D0 Note 5245

September 2006

Artificial neural network for photon selection.

Dmitry Bandurin

Abstract

Artificial neural network for selection of photon candidates in the Central and End calorimeter region is described in this note. We test the network on the data and Monte Carlo $Z \rightarrow ee$ events. The built neural network is applied to calculate the direct photon purities for some selected cuts on the neural network output value. In conclusion we show location of the built network and refer to the code that demonstrate the network application.

1 Introduction.

When discriminating between photons and background particles (π^0 as well as the neutral decay channels of η and K_s^0 mesons) we face a typical (for high energy physics) pattern recognition problem. The standard procedure for solving such a problem is the introduction of relevant cuts in the multi-dimensional data. Nowadays the application of a software-implemented artificial neural network (ANN) for pattern recognition is well known and usually gives the results that are superior to conventional approaches [3] – [7].

There is a wide spectrum of photon selection criteria that have been extensively studied by Photon ID group for the current p17 release. They are described in detail in this DØ note [1]. One of the purposes of this note is optimization of their application by building ANN that, on the one hand, would allow one to specify a criterion for increasing the photon purity and, on the other hand, to determine a photon fraction (purity) in the selected sample.

For this aim we are going to keep ideology of paper [2] and split the set of all available variables into two subsets: the first subset to be applied for the photon preselection and the second one for ANN training.

2 General description of ANN.

ANNs are often used to optimize a classification (or pattern recognition) procedure and was applied to many pattern recognition problems in high energy physics with a notable success. They usually have more input than output nodes and thus may be viewed as performing dimensionality reduction of input data set.

The ANN approach is a technique which assigns objects to various classes. These objects can be different data types, such as a signal and a background in our case. Each data type is assigned to a class which in the context of the given paper is 0 (zero) for the background (QCD events) and 1 (unity) for the signal (direct photons). Discrimination is achieved by looking at the class to which the data belongs. The technique fully exploits the correlation among different variables and provides a discriminating boundary between the signal and the background.

ANNs have an ability to learn, remember and create relationships amongst the data. There are many different types of ANN but the feed forward types are most popular in the high energy physics. Feed forward implies that information can only flow in one direction and the output directly determines the probability that an event characterized by some input pattern vector $X(x_1, x_2, ...x_n)$ is from the signal class.

2.1 Mathematical model of the neural network.

The mathematical model of the Neural Network (NN) reflects three basic functions of a biological neuron:

- sum up all the information arriving at inputs of the node/neuron;
- if sum is greater than some threshold, fire neuron;
- after firing, return to the initial state and send a signal to each of the neighboring neurons in the network.

The neuron with these characteristics is known as an elementary perceptron. The perceptron is a simple feed forward system with several input connections and a single output connection.

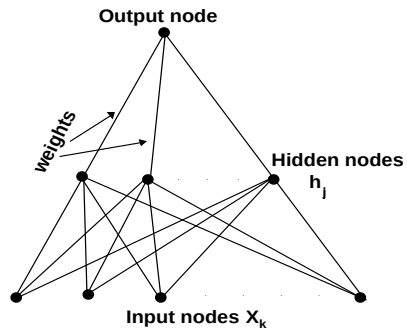

Fig. 1. Neural network with one layer of hidden units.

Mathematically the output can be written as

$$O(x_1, x_2, ...x_n) = g(\frac{1}{T}\sum_i \omega_i x_i + \theta). \tag{1}$$

Here g is a non-linear transfer function and typically takes the following form (sigmoid function)

$$g = \frac{1}{1 + e^{-2x}}, \tag{2}$$

$(x_1, x_2, ...x_n)$ is the input pattern vector, O is the output, ω_i and θ are independent parameters called weights (which connect the input nodes to the output node) and a threshold of the output node. Parameter $\beta = 1/T$ is called inverse temperature and defines the slope of g.

The pattern vector x_i is multiplied by the connection weights ω_i so that each piece of information appears at the perceptron as $\omega_i x_i$. Then the perceptron sums all the incoming information to give $\sum \omega_i x_i$ and applies the transfer function g to give the output (see (1)).

In a feed forward NN a set of neurons has a layered structure. Figure 2.1 shows the feed forward the NN with one hidden layer that is used here. In this case the output of NN is

$$O(x_1, x_2, ...x_n) = g(\frac{1}{T}\omega_j \sum_k g(\frac{1}{T}\sum_k \omega_{jk} x_k + \theta_j) + \theta), \tag{3}$$

where ω_{jk} is the weight connecting the input node k to the hidden node j and ω_j's connect the hidden nodes to the output node. θ_j and θ are the thresholds of the hidden and the output node respectively.

2.2 Learning of the perceptron.

The behavior of a perceptron is determined by independent parameters known as weights and thresholds. The total number of independent parameters in a neural network with a single layer is given

by:

$$N_{ind} = (N_{in} + N_{on}) \cdot N_{hn} + N_{ht} + N_{ot} \tag{4}$$

where N_{in} is a number of input nodes, N_{on} is a number of output nodes, N_{hn} is a number of nodes in a hidden single layer, N_{ht} is a number of thresholds in a hidden single layer, N_{ot} is a number of output thresholds.

Learning is the process of adjusting these N_{ind} parameters. During learning every perceptron is shown examples of what it must learn to interpret. It is fulfilled on the training set consisting of two parts: training data (a collection of input patterns to the perceptron) and a training target, which is a desired output of each pattern.

Mathematically, the goal of training is to minimize a measure of the error. The mean squared error function E averaged over the training sample is defined by equation (5)

$$E = \frac{1}{2N_p} \sum_{p=1}^{N_p} \sum_{i=1}^{N} (O_i^{(p)} - t_i^{(p)})^2, \tag{5}$$

where O_i is the output of the ith node of the NN in equation (3); t_i is the training target (in our case, 0 for the background and 1 for the signal); N_p is the number of patterns (events) in the training sample; N is the number of network outputs ($N = 1$ for our case).

There are several algorithms for error minimization and weight updating. Most popular are **Back propagation**, **Langevin** and **Manhattan** methods. In the last one the weight is updated during the learning by the following rule [1)]:

$$\omega_{t+1} = \omega_t + \Delta\omega \tag{6}$$

$$\Delta\omega = -\eta \cdot sgn[\partial E / \partial \omega] \tag{7}$$

where ω is the vector of weights and thresholds used in the network; t $(t+1)$ refers to the previous (current) training cycle and η is the learning rate which is decreased in the learning process.

3 MC and data samples.

In order to train the neural network we have used direct photons as signal and electromagnetic (EM) jets as background events (also called as just *em-jets*). The photon candidates in signal and background samples are taken from the photon+jet and the QCD events preselected (in order to have high final statistics) at the particle level [2], both generated in p17.09 version.

In our study we have also used real data, selected from "1EMloose" skim provided by Common Sample group. The data events were selected to satisfy a set of data quality criteria: just runs without

[1)] see [4] for a more complete description

calorimeter and CFT problems were considered, many early runs (up to 152649) are excluded. There should be fired at least one of the following EM triggers: EM_HI_SH, EM_HI, EM_MX_SH, EM_MX, EM_HI_2EM5_SH, E1_SHT20, E2_SHT20, E3_SHT20, E1_SH30, E1_L50, E1_SHT22, E2_SHT22, E3_SHT22, E1_SH35, E1_SHT25.

Before training neural network, we have preselected signal and background events by the following criteria. Photon candidates were formed from electromagnetic clusters of calorimeter cells within a cone of radius $\mathcal{R} = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} = 0.2$ with simple cone algorithm. To reject events with energy measurements biased by calorimeter module boundaries and structures, photon candidates are required to be in η and ϕ fiducial regions. Candidates were selected if there was significant energy fraction in the EM calorimeter layers (EMfrac> 0.96), and the probability to have a spatially matched track was less than 0.1%, and they satisfied the isolation requirement $Iso \equiv (E_{total}(0.4) - E_{EM}(0.2))/E_{EM}(0.2) < 0.07$, where $E_{total}(0.4)$ is the total energy in a cone with $\mathcal{R} = 0.4$ and $E_{EM}(0.2)$ is the EM energy within $\mathcal{R} = 0.2$. In CC region we have also limited from the top the energy weighted cluster phi-width in the finely-segmented EM3 layer by $Sigrphi < 14\ cm^2$, while in EC we have used parametrization $Sigrphi < 2.74\eta^2 - 16.3|\eta| + 25.0$ as well as $Sigrz < 5.96\eta^2 - 30.6|\eta| + 40.7$. For the CC region we have also used variables based on the central preshower detector (CPS): energy (E) weighted widths of CPS cluster $ERMS < 0.003$ [1].

In addition, we have required the event vertex to be within 50 cm of the nominal center of the detector along z-axis and should have at least 3 associated tracks. We reject events having too large missing E_T by the cut $E_T^{miss}/p_T^{\gamma} < 0.65$. Just events having at least one hadronic jet are selected.

The following four variables were used for ANN training: scalar sum of transverse momenta of the tracks that are within $0.05 < \mathcal{R} < 0.4$ (HC04_PT), the number of EM1 cells with energy greater than 300 MeV that belong to the EM cluster, fraction of EM cluster energy deposited at EM1 layer (again calculated for cells with $E > 300$ MeV), the scalar sum of the transverse momenta of tracks within $0.05 < \mathcal{R} < 0.4$. We have also considered possibility of building network with a larger track isolation cone of $0.05 < \mathcal{R} < 0.7$ (HC07_PT). The photon purity derived with such a network is about 10% better than with hollow cone $0.05 < \mathcal{R} < 0.4$ (Fig.16) while selection effciency drops by just 2%.

