-data, two ML-models where produced. One trained on PID-variables and one based on ISO variables. They are given the subscript W for W-data: RD(PID)_W, RD(ISO)_W. For Z-data only PID was trained on (RD(PID)_Z). Data selection in RD was meant to be based on data from $W \rightarrow e\nu$ decay only, because it wasn't possible to find other data. But late in the process, RD from $Z \rightarrow ee$ was discovered. Because of time constraints Z-data (Data from $Z \rightarrow ee$ decay) is only used to train a model and compare it to models trained in W-data (Data from $W \rightarrow e\nu$ decay) and evaluate based on Z-mass selection. And because only W-data was available in MC, labels were made with ML-models from W-data.

3.2 DATA SELECTION

3.2.1 *W data*

Like in MC the data is preselected by the ATLAS EGamma group and more specifically from Egam5 selections. The selections are therefore exactly the same as in section 2.3.1.

The total number of datapoints are 239.438. The distribution of data in tranverse energy can be seen in table 12

GeV	# Bkg	# Sig
80< E_T <100	89.922	20.248
100< E_T <150	76.315	13.352
150< E_T <200	21.263	2.482
200< E_T	14.911	945
Total	202.411	37.027

Table 12: Signal and background in W RD as a function of E_T. Predictions by **MC(PID)**$_W$

GeV	# Bkg	# Sig
80< E_T <100	85.393	24.777
100< E_T <150	73.341	16.326
150< E_T <200	20.638	3.107
200< E_T	14.526	1.330
Total	193.898	45.540

Table 13: Signal and background in W RD as a function of E_T. Predictions by **MC(ISO)**$_W$

3.2.2 *Z data*

The point of getting Z-data was to evaluate ML-models through the Z-boson mass peak at 91 GeV.

The number of datapoints are: 2.511.148.

The data is from Egam1 selections. Electrons in Egam1 datasets are chosen based on one of the following criteria:

- 2 electrons, one tight or LH tight,$pT > 24.5$ GeV, one medium or LH medium, $pT > 19.5$ GeV
- medium or LH medium, $pT > 19.5$ GeV
- one medium or LH medium, $pT > 24.5$ GeV, one with $pT > 6.5$ GeV
- one electron, medium or LH medium, $pT > 24.5$ GeV, and one photon with $ET > 14.5$ GeV

and that the invariant mass of at least one pair must be greater than 50 GeV. Like with other data only electrons with 80 GeV are chosen. Further selections was done the same way as with data from $W \to e\nu$.

3.3 MAKING LABELS IN RD

3.3.1 *Problems with Tag-and-Probe for W data*

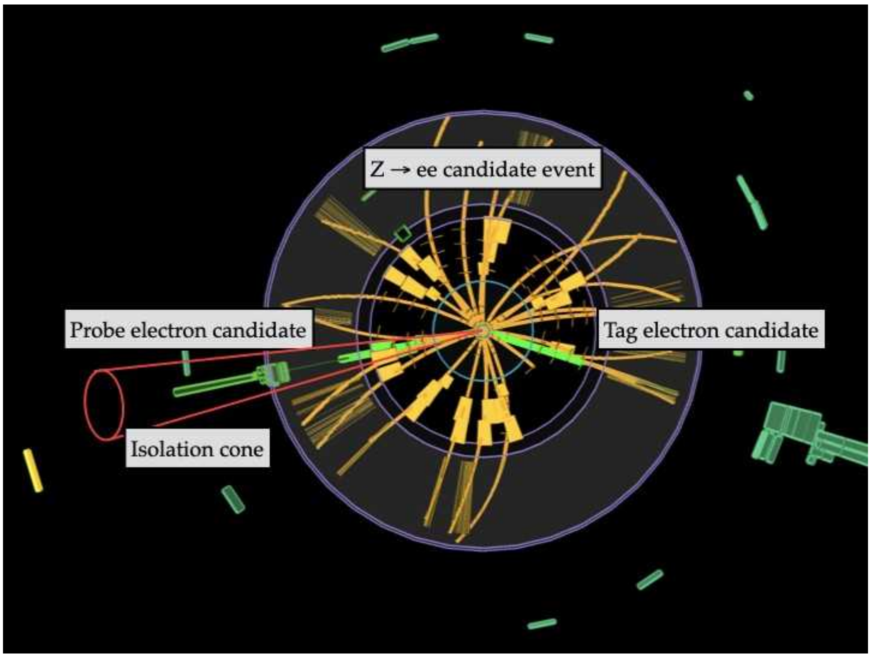


Figure 23: *Picture of an event with illustration of Tag and Probe. Here a Tag is an electron candidate selected with high certainty. If there is another candidate which is isolated it is called a Probe. If the Tag and Probe system has an invariant mass close to the Z-mass, the probe is likely an electron. For $W \rightarrow e\nu$ it was tried to have the tag as the E_T^{miss} and the probe as the electron and then compare it to the transverse W-mass, but with not accurate results. The advantage of Tag and Probe is that it is a relatively unbiased method of selecting electron labels for machine learning.*

Tag-and-Probe is a method to select electrons in RD from the LHC. If two electron candidates in an event (collision) have the combined energy of the Z-mass or W-mass it is likely an electron for $Z \rightarrow ee$ or $W \rightarrow e\nu$. For each event in the ATLAS detector a search will be done for an electron candidate in that event, which is called a Tag. If there is a Tag-electron, a search for another electron with another set of requirements is done, and if the search is successful that electron will be called a Probe.
Tag-and-Probe has proved efficient in Z-data. It has not been proven efficient with $W \rightarrow e\nu$ data, but was tried in this project with $W \rightarrow e\nu$.

The method was to find an electron and positron in one event, and require one of them to be a Tag. For another electron in the event to be a Probe the total transverse energy of the Tag and the Probe and the E_T^{miss} (missing transverse energy of an event) had to be close to the transverse mass of the W. Unfortunately the mass distribution did not show a satisfying peak in the W mass, and therefore the Tag-and-Probe method was abandoned. The reason for this could be low efficiency of the E_T^{miss} trigger in ATLAS (see figure 24). The E_T^{miss} trigger reaches an efficiency of 88% at 150 GeV and most data was below 150 GeV. Another reason could be that the trigger is executed by other events than E_T^{miss}.
Tag-and-Probe wasn't used with Z-data, because for high energy electrons above 80 GeV from $Z \to ee$ are very scarce.

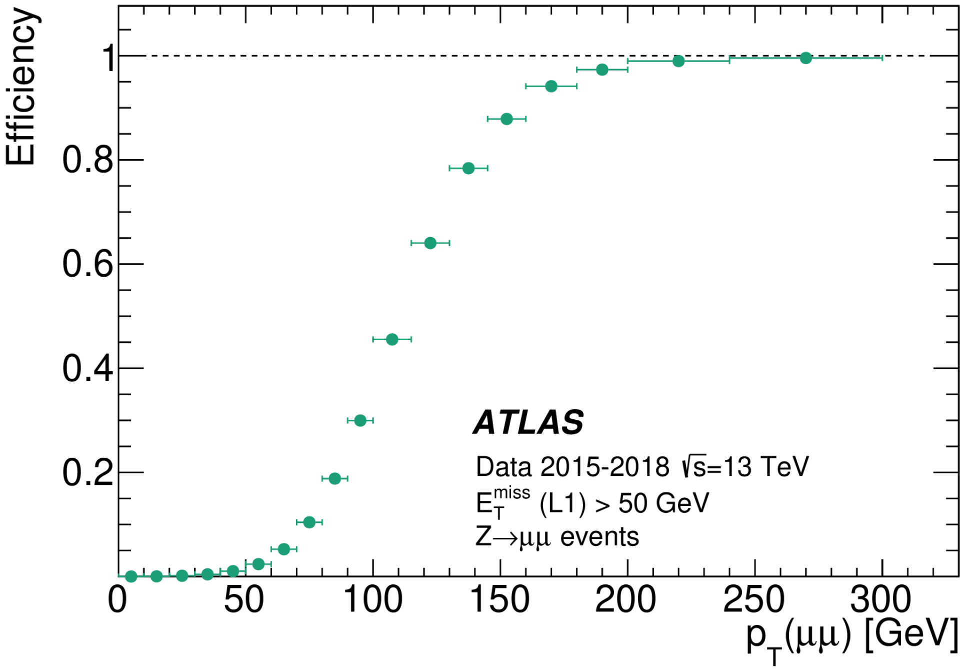


Figure 24: *The ATLAS E_T^{miss} trigger efficiency as a function of energy.*

3.4 initial labels in w - data

Because labels with Tag-and-Probe wasn't possible, initial labels in RD was made with predictions from ML-models trained on MC. For training on PID variables, MC(ISO)_W was used to make labels and for training on ISO-variables in RD, MC(PID)_W was used to predict labels. This was done because is is important the training model has no or little preexisting information about the data it is training on. Otherwise it will learn less about the new data and become a weaker learner. The initial predictions of MC(ISO)$_W$ on RD can be seen in figure 25 and for MC(PID)$_W$ in figure 26.

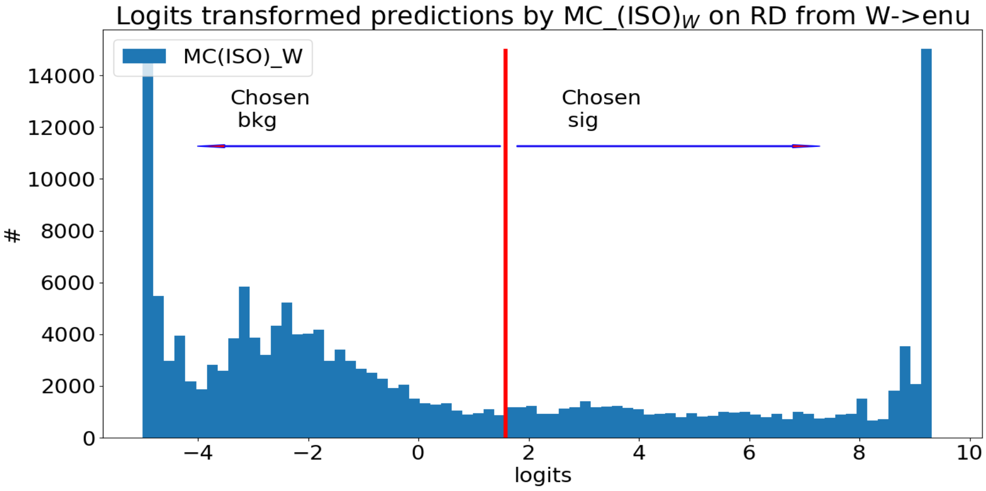


Figure 25: *Initial predictions in W RD by MC(ISO)$_W$. The red line is the initial guess, and separates predicted signal from predicted background.* Initially everything less than the lines x-position is assumed background and everything bigger is assumed signal.

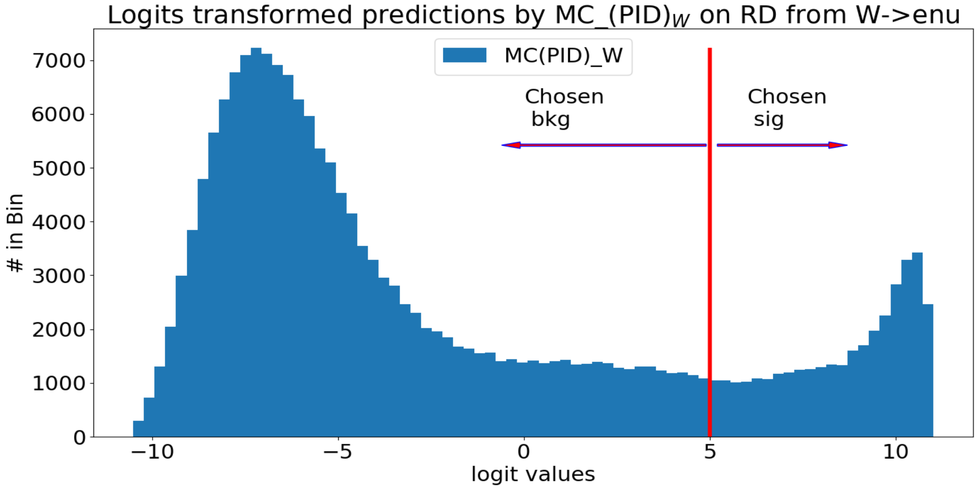


Figure 26: *Initial predictions in W RD by MC(PID)$_W$. The red line is the initial guess, and separates predicted signal from predicted background.*

What is worth noting in figure 25 and 26 of MC(PID)$_W$ and MC(ISO)$_W$ predictions on RD is that there are peaks at each

side of the spectrum, which likely represents signal and background like in MC. The MC(ISO)$_W$-predictions are more 'bumpy', while the MC(PID)$_W$-predictions are more smooth. The cuts between signal and background include a lot of data outside the peaks, when everything before the cut is signal and after the cut is background. It was tried to make different cuts like the ones seen in figure 31 and 32 before reweighing, but did not result in much improvement.
In figure 27 a drawing of how the initial labels are made is shown. The arrows represent ML-models trained on the variables they originate from. So a ML-model trained on ISO-variables is used for making labels (Truth) for PID-variables in RD, and vice versa.

