Online particle detection with Neural Networks based on topological calorimetry information

T Ciodaro[1], D Deva[1], JM de Seixas[1], D Damazio[2]

[1]Federal University of Rio de Janeiro
[2]Brookhaven National Laboratory

E-mail: ciodaro@cern.ch, dhiana.deva@cern.ch, seixas@lps.ufrj.br, damazio@cern.ch

Abstract. This paper presents the latest results from the Ringer algorithm, which is based on artificial neural networks for the electron identification at the online filtering system of the ATLAS particle detector, in the context of the LHC experiment at CERN. The algorithm performs topological feature extraction using the ATLAS calorimetry information (energy measurements). The extracted information is presented to a neural network classifier. Studies showed that the Ringer algorithm achieves high detection efficiency, while keeping the false alarm rate low. Optimizations, guided by detailed analysis, reduced the algorithm execution time by 59%. Also, the total memory necessary to store the Ringer algorithm information represents less than 6.2 percent of the total filtering system amount.

1. Introduction

Pattern recognition based on artificial neural networks is widely used in many scientific areas achieving good performance [1]. For example, neural networks can be found in several high-energy physics experiments, for particle identification [2], particle tracking [3], and other related issues [4]. The experiments in the Large Hadron Collider experiment (LHC), in CERN [5], have used artificial neural networks widely, e.g. [6, 7].

The LHC accelerates and collides proton beams at an event rate of 40 MHz, with up to 14 TeV center-of-mass energy (currently, it operates at 7 TeV). At this unprecedent energy, interesting physics processes never seen before, such as the Higgs boson, can be searched for.

ATLAS is the largest LHC particle detector [8]. In order to provide detailed information about the collision subproducts, ATLAS is split into three main subdetectors: the inner detector, which is responsible for tracking particles; the calorimeters, responsible for measuring their energy; and the muon chambers, which aim at tracking and measuring the momentum of muons. figure 1 shows the ATLAS detector.

Altogether, ATLAS produces around 1.5 MB of information per collision event [9], which must be stored and analyzed. Considering the LHC collision rate, the total ATLAS data flow is around 60 TB/s. As much of the events don't contribute to the new physics which is the goal of the experiment, a three level online filtering system (trigger) was designed to select interesting events [9].

This article presents the latest results of the Ringer algorithm [10], which combines both feature extraction, from the ATLAS calorimetry system, and neural networks to select electrons from the background at the ATLAS second level trigger system. Optimizations are applied

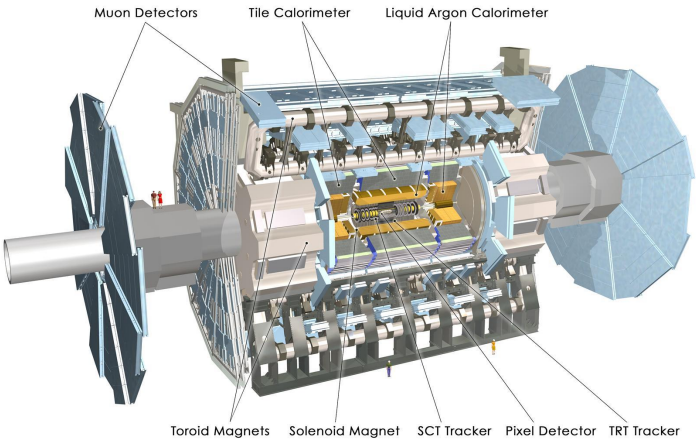


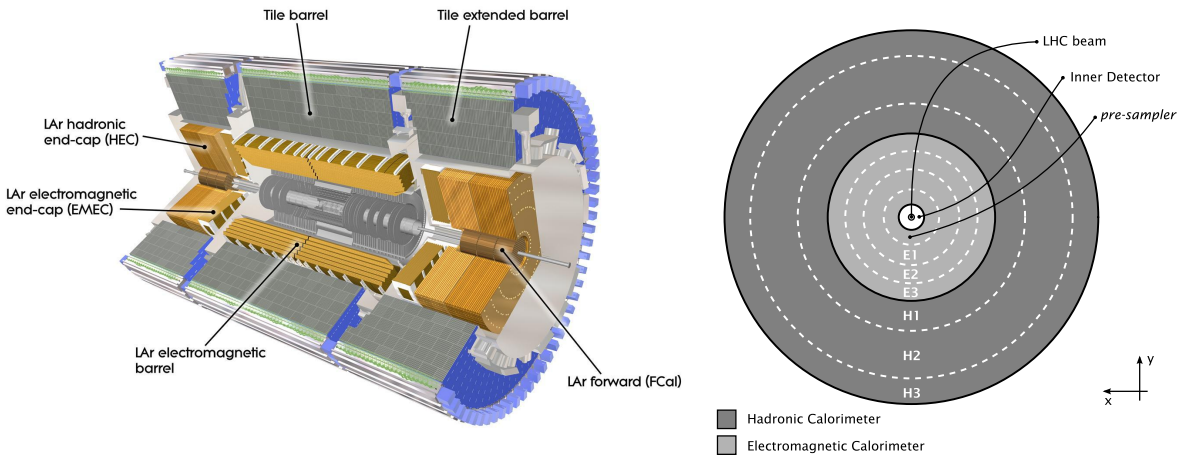
Figure 1. The ATLAS detector.