To verify the MC/data agreement with respect to these variables we have tested them on the $Z \to ee$ events. The normalized distributions of those variables for electrons from the $Z \to ee$ events in MC and data are shown in Figs. 2 and 3. Electrons from Z^0 decay are required to be within $|\eta_{det}| < 1.0$ (CC) or $1.5 < |\eta_{det}| < 2.5$ (EC), within Z invariant mass window $|M_{inv} - M_Z| < 7$ GeV, and with $25 < p_T^e < 80$ GeV.

Normalized distributions of the number of signal and background events over same variables are shown in Figs. 4 and 5 for CC and EC regions, respectively.

Instead of direct application of cuts on these three additional variables they were used to build ANN that can accumulate a power of all the variables and criteria on them. ANN is then can be applied for

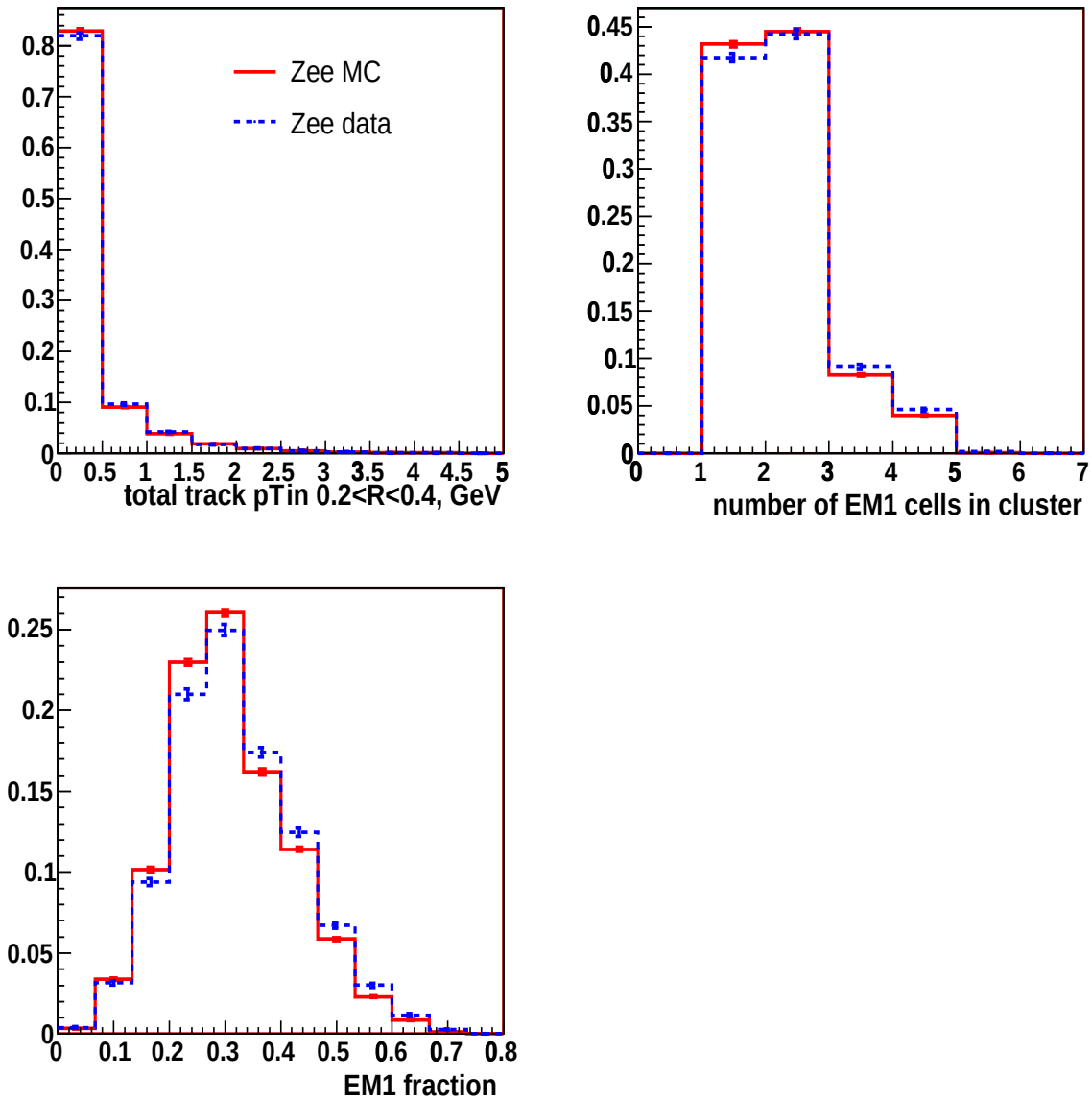

Figure 2: Normalized distributions of the number of events over three variables for $Z \to ee$ electrons from in MC (red full line) and data (blue dotted line) events in CC region ($|\eta_{det}| < 1.0$).

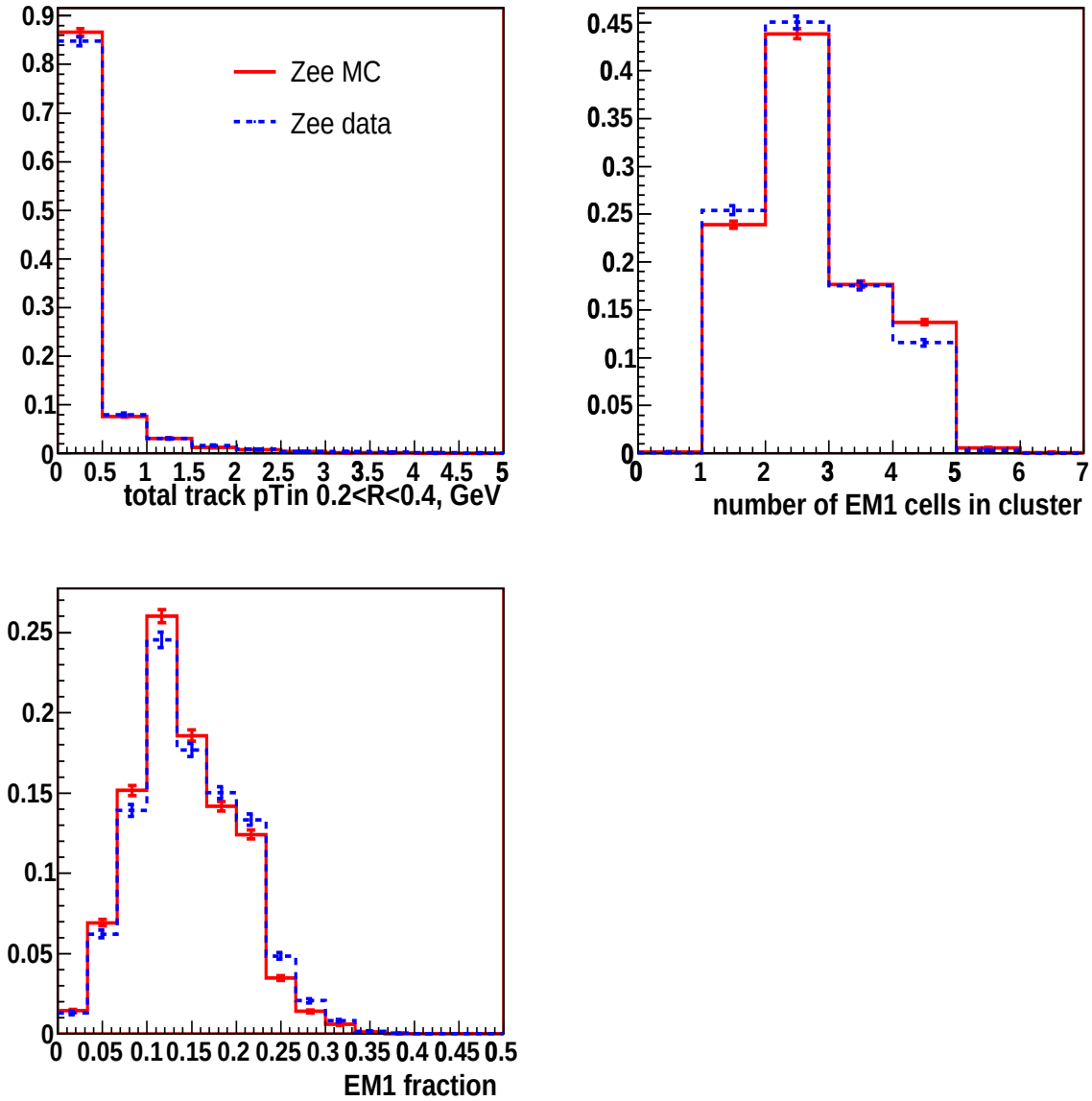

Figure 3: Normalized distributions of the number of events over three variables for $Z \to ee$ electrons from in MC (red full line) and data (blue dotted line) events in EC region ($1.5 < |\eta_{det}| < 2.5$).

