Denoising Autoencoder to Mitigate Crosstalk Effects in a Liquid Argon Electromagnetic Calorimeter

Ícaro Nascimento Queiroz, Antônio Carlos L. F. Júnior, Eduardo F. de Simas Filho, Marton S. dos Santos
Laboratório de Sistemas Digitais - LSD UFBA, Salvador, Brazil
icaro.queiroz@ufba.br, antonio.lopes@ufba.br, eduardo.simas@ufba.br, martonsandesd@ufba.br

***Abstract*—The calorimeters in ATLAS, one of the experiments carried out at CERN's LHC, can detect the energy levels generated by the collision of various particles, thus enabling the computational reconstruction of the energy value deposited by the particle and its flight time. One of the problems observed in the sensor elements present in ATLAS is the phenomenon of crosstalk, caused by the high energy levels generated by collisions and the inclusion of a series of uncertainties into the computational reconstruction of the collision. In this article the approach suggested to mitigate these problems is the use of a Denoising Autoencoder neural network, trained on simulated data extracted from the Lorenzetti framework, developed to generate synthetic data relating to particle collisions within ATLAS. Through the use of these neural networks, we sought to carry out a neural network filtering process, using several convolutional and pooling layers, to achieve energy values closer to the true energy value.**

***Index Terms*—calorimeter, lorenzetti, simulator, crosstalk, neural networks, autoencoder, denoising**

I. Introduction

The field of study of particle and high-energy physics seeks to understand the behavior of particles and their interactions, to observe and understand the behavior of matter, as described by the theory of the standard model of elementary particles [1]. The CERN (*Conseil Européen pour la Recherche Nucléaire*) has a particle accelerator, the LHC (Large Hadron Collider), a structure installed in a 26.7 kilometer tunnel made up of various experiments aimed at studying high-energy phenomena and observing particles. One of the experiments carried out is ATLAS (A Toroidal LHC ApparatuS), which can be seen in Figure 1, carried out to detect and characterize particles generated through proton-proton collisions [2], where proton beams are capable of generating an energy value of up to 14.8 TeV (Tera electronvolt). The ATLAS has a calorimeter with a cylindrical structure, developed with a great detection potential, including Higgs bosons and supersymmetric particles, and a solenoid structure capable of generating a 2 Tesla magnetic field and is separated into different regions, each of which has different functions in the detection process [3]. Charged particles passing through electromagnetic calorimeter cells generate high levels of energy deposition. This event generates excitations in neighboring cells, a phenomenon called crosstalk, which ends up distorting the energy readings. Crosstalk has different types of contributions: capacitive, inductive, and resistive, coming from the materials that make up the detector cells [4]. These uncertainties generated in the readings can affect the characterization of incoming particles from collisions. One of the approaches devised to mitigate the effects of crosstalk on calorimeter readings is the use of neural networks, based on data simulated using Lorenzetti, a framework developed to simulate the energy readings taken in the ATLAS calorimeter using computational methods [5], taking into account the modeling of the systems present in the calorimeter and the physical phenomena involved, including the phenomenon of crosstalk. Previous work, such as that done by [2] proposes reducing the effect of crosstalk through supervised learning using MLP neural networks. This work proposes a neural network model to mitigate the effects of crosstalk in calorimeter readings using an autoencoder architecture. The architecture proposed for processing the simulated Lorenzetti data is a two-dimensional convolutional autoencoder, which is a type of neural network developed for the purpose of extracting information and characteristics from an input in an unsupervised manner [6]. For autoencoder input, samples from the simulation process were used, arranged in a matrix format with dimensions of 7×7, using a denoising autoencoder architecture to reduce noise, without using real data in the prediction process.

II. Calorimeters

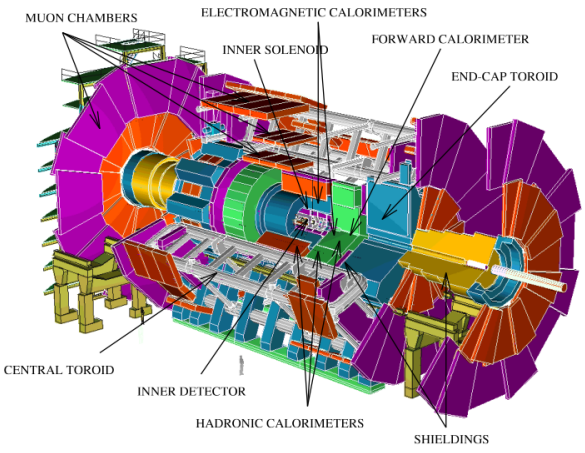


Fig. 1: ATLAS Experiment Structure [2]