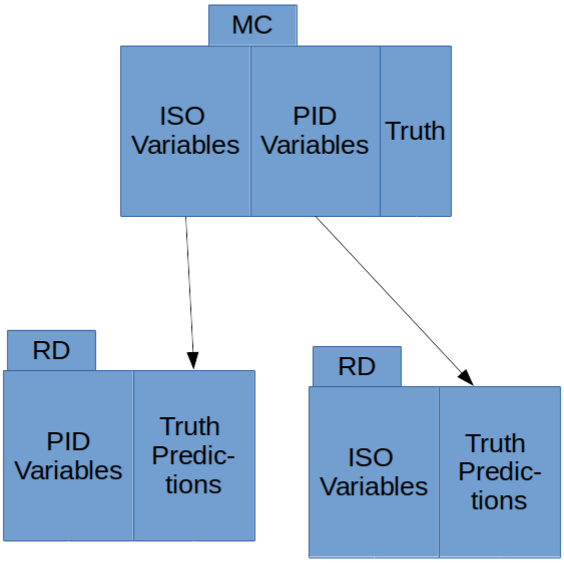


Figure 27: *The factory line of making labels in RD. In MC the labels (Truth) is known, in RD the labels come from predictions by MC trained ML models. The arrow represent ML models predicting truth labels.*

3.5 INITIAL LABELS IN Z - DATA

Labels and predictions in Z-data is made in a similar way. The plots can be seen in the appendix.
Z-data had a lot more data points and a very high signal percentage.

GeV	**# Bkg**	**# Sig**
80<E_T <100	299.260	877.533
100<E_T <150	305.866	628.356
150<E_T <200	118.522	128.308
200<E_T	103.807	49.496
Total	827.455	1.683.693

Table 14: Signal and background for Z RD as a function of E_T. Labels are made from a separation value in logit transformed predictions by MC(ISO)$_W$ model

3.6 RD VARIABLE DISTRIBUTIONS

The variable distributions for both Z- and W-data are in the appendix .4.

3.7 REWEIGHING IN RD

Reweighing for RD for data with labels from MC(ISO)$_W$ is shown in figure 28. For Z-data an error was made, that resulted in Z not being reweighted in $\langle \mu \rangle$. This was discovered late. Because Z-results were good, it was tested whether it had an impact on W-data if $\langle \mu \rangle$ was not reweighted. This did not yield any significant difference in AUC.

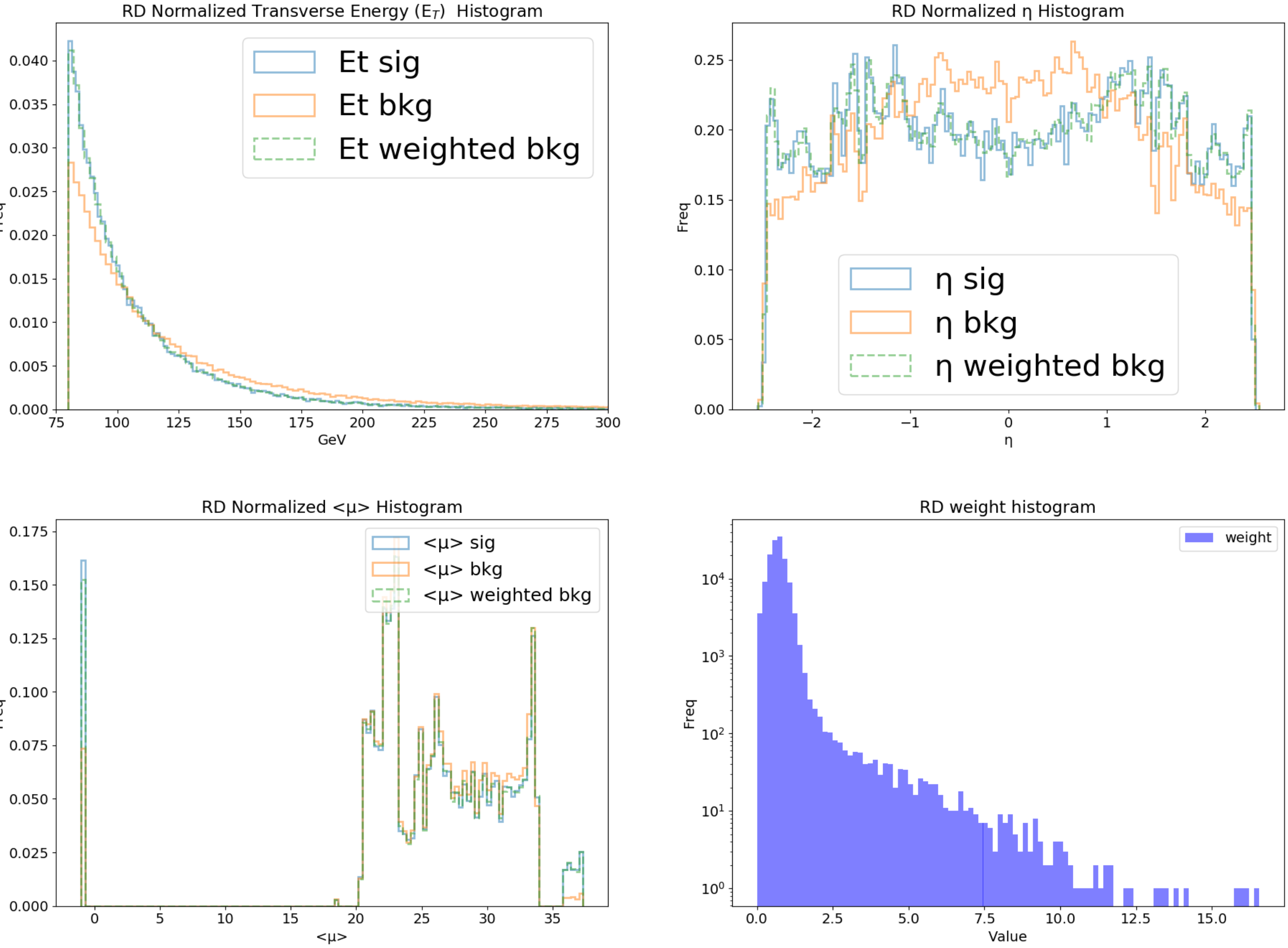


Figure 28: *$W \to e\nu$ .Plots of E_T, η, $\langle\mu\rangle$ and weights. The weights are calculated to make background look like signal. Data is RD with labels from MC(ISO)$_W$*

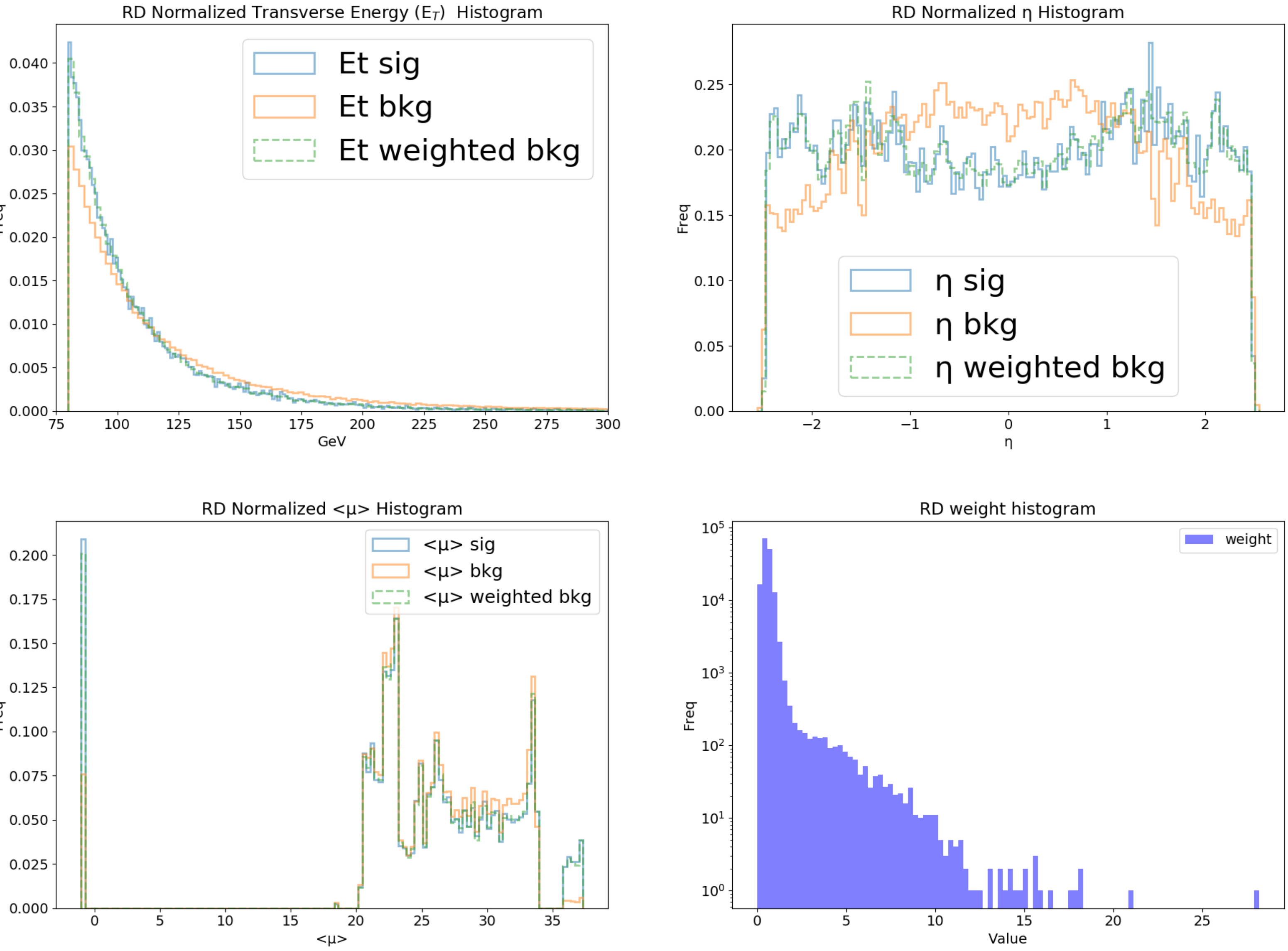


Figure 29: *$W \rightarrow e\nu$. Plots of E_T, η, $\langle\mu\rangle$ and weights. The weights are calculated to make background look like signal. Data is RD with labels from MC(PID)$_W$*

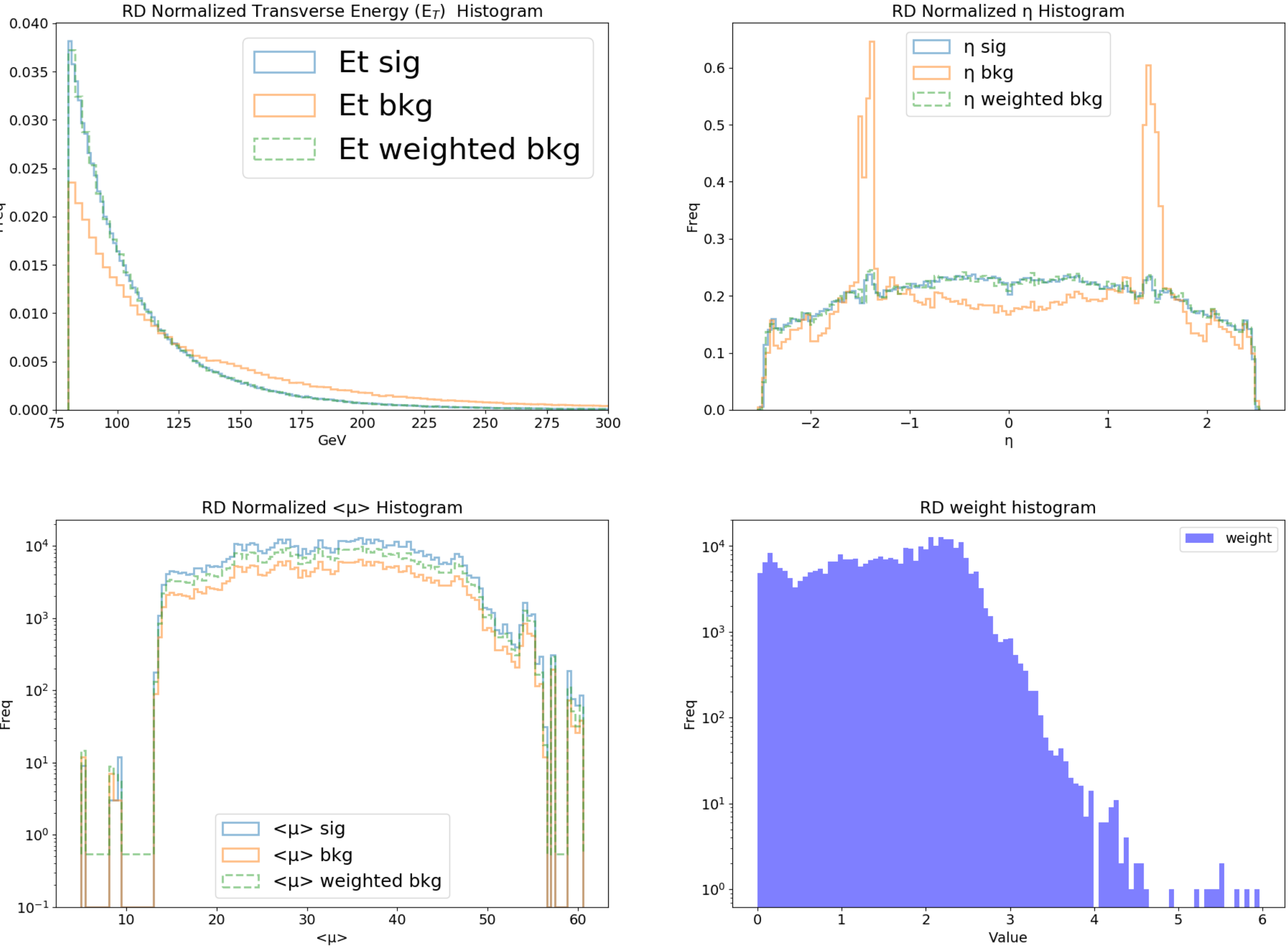


Figure 30: $Z \rightarrow ee$.*Plots of* E_T, η, $\langle\mu\rangle$ *and weights. The weights are calculated to make background look like signal. Data is RD with labels from MC(ISO)*$_W$

3.8 DECORRELATION

Since MC(PID)$_W$ predictions (abbrevated *pid* in the following equations) and MC(ISO)$_W$ ML-model predictions (abb. *iso*) are correlated as seen in table 11, decorrelation will make the labels better to train on. There is a mathematical framework for decorrelating two variables linearly:

Defining

$$x = \frac{pid}{\sigma_{pid}} \qquad \text{and} \qquad y = \frac{iso}{\sigma_{iso}}$$

Then

$$\sigma(x) = \sigma(y) = 1$$

and

$$cov(x, y) = corr(x, y)$$

Where cov is covariance, corr is pearson correlation (ρ) and σ is the variance. This means the variable

$$y' = y - corr(x, y) \cdot x \tag{2}$$

is linearly decorrelated from x.