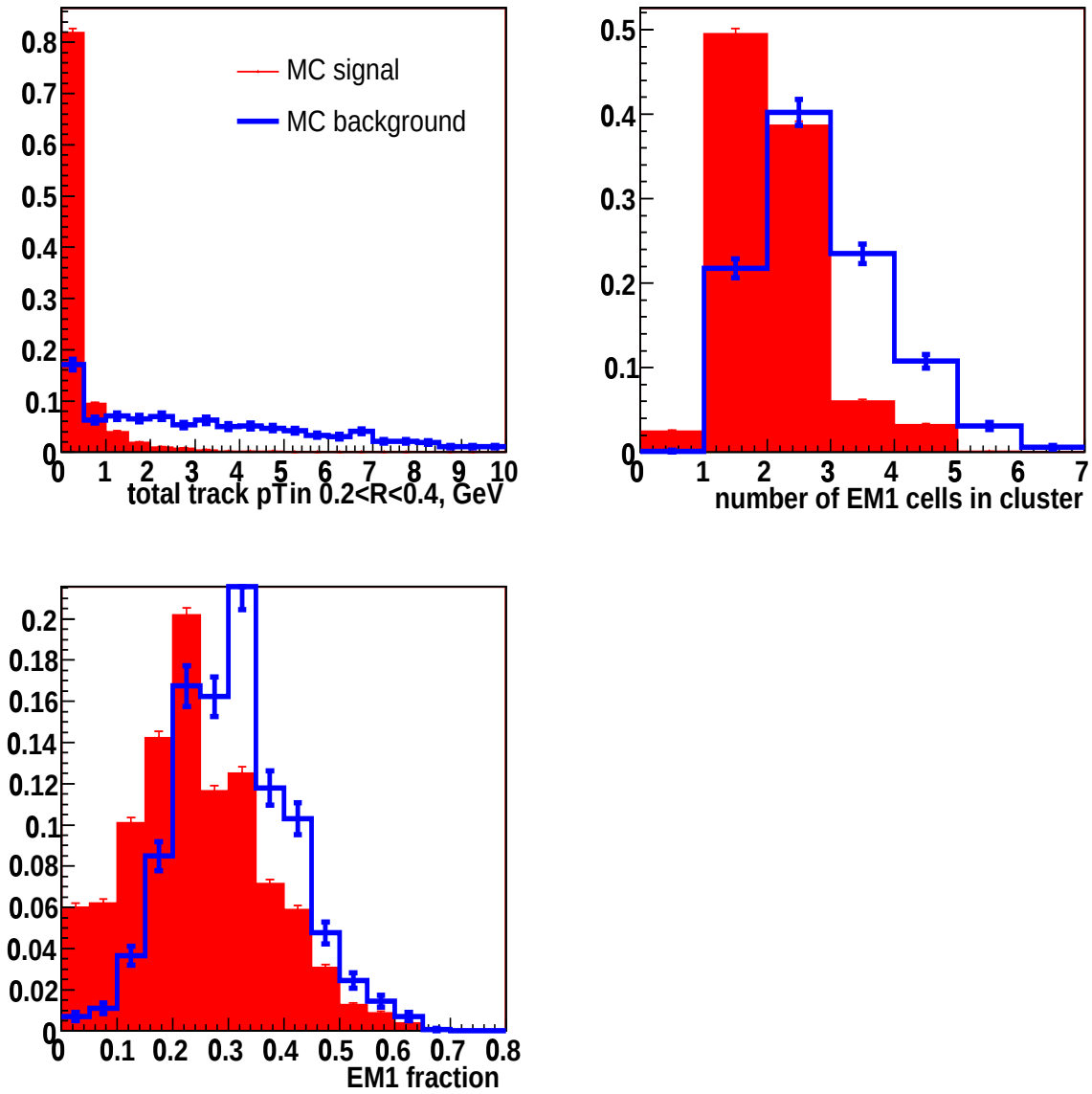

Figure 4: Normalized distributions of the number of events over three variables for signal (red filled histogram) and background MC events (blue full line) events in CC region ($|\eta_{det}| < 1.0$).

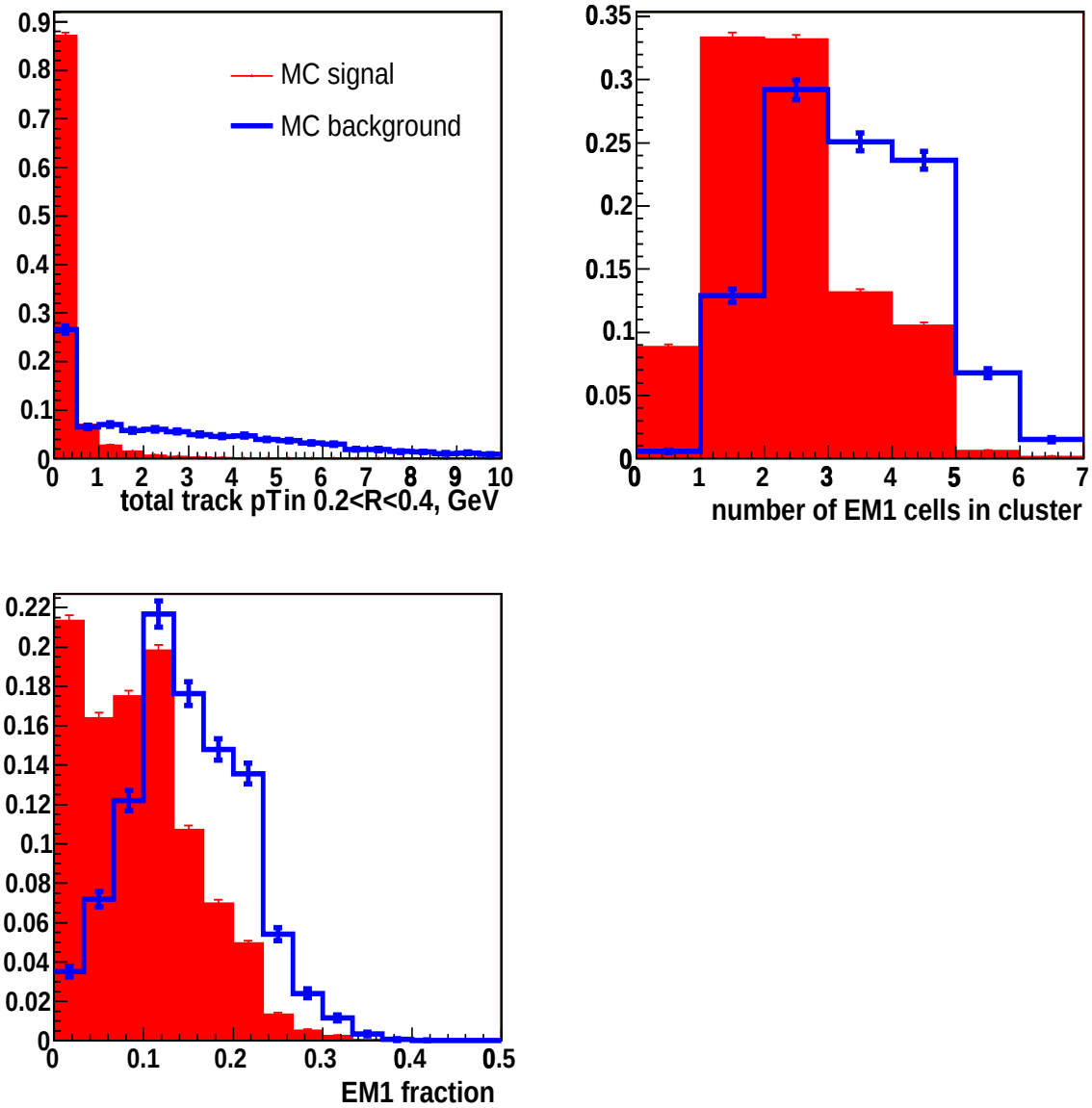

Figure 5: Normalized distributions of the number of events over three variables for signal (red filled histogram) and background MC events (blue full line) events in EC region (1.5¡$|\eta_{det}| < 2.5$).

an additional selection criterion and calculation of photon purity.

4 ANN training and testing.

There a lot of software packages that reflect ANN ideology and have been applied for many physical applications. We have chosen among them for building ANN the JETNET package [8] of version 3.5 with ROOT interface [8].

The Manhattan algorithm [9] for weight updating was used at the training stage. The network is trained to produce 1 (unity) in case of signal (direct photon from "γ^{dir} + jet" events) and 0 (zero) in case background events. We have taken just one hidden layer with five units. Thus, ANN with architecture 3-5-1 is used here. To ensure convergence and stability, the total number of training patterns (events) must be significantly ($> 20 - 30$ times) larger than the number of independent parameters given in (4). For the ANN training in CC and EC regions we have used about 20 000 (30 000) MC signal and about 7 000 (9 000) background events from the interval of $40 < p_T^\gamma < 60$ GeV in CC (EC) regions, i.e. more than 100 patterns per a weight.

The ANN output for signal and background events for photon candidates win $40 < p_T^\gamma < 60$ GeV in CC is shown in Fig. 6 and in Fig. 7 with photon candidate in EC. The network output for the data preselected by the same main selection cuts (section 3) are also shown on this figure (black histogram).

