Electron/Photon Ambiguity Resolution Using Neural networks For ATLAS Experiment

Nutthawara Buatthaisong, Khon Kaen University, Thailand

September 4, 2019

Abstract

It is difficult to separate photons and electrons. The current ambiguity resolver classifies up to 10 percent of photon wrongly at high interaction number per bunch crossing $< \mu >$. To minimize misidentified particles, a supervised neural network is applied to improve the classification of electron and photons. The neural network is trained with pile-up and no pile-up data condition of 2017. Pile-up training shows lower number of wrong categorised objects. From the results, fake efficiency of the electron type decreases to under 1 percent, and fake efficiency of photons can be reduced from up to 10 percent to lower than 4 percent.

Contents

1 Introduction

Photon and electron are important in the ATLAS experiment at Large Hadron Collider (LHC). They participate in several Standard Model interactions, for example, $H \to \gamma\gamma$ from proton - proton collision. Photons and electrons have similar signature in the ATLAS detector due to the conversion of the photon. Both of them have energy deposits or clusters in the Electronmagnetic (EM) Calorimeter, but the electron is defined as the clusters matched to tracks, and photon clusters are matched to a conversion vertex. For high interactions per bunch crossing (μ), up to 10 percent of total photons is identified as an electron. The neural network is applied to the decrease number of misidentified photons, and turn the photons into ambiguous type which means the particle can be both a photon or an electron instead of an exact electron.

2 ATLAS Detector

The ATLAS detector is one of the largest detector located at the Large Hadron Collider at CERN. The detector has symmetric cylindrical shape with length and radius of 44 m and 11 m, respectively as shown in Figure 1. ATLAS detector is a multi purposes detector. The detector makes it possible to study various Standard Model interactions and search hints of Beyond Standard Model physics. The detector composes of stacks of sub-detectors, including four major layers, Inner Detector (ID), Electromagnetic (EM) Calorimeter, Hadronic Calorimeter, and Muon Spectrometer.

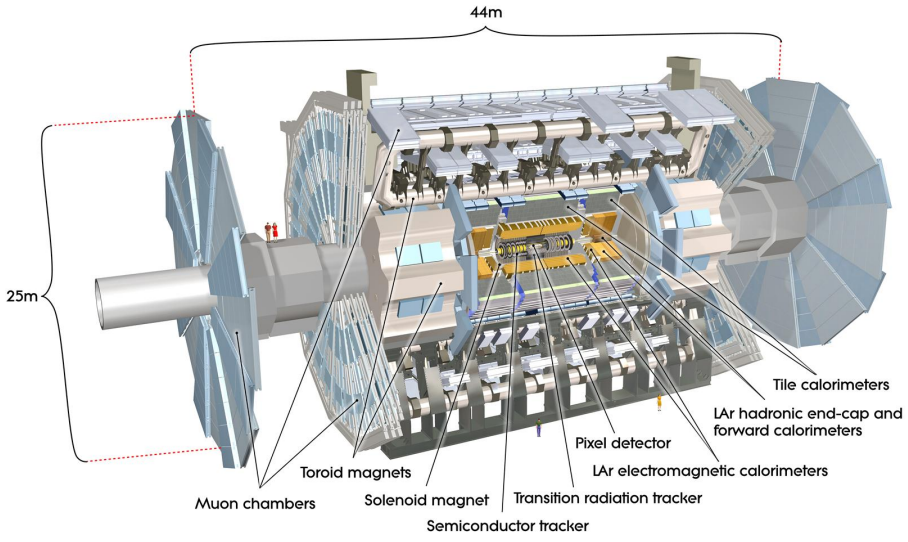

Figure 1: The ATLAS detector schematic diagram [1]

3 Electron and Photon Reconstruction

In 2017, the electron - photon reconstruction algorithm is improved to have variable-sized energy deposits or clusters in the EM calorimeter [2] known as topo-cluster. Electrons in ATLAS detector are defined to have a cluster in the calorimeter and tracks found in the Inner detector. Photons can be categorised to converted and unconverted photons due to photon conversion. A converted photon has a cluster matched with tracks and a conversion vertex. An unconverted photon is a cluster that is not matched to a track or a vertex as shown in Figure 2.