for improving the processing speed and the whole detection performance is evaluated through algorithm implementation at the trigger framework using simulated data, which include pileup effects.

The article is divided as follows. Section 2 describes the ATLAS calorimeters and section 3 details the ATLAS trigger system. The Ringer algorithm implementation is presented in section 4. Section 5 discusses the methodology and the results achieved, while conclusions are derived in section 6.

2. Calorimetry

The ATLAS calorimetry system [11] is divided into two sections: electromagnetic and hadronic, as can be seen on figure 2(a). The calorimeters are segmented in-depth into layers. The granularity (number of cells per layer) is layer dependent and it was designed to efficiently characterize the desired particles.



(a) (b)

Figure 2. The ATLAS calorimeters (a) and the schematic cross-sectional view of the ATLAS detector (b).

The electromagnetic calorimeter uses liquid Argon technology [12] and is divided into two parts: the barrel and the end-caps. The electromagnetic calorimeter also has an additional pre-sampler layer, which is used to correct for energy lost in dead materials in front of the calorimeter (solenoid, cryostat, ...), totalizing four layers. The electromagnetic section comprises, approximately, 180,000 readout cells.

The hadronic calorimeter, however, uses two technologies: scintillating tiles (TileCal) [13] in the barrel region, and liquid Argon in the hadronic end-cap regions. The TileCal is segmented into three layers. The electromagnetic and hadronic calorimeters cover a region of $-\pi < \varphi < \pi$ and $-2.5 < \eta < 2.5$. The region $|\eta| > 2.5$ is covered by the forward calorimeter, which, however, is not used in this work. Also, the calorimeter has a gap (crack region) in $|\eta| \approx 1.4$, which permits services and infrastructures to reach the most inner parts of ATLAS.

3. Trigger system

The trigger system is split into three cascaded levels, each with its own maximum event rate and latency requirements [9]. Eventually, if the event is accepted by all three levels, data from the entire ATLAS detector are stored for offline analysis [9]. Figure 3 shows the ATLAS trigger system diagram.

3.1. The L1 trigger

The first level of the ATLAS trigger system (L1) is based on highly programmable devices and its main objective is to reduce the 40 MHz event rate to no more than 75 kHz. The L1 identifies the regions at the detector with possibly interesting physics (RoIs, Regions of Interest) [14]. The time budget for the L1 trigger decision (latency) is 2.5μs.

In order to achieve the highest event rate reduction, in the shortest time possible, L1 has access to compacted information from the detector. Only calorimetry and muon information is used. In the calorimeters, trigger tower regions are defined in the $\eta \times \varphi$ plane (0.1×0.1). The energy of the cells inside a trigger tower is summed up, for both electromagnetic and hadronic sections. For the muon chambers, low precision muon transverse momentum information is used. Electrons and other electromagnetic objects are the most frequent signature accepted by L1 [14], and this is the reason for their study in this work.

3.2. The High Level Trigger

The ATLAS High Level Trigger (HLT) is formed by the second level trigger (L2) and the event filter (EF) and it is implemented in software [15]. The HLT operates on the RoIs selected by L1, accessing the detector information in its full resolution.

Large computer farms are used to support the HLT framework. The event processing is split into several processing units according to their usage. For the L2, the mean processing time available is 40 ms, and for the EF, it is 4 s.

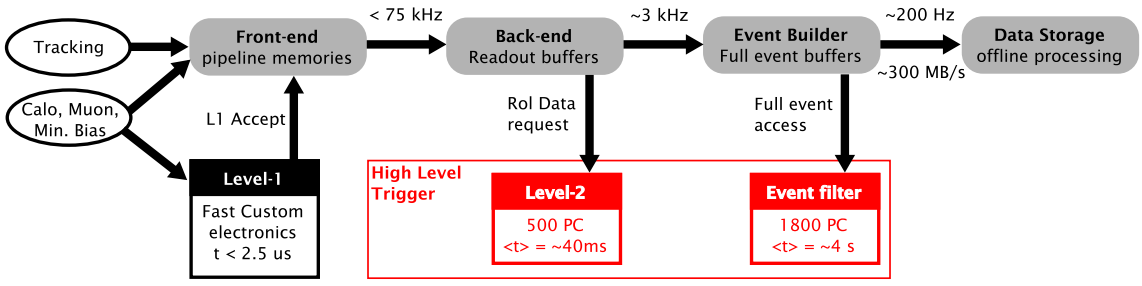


Figure 3. The ATLAS trigger system.