Calorimeters play an important role in the ATLAS experiment, measuring particles such as photons, electrons, hadrons, and jets. The detection information from the calorimeters is analyzed by the trigger system, which indicates whether or not an event of interest has occurred. The ATLAS electromagnetic calorimeter covers the $|\eta| < 3.2$ region and is a liquid argon sampling calorimeter. The other detection system uses a hadronic calorimeter, covering the region of $|\eta| < 1.7$ [7]. Its type of detection is based on the use of flat plastic scintillators, and its signals are read via a fiber optic system. The active and passive layers, the electrodes, and the absorbers present in the ATLAS electromagnetic calorimeter barrel are arranged in an accordion shape, where the folds are perpendicular to the path of the particle. The absorbers illustrated on Figure 2, are separated by a layer filled with liquid argon, while the electrodes of the reading system are separated by layers of Kapton [8]. To prevent the structure from deforming due to the weight of the absorbers, a layer of stainless steel is bonded with a fibreglass-based material. In the calorimeter coordinate system, the origin is defined as the point of interaction, located in the center of the detector. For particle detection the polar coordinate system is used, where r is the distance between the particle and the interaction point, ϕ is the azimuthal angle in the plane transverse to the beam pipe, and θ is the angle concerning the z-axis. Another important variable is the pseudorapidity η, given by Equation (1).

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right) \tag{1}$$

III. Crosstalk in High Energy Calorimeters

The electrons from the ionization carried out in the electromagnetic calorimeter deflect towards the electrodes, which measure the signal proportional to the energy deposited. From these signals, the energy value is reconstructed, using the Optimal Filtering (OF) technique. The optimal filter algorithm estimates the value of the energy amplitude A, defined in Equation (2) and particle flight time τ, defined in Equation (3), for n samples of a cluster, where s_i are the energy values contained in the samples and p are the energy pedestal values estimated through the average of the events that have occurred. The coefficient a_i is used to estimate the energy amplitude, while a coefficient b_i is used for flight time [9].

$$A = \sum_{i}^{n} a_i(s_i - p) \tag{2}$$

$$\tau = \frac{1}{A}\sum_{i}^{n} b_i(s_i - p) \tag{3}$$

Due to the type of material in the detection cells, unwanted couplings are generated, such as capacitive, inductive, and resistive couplings. The capacitive crosstalk is the most strong type in hadronic calorimeters [10], due to the spaces between the cells. The middle and front regions of the calorimeter have resistive connections on the electrodes, which means that resistive crosstalk is observed more often in this region. Inductive crosstalk is more common in the region between the back and middle of the calorimeter, which occurs due to the mutual inductance between cells and the ground [11]. The crosstalk between cells is illustrated in Figure 3. When the particles collide with the detector cells, there is a large deposition of energy, and due to the existing couplings, the phenomenon of crosstalk induces a mutual excitation from one cell to the other cells in its neighborhood. This ends up directly influencing the energy readings and, consequently, the process of estimating the quantities relating to the particles.

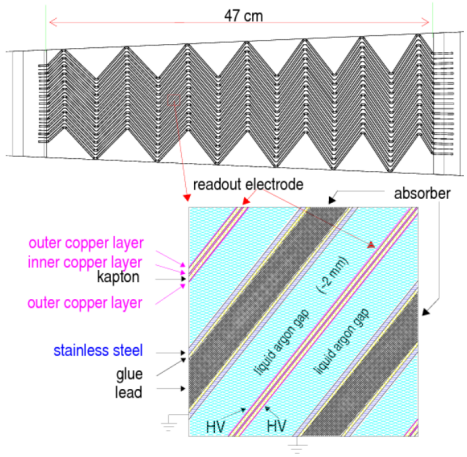


Fig. 2: Electromagnetic Barrel Cells layers diagram [8]

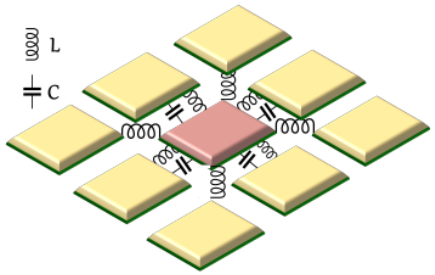


Fig. 3: XT between the calorimeter cells [2]