To verify the MC/data agreement, the built network was tested on the $Z \to ee$ MC/data events. As is seen from Fig. 8, the network shows very good agreement on the MC/data electrons. For example, the difference in the electron selection efficiencies in CC region after cuts on NN output $O_{\rm NN} > 0.7$ turned out to be 91.7% for MC and 90.5% for data. The difference of 1.2% can be taken as a systematic uncertainty for this $O_{\rm NN}$ cut. The output for ANN built for EC region is shown in Fig. 9. The electron selection efficiencies for cut $O_{\rm NN} > 0.7$ are 93.8% and 93.1%, in MC and data events respectively.

Figs. 11 and 10 show dependence of electron selection efficiency on electron p_T for the cut $O_{\rm NN} > 0.7$ for CC and EC regions. The efficiencies are calculated w.r.t. main preselection criteria of section 3.

At Figures 12 and 13 we present the selection efficiency w.r.t. cut $O_{\rm NN} > 0.7$ as a function of p_T^γ for a photon candidate in CC and EC regions. The three curves correspond to direct photon, em-jet and photon candidates from data. The efficiencies are calculated after events selection by set of main cuts from section 3.

5 Application: estimation of direct photon purity.

As an example of application of the built ANNs, let us apply them to obtain a direct photon purity in the selected samples. The photon purity ($\mathcal{P}$), defined as the ratio of signal to signal plus background,

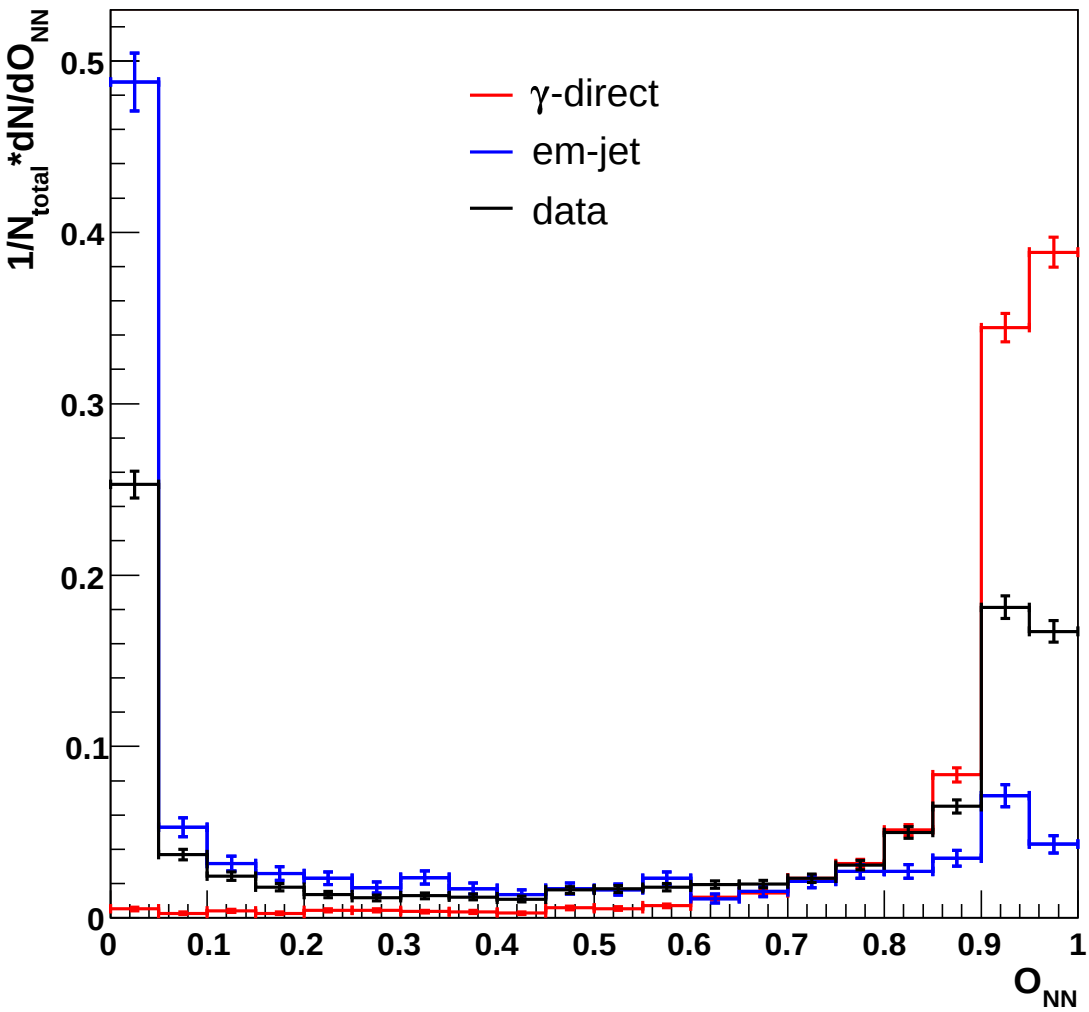

Figure 6: Normalized distribution of ANN output for MC signal (red), background (blue) and data (black) events selected after application of the main selection criteria (section 3) with $\mathbf{40 < p_T^e < 60}$ GeV and $|\eta^\gamma| < 1.0$ is shown.

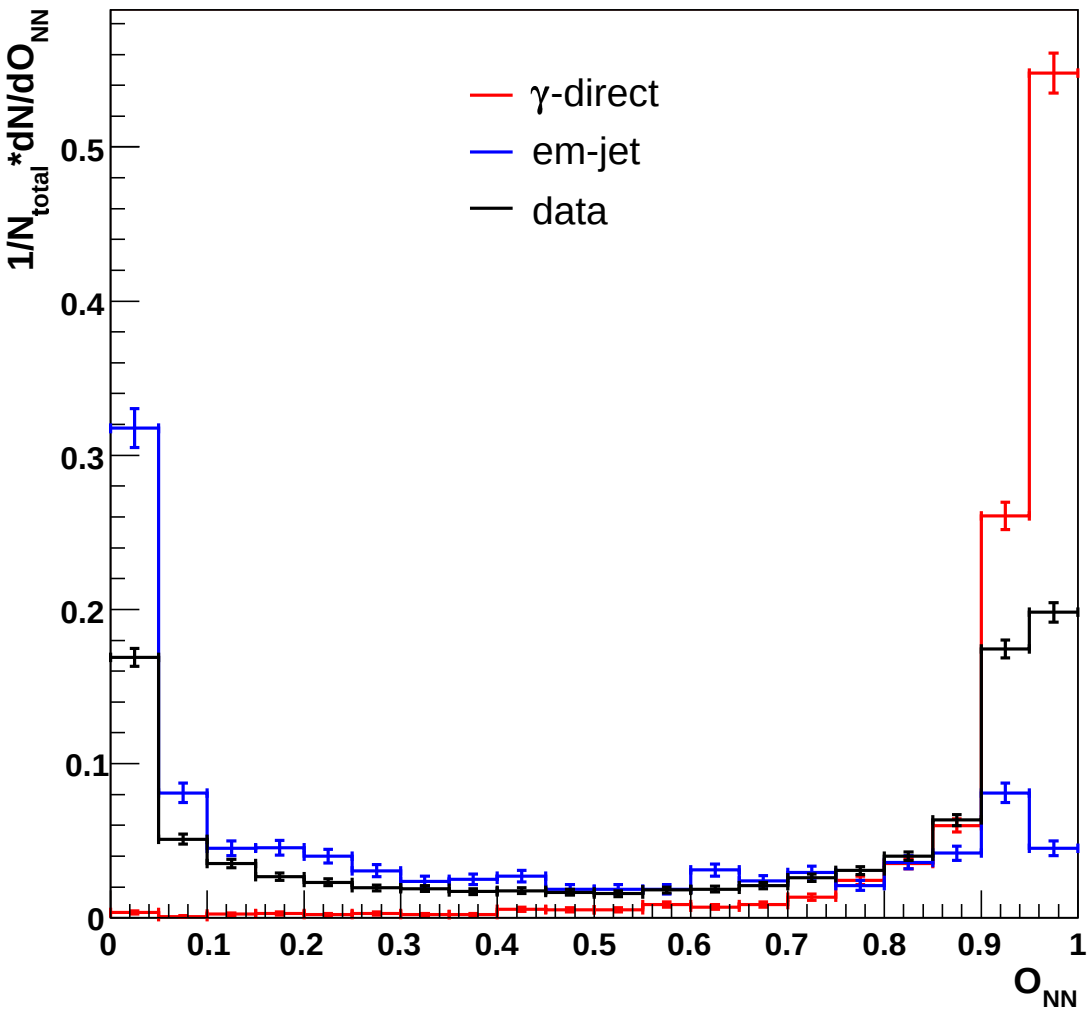

Figure 7: Normalized distribution of ANN output for MC signal (red), background (blue) and data (black) events selected after application of the main selection criteria (section 3) with $\mathbf{40 < p_T^e < 60}$ GeV and $1.5 < |\eta^\gamma| < 2.5$ is shown.