Proof
The proof of this can be made by calculating the covariance or correlation between y' and x and see that it is zero:

$$cov(y', x) = E[(x - \mu_x)(y' - \mu_{y'})]$$

where E[] means expectation value and μ is the mean. Inserting $y' = y - \rho_{x,y} \cdot x$:

$$cov(y', x) = E[(x - \mu_x)(y - \rho_{x,y} \cdot x - E[y - \rho_{x,y} \cdot x])]$$

This expression can be grouped together as:

$$cov(y', x) = cov(x, y) - \rho_{x,y} \cdot (E[x^2] - E[x]^2) = cov(x, y) - \rho_{x,y} \cdot \sigma_x^2$$

Since $\sigma_x = 1$ and $cov(x, y) = \rho_{x,y}$ it is zero
. □

Final decorrelated variable
Inserting values for x and y into eq. 2 we get:

$$iso'_{bkg/sig} = \frac{iso_{bkg/sig}}{\sigma_{iso}} - \rho_{pid,iso} \cdot \frac{pid_{bkg/sig}}{\sigma_{pid}}$$

which is decorrelated from $x = \frac{pid_{bkg/sig}}{\sigma_{pid}}$ and $pid_{bkg/sig}$ because multiplying $\rho(iso'_{bkg/sig}, \frac{pid_{bkg/sig}}{\sigma_{pid}})$ with σ_{pid} is still zero. Multiplying one of the variables in ρ with a constant is the same as multiplying ρ with a constant

$$k \cdot \rho(a,b) = \rho(a, k \cdot b)$$

and therefore we can get a pretty final expression which is decorrelated from pid:

$$iso'_{bkg/sig} = iso_{bkg/sig} - \frac{\sigma_{iso}}{\sigma_{pid}} \cdot \rho_{pid,iso} \cdot pid_{bkg/sig} \tag{3}$$

Here σ_{pid} and σ_{iso} should be for background or signal, but writing $\sigma_{pid,bkg/sig}$ was deemed to be too long.

Decorrelating in practice
In practice there can be a large discrepancy in correlation values in signal and correlation values in background. When decorrelating signal and background individually, the linear 'movement' in equation 3 can move the signal into the background or vice versa. Therefore a more wholesome approach was done. Instead of decorrelating signal and background separately. Signal and background was decorrelated together with ρ and σ in equation 3 replaced by a sigmoid function of the total MC(ISO)$_W$ score.

$$f(x, iso_{tot}) = x_{sig} + \frac{x_{bkg} + x_{sig}}{1 + e^{(iso_{tot}+\mu)/\sigma}} \tag{4}$$

In equation 4 μ and σ should be the average and standard deviation of the total iso-scores. But a lower pearson correlation coefficient was found by making a gridsearch over the mu-σ space. x can be σ and ρ in equation 3

3.9 CLEANING DATA

The peaks at large prediction score values are assumed to represent signal, and the peaks at low values are assumed

to represent background. In data the separation is far from as crystal clear as in MC. There are a lot 'dirty' events in RD, where maybe something happened in the detector that corrupted the output. That means the data has to be cleaned. This is done by removing the dirty data through cuts. The cuts then determine signal and background.
You might argue that reweighing should be done after cleaning. It was tested and there was not a significant drop in AUC-scores if cleaning was done after reweighing. It also was easier and faster.

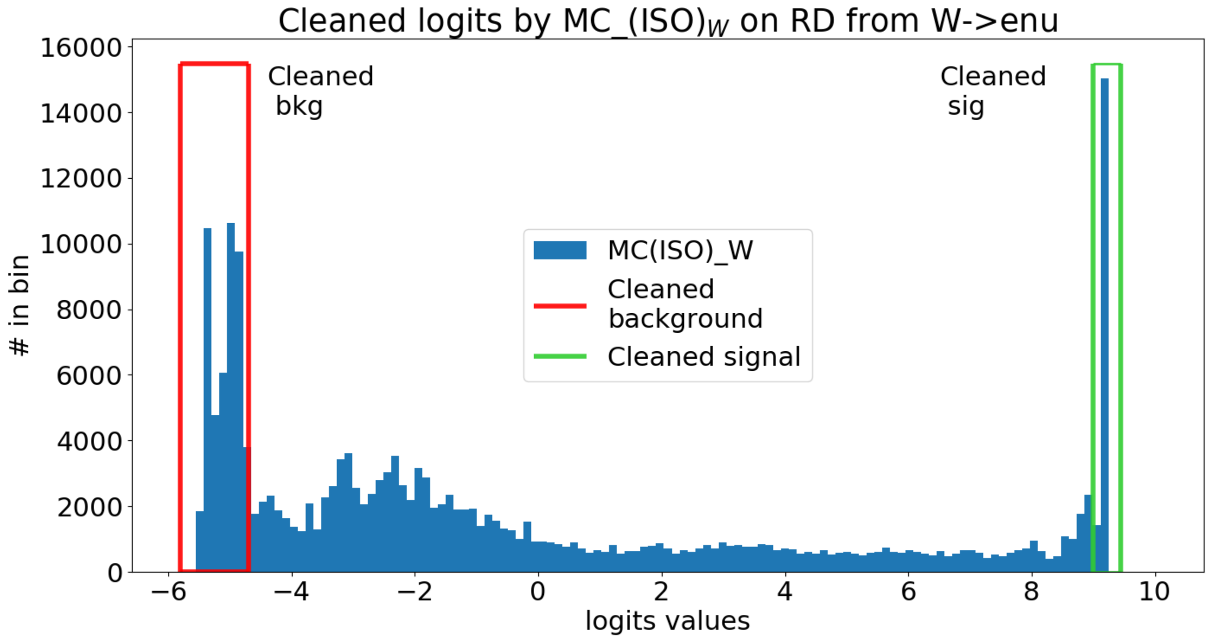


Figure 31: *The logit transformed scores of predictions on RD from W → eν . The predictions are made with MC(ISO)$_W$*

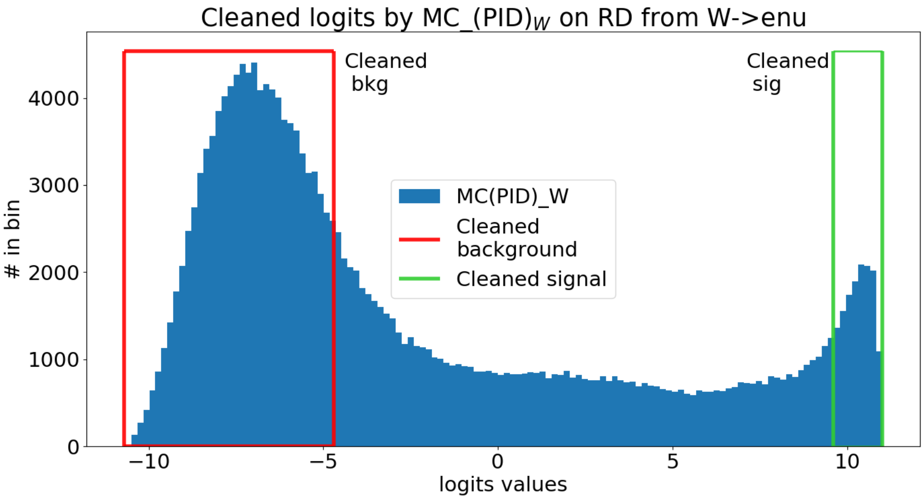


Figure 32: *The logit transformed scores of predictions on RD. The predictions are made with MC(PID)$_W$. After cleaning only the red (background) and green (signal) colored section is trained on and the middle is removed.*

3.10 ROC CURVES IN RD

Like in MC, two ML-models where trained: RD(PID)$_W$ and RD(ISO)$_W$. RD(PID)$_W$ is trained on labels by MC(ISO)$_W$, and RD(ISO)$_W$ is trained on labels by MC(PID)$_W$.
For Z-data a model (RD(PID)$_Z$) was trained on data with labels by MC(ISO)$_W$

RD(PID)$_W$
The ROC curve for a boosted decision tree trained on the set of cleaned PID variables can be seen in figure 33. With a AUC score of 0.9998 the MC(PID)$_W$ model is very good at separating signal from background. What is not so good is that the ATLAS Likelihood points for the data are not so. The ATLAS LH-points should lie above the ATLAS LH control variable

	Tuned LGBM HPs
RD(ISO)$_W$	learning_rate: 0.100 num_leaves : 25
RD(PID)$_W$	learning_rate: 0.002 num_leaves: 2
RD(PID)$_Z$	learning_rate: 0.035 num_leaves: 5

Table 15: *Trained ML-models in RD.*

(blue curve), since the points are based on the same variables but with additional variables added to the LH-points. I have tried testing different model improvements, but not found an explanation for this. When looking at the same plot in Z-data, figure 35 (labels also made with MC(ISO)$_W$ and trained on PID variables), there the LH-points behaves like they should, which makes this look strange.
One possible explanation could be that the cuts in the ATLAS LH points worsen the performance in W-data at high energies.

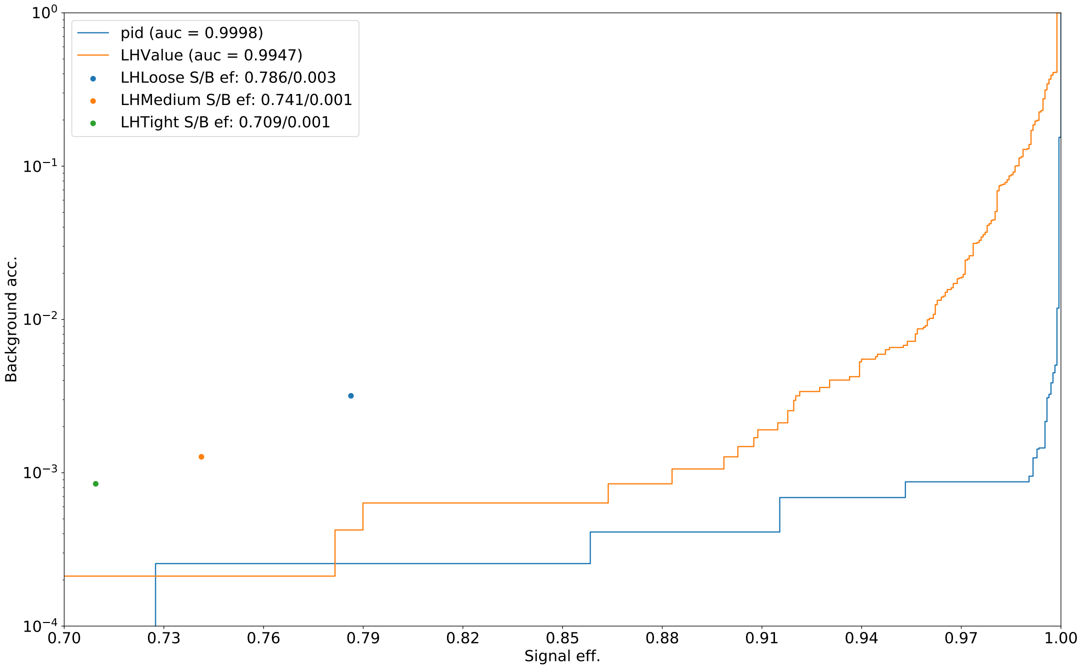


Figure 33: *RD W → eν. ROC curve of RD(PID)$_W$ model, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS likelihood points. Labels are made with MC(ISO)$_W$-model*

RD(ISO)$_W$
The ROC curve can be seen in figure 34. Compare the LH-points to figure 33 and here the they look more as one would expect. This indicates that when making labels a ML-model based on PID-variables in MC then it agrees with the LH-points, which is based on a lot of the same variables.
The ATLAS ISO curve is from $ptcone40/E_T$ which is the current method for separating electrons from background in the ATLAS detector.