Differently from L1, where only a rough classification is possible, the L2 runs specialized algorithms, which use information from the different subdetectors (calorimeters, tracking and muon chambers). The algorithm's complexity is further increased at the EF, where adapted versions of the offline reconstruction algorithms are executed. Accessing the full event data, the EF can better exploit the physics properties of each event.

3.2.1. Calorimeter HLT: The calorimeter High Level Trigger (HLTCalo) is a set of algorithms that selects particles by performing different clustering methods at the cell level. An optimized infrastructure, common for L2 and EF, for data retrieval is available for all trigger algorithms.

The HLTCalo algorithms are split into two parts: feature extraction (FeX) and hypothesis testing (Hypo). The FeX part accesses the detector data, through the common HLTCalo infrastructure, and calculates relevant variables needed for particle identification. These variables are used by the Hypo part to classify the RoI and, eventually, to accept or reject event.

3.3. Athena

All levels of processing of ATLAS data, from high level trigger to event simulation, reconstruction and analysis, take place within the Athena framework [16]. Athena is fully implemented in objected oriented language (C++ and python) and it is structured into projects and packages.

The HLT community maintains a set of algorithms to emulate the trigger system, which runs after the L1 event acceptance. Also, common tools are provided for analyzing the trigger performance for the associated physics. New algorithms can be developed for running at the HLT, as long as they are implemented within the Athena framework, following some standard procedures and common methods.

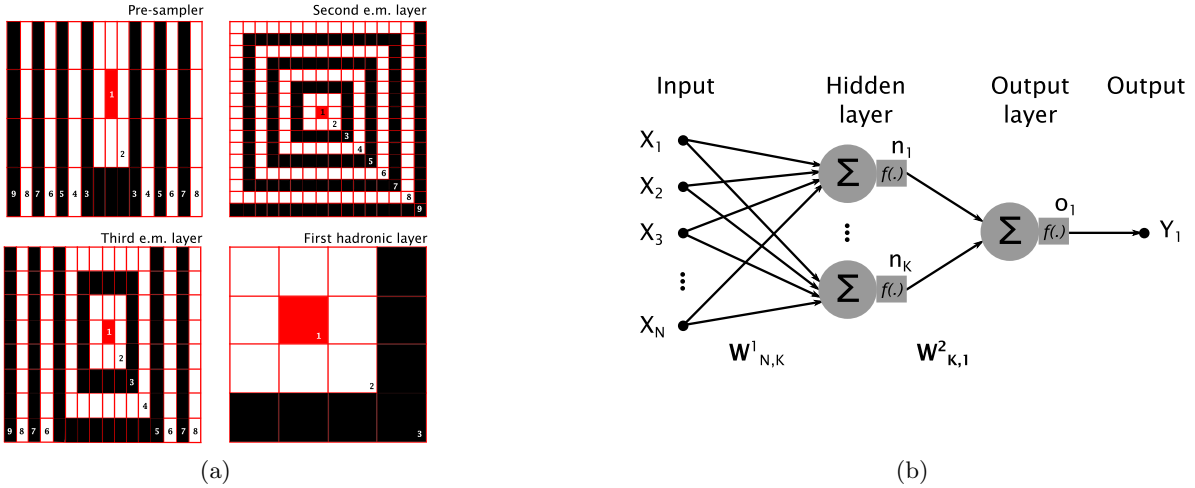
3.4. Electron identification

According to the expected event rates in L1, the detection of electromagnetic particles (electrons and photons) is extremely important for the identification of interesting physics channels. A slice of the trigger system is fully dedicated to the detection of such particles (electron/photon slice) [17].

In order to control the electron trigger flow, linear thresholds are applied to the quantities extracted from electromagnetic showers in HLT. These thresholds can also change during the experiment, in order to allow both detection efficiency and false alarm rate to be controlled. If less background is desired, tighter criteria for electron identification are applied, but also more electrons are discarded. Otherwise, loose criteria are selected and more electrons (and also fake electrons from the background), are triggered. Through such selection, the experiment can tune the trigger rate and other physics demands optimizing the trade-off between particle identification efficiency and speed [15].