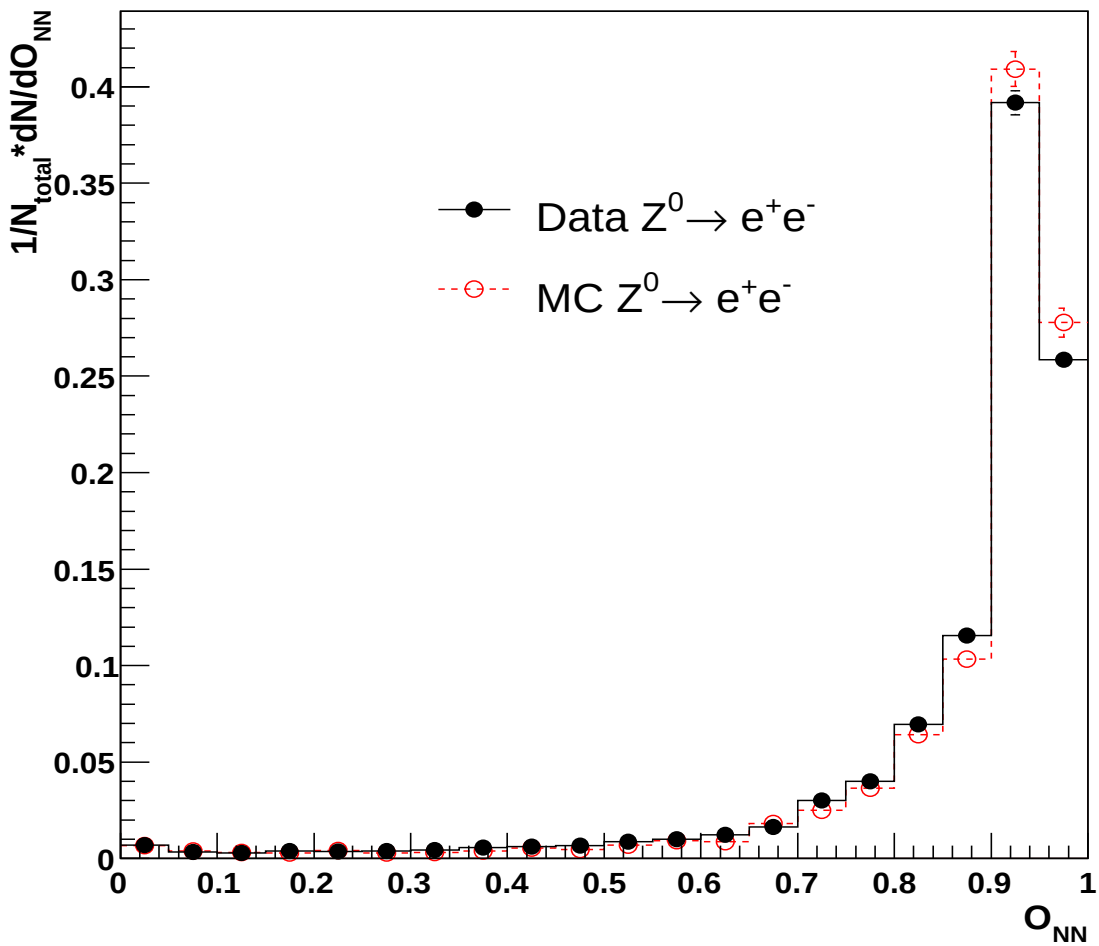

Figure 8: Normalized distribution of ANN output for data (red) and MC (blue) $Z^0 \to ee$ events with $\mathbf{20} < \mathbf{p_T^e} < \mathbf{80}$ GeV and $|\eta^e| < 1.0$ is shown.

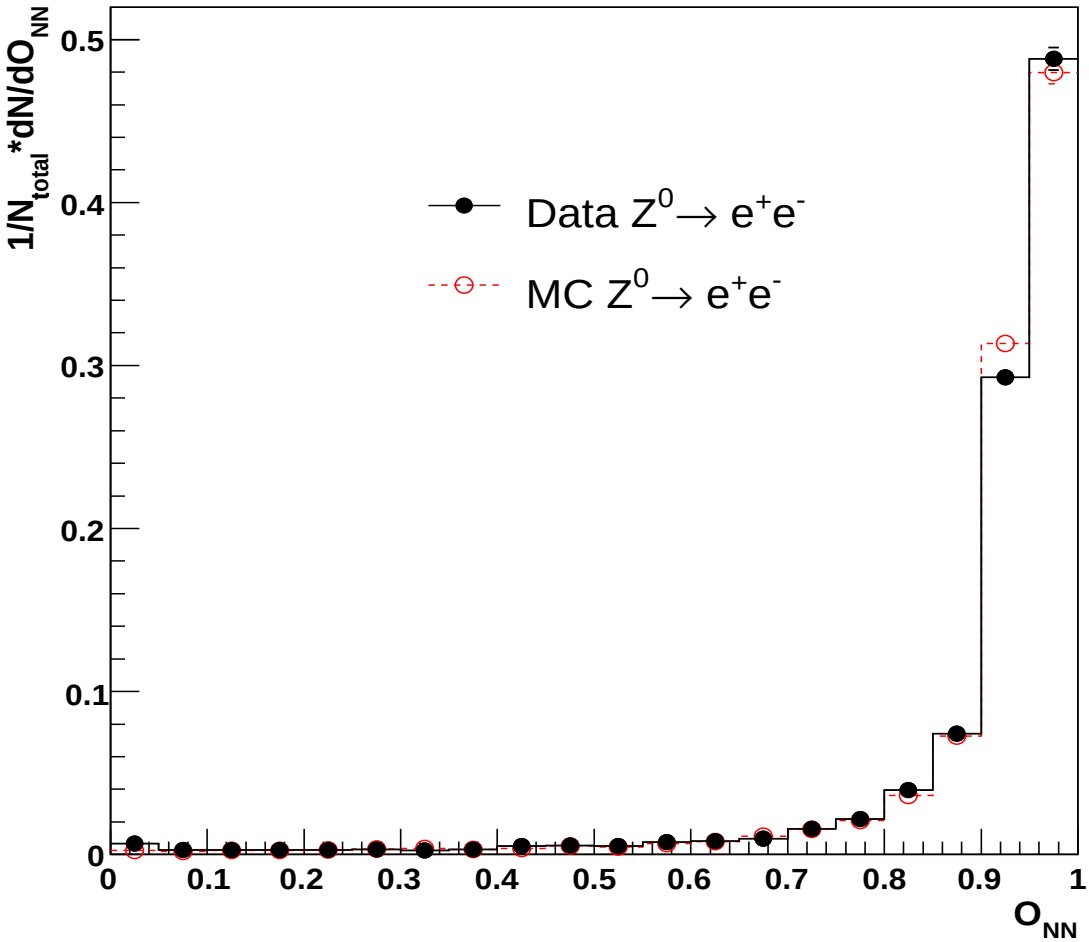

Figure 9: Normalized distribution of ANN output for data (red) and MC (blue) $Z^0 \to ee$ events with $\mathbf{20} < \mathbf{p_T^e} < \mathbf{80}$ GeV and $1.5 < |\eta^e| < 2.5$ is shown.