RD(PID)$_Z$

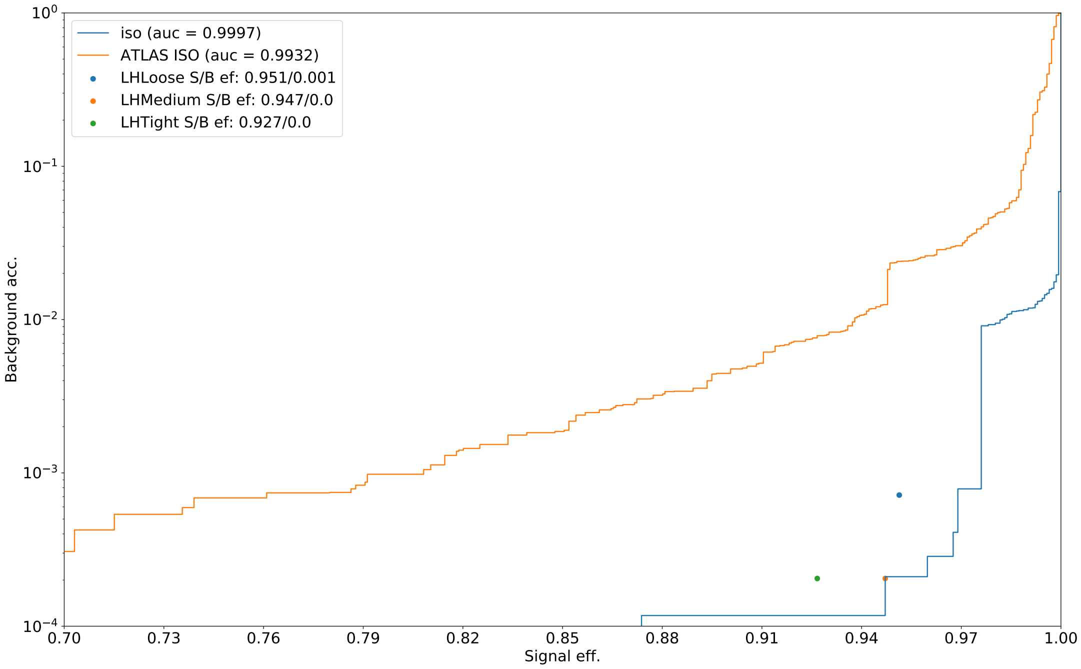


Figure 34: *RD $W \rightarrow e\nu$. ROC curve of RD(ISO)$_W$, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS Likelihood points. Labels are made with MC(PID)$_W$-model*

Data from $Z \rightarrow ee$ and labels made with an MC(ISO)$_W$-model (trained on $W \rightarrow e\nu$) is seen in figure 35. What we see is RD(PID)$_Z$ outperforming the ATLAS LH variables. With an AUC of 0.9991, RD(PID)$_Z$ does a well job of separating signal from background. The LH-points here does more as one would have expected.

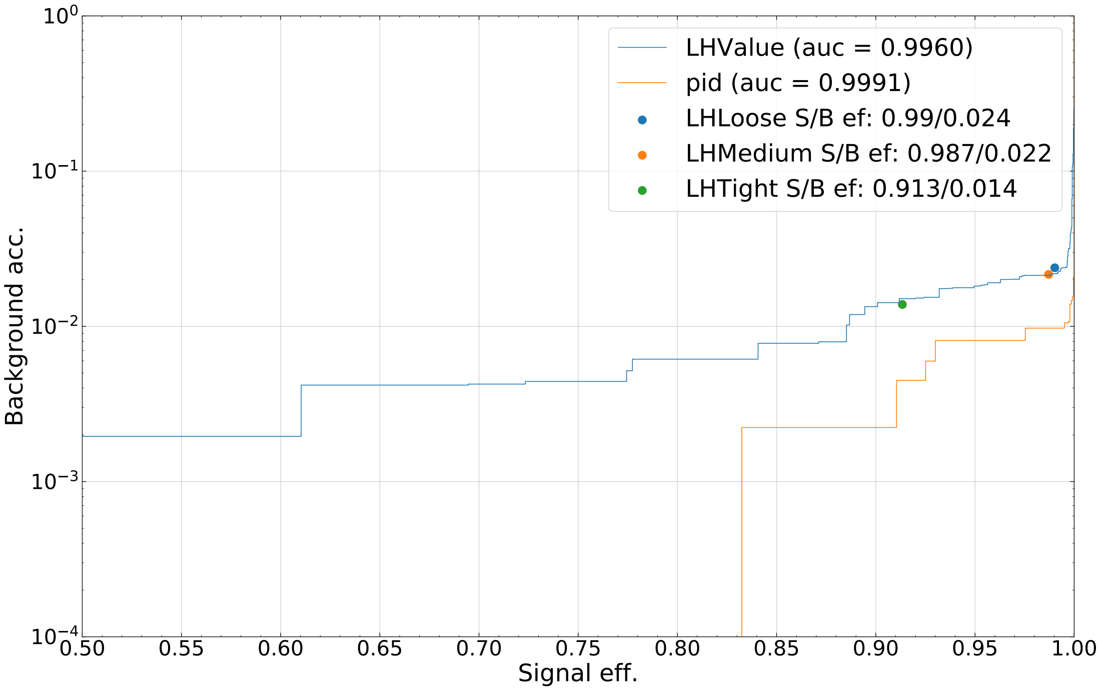


Figure 35: *RD $Z \rightarrow ee$. ROC curve of RD(PID)$_Z$ model, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS likelihood points. Labels are made with MC(ISO)$_W$-model*

3.11 DECORRELATION RESULTS

Previous results (AUC and ROC-curves) were not decorrelated. Decorrelation have been successful before by Benjamin Henckel on Z-data, where it was used to decorrelate labels made with ML-models but also requiring electrons coming from a Z-boson (Z-mass peak). No Z-mass peak or W-mass peak was possible so labels were only made with MC trained ML-models. In general decorrelation was not successful in this project. It was probably due to the fact that pearson correlation scores was high. The best result I got was with labels made by a different MC-model, which will be called MC(ISO)$_{W75}$. The 75 comes from it was trained on 7.5 mio. events. This did worse in not decorrelated data, maybe because of the high signal/background ratio.

3.11.1 *Data from $W \rightarrow ev$*

Labels for data from $W \rightarrow ev$ were made with MC(ISO)$_{W75}$ model and then decorrelated: The steps taken are:

1. Train MC(ISO)$_{W75}$ ML-models in MC (Done in the MC chapter)
2. Make initial labels in RD using the MC(ISO)$_{W75}$ predictions choosing signal/background cut.
3. Use a MC(PID)$_{W75}$ model to make predictions in same data
4. Decorrelate initial predictions with the decorrelation framework described in section 3.8.
5. Calculate weights
6. Clean data
7. Train an ML-model based on PID-variables
8. Evaluate using ROC curves, AUC, ATLAS LH-points.

The next couple of subsections will describe it in more detail.

Decorrelated predictions for $W \to e\nu$
Separating signal and background at a logit score of 4.8 for MC(ISO)$_{W75}$ predictions gives a pearson correlation coeffcient for signal of 0.18, and background of 0.82. But after decorrelation, the correlation is -0.003 for background and -0.007 for signal with $mu = 10$ and $\sigma = 2.5$ in equation 4. But looking at the distance correlation coefficients in table 11, they are not completely decorrelated. According to distance correlation background is still heavily correlated and signal to a lesser degree.

Before decorrelation:

	Background	Signal
Pearson corr. koeff.	0.82	0.18
Distance corr. koeff.	0.80	0.18

After decorrelation:

	Background	Signal
Pearson corr. koeff.	-0.003	-0.007
Distance corr. koeff.	0.48	0.18

Table 16: Correlation coefficients between predictions by MC(ISO)$_W$ and MC(PID)$_W$ in RD

The decorrelated MC(ISO)$_{W75}$-predictions in figure 36, show that heavily correlated signal and background points have been moved by decorrelation. They should be compared to figure 41 (in the Appendix) of predictions before they are decorrelated to see what decorrelating does to the distributions. They still have some of the same "bumps" in the histogram, but they have been smoothed out.

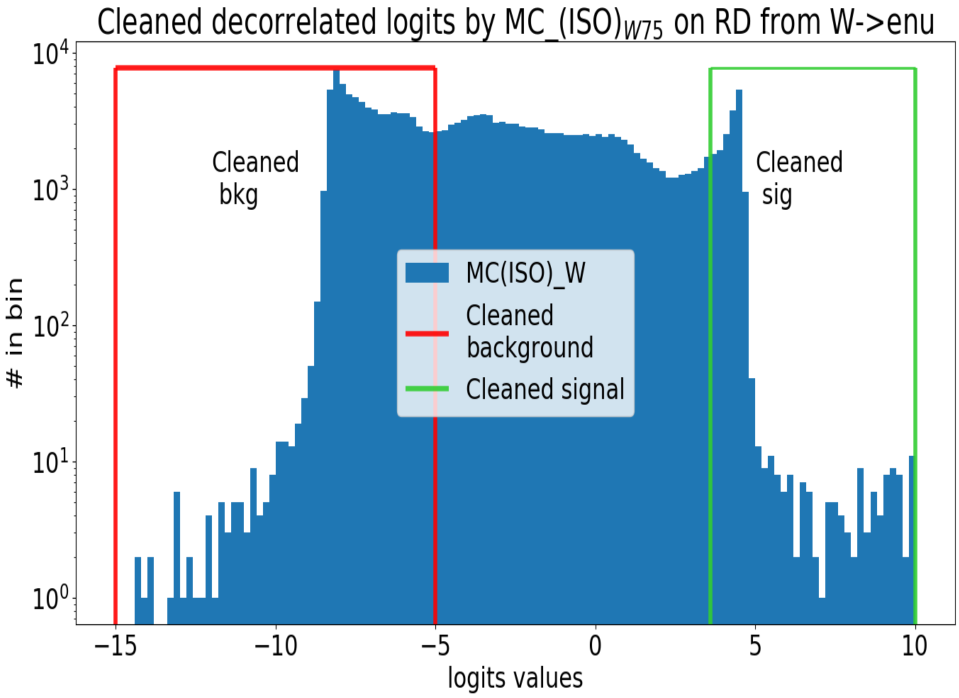


Figure 36: *Histogram of decorrelated MC(ISO)$_{W75}$ prediction scores after logit transformation. The signal and background that is left after cleaning of the data is marked with green for signal and red for background*

2D Plots Of MC(PID)$_{W75}$ And Decorrelated MC(ISO)$_{W75}$ Predictions

The 2D plot of the decorrelated MC(ISO)$_{W75}$ and MC(PID)$_{W75}$ (not decorrelated) predictions in figure 37 look very different from the 2d plot of the MC(ISO)$_W$ and MC(PID)$_W$ before MC(ISO)$_{W75}$ is decorrelated. The change to the background looks like it gets a strong linear correlation, even though it should be linearly decorrelated. But it shows the weaknesses of the linear decorrelation method. RD has a lot of dirty events. What could go wrong is that both models don't know how to deal with these dirty events and the predictions becomes correlated. This leads to a high shift in many of the data points and background and signal get mixed up.

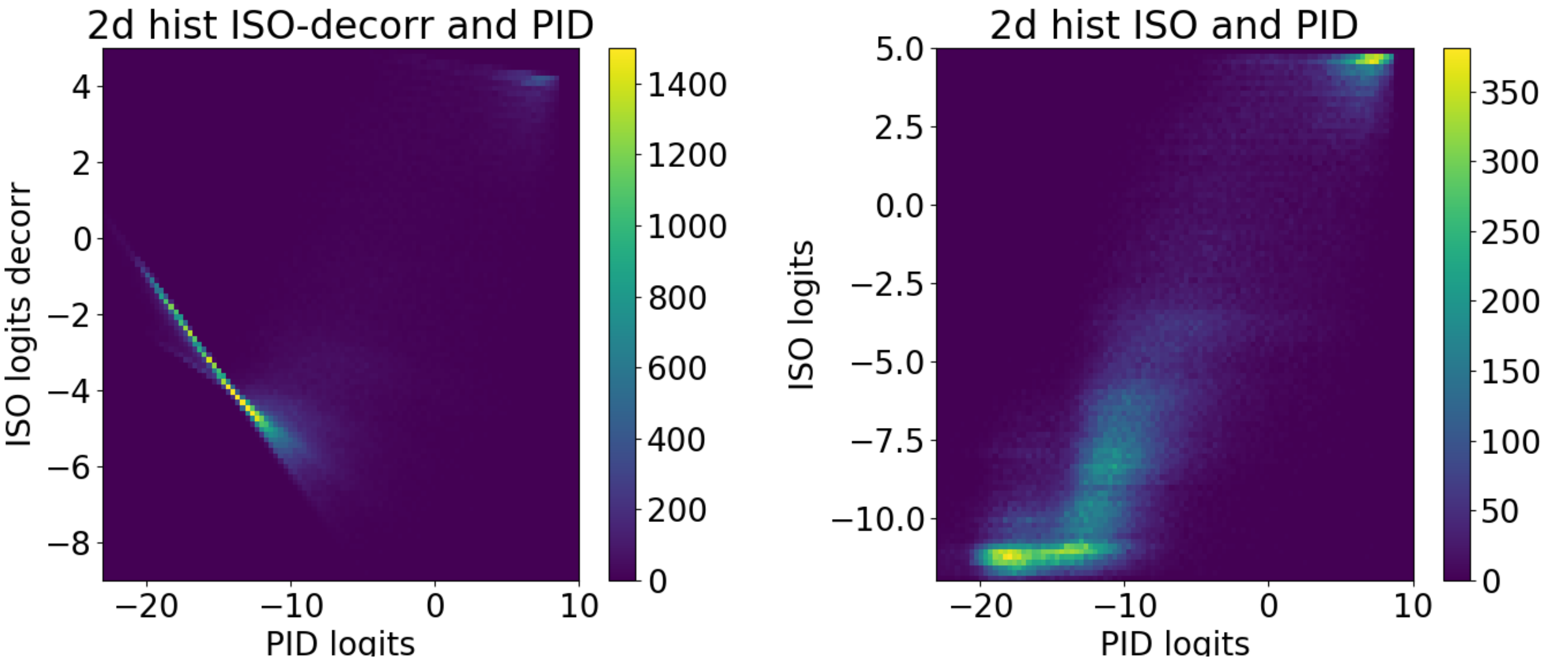


Figure 37: *2D histograms of MC(PID)$_{W75}$ and MC(ISO)$_{W75}$ predictions on RD. To the left, MC(ISO)$_{W75}$ has been decorrelated. To the right is before decorrelation*

AUC and ROC-curve

The AUC has dropped to 0.9987 compared to not decorrelated AUC. It is expected that the AUC would drop since decorrelation takes away information. But the LH-points have a little higher value than before. PID still beats the LH-points and the LH-variable.

3.11.2 *Decorrelation In Data From Z → ee*

RD(PID)$_Z$ trained on decorrelated labels from MC(ISO)$_{W75}$, did not get a good AUC. Despite creating linearly uncorrelated MC(ISO)$_{W75}$-model scores for labels, AUC fell to 0.98 and the Z-peak from this model was also low. I will therefore not spend much time on this.