Offline algorithms extrapolate tracks from the inner detector to the electromagnetic calorimeter, improving the characterization of the electromagnetic showers. The standard quantities extracted can be used as guidelines for trigger performance evaluation. In particular, the R_{core} quantity extracted offline is used in this study, which is defined as follows:

- R_{core} $\rightarrow$ characterizes the shower spread at the second electromagnetic calorimeter layer: $R_{core} = E_{3\times7}/E_{7\times7}$, where $E_{m\times n}$ is the energy deposited in a region of $m \times n$ cells in the $\eta \times \varphi$ plane around the extrapolated point of impact in the calorimeter of the associated track.



(a) (b)

Figure 4. Illustration of possible rings from the Ringer feature extraction (a) and an example of a feedforward neural network for the Hypothesis testing (b).

4. The Ringer algorithm

The Ringer algorithm is an alternative L2 algorithm for electron identification [10]. Feature extraction is performed through data description by means of concentric rings extracted from seven calorimeter layers of a selected L1 RoI. Hypothesis testing is applied through an artificial neural network [18] operating on the extracted calorimetry features (ring sums).

4.1. Feature extraction

The energy sum over all the cells in each of the rings in an RoI are calculated. In each calorimeter layer, the algorithm searches for the most energetic cell, which defines the first energy sum. The energies of the outer cells with respect to the previous ring are summed up to form the next ring energy sum. This procedure is implemented until the RoI is covered, or a maximum number of rings is reached.

Figure 4(a) shows some ring examples. It can be seen that, in some cases, the ring is completely conceptual, as some are not closed rings. For RoI description, a total of 100 rings are used and each layer contributes with a fixed number of rings (see table 1).

4.2. Hypothesis testing

The extracted rings are presented to a three-layer feedforward neural network (Multi-Layer Perceptron, MLP), whose architecture is illustrated in figure 4(b). The network is trained offline and its parameters are set before the start of the experiment. Several normalizations have been tested to improve the discrimination performance [10]. The present work implemented an event normalization, which consists of dividing each ring sum component by the total event energy.

Table 1. Number of rings per calorimeter layer.

Layer	Electromagnetic Calorimeter				Hadronic Calorimeter		
	pre-sampler	EM 1	EM 2	EM 3	HAD 1	HAD 2	HAD 3
# Rings	8	64	8	8	4	4	4

4.3. Implementation
The Ringer algorithm is implemented on the Athena framework, which allows the algorithm to run on both real and simulated data. Several optimizations have been performed to ensure that the Ringer implementation respects the time constraints of the L2 trigger. Also, the total amount of data required to store the Ringer information must be reasonable. Results from tests on such requirements are discussed in the following sections.

5. Performance
The Ringer algorithm performance is first evaluated using 2010 14 TeV Monte Carlo simulations of electrons and particle jets (background). Electrons are from filtered simulated $Z \rightarrow ee$ events. In both simulations, pileup is considered (the current event might have information from previous events), which tends to deteriorate the trigger performance.

Time performance is evaluated through detailed analysis using the Valgrind tool [19], which is a framework specially designed to analyze code implementation. Through this framework, it is possible to reduce code bottlenecks, optimize function calls and spot unnecessary memory consumption and memory leaks. Athena has a patch for Valgrind analysis, such that the Ringer implementation can be analyzed within the ATLAS software framework.

For the hypothesis testing, (trained) neural network parameters (weights and biases) are exported and uploaded to the Athena framework.

5.1. Methodology
In Z→ee events there will be detector activity registered which is not related to the electrons (for instance it could originate from jets occurring elsewhere in the event). In order to consider only RoIs that correspond to electrons, we use only RoIs (from simulated events) that match the true direction of an electron (defined as $\sqrt{\Delta\eta^2 + \Delta\varphi^2} < 0.05$ between the directions of the RoI and of the actual electron). For the jet simulation, all fake electron RoIs (accepted by a simulated L1 trigger) are considered background. In total, nearly 7k events are selected for electrons and 5k for jets, which are split into half, one forming the training dataset, and the other forming the testing dataset. While the training dataset is used to adapt the MLP parameters, the testing dataset is used to evaluate the MLP generalization (capacity to generalize what was learnt from the training dataset). Also, the testing dataset is used as validation set to avoid overtraining [18].

The resilient back-propagation algorithm [20] is used for training the MLP in batch mode, because of its fast convergence. Also, the MLP is built with one hidden layer only, in order to keep its architecture simple (interesting for online operation, because of its speed). The hyperbolic tangent is used as activation function in both hidden and output layers, where electrons and jets are targeted to the output values '1' and '-1', respectively. The training cost function is the mean squared error between the network output and the desired target. Previous studies have shown that 10 neurons at the hidden layer gives the best trade-off between performance and architecture complexity. In order to avoid local minima, the training algorithm was initialized randomly 100 times and only the network with the best performance was considered.