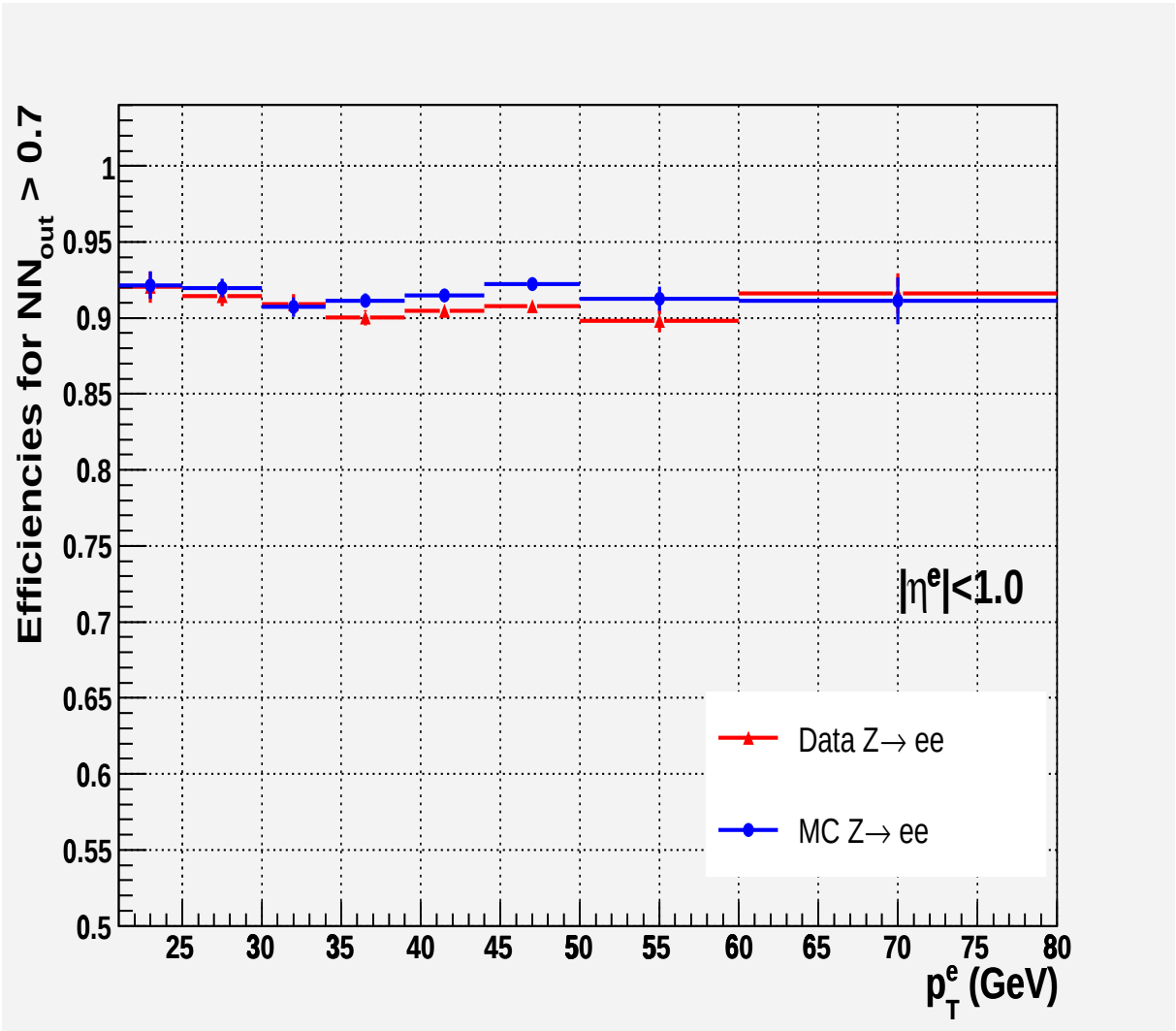

Figure 10: Electron selection efficiency for the cut $O_{\rm NN} > 0.7$ in EC region for data (red) and MC (blue) $Z^0 \to ee$ events for CC region ($|\eta^e| < 1.0$).

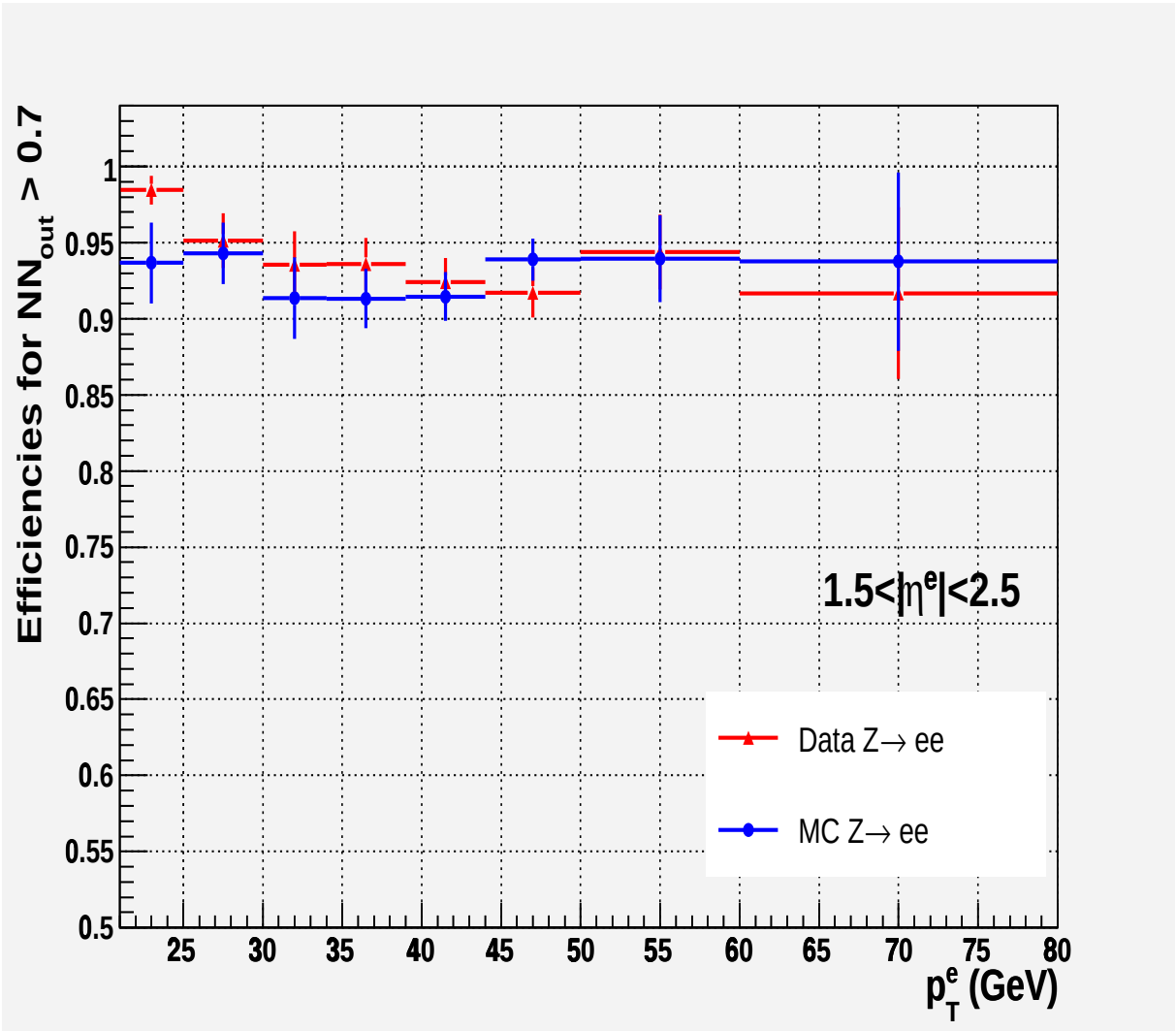

Figure 11: Electron selection efficiency for the cut $O_{\mathrm{NN}} > 0.7$ in EC region for data (red) and MC (blue) $Z^0 \to ee$ events for CC region ($1.5 < |\eta^e| < 2.5$).

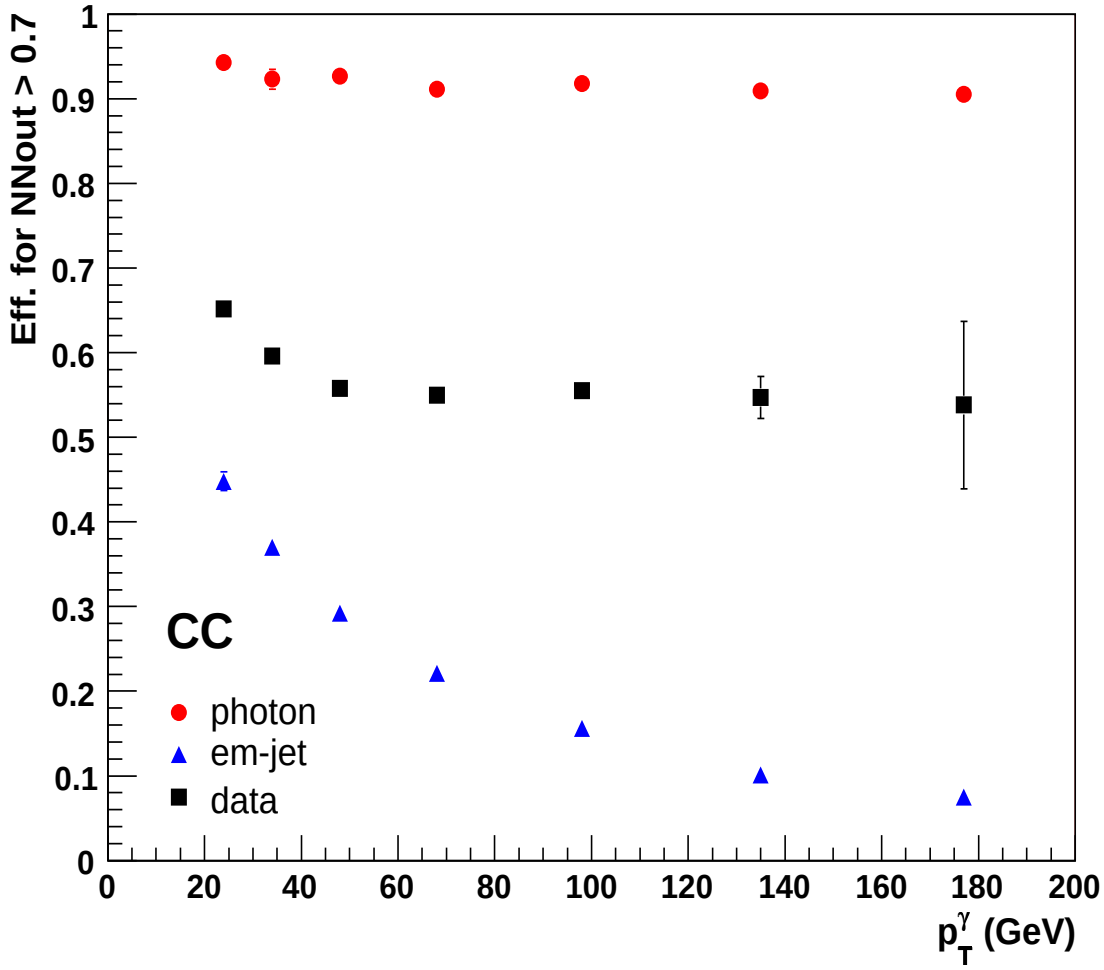

Figure 12: Selection efficiency w.r.t. the cut on ANN output $NN_{output} > 0.7$ as a function of p_T^γ in *CC region* for direct photon, em-jet and candidates from data.