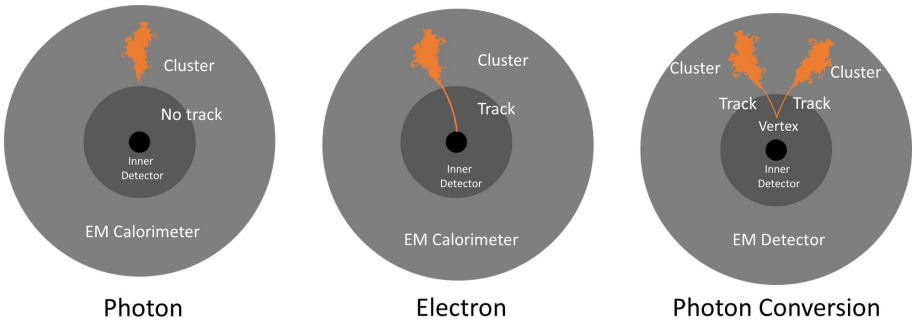

Figure 2: Photon, electron, and photon conversion in ATLAS detector

The algorithm of photon-electron reconstruction is shown in Figure 3. Firstly, the algorithm prepares tracks and clusters. In the EM calorimeter, topo-clusters are selected and reconstructed. The cluster is matched to tracks in Inner detector in case of an electron. For photons, it builds the conversion vertex (or vertices) and then matches the vertex to the cluster. The superclusters, object seed and satellite clusters, of electron and photon are built separately after that. The algorithm seeds photon supercluster and electron superclusters from the topo-cluster, add secondary clusters, and applies the calibrations and corrections to energy and position. The electron tracks and the conversion vertices are matched to superclusters. Then the ambiguity resolver which is a tool to distinguish between photons and electrons is applied, and electrons and photons for analysis are built and calibrated in following step. After that the discriminating variables are calculated and used to separate the objects from the background. Lastly, particle identification will be applied for all objects in the system.
For photon and electron identification during the reconstruction, the ambiguity resolution algorithm is applied [3]. The classification algorithm can divide the objects in the system into 3 types, photon, electron, and ambiguous which is a state that the object can be identified exactly as a photon or electron. The procedure of the identification is shown in Figure 4. First, we have input objects which are then matched to the seed cluster to photon and electron. The photon seed has no track, and the electron seed is matched to the track. Then, the matched objects are checked that their tracks have hits in Silicon layer (Si). For the case that has Si hits, photons must have 2 tracks due to the conversion, no hits in Pixel layer, and conversion tracks. When the objects have one