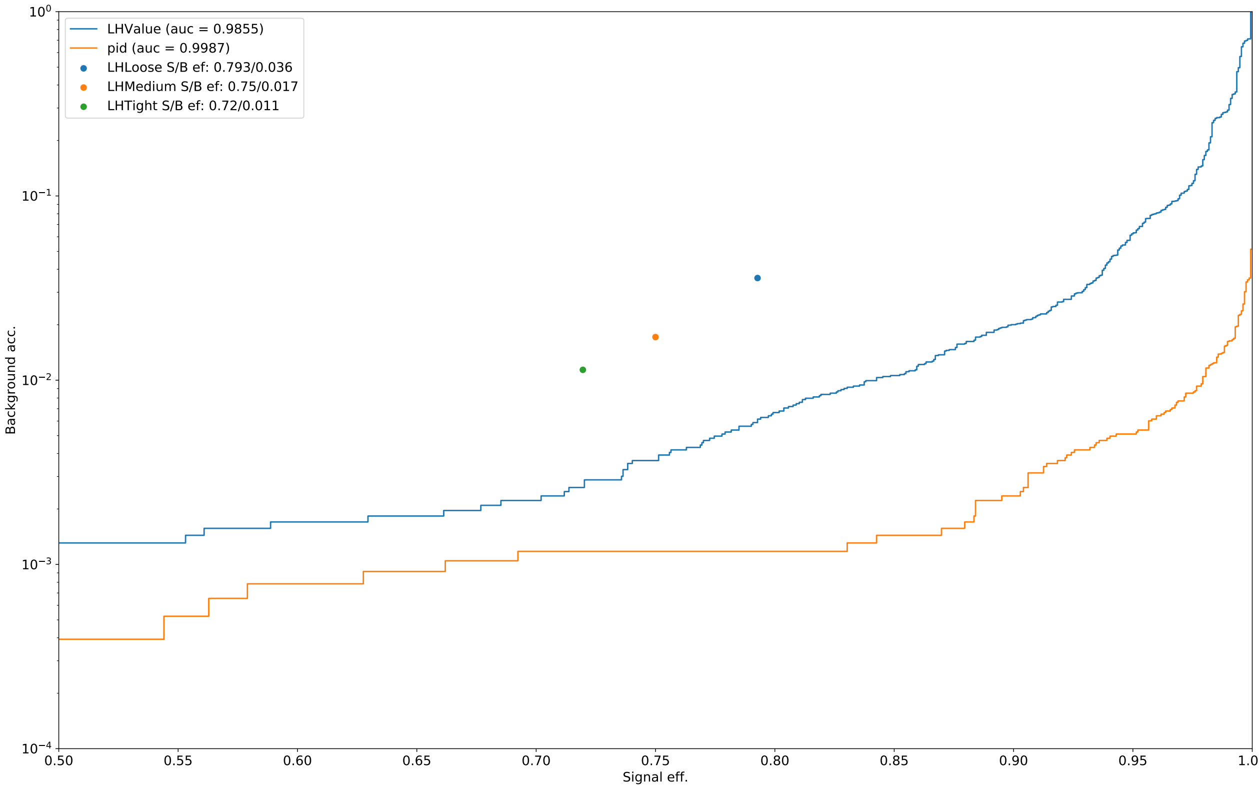


Figure 38: *ROC-curve from PID model trained on MC(ISO)$_{W75}$ labels linearly decorrelated from MC(PID)$_{W75}$ predictions*

3.12 EVALUATION WITH Z-PEAKS

One of the advantages of using data from $Z \rightarrow ee$ is that you can evaluate ML-models on how good they are to predict electrons from the size of the peak around the Z-mass (91 GeV). This is a great advantage because, requiring the combined mass of two electrons to be in the vicinity of the mass of the Z-boson is an unbiased way of making labels in RD. But with high energy electrons above 80 GeV, there is very few that combine to a total mass of the Z-boson. But there are a few, and therefore this gives an extra way to evaluate the efficiency of different models.
The function used to fit the peak was a straight line (to represent background) added a gaussian multiplied by the error-function [27] to give it a bit of a tail:

$$l(a, b) = ax + b \tag{5}$$

$$f(h,\sigma,\tau,\mu) = h \cdot \frac{\sigma}{\tau} \cdot \sqrt{\frac{\pi}{2}} \cdot \exp\left[\frac{1}{2} \cdot \left(\frac{\sigma}{\tau}\right)^2 - \frac{x-\mu}{\sigma}\right] \cdot erfc\left(\frac{1}{\sqrt{2}}\left(\cdot\frac{\sigma}{\tau} - \frac{x-\mu}{\sigma}\right)\right) \tag{6}$$

where erfc is given by (1 - the errorfunction):

$$erfc(x) = 1 - erf(x) \tag{7}$$

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \tag{8}$$

The total fitting function is given by:

$$F(a,b,h,\sigma,\tau,\mu) = l(a,b) + f(h,\sigma,\tau,\mu) \tag{9}$$

This function (9) gives a rough fit to the peak and results in a rough estimate. Therefore results are only indicative and not perfectly matching data. From figure 39 it shows that the best model to predict from electrons $Z \rightarrow ee$ is surprisingly MC(ISO)$_W$. It is marginally better than RD(PID)$_Z$. This suggests that correlations aren't that important at high energy. For clarity reasons other models where not included in the plot, they were all between the ATLAS LH Loose and RD(PID)$_Z$ fit.

It should be noted, that there is not a lot of data. The max peak has only 175 electrons (out of about 1 mio. individual events).

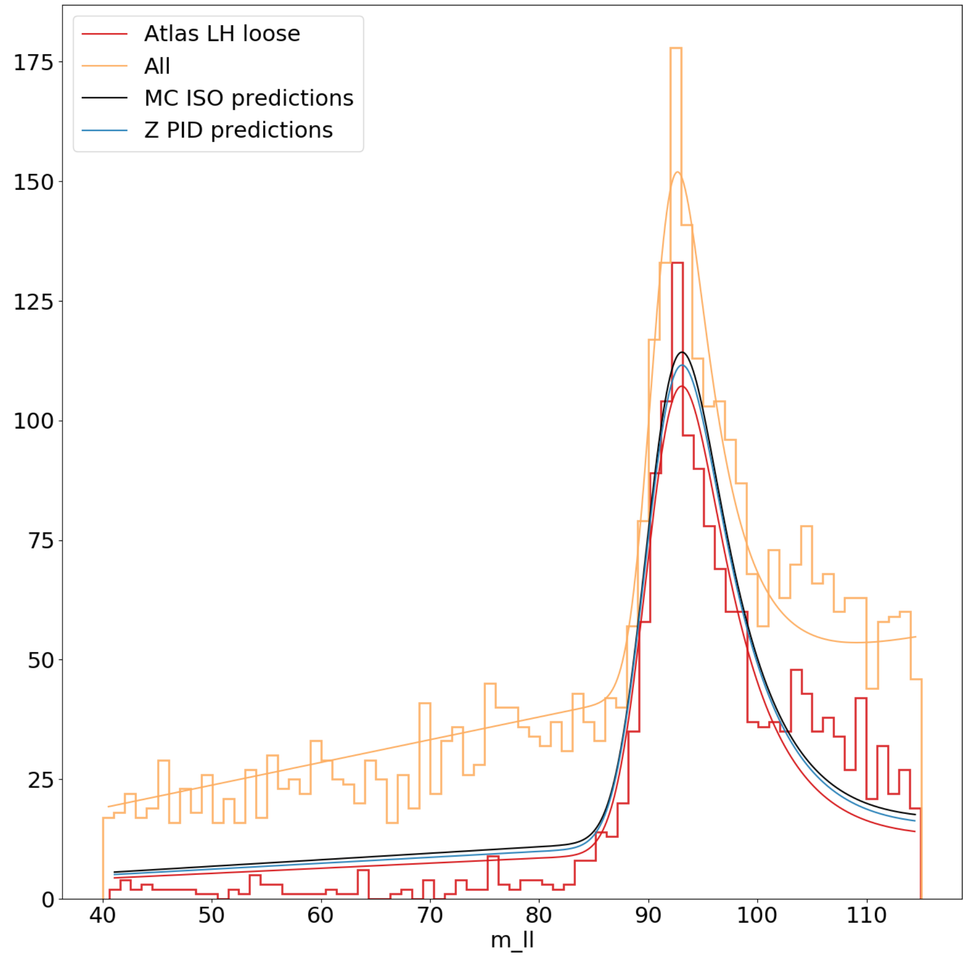


Figure 39: *Z-mass peak for different models. The fits are fitted to the histogram bars. Here is the MC(ISO)$_W$ predictions, RD(PID)$_Z$ predictions, ATLAS Loose likelihood, and histogram of all events. RD(PID)$_Z$ and MC(ISO)$_W$ are very close to eachother. Loose histogram is also included to represent the other histograms. To include them all would not yield clarity.*

3.13 DISCUSSION

Ways to improve the investigation into ML-models performance in high energy could be to change the weights in E_T. The fact that the amount of electrons with high E_T drops exponentially could be mitigated by giving higher weights to electrons with higher energy. Another way that might be possible is to create data with GANS (Generative Adverserial Neural Networks). For decorrelation could be used Neural networks.

3.14 CONCLUSION

It was hard to make clear conclusions because of the large correlation between the ISO variable- and the PID variable based ML models. AUC might be good for a model but the LH-points were bad. The large correlation between points made it impossible to get good AUC scores and Z-peaks for ML-models trained on decorrelated labels. Linear decorrelation works if the data is not too correlated (maybe for ρ less than 0.40), but with a correlation at 0.60-0.80, decorrelation removes too much information and background and signal gets mixed into each other.

To make labels alone with a ML-model trained on MC seems not optimal. It is seen in the Z-mass distribution, where a MC trained ML-model performs the best in RD. It should be noted though that the Z-mass distributions also are low on data, and the differences between the ML-models are small. The need for labels made from Z or W distributions is important. This is one of the struggles of high energy data.
There is still improvements though compared to the ATLAS LH.

ACRONYMS

ATLAS A Toroidal LHC ApparatuS. 3, 5, 9, 10, 11, 12, 15, 17, 18, 20, 21, 23, 28, 29, 30, 38, 41, 43, 44, 54, 55, 56, 58, 62, 64, 66, 67, 68, 71

BDT Boosted Decision Trees. 5

HP Hyper Parameters for Machine Learning algortihms. 24, 36

ISO Variables based on isolation in the ATLAS detector. 5, 30, 41, 44, 46, 64

LH Likelihood. 5, 20, 21, 31, 38, 42, 54, 55, 58, 60, 64

LHC Large Hadron Collider. 5, 6, 9, 10, 20, 43

MC Data based on simulation using the Monte Carlo method. 5, 28, 29, 30, 35, 41, 44, 46, 52, 54, 55, 57, 64, 67, 69, 77

ML Machine Learning. 3, 5, 6, 22, 23, 24, 25, 26, 28, 30, 35, 36, 41, 44, 46, 51, 54, 55, 57, 58, 61, 64, 66, 67, 69, 79

PID Variables based on particle identification in the ATLAS detector. 5, 30, 31, 30, 36, 41, 44, 46, 54, 55, 58, 64, 66, 69, 77, 79

RD Real Data. Data collected from collisions in the ATLAS detector at the LHC. 28, 30, 35, 36, 38, 41, 43, 44, 45, 46, 47, 52, 53, 58, 59, 61, 64, 67, 68

LIST OF FIGURES

Figure 1 *Fundamental particles in the Standard Model* 7

Figure 2 *Overview of cross-section measurements. Except for the Jets and proton-proton they all include possible decays to electrons* 7

Figure 3 *Exotic searches that include the possibility of new particles. Many of them include high energy electrons. The yellow bars show the energy range in which the models are excluded at 95% confidence level.* 8

Figure 4 *The Large Hadron Collider [28]* 9

Figure 5 *The ATLAS detector [28]* 10

Figure 6 *Pseudorapidity* 11

Figure 7 *Overview of the Inner Detector [9]* 11

Figure 8 *Drawing showing the path a 10 GeV particle traverses in the barrel part of the ID[9]* 12

Figure 9 *ATLAS calorimeters [9]* 13

Figure 10 *ECAL barrel module[9]* 14

Figure 11 *Electrode structure of the ATLAS electromagnetic calorimeter [12]* 15

Figure 12 *Overview of the Muon System in the ATLAS detector. Picture source [9]* 17

Figure 13 *Example of decision tree. Source: [30]* 23

Figure 14 *Cross entropy of possible ML values to signal (y=1) and background (y=0)* 25

Figure 15 LighGBM compared to other tree algorithms 27

Figure 16 *The distributions of the isolation variables* 33

Figure 17 33

Figure 18 *The distributions of the PID variables, with weights from reweighing. The rest of the 24 variables can be found in the Appendix* 34

Figure 19 *Histograms of signal, background and weighted background of $\langle\mu\rangle$, E_T and η. The weight distribution is in the bottom right* 36

Figure 21 *ROC curve of ATLAS Likelihood value, pid ($MC(PID)_W$), iso ($MC(ISO)_W$), and the Tight, Medium and Loose ATLAS likelihood points* 39

Figure 22 *2d histogram of logit-transformed predictions from $MC(PID)_W$ and $MC(ISO)_W$* 40

Figure 23 *Picture of an event with illustration of Tag and Probe. Here a Tag is an electron candidate selected with high certainty. If there is another candidate which is isolated it is called a Probe. If the Tag and Probe system has an invariant mass close to the Z-mass, the probe is likely an electron. For $W \rightarrow e\nu$ it was tried to have the tag as the E_T^{miss} and the probe as the electron and then compare it to the transverse W-mass, but with not accurate results. The advantage of Tag and Probe is that it is a relatively unbiased method of selecting electron labels for machine learning.* 43