The SP index is used to evaluate the network performance:

$$SP = \sqrt{\sqrt{P_e \times (1 - F_j)} \times \left(\frac{P_e + (1 - F_j)}{2}\right)} \quad (1)$$

where P_e is the electron detection efficiency and F_j is the jet false alarm rate. Previous studies concluded that this index is the best compromise between electron detection and jet rejection, when compared to their average value, for instance [21]. The SP index results in '1' if all electrons are accepted and all jets are rejected. If all electrons are rejected, or all jets are accepted, it

results in '0'. The MLP output is compared to a threshold in order to finally classify the input as electron or jet. The discrimination threshold is selected in order to maximize the SP index.

5.2. Relevance mapping

A relevance mapping [22] can reduce the MLP complexity by selecting only the input variables (rings) which are most relevant for the discrimination process. Here, the relevance R_i of a given input variable i is measured from the relative difference between the MLP SP index and the SP index achieved when the input variable is fixed to its mean value:

$$R_i = \frac{SP_{MLP} - SP_i}{SP_{MLP}} \tag{2}$$

Rings having relevance values below a given threshold are removed from the training and test sets and a new MLP is designed. Note that, if negative relevant values are found, it means that the given input variable confuses the MLP, and its removal may increase performance.

5.3. Tag-and-probe

In real ATLAS data (as opposed to simulated data) it is not known if a given trigger RoI is truly from an electron or a jet. Thus, a data-driven method for efficiency estimation must be employed. The so-called Tag-and-probe method is often used, as it is extremely suitable for physics processes characterized by a double-object final state signature (like the $Z \rightarrow ee$ process) [23].

The method consists in evaluating the efficiency of detecting 'probe' objects within a properly 'tagged' sample of events. For the L2 online trigger, it consists of selecting possible electron pair candidates with invariant mass consistent with the Z boson mass and which have been accepted by the electron offline algorithm. The event is tagged if one of the electrons (at least) is accepted by a L2 tight electron criteria. The probe object is selected according to the Ringer algorithm hypothesis testing, considering the discriminator threshold that maximizes the SP index. Only the test set is used to evaluate the Ringer performance with the Tag-and-Probe method.

5.4. Results

The Ringer algorithm implementation was evaluated with respect to the total execution time and amount of data needed to save its information (ring sums and neural network output). The Valgrind tool did not report any memory leak after several events. The total time needed to run the entire L2 electron algorithm was measured. After a total of $\sim$17k RoI's, the new Ringer implementation (after the Valgrind analysis) is estimated to take, on average, 1.075 ms to fully execute (feature extraction and hypothesis testing), while before the Valgrind analysis it took, on average, 1.445 ms. The detailed analysis led to a 59% faster execution time. Also, the new Ringer implementation adds an overhead of $\sim$0.3 ms per RoI when compared to a standard L2 algorithm. With respect to the ring energy sums and neural network output, the Ringer information represents 6.2% of the total amount of stored data used by electron/photon algorithms.

The neural network performance for the training set indicates a 5.18% false alarm probability (jets misclassified as electrons), for a fixed electron detection probability of 98.00%. Figure 5 shows the neural network output for the training set. Notice that most of the electrons were correctly mapped to the output value '1', while jets were mapped to '-1'. The threshold needed to maximize the SP index is also shown, together with the threshold needed to achieve 98.00% electron detection. The maximum SP index translates into a 97.28% electron detection probability, against a 4.00% false alarm probability.

In order to reduce the network complexity, the input variables (ring energy sums) were ranked according to their relevance. Figure 6(a) shows the relevance for each ring. The full vertical lines indicate the end of a calorimeter layer and the beginning of the next one. By changing