The composition of the readout sample i in each cell j, that receives the XT contribution from another cell k is given by Equation (4). S_j^{rec} is the energy read out by the detector cell, S_j^{truth} is the true energy deposited in the cell by the particle, plus the XT contributions generated by crosstalk in $XT_{k\to j}$, where the b_j signal is the influence of electronic noise in cell j [9]. The characteristics of the crosstalk signal are illustrated in Figure 4.

$$S_j^{rec}(t_i) = S_j^{truth}(t_i) + \sum_{k} XT_{k\to j}(t_i) + b_j(t_i) \tag{4}$$

IV. Used Dataset

The data used in this work have been obtained from the Lorenzetti electron shower simulator for single electron production. This software is based on the technologies used

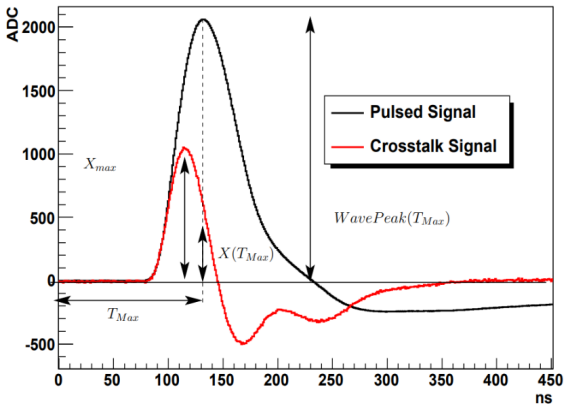


Fig. 4: Amplitudes of Pulsed Signal and the crosstalk [4]

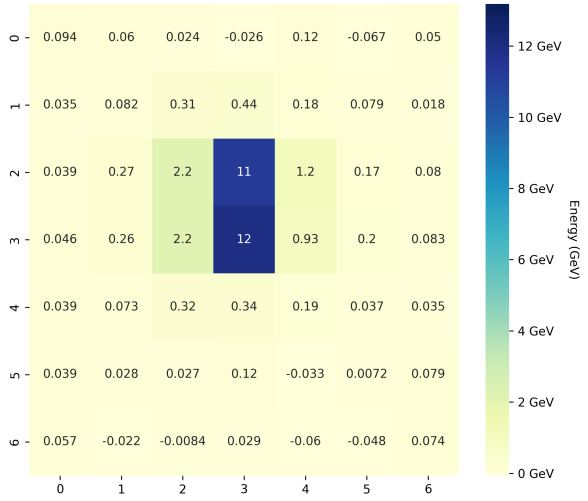
for particle detection and the reconstruction of collision events, allowing the behavior of the calorimeter to be simulated, generating synthetic data relating to the detection of energy levels. In addition, Lorenzetti is also capable of simulating crosstalk, providing a simulation with a higher degree of fidelity [5]. The energy data generated by Lorenzetti are presented in the form of clusters, where a simulated event occurs in a group of cells in a η region of the calorimeter. Two different types of files generated by the Lorenzetti simulator were used to apply the mitigation processes using neural networks, both in pickle format. The pickle format is used for serializing data structure objects in Python, where *pickling* is the serialization of data into a sequence of bytes and *unpickling* is the inverse process of this operation [12].

- **Samples File**: This file contains 99492 distinct single-electron events, at the electromagnetic second layer of the barrel calorimeter, produced one by one. In each event, a 7×7 cluster around the hottest cell is generated. In this layer, the cells are the size $(\eta \times \phi)$ $\frac{\pi}{128} \times 0.025$. Where it is possible to estimate the particle's energy amplitude and time of flight values using an optimal filter algorithm.
- **Amplitudes File**: This file also contains 99492 samples, but this file only contains the energy amplitude values, both for the true energy values and crosstalk.