Figure 24 *The ATLAS E_T^{miss} trigger efficiency as a function of energy.* 44

Figure 25 *Initial predictions in W RD by $MC(ISO)_W$. The red line is the initial guess, and separates predicted signal from predicted background.* Initially everything less than the lines x-position is assumed background and everything bigger is assumed signal. 45

Figure 26 *Initial predictions in W RD by $MC(PID)_W$. The red line is the initial guess, and separates predicted signal from predicted background.* 45

Figure 27 *The factory line of making labels in RD. In MC the labels (Truth) is known, in RD the labels come from predictions by MC trained ML* models*. The arrow represent ML* models *predicting truth labels.* 46

Figure 28 *$W \rightarrow e\nu$ .Plots of E_T, η, $\langle\mu\rangle$ and weights. The weights are calculated to make background look like signal. Data is RD with labels from $MC(ISO)_W$* 48

Figure 29 *$W \rightarrow e\nu$. Plots of E_T, η, $\langle\mu\rangle$ and weights. The weights are calculated to make background look like signal. Data is RD with labels from MC(PID)$_W$* 49

Figure 30 *$Z \rightarrow ee$ .Plots of E_T, η, $\langle\mu\rangle$ and weights. The weights are calculated to make background look like signal. Data is RD with labels from MC(ISO)$_W$* 50

Figure 31 *The logit transformed scores of predictions on RD from $W \rightarrow e\nu$. The predictions are made with MC(ISO)$_W$* 53

Figure 32 *The logit transformed scores of predictions on RD. The predictions are made with MC(PID)$_W$. After cleaning only the red (background) and green (signal) colored section is trained on and the middle is removed.* 54

Figure 33 *RD $W \rightarrow e\nu$. ROC curve of RD(PID)$_W$ model, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS likelihood points. Labels are made with MC(ISO)$_W$-model* 55

Figure 34 *RD $W \rightarrow e\nu$. ROC curve of RD(ISO)$_W$, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS Likelihood points. Labels are made with MC(PID)$_W$-model* 56

Figure 35 *RD $Z \rightarrow ee$. ROC curve of RD(PID)$_Z$ model, ATLAS Likelihood variable and Tight, Medium and Loose ATLAS likelihood points. Labels are made with MC(ISO)$_W$-model* 57

Figure 36 *Histogram of decorrelated MC(ISO)$_{W75}$ prediction scores after logit transformation. The signal and background that is left after cleaning of the data is marked with green for signal and red for background* 59

Figure 37 *2D histograms of MC(PID)$_{W75}$ and MC(ISO)$_{W75}$ predictions on RD. To the left, MC(ISO)$_{W75}$ has been decorrelated. To the right is before decorrelation* 60

Figure 38 *ROC-curve from PID model trained on MC(ISO)$_{W75}$ labels linearly decorrelated from MC(PID)$_{W75}$ predictions* 61

Figure 39 *Z-mass peak for different models. The fits are fitted to the histogram bars. Here is the MC(ISO)$_W$ predictions, RD(PID)$_Z$ predictions, ATLAS Loose likelihood, and histogram of all events. RD(PID)$_Z$ and MC(ISO)$_W$ are very close to eachother. Loose histogram is also included to represent the other histograms. To include them all would not yield clarity.* 63

Figure 40 *Predictions from MC(ISO)$_W$ model on data from Z → ee . The red line represents the initial guess of what separates signal from background* 76

Figure 41 *Initial predictions by MC(ISO)$_{W75}$ on W-data from RD. Used for decorrelation* 77

Figure 42 *PID MC Variable distributions*. The last 16 of the 25 variables in PID. The other 9 are in the MC chapter 78

Figure 43 *Isolation variable distributions in W-data with labels made by MC(PID)$_W$*. They are used to train RD(ISO)$_W$ ML-models 80

Figure 44 *9 PID variable distributions from W-data with labels made by MC(ISO)$_W$*. They are used to train RD(PID)$_W$ ML-models 81

Figure 45 *16 PID variable distributions from W-data with labels made by MC(ISO)$_W$*. They are used to train RD(PID)$_W$ ML-models 82

Figure 46 *9 variable distributions from PID in Z-data with labels made by MC(ISO)$_W$*. They are 9 out of 25 variables that are used to train RD(PID)$_Z$ ML-models 83

Figure 47 *16 variable distributions from PID in Z-data with labels made by MC(ISO)$_W$* 84

BIBLIOGRAPHY

[1] B. R. Martin Nuclear and particle physics 2009 *John Wiley and Sons Ltd*
ISBN 978-0-470-74275-4 11 Mar 2021

[2] Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC 31 AUG 2012 *arXiv:1207.7214v2[hep-ex]* 11 Mar 2021

[3] Observation of a new boson at a mass of 125 GeV
with the CMS experiment at the LHC 28 JAN 2013
arXiv:1207.7235v2 [hep-ex] 11 Mar 2021

[4] Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton–proton collision data at $\sqrt{s} = 13$ TeV
arXiv:1902.04655v2 [physics.ins-det]
9 Aug 2019

[5] Search for new high-mass phenomena in the dilepton final state using 36 fb^{-1} of proton–proton collision data at $\sqrt{s} = 13 TeV$ with the ATLAS detector
arXiv:1707.02424v2 [hep-ex]
15 Nov 2017

[6] Advacam Wins Tendering of CERN LHCb VELO Upgrade.
http://advacam.com/1576.html. April 18, 2018
Accessed Sep 6, 2020

[7] Detector and Technology
https://atlas.cern/discover/detector. 2020
Accessed Sep 7, 2020

[8] The Inner Detector — Atlas website
https://atlas.cern/discover/detector/inner-detector. 2020
Accessed Dec 8, 2020

[9] The atlas experiment at the cern large hadron collider.
Journal of Instrumentation. 2008
The ATLAS Collaboration et al 2008 JINST 3 S08003

[10] Production and integration of the ATLAS Insertable B-Layer.
Journal of Instrumentation. 2018
B. Abbott et al 2018 JINST 13 T05008

[11] ACT Lectures on Detectors - Calorimeters (2/5)
https://indico.cern.ch/event/115059/. 2011
Philippe Bloch (CERN)

[12] Calorimetry for particle physics
REVIEWS OF MODERN PHYSICS, VOLUME 75. 2003
Christian W. Fabjan and Fabiola Gianotti (CERN)

[13] Atlas Cern news
https://atlas.cern/updates/atlas-news/new-sub-detector-atlas/. 2014

[14] The Muon System
https://atlas.cern/discover/detector/muon-spectrometer. 2020
Accessed Oct 6, 2020

[15] Electron reconstruction
Eur. Phys. J. C 79:639
https://doi.org/10.1140/epjc/s10052-019-7140-6 . 2019
Accessed Oct 9, 2020

[16] Electron efficiency measurements with the ATLAS detector using the 2015 LHC proton-proton collision data
ATLAS -CONF-2016-024 June 2016
Accessed Oct 16, 2020

[17] Link to the ATLAS definitions of truthtype (Particle type) and origin of particle (Particle Origin)
https://gitlab.cern.ch/atlas/athena/blob/master/PhysicsAnalysis/MCTruthClassifier/M

[18] Wikipedia Distance correlation
https://en.wikipedia.org/wiki/Distance_correlation

[19] EGAM file derivation descriptions
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaxAODDerivations
.

[20] Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1
ATLAS Collaboration

Eur. Phys. J. C (2017) 77:490 DOI 10.1140/epjc/s10052-017-5004-5

The European Physics Journal C. 2017
Accessed Dec 7, 2020

[21] Boosting Decision Trees
Drucker, Harris and Cortes, Corinna Advances in Neural Information Processing Systems.
vol 8, p 479-485

1995
Accessed Dec 7, 2020

[22] The Elements of Statistical Learning, Data Mining, Inference, and Prediction
Trevor Hastie, Robert Tibshirani and Jerome Friedman-Drucker

Second Edition

Springer Series in Statistics

Springer New York (April 2017).
ISBN: 9780387848570

Accessed Dec 7, 2020

[23] Particle identification with the ATLAS electromagnetic calorimeter.

TRDs for the Third Millenium - 3rd Workshop on Advanced Transition Radiation

Detectors for Accelerators and Space Applications, Sep 2005, Brindisi, Italy. pp.321-325, ff10.1016/j.nima.2006.02.151ff. ffin2p3-00105510f.

[24] Particle detectors and accelerators, Lecture notes.
Peter Hansen
Second edition. University of Copenhagen, 2015

[25] Reference to wtots1 variable description
http://cds.cern.ch/record/1115352/files/ATL-SLIDE-2008-072.pdf

[26] LightGBM Microsoft Corporation. Revision e5c3f7e7 2021
https://lightgbm.readthedocs.io/en/latest/

[27] Exponentially modified Gaussian distribution
https://en.wikipedia.org/wiki/Exponentially_modified_Gaussian_distribution

[28] Standard-Model fundamental particles pic
https://home.cern/science/physics/standard-model. 2020
Accessed Sep 8, 2020

[29] ATLAS detector picture
https://atlas.cern/discover/detector. 2020
Accessed Sep 10, 2020

[30] Decision Tree Picture
https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 . 2017
Accessed Oct 20, 2020

[31] Measurement of the H→ WW* Branching Ratio at 1.4 TeV using the semileptonic final state at CLIC
A. Winter, N. Watson University of Birmingham, United Kingdom CLICdp-Note-2016-003. 2020
Accessed Dec 12, 2020

Appendix

Name:	**Value:**	**Name::**	**Value:**
boosting	gbdt	*objective*	binary
metric	none	*tree_learner*	serial
device_type	cpu	*data*	
valid		*num_iterations*	100
learning_rate	0.1	*num_leaves*	31
num_threads	15	*max_depth*	-1
min_data_in_leaf	20	*min_sum_hessian_in_leaf*	0.001
bagging_fraction	1	*bagging_freq*	0
bagging_seed	3	*feature_fraction*	1
feature_fraction_seed	2	*early_stopping_round*	0
max_delta_step	0	*lambda_l1*	0
lambda_l2	0	*min_gain_to_split*	0
drop_rate	0.1	*max_drop*	50
skip_drop	0.5	*xgboost_dart_mode*	0
uniform_drop	0	*drop_seed*	4
top_rate	0.2	*other_rate*	0.1
min_data_per_group	100	*max_cat_threshold*	32
cat_l2	10	*cat_smooth*	10
max_cat_to_onehot	4	*top_k*	20
monotone_constraints		*feature_contri*	
forcedsplits_filename		*refit_decay_rate*	0.9
verbosity	-1	*max_bin*	375
min_data_in_bin	3	*bin_construct_sample_cnt*	200000
histogram_pool_size	-1	*data_random_seed*	1
output_model	LightGBM_model.txt	*snapshot_freq*	-1
input_model		*output_result*	LightGBM_predict_result.txt
initscore_filename		*valid_data_initscores*	
pre_partition	0	*enable_bundle*	1
max_conflict_rate	0	*is_enable_sparse*	1
sparse_threshold	0.8	*use_missing*	1
zero_as_missing	0	*two_round*	0
save_binary	0	*enable_load_from_binary_file*	1
header	0	*label_column*	
weight_column		*group_column*	
ignore_column		*categorical_feature*	
predict_raw_score	0	*predict_leaf_index*	0
predict_contrib	0	*num_iteration_predict*	-1
pred_early_stop	0	*pred_early_stop_freq*	10
pred_early_stop_margin	10	*convert_model_language*	
convert_model	gbdt_prediction.cpp	*num_class*	1
is_unbalance	0	*scale_pos_weight*	1
sigmoid	1	*boost_from_average*	1
reg_sqrt	0	*alpha*	0.9
fair_c	1	*poisson_max_delta_step*	0.7
tweedie_variance_power	1.5	*max_position*	20
label_gain		*metric_freq*	1
is_provide_training_metric	0	*eval_at*	
num_machines	1	*local_listen_port*	12400
time_out	120	*machine_list_filename*	
machines	*gpu_platform_id*	-1	
gpu_device_id	-1 *gpu_use_dp*	0	

Table 17: Default lightGBM hyper parameter values for all models used in this project

Egam5 trigger python script link
Link to trigger list.: https://svnweb.cern.ch/trac/atlasoff/browser/PhysicsAnalysis/DerivationFramework/DerivationFrameworkEGamma/trunk/share/EGAM5.py#L23

$HLT\_e60\_lhloose\_xe60noL1||HLT\_e120\_lhloose||HLT\_j80\_xe80||HLT\_xe70$ ('xe' means MET)

.1 z-data initial predictions

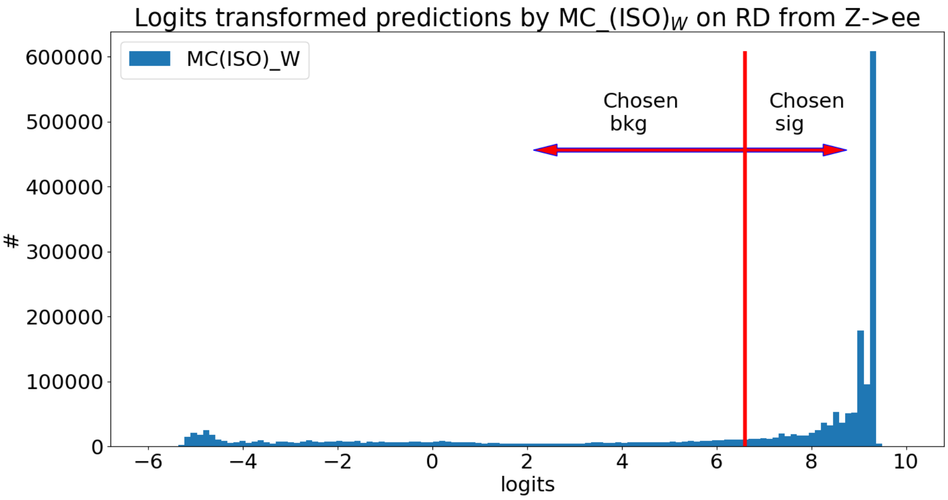


Figure 40: *Predictions from MC(ISO)$_W$ model on data from $Z \rightarrow ee$. The red line represents the initial guess of what separates signal from background*