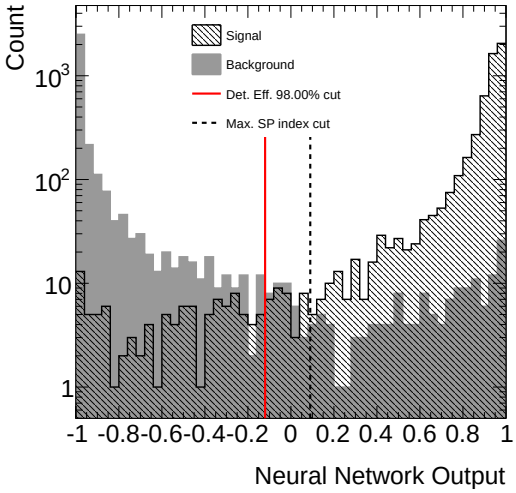
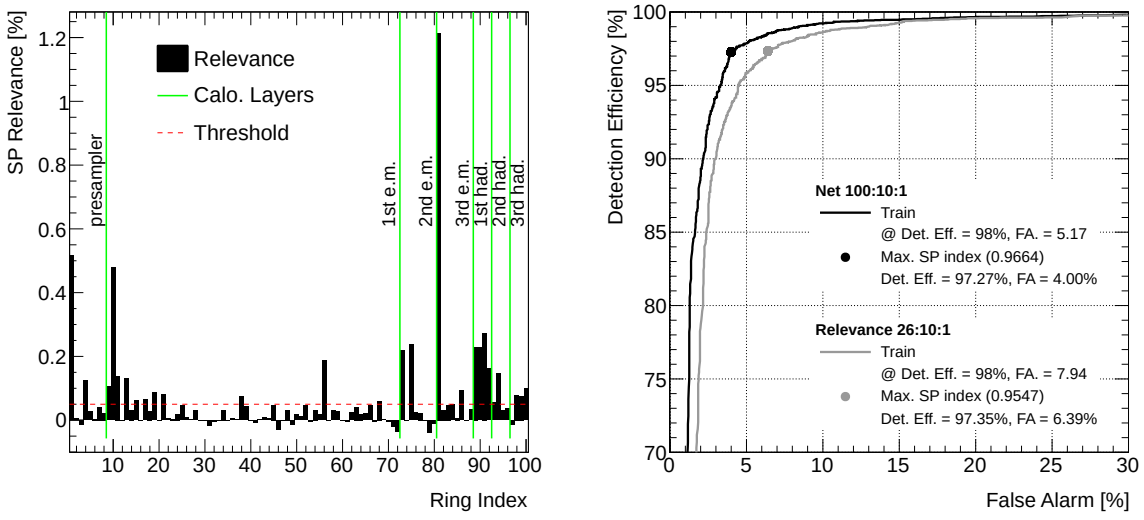


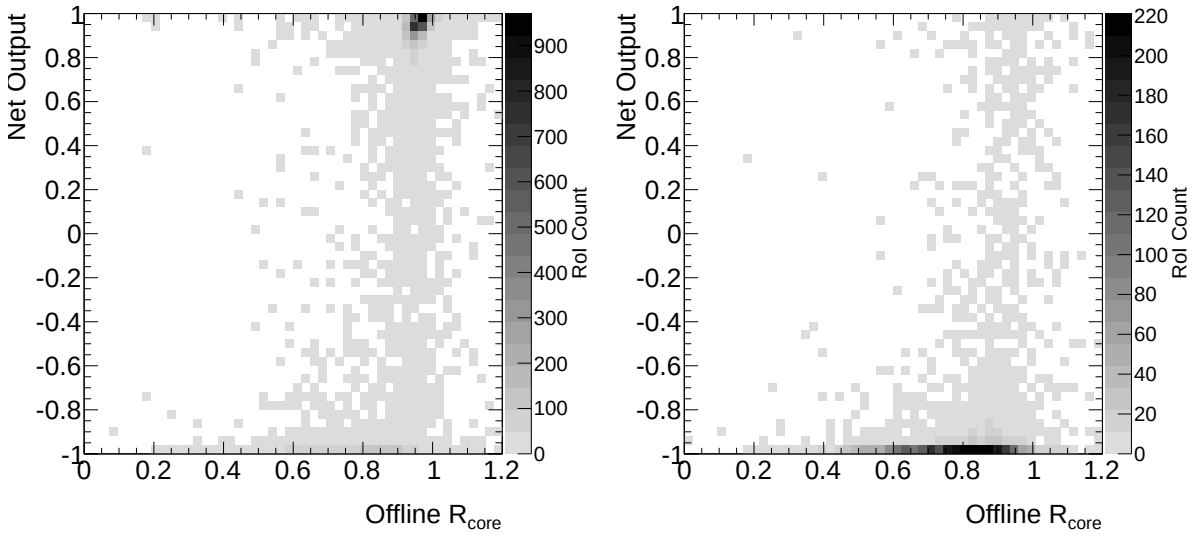
Figure 5. Neural network output for the training set over the MLP trained with full ring description.

the relevance threshold, the number of rings considered relevant for the discrimination can be adjusted. In order to achieve a reduction of ∼70% of the total amount of rings, for instance, a relevance threshold of 0.05% is required (dashed horizontal line). Note that some rings have negative relevance, meaning that they confuse the particle classification, instead of assisting it. In terms of performance, figure 6(b) shows the ROC (Receiver Operating Characteristic) [24] curve achieved after training an MLP using only the relevant rings as inputs (again initializing 100 times to avoid local minima). The ROC curve for the MLP using the entire ring set is



(a) (b)

Figure 6. Relevance mapping using the training dataset (a) and the resulting ROC curve for MLPs using only relevant rings, and using full ring information (b).