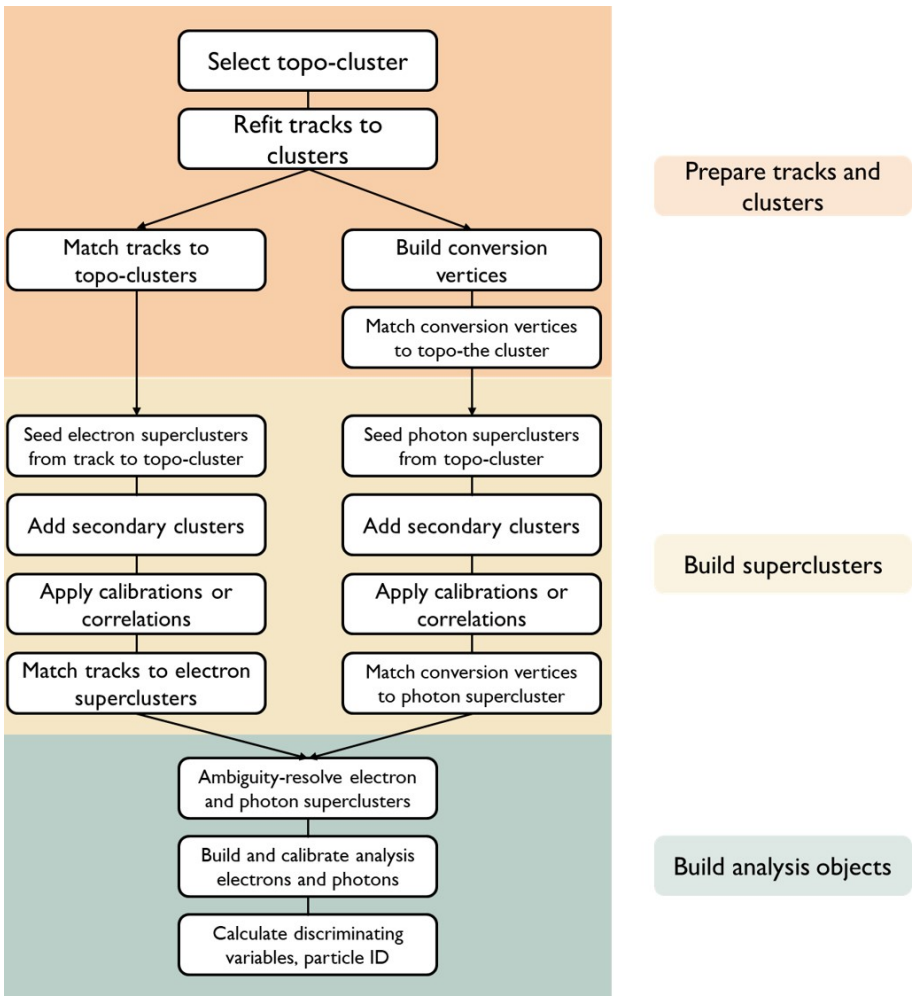

Figure 3: Algorithm of photon and electron reconstruction [3]

of these conditions, the tracks with transverse momentum (P_t) less than 2 GeV, or the ratio of energy in the supercluster to the track momentum (E/P) greater than 10, or no Pixel hits, they will be classified into ambiguous type. The objects that hav a hit in the Pixel nearest to the beam-line along the track will be also classified into ambiguous type. For an electron, there must not be two tracks that are matched to a vertex, the both tracks should not have an innermost hit, and the distance between radial conversion vertex position to the minimal position of a track hit should be less than 40 nm. All of these factors are further used to split to electron, photon, and ambiguous.

4 Neural network

Neural networks or artificial neural networks are a set of algorithm that mimics biological neuron or neural networks. The algorithm is designed to remember the pattern of input

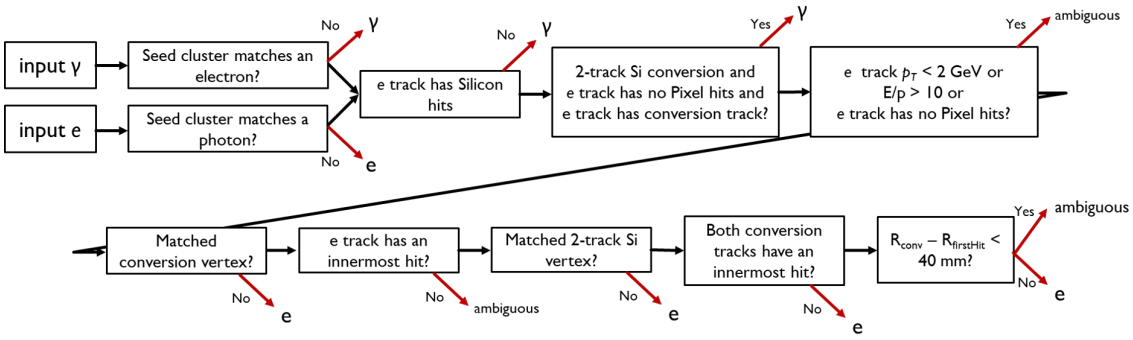

Figure 4: Ambiguity resolution logic of electron and photon in current algorithm [3]

data or create a mathematical model. Neural networks can be used to cluster and classify the data. To cluster the data, the networks will find the similarities among the unlabelled data. This is called unsupervised learning. In other hand, in the process of classification, the neural networks need to be trained by the labelled input data, then it will create the pattern for predicting unlabelled data as shown in Figure 5. This process is also known as supervised learning.