.2 w-data mc(iso)w75 initial predictions

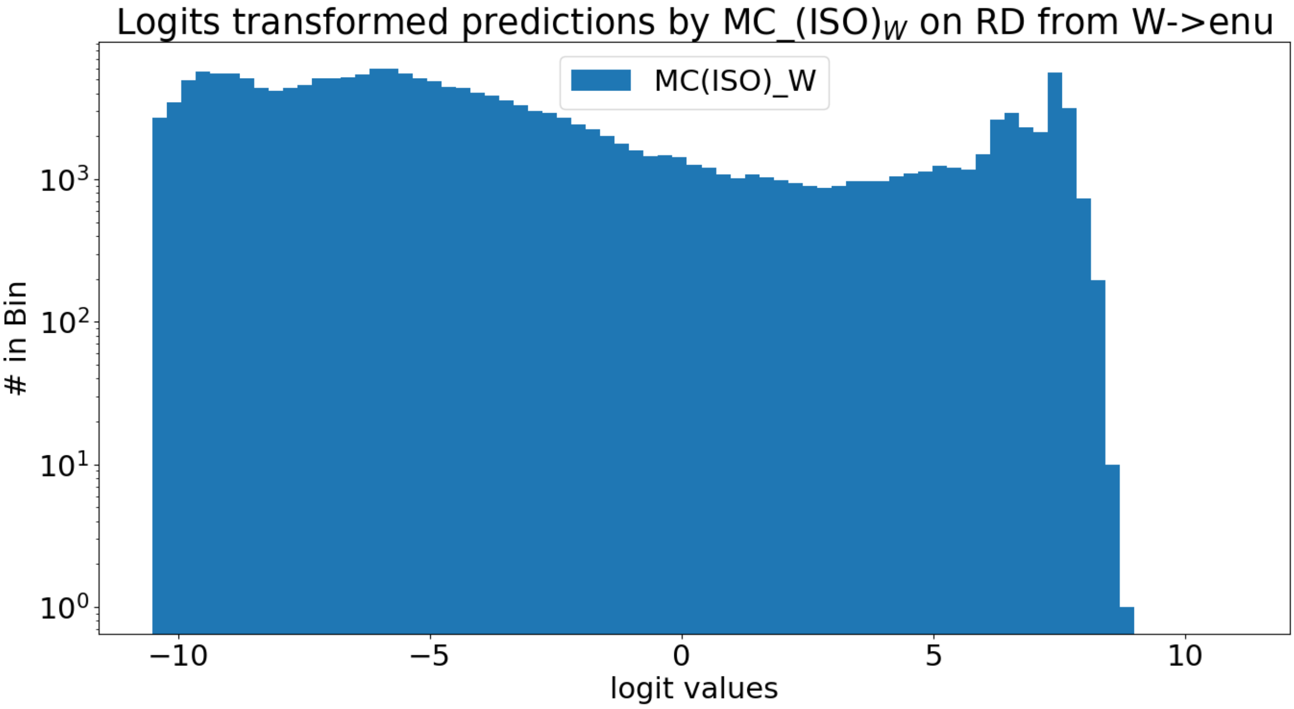


Figure 41: *Initial predictions by MC(ISO)$_{W75}$ on W-data from RD. Used for decorrelation*

.3 VARIABLES DISTRIBUTION IN MC PID

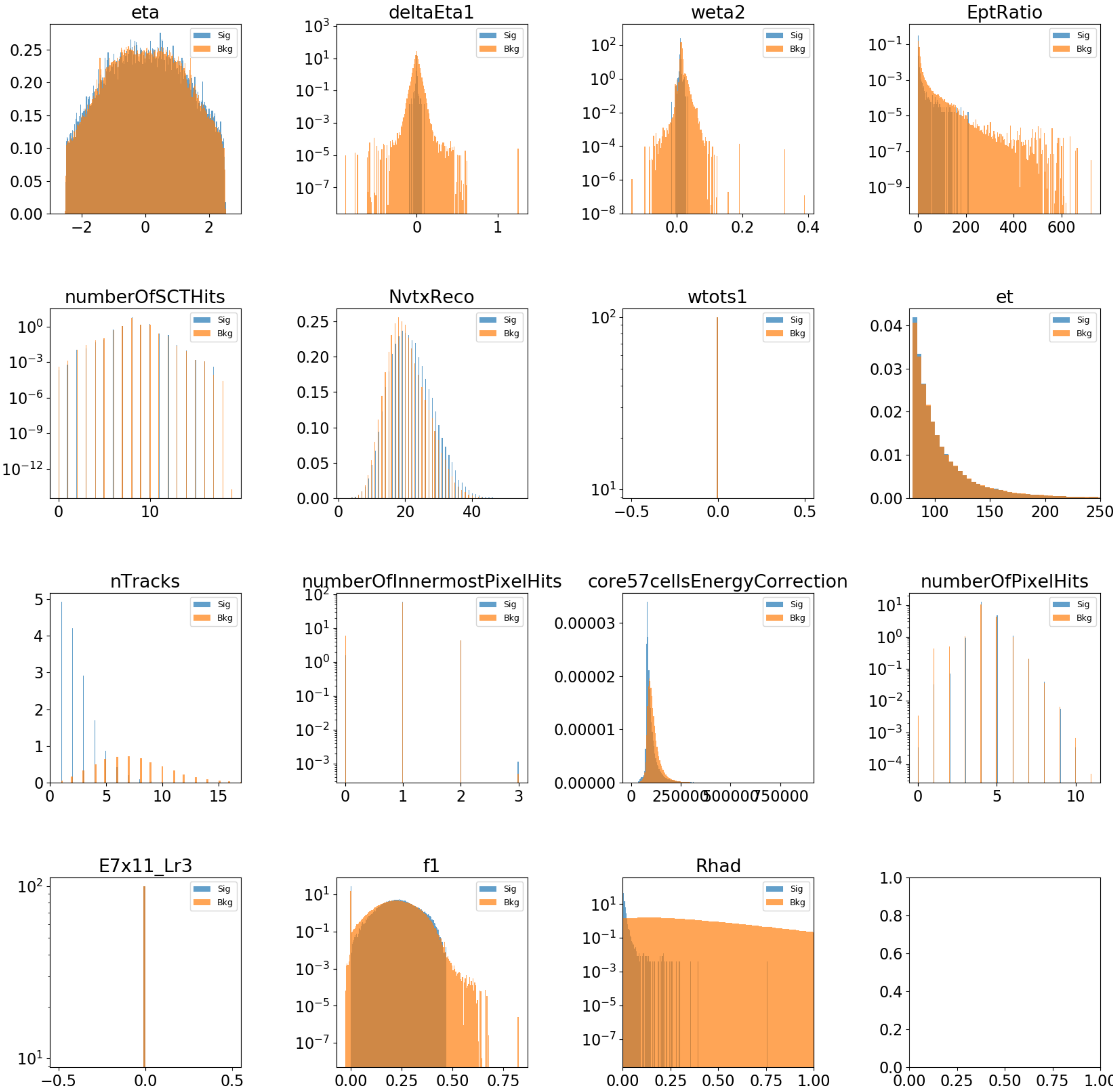


Figure 42: *PID MC Variable distributions*. The last 16 of the 25 variables in PID. The other 9 are in the MC chapter

.4 VARIABLES DISTRIBUTIONS IN RD

.4.1 *W-data*

.4.2 *Z-data*

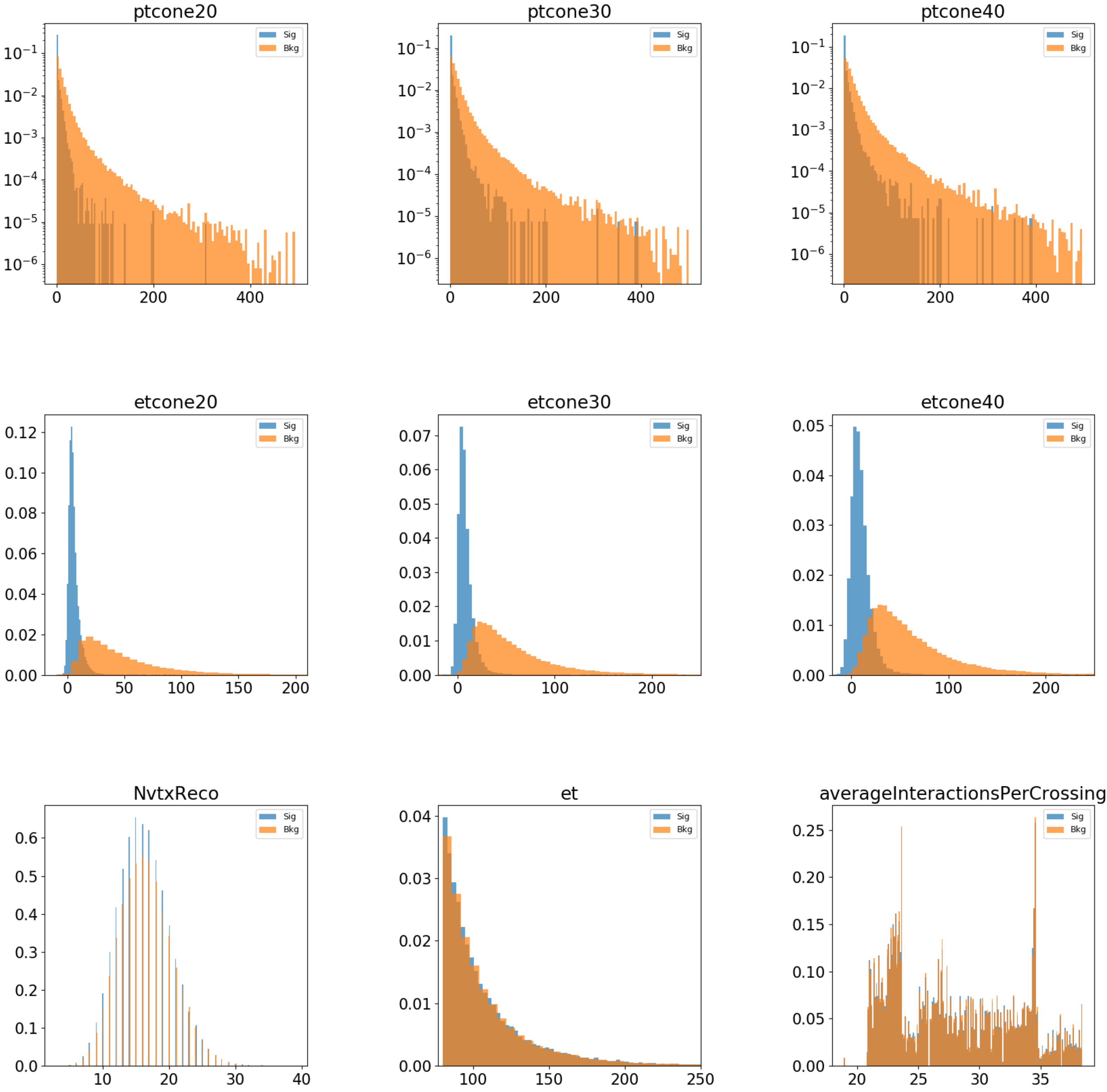


Figure 43: *Isolation variable distributions in W-data with labels made by MC(PID)$_W$*. They are used to train RD(ISO)$_W$ ML-models

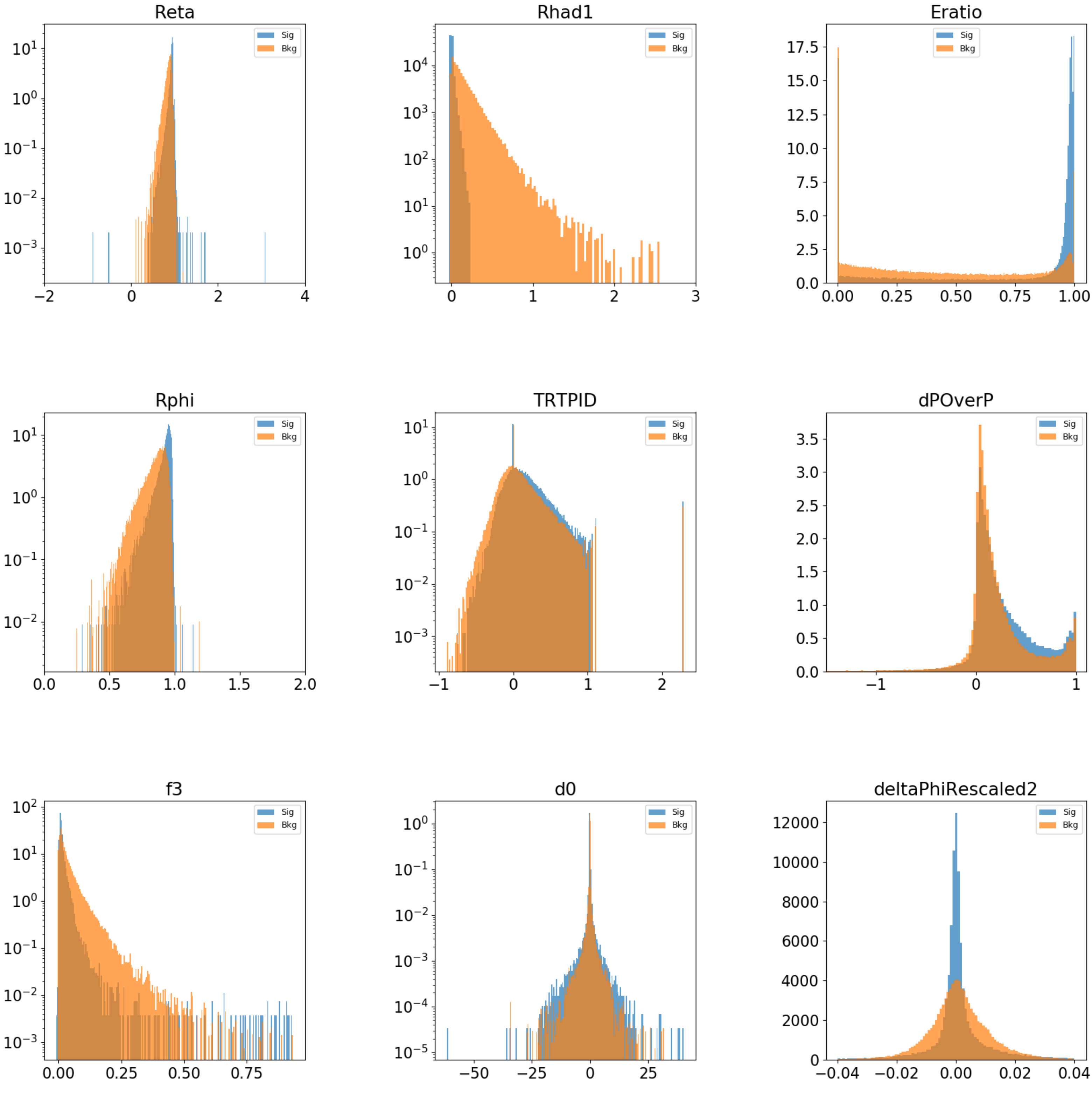


Figure 44: *9 PID variable distributions from W-data with labels made by MC(ISO)$_W$*. They are used to train RD(PID)$_W$ ML-models