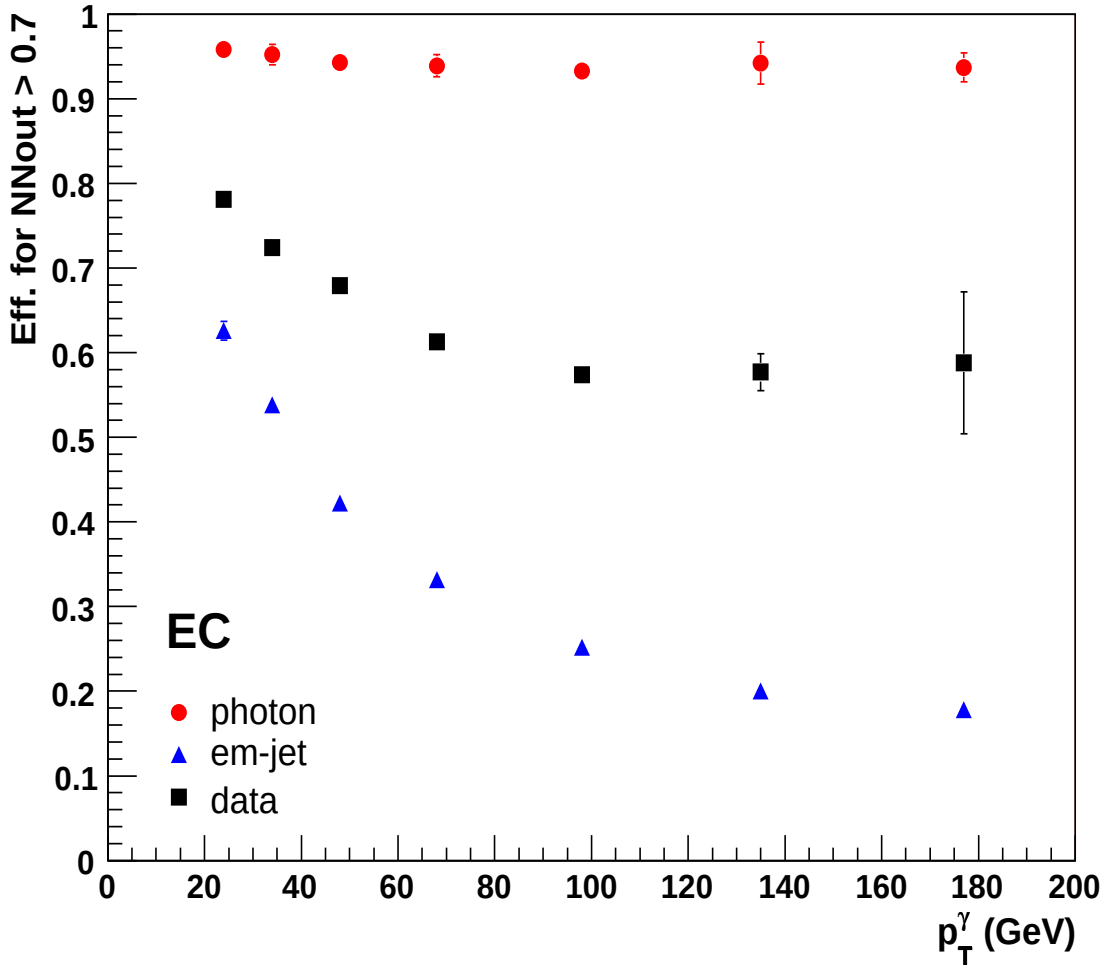

Figure 13: Selection efficiency w.r.t. the cut on ANN output $NN_{output} > 0.7$ as a function of p_T^γ in *EC region* for direct photon, em-jet and candidates from data.

can be determined statistically for each p_T^γ bin. We can do it using "template method" by fitting sum of MC signal and background events to data. The fitting can be performed using TFractionFitter program from ROOT package based on HMCMLL [10] routine from HBOOK. Distributions of the number of events as a function of $O_{\rm NN}$ (with $O_{\rm NN} > 0.7$) are shown for data and MC in Fig. 14 for six p_T^γ intervals. The MC signal and background events in this figure were weighted by the fractions of signal and background events resulted from the fit. The data are well described by the sum of MC signal and background samples, with $\chi^2/ndf \leq 1$ for all p_T^γ intervals.

For those examples the photon purities are given at Fig. 15 by the middle line. Fig. 15 shows also dependence of direct photon purity (found by fitting with the template method) vs. p_T^γ for other two $O_{\rm NN}$ cuts.

In Fig. 16 we compare the direct photon purity as function of p_T^γ for track isolation inside two cones: $R = 0.4$ and $R = 0.7$ (case of $O_{\rm NN} > 0.7$ is considered).

6 Conclusion.

In this paper we have considered a way of optimization of Photon ID criteria [1] by building the artificial neural network for photons in CC and EC regions. Note, it is very important that the constructed ANNs work on the top of strong preselection criteria, described in section 3. Thus, the signal/background discrimination which can be achieved with ANN is additional to that obtained with the main preselection criteria.

We hope that the built network can be applied for many physical applications, for example, such as selection and determination purity in γ+jet, $\gamma+$ heavy flavor jet or $\gamma\gamma$ events.

More information and examples of building, testing and application of ANN can be also found at www-page *http://www-d0.fnal.gov/ bandurin/MyHome/ann.html.*

The constructed ANNs and examples of its usage can be found at this page:
http://www-d0.fnal.gov/ bandurin/ANN_Note/.
Here file NNout_CC_hc04.C contains ANN built using total p_T of tracks inside cone $R = 0.4$, EM cluster p_T fraction at EM1 layer and the number of EM cluster cells at EM1 layer (see section 3), file NNout_CC_hc07.C contains analogous ANN built for cone $R = 0.7$, while NNout_EC_hc04.C and NNout_EC_hc07.C describe ANN done for EC region. File ANN_example.C shows example of the code that uses one of those ANNs and writes obtained ANN output in an external file for following analysis.

Acknowledgments

The author is greatly thankful to Photon ID group, especially to Yuri Maravin, Oleksiy Atramentov as well as Michael Begel for many useful discussions and fruitful suggestions.

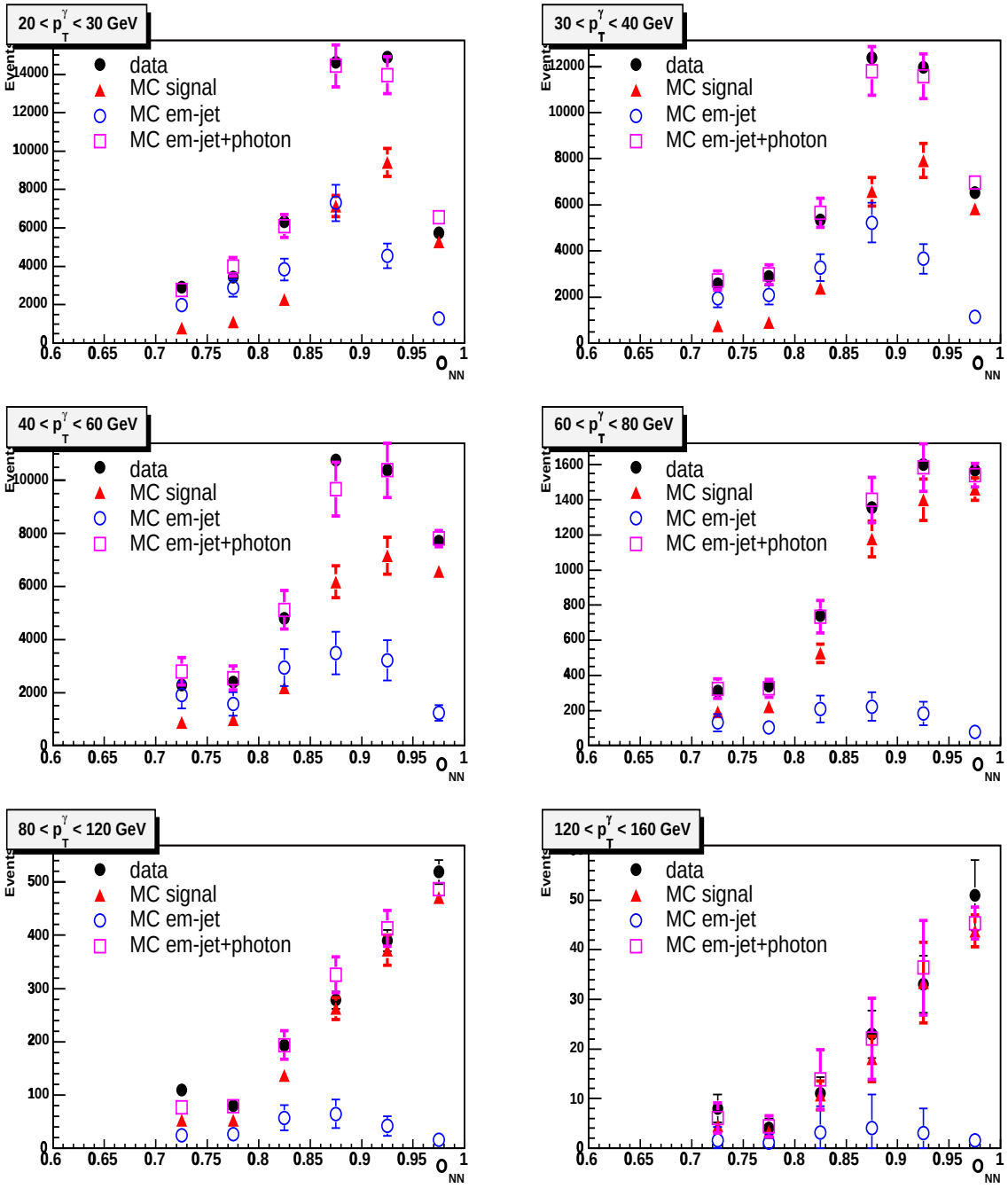

Figure 14: Distribution of the number of events in data (•) as a function of the NN output ($O_{\rm NN}$) for six p_T^γ intervals in CC region. The contributions from MC background (○), signal (△) and summed MC signal and background (□) are also shown. The MC points were weighted according to the fitted purity (the errors shown are statistical).