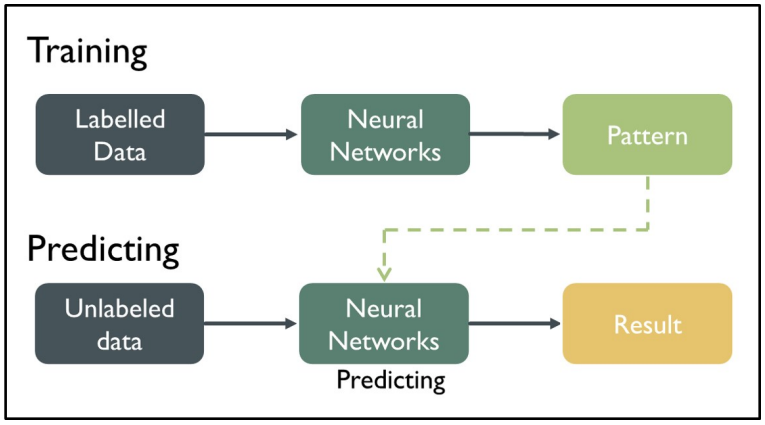

Figure 5: Process of supervised learning in neural networks

Neural network has basically 3 layers, including input layer, hidden layer, and output layer as shown in Figure 6. Input layer composes of neurons. The role of the input layers is to bring the data to the hidden layer for the further processing. Hidden layer (layers) is a layer in between input layer and output layer. This layer can build the model by a set of weight, bias, and activation function. The last layer is the output layer. The role of the output layer is to give the outputs to user.
The simplest architecture of neural network consists of one neuron which is called perceptron as shown in Figure 7. Inputs in this system are x_1 and x_2 which can be real number. The data is weighted by w_1 and w_2, and then brought together for summation

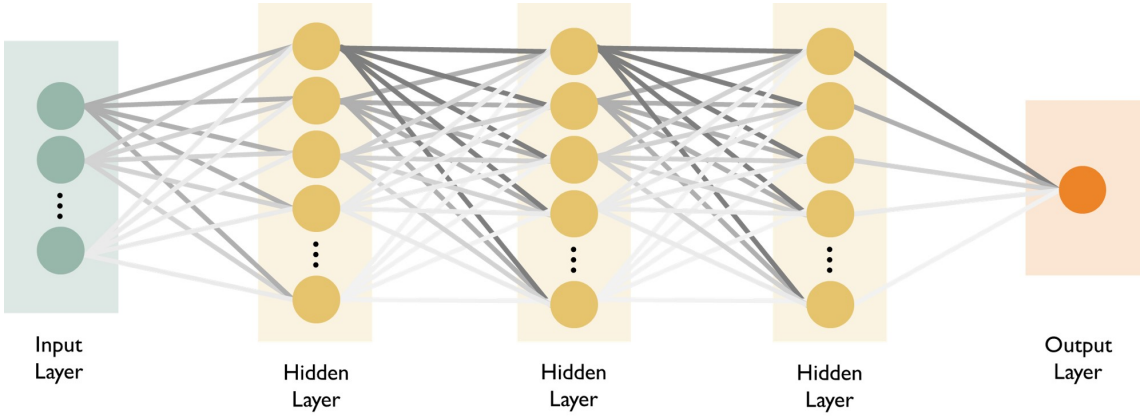

Figure 6: Neural network architecture

(z) with bias (b). The output is calculated from activation function (a) which depends on z. Activation function or transfer function is used to decide when to activate the output node. There are several activation functions which commonly used today, such as sigmoid function and ReLu function. Sigmoid function or logistic function is a non-zero derivative function which can be written as

$$a(z) = \frac{1}{1 - exp(-z)}. \tag{1}$$

However, sigmoid function also causes small or vanished gradient (also called vanishing gradient problem). The effect is that the networks learn slower or refuse to learn. Rectified Linear Unit or ReLu function is non-linear function which can be written as

$$a(z) = max(0, 1). \tag{2}$$

ReLU function gives value only when z is positive, and 0 otherwise. The function benefits for multi-layer neural networks.

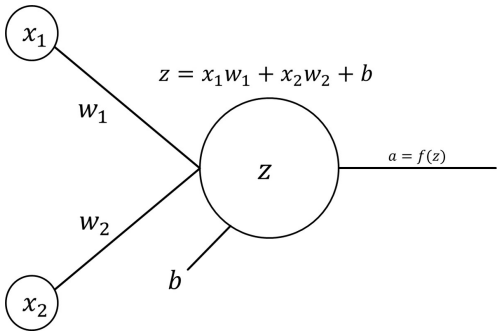

Figure 7: The simplest neuron called perceptron

5 Current Ambiguity Resolution

In the ATLAS detector, photons are able to interact with the detector material and produce positron-electron pairs. The interaction causes the similar signature to electrons in the detector. Moreover, there is noise in the system induced by pile-up. Pile-up means that there are multiple pp interactions in the same bunch crossing or nearby the crossings.
For high mean number of interactions per bunch crossing $< \mu >$, there is an increase of the fraction of misidentified photons and total photons. The fraction is also known as fake fraction or fake efficiency. As shown in Figure 8, the fraction rises up from 4 percent to 9 percent with increasing $< \mu >$, while the electron fake fraction is constant at approximately 2 percent. The higher value of fake fraction implies that there is lower electron-photon identification capability. Aim of this work is to reduce the fake fraction, especially fake fraction of photon by classify the objects into the ambiguous type.