(a) (b)

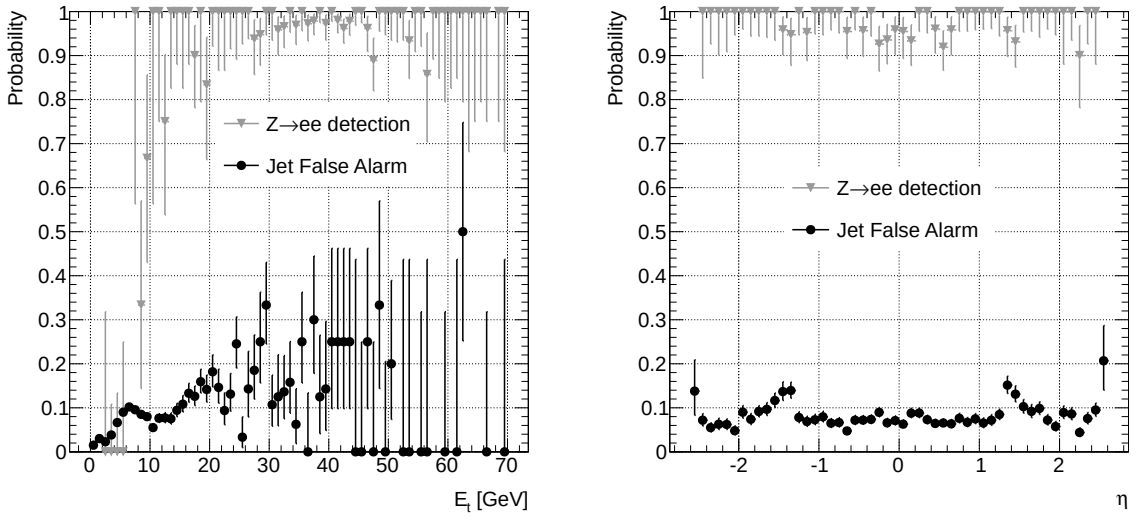
Figure 7. MLP output as a function of the electron offline R_{core} discriminant variable for electrons (a) and jets (b).

also shown for the sake of comparison. It can be seen that the ∼70% information reduction increased the false alarm rate by less than 3% (fixing the electron detection efficiency to 98%), while the SP index is slightly reduced (∼0.1). The relevance threshold value can be used as an adjustable parameter to regulate the MLP performance in a compromise with the Ringer hypothesis test speed and the total false alarm rate.

In order to extract physics conclusions, the neural network output can be compared to some key offline electron variables for a given RoI. Figure 7 shows the MLP trained without the relevance cut (full ring information) with respect to the R_{core} offline variable (see section 3.4), for both electrons and jets. The results are considering the test set, in order to evaluate the MLP generalization. The R_{core} measures the electromagnetic shower spread. Electrons produce narrower showers, leading to an R_{core} value close to unity, while jets tend to form wider showers, thus with a R_{core} value far from unity. It can be seen from figure 7(a) that the R_{core} value is close to unity for most of the electrons that pass the MLP cut. This is particularly interesting and useful when real data is considered, as the actual RoI classification is unknown. The same result can be seen for the rejection of jets, on figure 7(b), where most of the jet RoI's are correctly classified by the Ringer hypothesis test, as their R_{core} value is mostly around 0.8.

Finally, the performance with respect to the Tag-and-Probe method was evaluated. As described before (see section 5.3), the Tag-and-Probe is a data-driven method for efficiency estimation, and it is an important performance estimator used in real data. Figure 8(a) shows the Ringer performance and false alarm rate as a function of the RoI transverse energy. RoIs near the crack region ($|\eta| \approx 1.5$) and the end-cap ($|\eta| \approx 2.5$) are excluded. It can be seen that the Ringer performance in the $Z \rightarrow ee$ channel is very good for most of the energy range. On average, for the transverse energy range 20 GeV$< E_t <$50 GeV, the Ringer algorithm achieves an electron detection efficiency of 98.24±2.54%, for a 15.78±10.16% false alarm probability. More statistics is needed to increase the measurement accuracy.

Similarly, the performance for the electron detection based on the Tag-and-Probe method and the false alarm rate as a function of η is shown on figure 8(b). Only RoI's whose respective transverse energy is greater than 20 GeV are considered. It can be seen that the performance



(a) (b)

Figure 8. MLP performance using the and the Tag-and-Probe method as a function of the RoI offline transverse energy (a) and η (b).

over the $Z \rightarrow ee$ channel is almost uniform in η (97.34±2.82% electron detection efficiency against 8.08±2.46% false alarm probability, considering $|\eta| < 1.5$). For the false alarm rejection, the regions around the calorimeter crack ($|\eta| \approx 1.5$) and at the far end of the calorimeter suffer from a higher false alarm probability. This is expected due to the missing information and limited energy resolution in these regions.