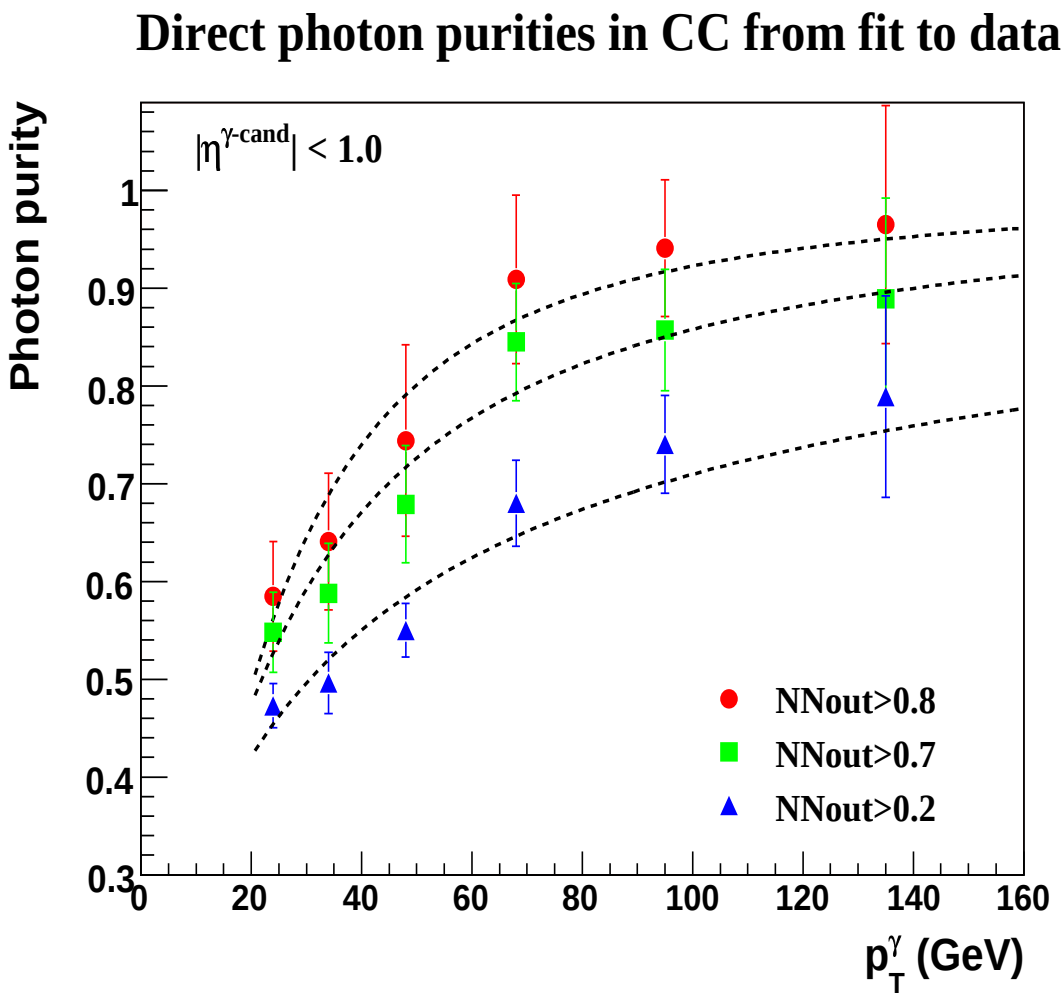

Figure 15: Direct photon purity as a function of p_T^γ in CC, found for three $O_{\rm NN}$ cuts.

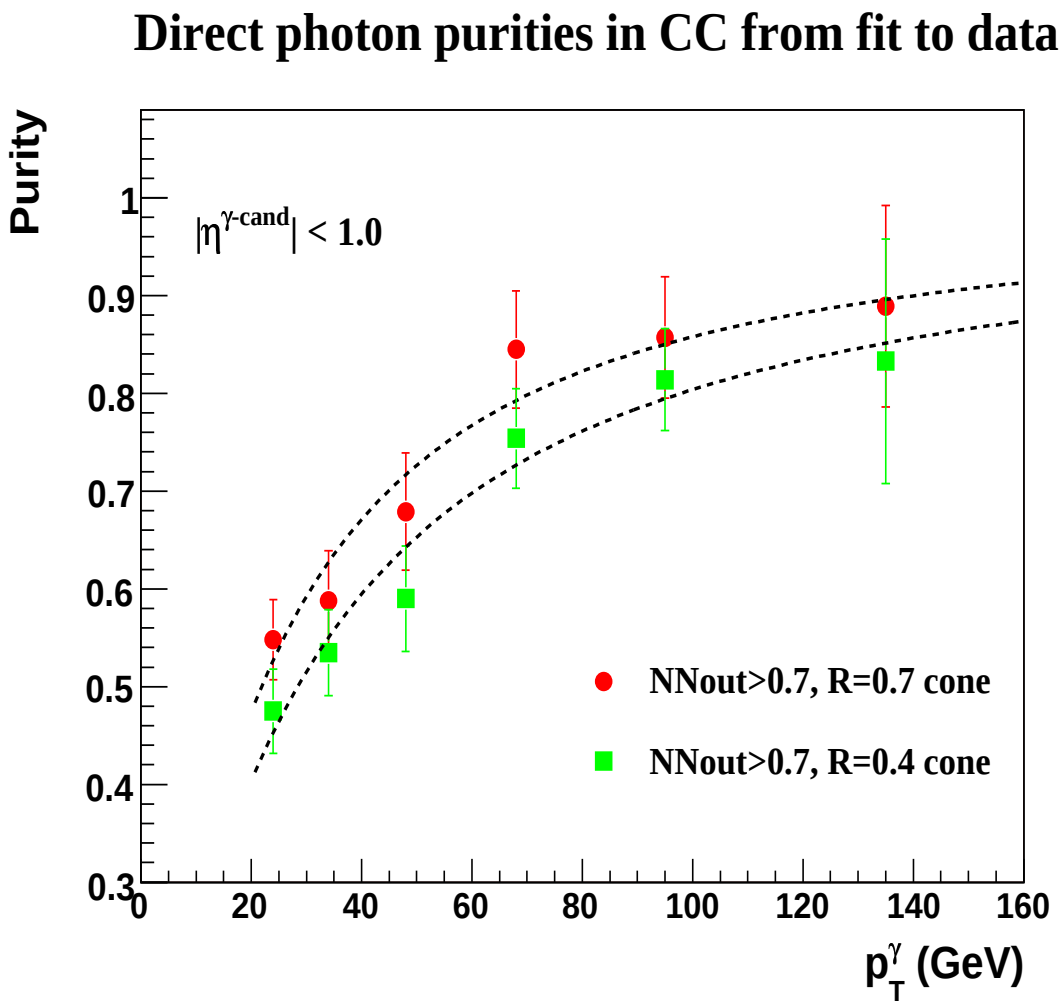

Figure 16: Direct photon purity as a function of p_T^γ in CC, found for two track isolation cones (ANN cut $O_{\rm NN} > 0.7$) is used.

References

[1] "Photon Identification in p17 data", DØ Note 4976.

[2] "Measurement of the isolated photon cross section in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV.", DØ Note 4672, Phys.Lett. **B639**, (2006)151, hep-ex/0511054.

[3] P. Bhat, L. Lonnblad, K. Mejer, K. Sugano, "Using Neural Networks to identify jets in hadron-hadron collisions", DESY Note DESY 90-144, Lund University Preprint LU-TP 90-13.

[4] Proc. of CERN School of Computing, 1991, Ystad, Sweden, CERN 92-02, p.113 – 170.

[5] L. Lonnblad, C. Peterson and T. Rognvaldsson, "Finding gluon jets with a neural trigger", Phys.Rev.Lett, **65**, p. 1321 – 1324, 1990.

[6] L. Lonnblad, C. Peterson and T. Rognvaldsson, "Using neural network to identify jets", Nucl.Phys. **B 349**, p. 675, 1991.

[7] D. Bandourin and N. Skachkov , JHEP 0404:007 (2004), hep-ex/0108051.

[8] C. Peterson, T. Rognvaldsson and L. Lonnblad, "JETNET 3.0. A versatile Artificial Neural Network Package", Lund University Preprint LU-TP 93-29. Version 3.5 is used here.

[9] C. Peterson and E. Hartman, Neural Network **2** (1989)475.

[10] R. Barlow, C. Beeston, Comp.Phys.Comm. **77** (1993)219-228. See also p.120-131 of HBOOK manual.