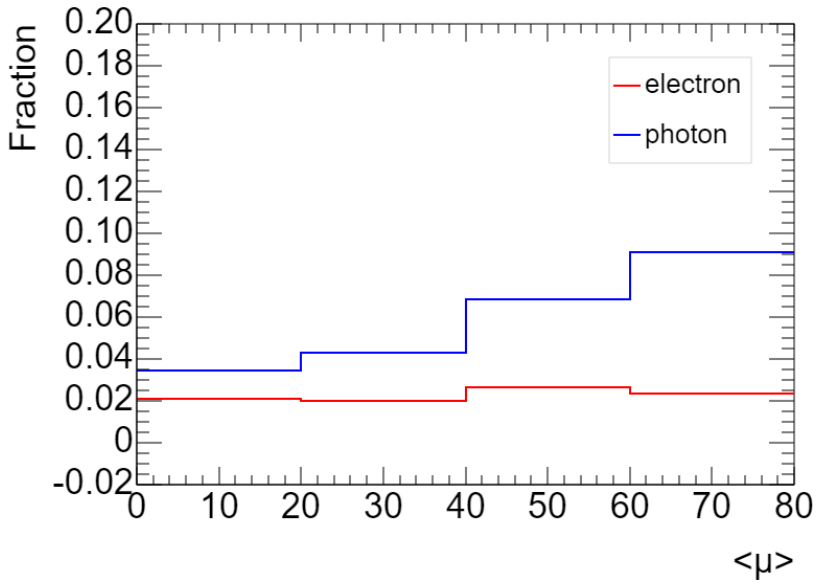

Figure 8: Fake fractions of misidentified photons to total number of photons and misidentified electrons to total number of electrons

6 Neural networks for Electron-Photon Identification

We use neural networks to classify the objects into photons and electrons by using new discriminant variables. The reduced fake efficiencies of electrons and photons are expected to be minimized.

6.1 Neural network Architecture

The photon and electron identification are run parallel in an algorithm with Keras [4]. The algorithm has the Keras Sequential model which composes a linear stack of hidden layers. Core layer of the neural network is Dense layer. Dense layer means there are fully connected nodes between layers. Each hidden layer composes of 64 nodes and its dimension equals to the number of discriminant parameters for the first hidden layer. The last layer is the output layer. Number of output node relies on the number of categories, and in this case is one node. The architecture of the neural network constructed in this work is similar to the neural network shown in Figure 6. For the compiler, we use Binary Crossentropy as a loss model and adam as an optimizer. In preprocessing data step, we scale the data using MinMaxScaler from scikit-learn (sklearn) package [5].

6.2 Discriminant Variables

We use the same discriminant variables applying to the classification neural networks in electron-photon identification are described in Table 1.

Table 1: Discriminant variables used for the classification of the neural networks

Variables	Description
trkPixelHits	Number of hits of the track in Pixel Layer in the Inner detector. Photon tends to have zero hits or lower number than electron
trkSiHits	Number of hits of the track in Silicon Layer in the Inner detector.Photon tends to have zero hits or lower number than electron
trkNhitsInnerMost	The particle has at least a track and innermost hits in the Inner Detector. Most of photons are failed in this conditions.
track_ep	The ratio of the cluster energy to the track momentum.
track_pt	Transverse momentum of the track
dR_track1_track2	The distance between the track and another track
dR_track1_cluster	The distance between the track and the cluster
radius_firstHit	The radial distance of the first hit of the track

For each discriminant variable, we compare the histogram of no pile-up to pile-up one and find only a difference. For example, the normalised plots of 'trkPixelHits' variable show the number of hits in Pixel layers. We find small amounts of photons of pile-up data have more number of hits as shown in Figure 9.