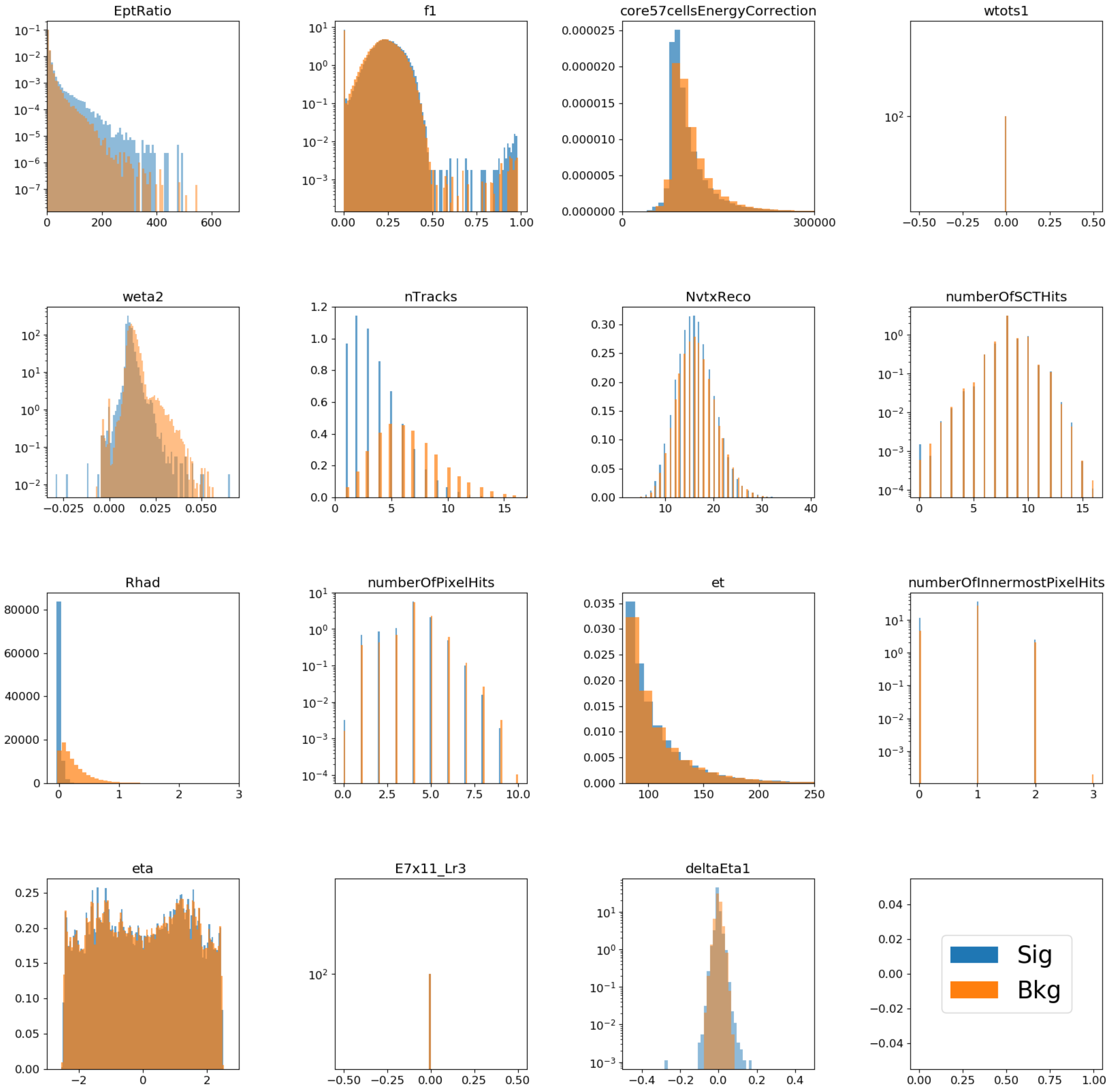


Figure 45: *16 PID variable distributions from W-data with labels made by MC(ISO)$_W$.* They are used to train RD(PID)$_W$ ML-models

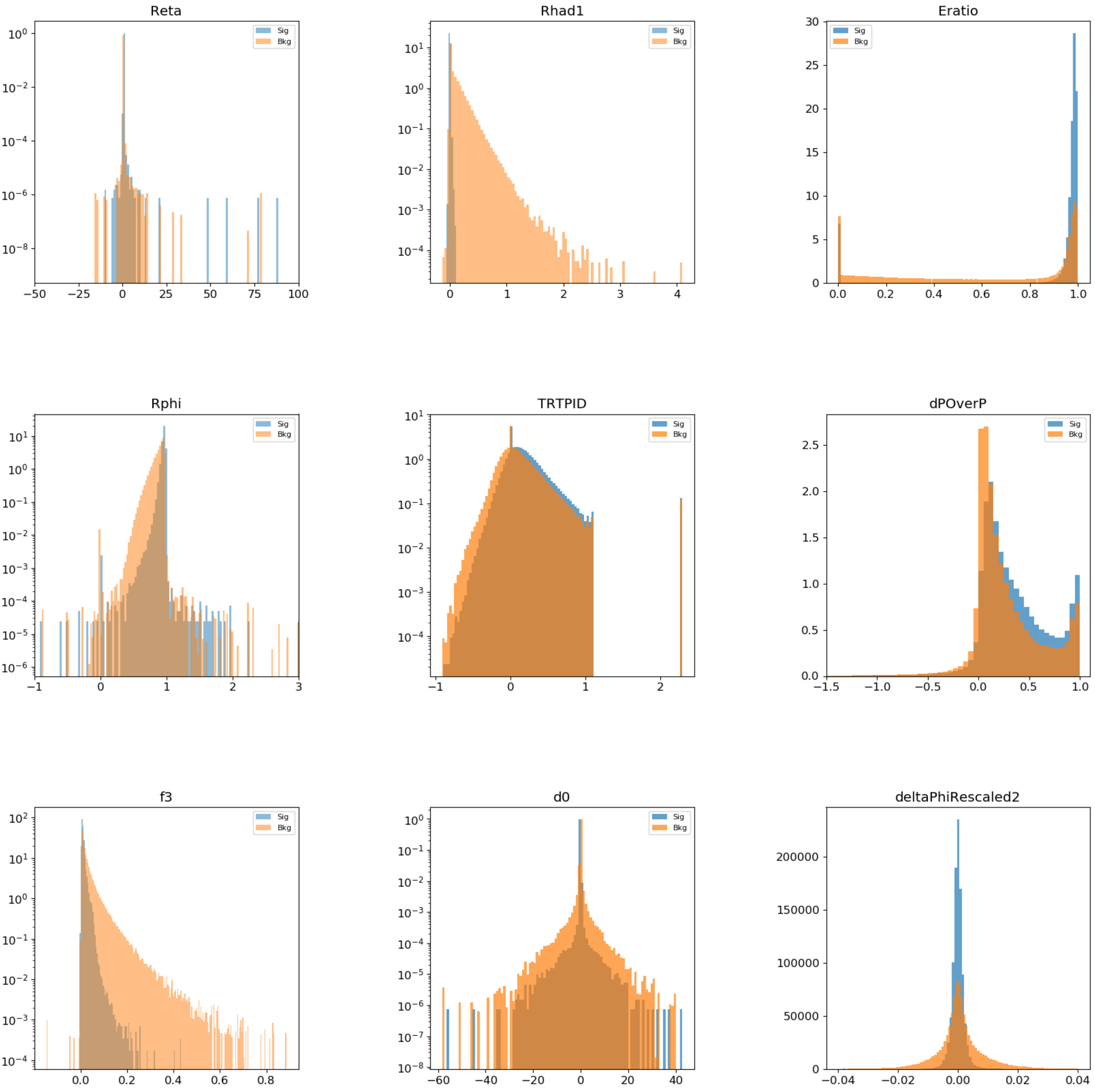


Figure 46: *9 variable distributions from PID in Z-data with labels made by MC(ISO)$_W$*. They are 9 out of 25 variables that are used to train RD(PID)$_Z$ ML-models

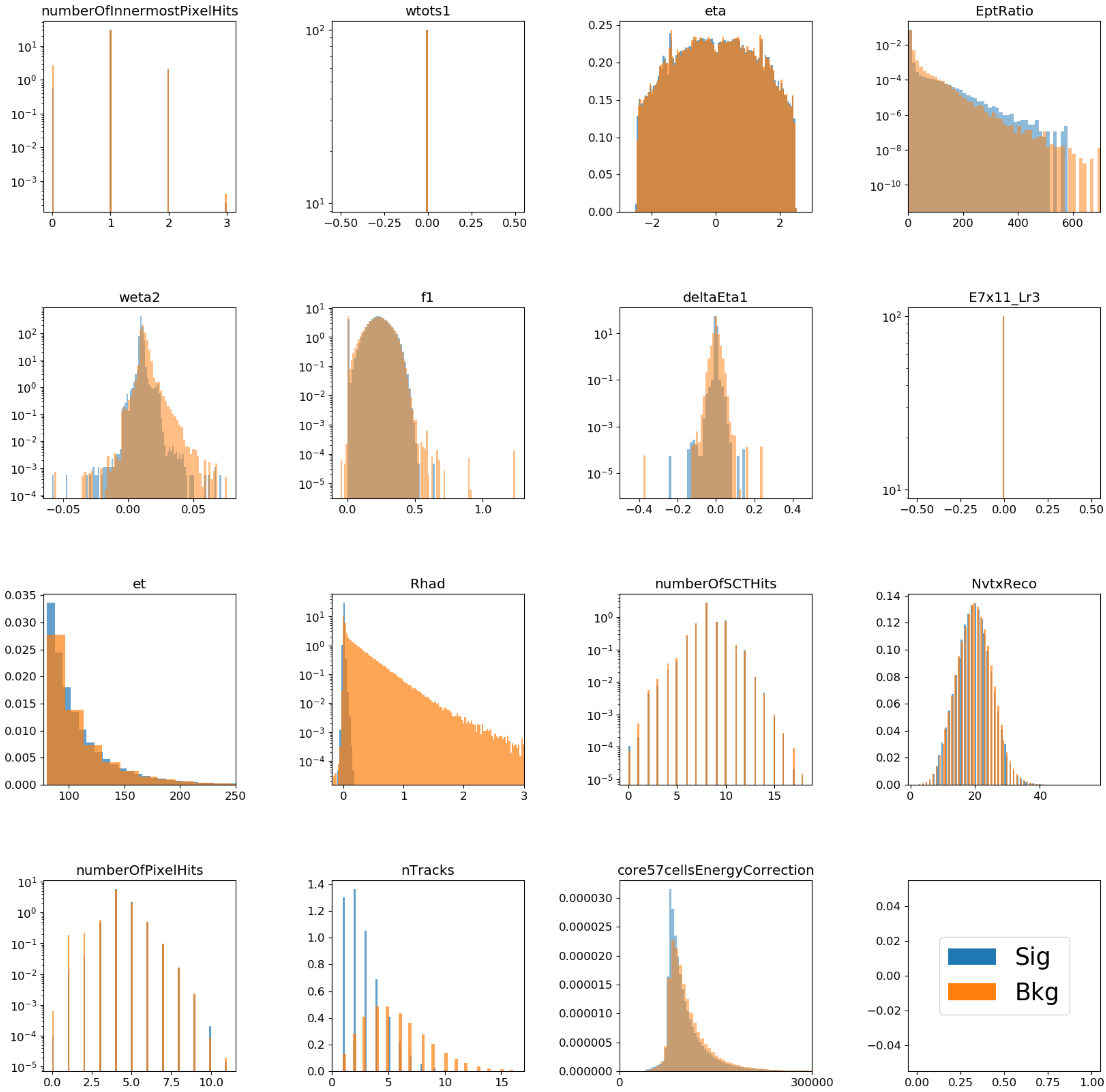


Figure 47: *16 variable distributions from PID in Z-data with labels made by MC(ISO)$_W$*