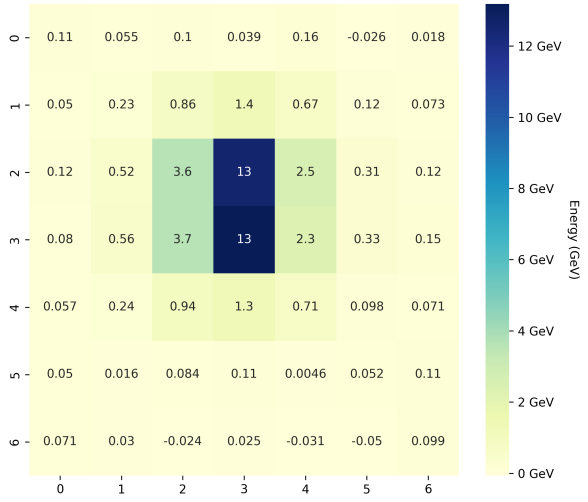
Figure 5 shows how the data are extracted from these files in heatmap format for a 7×7 cell cluster. The samples used in this work refer to the calorimeter η regions between -0.08750 and 0.00875, and between 1.4603 and 1.6075 for ϕ, referring to the central region of the calorimeter.

V. Autoencoder to Mitigate Crosstalk Effects

The neural network structure proposed to carry out the crosstalk mitigation study was Autoencoder. Generally speaking, neural networks work with supervised learning, where a predicted value $\mathbf{X}_i$ has an expected output $\mathbf{Y}_i$ [14], where learning will be based on the relationship between the input data and the expected labels. However, autoencoders work in such a way as to reconstruct the input data $\mathbf{X}_i$ with the least possible error [15]. Two elements are important in defining the



(a) True energy



(b) Crosstalk contribution

Fig. 5: Heatmaps from a 7 × 7 cluster: (a) true energy and (b) crosstalk contributions.

operating structure of an autoencoder: encoder and decoder, as seen in Figure 6. The encoder, responsible for the input compression process, can be described in Equation (5), where $\mathbf{H_i} \in \mathbb{R}^{m \times q}$, the latent representation, is the output of the encoder, which is validated against the input $\mathbf{X}_i \in \mathbb{R}^{m \times n}$ where $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ for each row of the input $\mathbf{X}_i$. The decoder is responsible for expanding the compressed representation of the input $\tilde{\mathbf{X}}_i$ and can be represented by Equation (6) [15].

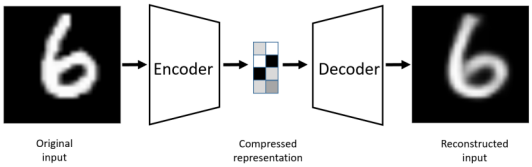


Fig. 6: Autoencoder structure [6]

$$\mathbf{H}_i = g(\mathbf{X}_i) \tag{5}$$

$$\tilde{\mathbf{X}_i} = f(\mathbf{H}_i) = f(g(\mathbf{X}_i)) \tag{6}$$

In this way, training an autoencoder seeks to find functions f and g that satisfy the Equation (7), where Δ is a loss function, which will penalize the network according to the difference between input and output [15].

$$\arg\min_{f,g} < [\Delta(\mathbf{X}_i, f(g(\mathbf{X}_i))] > \tag{7}$$

Autoencoders are categorized into different groups: Regularized Autoencoders, which are suitable for learning representations for classification tasks; Variational Autoencoders, capable of creating generative models [13]; Denoising Autoencoders, used to create filtered reconstructions of an input through noisy samples intentionally inserted into its training, to minimize the difference between a noisy input and the output [16]. The architecture chosen for the mitigation process was the two-dimensional convolutional Denoising Autoencoder, due to its specificity for removing noise, and its ability to evaluate the outputs in relation to the target value during its training process. The loss functions check the difference between an output sample $\tilde{\mathbf{X}}_i$ and an input sample $\mathbf{X}_i$, shown in Equation 8, the g and f functions are obtained in the dense layers of the network. The autoencoders work in such a way as to search for the weights that generate the smallest difference values between $\mathbf{X}_i$ and $\tilde{\mathbf{X}}_i$ according to some metric.

$$\mathbb{E}[\Delta(\mathbf{X}_i, g(f(\mathbf{X}_i)))] \tag{8}$$

The loss function of a denoising Autoencoder is commonly defined as a Frobenius Norm shown in Equation 9, where X and $\hat{X}$ are the clean and noisy input data respectively [17]. The Denoising Autoencoder structure is shown in Figure 7.

$$L_{DAE}(\mathbf{X}, \mathbf{X}') = \min(||\mathbf{X} - \mathbf{X}'||_F^2) \tag{9}$$