6.3 Training

The neural networks for classification is a supervised learning network. The algorithm is trained with labelled data which is a single particle gun Monte Carlo sample with and

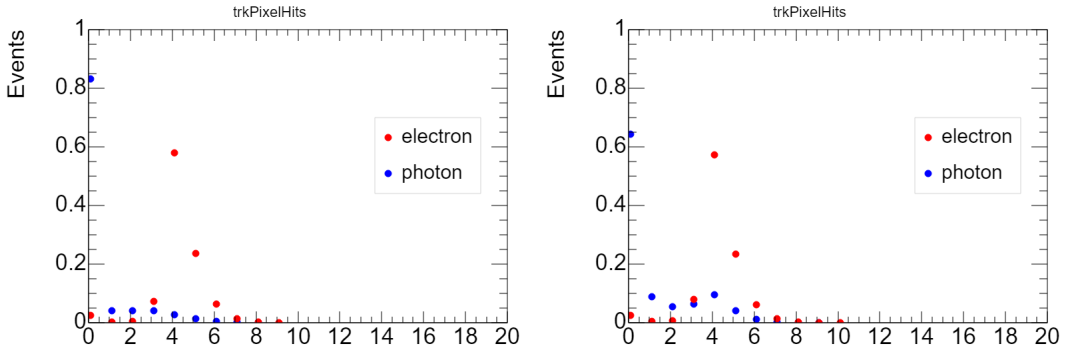

Figure 9: Histograms of 'trkPixelHits' variable shows the number of Pixel hits of no pile-up (left) and pile-up (right).

without pile-up condition of 2017 [3]. In this work, we construct 2 sets of experiments which are trained with zero-μ without pile-up and non-zero-μ with pile-up labelled data.

- No pile-up cluster ($\mu = 0$) The data set is reconstructed under the condition of low-μ or zero-μ. Being trained by this data set, the algorithm will learn from a simpler case.
- Pile-up cluster with $0 < \mu \leq 80$ The train data set reconstructed under the same condition as the test set is more complicated than no pile-up.

Furthermore, we split the data into feature set which contains the discriminant variable values and target set containing the target score of the trained objects.

7 Results

7.1 ROC Curve

Receiver operating characteristic (ROC curve) implies the classification quality of the neural network. Higher quality of classification will show the constant ratio of background rejection and signal efficiency around 1.0, and suddenly dropped down at signal efficiency equals to 1.0. From Figure 10, we find that in case that the algorithm is trained using pile-up set gives higher background rejection value for high signal efficiency while the algorithm trained with no pile-up set drops down.

7.2 Overtraining Test

The neural network role is to find the mathematical or statistical model to fit the data. Overtraining or overfitting problem is that the neural network adapts the model to fit closely or perfectly to data included noise. We monitor the overtraining problem by plot the event distribution for train data set compared to test set. From Figure 11, there is

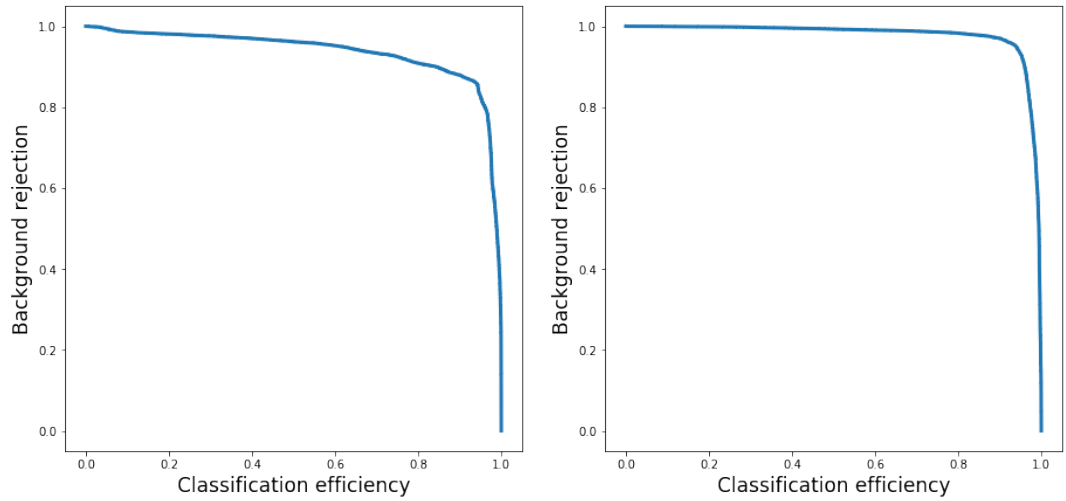

Figure 10: ROC curve plots show comparison between ratio of background rejection to signal efficiency of no pile-up (left) and pile-up (right).

no overtraining problem. However, the pile-up plot shows better fitting compared to no pile-up one.