6. Conclusions

Results from the ATLAS HLT Ringer algorithm performance were shown, with respect to the implementation in the ATLAS Athena framework, and considering the detection performance in the $Z \rightarrow ee$ channel, including pileup. In addition, a methodology for reducing the neural network complexity, by ranking the input variables according to their relevance, was studied. Finally, the relevance study reduced the MLP input dimension in ∼70%, with a small impact on detection performance.

The detailed Valgrind analysis led to an optimized implementation, which became 59% faster, while the total amount of data needed to store the Ringer information was kept low (6.2% of the total used by the electron trigger slice).

The detection performance was evaluated from 2010 14 TeV Monte Carlo simulations. The Ringer algorithm showed high performance also when a data-driven method for performance estimation is considered (Tag-and-Probe). Crosscheck with the offline R_{core} discriminant variable shows good agreement with the MLP response for electrons and jets.

The results presented here show that it is promising to use the Ringer algorithm for electron identification in the trigger system for collider experiments.

Acknowledgements

The authors would like to thank the Brazilian funding agencies CNPq, CAPES, FINEP, FAPERJ and RENAFAE, as well as CERN and the European Union, for the support to this work. We also thank our colleagues from the ATLAS/TDAQ collaboration for the interesting discussions

concerning this work.

References

[1] Bishop C M 1994 *Review of Scientific Instruments* **65** 1803
[2] Abramowicz H *et al.* 1995 *Neural network based electron identification in the ZEUS calorimeter* vol 1 p 763
[3] Badalà A, Barbera R *et al.* 2003 *Nucl. Instrum. Methods Phys. Res.* A **502** 503
[4] Bortolotto C, Angelis A D, Groot N D and Seixas J 1992 *Int. J. Mod. Phys.* C **3** 733
[5] Evans L *et al.* 2008 *Journal of Instrumentation* **3** S08001
[6] Prevotet J C, Denby B *et al.* 2003 *Nucl. Instrum. Methods Phys. Res.* A **502** 511
[7] McGlone H M 2009 *Neural Network Analysis in Higgs Search using* $t\bar{t}H, H \rightarrow b\bar{b}$ *and TAG Database Development for ATLAS,* Ph.D. thesis , University of Glasgow (*Preprint* `http://cdsweb.cern.ch/record/1287877/files/CERN-THESIS-2005-062.pdf`)
[8] The ATLAS Collaboration 2008 *Journal of Instrumentation* **3** 1
[9] Kordas K *et al.* 2007 *Nuclear Physics B - Proceedings Supplements* **172** 178
[10] dos Anjos A R, Torres R C, J M de Seixas B C F and Xavier T C 2006 *Nucl. Instrum. Methods Phys. Res.* A **559** 134
[11] Wigmans R 2000 *Calorimetry: energy measurement in particle physics* (Oxford: Clarendon Press)
[12] Zhang H *et al.* 2010 *The ATLAS Liquid Argon Calorimeter: Overview and Performance* vol 1 pp 1–6
[13] Ariztizabal F *et al.* 1994 *Nucl. Instrum. Methods Phys. Res.* A **349** 384
[14] Berge D, Aielli G, Andrei V *et al.* 2007 *Nucl. Instrum. Methods Phys. Res.* A **581** 476
[15] Stelzer J *et al.* 2011 *Journal of Physics: Conference Series* **331** 022026
[16] Cranmer K 2008 *Nuclear Physics B, Proceedings Supplements* **177** 126
[17] The ATLAS Collaboration 2010 Performance of the electron and photon trigger in p-p collisions at a centre of mass energy of 900 GeV Tech. Rep. ATLAS-CONF-2010-022 CERN Geneva
[18] Haykin S 2008 *Neural Networks and Learning Machines* (Prentice Hall)
[19] Nethercote N and Seward J 2007 *Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation* vol 1 pp 1–12
[20] Riedmiller M and Braun H 1993 *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm* pp 586–591
[21] Torres R C, de Lima D E F, Simas Filho E and Seixas J M 2009 *IEEE Nuclear Science Symposium Conference Record* **1** 530–536
[22] Gruber A *et al.* 1994 *New Computing Techniques in Physics Research* **1** 429
[23] Straessner A and Schott M 2010 *Journal of Physics: Conference Series* **219** 032023
[24] van Trees H 2001 *Detection, estimation, and modulation theory part I* (John Wiley & Sons)