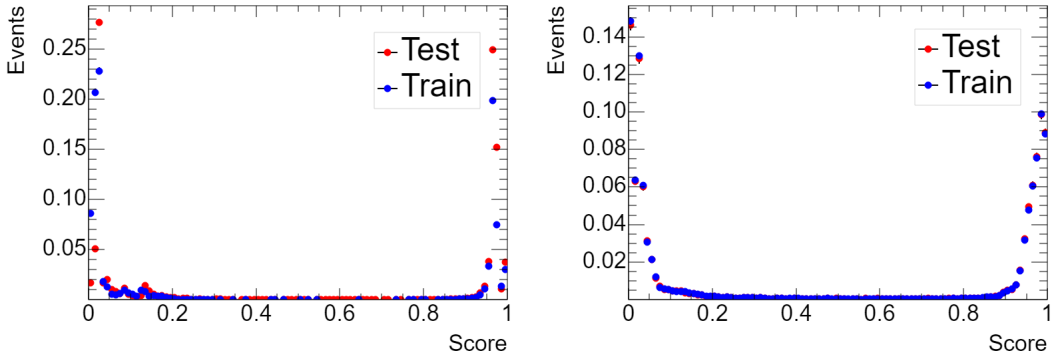

Figure 11: Overtraining plots of no pile-up (left) and pile-up (right)

7.3 Event Distribution

The algorithm is given that score equals to 0 for photon and 1 for electron as labelled in target set. As illustrated in Figure 12, it is almost the same distributions of the predicted score for both cases. We can apply the electron and photon cut off to eliminate some misidentified particles and turn it into ambiguous type.

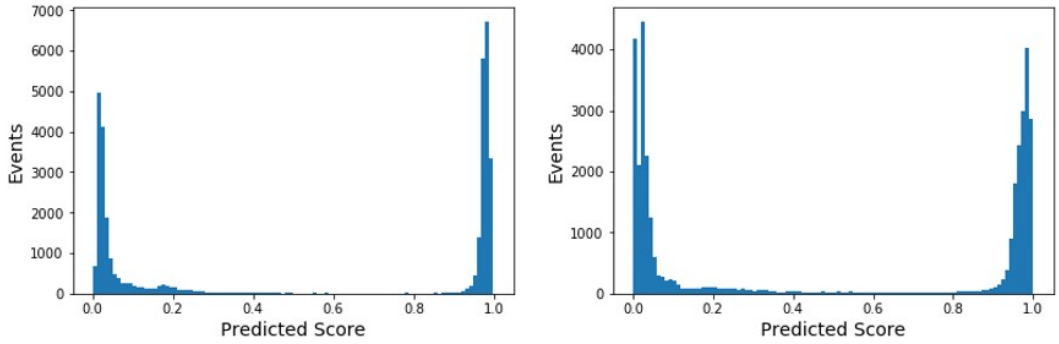

Figure 12: The distribution of event predicted scores range from 0 (photon) to 1 (electron) of no pile-up (left) and pile-up (right) data sets

7.4 Identification Efficiency

The efficiency of the algorithm is defined by the fraction of correct identified photons and total truth photons and correct identified electron and total truth electrons. On the other hands, the efficiency also can be calculated from misidentified objects in the system as known as fake efficiency. For electrons, the fake efficiency can be written as

$$\text{fake eff} = \frac{\#\text{ of e identified as } \gamma}{\#\text{ of total e}}, \tag{3}$$

and for photons, the fake efficiency can be similarly written as

$$\text{fake eff} = \frac{\#\text{ of } \gamma \text{ identified as e}}{\#\text{ of total } \gamma}. \tag{4}$$

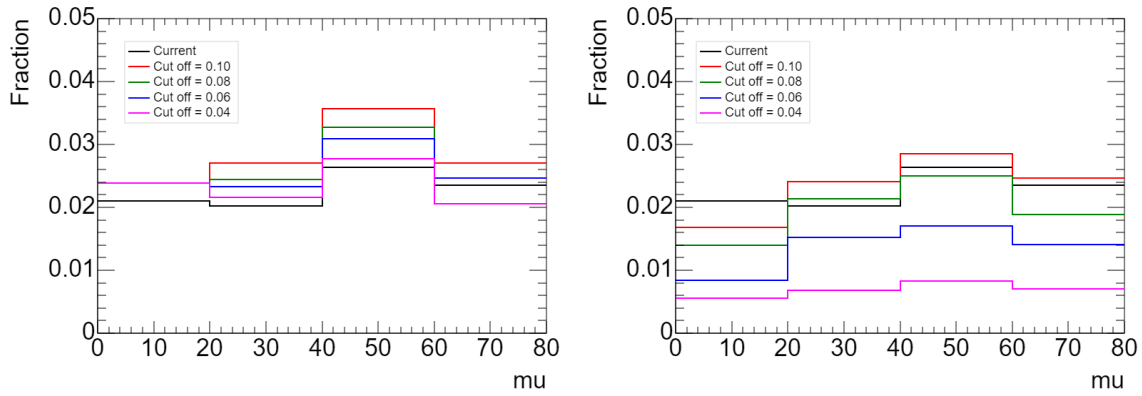

Figure 13: Fake fraction of electrons trained with no pile-up (left) and pile-up (right) which are identified as a photon

Photon cut offs are applied to classify the objects into a photon. We count the number of electrons that are identified as photon under the cut offs. Figure 13 shows the ratio of misidentified electrons by different cut off to the total number of electrons. Black line is the efficiency of current ambiguity resolver, and other lines are the efficiency after identified by the network and applied the different cut off. From the results, the lower cut off for photons can decrease the number of misidentified electrons. We find the difference between trained with no pile-up and pile-up results. For the same value of cut off, trained with pile-up algorithm is seemed to give lower fake efficiency which means less number of misidentified electrons.

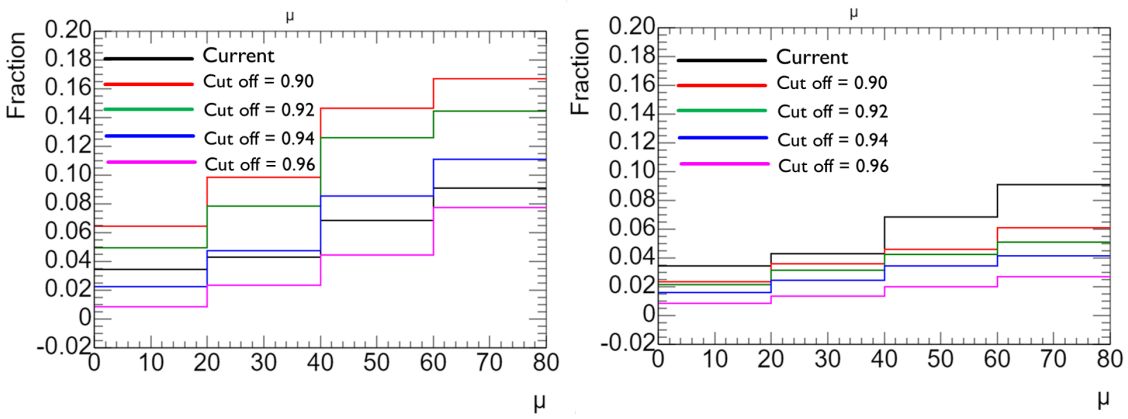

Figure 14: Fake fraction of photons trained with no pile-up (left) and pile-up (right) which identified as a electron

We also apply several values of electron cut off to see the number of photons contaminated in electron zone. From Figure 14, the histogram shows the value of photon fake fraction or efficiency compared to μ. For high-μ, there are more photons mislabeled. We find the trend that for higher cut off applied it will be less misidentified photons. In the case of training with no pile-up, the results will be better than the current only when applied high cut off and significantly higher compared to pile-up training.

8 Conclusion

Due to the difficulty to separate photon and electron, the neural networks seem to improve the current ambiguity resolver. We find the more number of misidentified photons and electrons at higher number of interactions per bunch crossing. Trained with pile-up data algorithm shows better results. For the same cut off, pile-up training still shows lower fake efficiency which means lower objects that are identified as another type. We can classify those assigned to the wrong category into ambiguous type particles, therefore the number of particles in the system still the same.

References

[1] Georges Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003. 437 p, 2008. Also published by CERN Geneva in 2010.

[2] Morad Aaboud et al. Measurement of the photon identification efficiencies with the ATLAS detector using LHC Run 2 data collected in 2015 and 2016. *Eur. Phys. J.*, C79(3):205, 2019.

[3] Georges Aad et al. Electron and photon performance measurements with the ATLAS detector using the 2015-2017 LHC proton-proton collision data. 2019.

[4] Karlijn Willems. Keras tutorial: Deep learning in python, Feb 2019.

[5] F. Pedregosa and other. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.