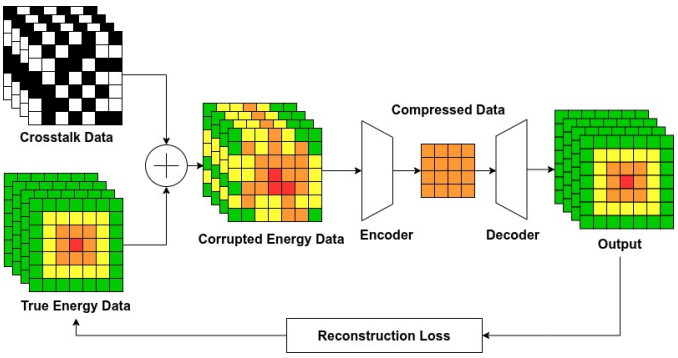


Fig. 7: Denoising Autoencoder structure

The implementation of the autoencoder was divided into several specific layers for both the encoder and the decoder. The input sample dimensions for the autoencoder are $7 \times 7 \times 1$.

- **Conv2D convolution** - The convolutional layers used in the network architecture have the function of extracting features from the input [18], using the 64 and 86 filters in the input convolution process of the encoder and after the pooling process, respectively.
- **Max Pooling MaxPooling2D** - The pooling layer uses the max pooling process, which extracts patches from the feature map, keeping the maximum value of each patch and discarding the rest [18].
- **Conv2D_Transpose deconvolution** - The primary purpose of the deconvolution process is to reconstruct the features that were extracted after the pooling and convolution operations carried out on the encoder. The output of this layer is the same size as the encoder output, followed by an upsampling layer that increases the data size.

The dimensions of the outputs of each network layer are described in Table I. A second deconvolution layer performs the feature reconstruction process now with the expanded data, reducing the number of filters to a single unit, returning the sample to its original dimension with a final cropping layer, where the final data have a dimension of $7 \times 7 \times 1$. This combination of layers to structure the encoder and decoder structures, as shown in [19], is able to learn to remove the noise present in the samples. The convolutional layers are responsible for detecting patterns present in the images, where the first convolutional layers are responsible for detecting local patterns. In contrast, the following convolutional layers are responsible for detecting broader patterns, which contain the patterns detected by the first layer [19]. The network architecture implemented to mitigate the effects of crosstalk is shown in Figure 8.

TABLE I: Hyperparameters and Configuration Details

Layer	Layer Dimension
Input	$7 \times 7 \times 1$
Convolution	$7 \times 7 \times 64$
Max Pooling	$4 \times 4 \times 64$
Convolution	$4 \times 4 \times 86$
Convolution Transpose	$4 \times 4 \times 86$
Up Sampling	$8 \times 8 \times 86$
Convolution Transpose	$8 \times 8 \times 1$
Cropping	$7 \times 7 \times 1$

VI. Results

The results were individually analyzed for each cell, which can provide a reasonable understanding of how the neural network carried out the mitigation process according to the energy level present in the cell and how close the predicted value is to the actual energy value. The energy values tested for the proposed network were in the order of 50 GeV for the total sum of energy in the cluster. The analysis presented in the research was carried out in the center cell of the 7×7 cluster, and the ring referring to the first neighborhood, the region

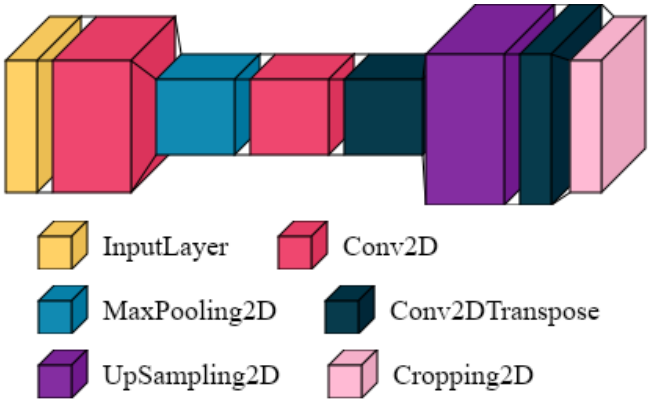


Fig. 8: Autoencoder architecture used

with the highest energy deposition in the simulated samples is the region of interest for analysis in this work, represented in Figure 9. The prediction test with Lorenzetti data was

-1,1	0,1	1,1
-1,0	0,0	1,0
-1,-1	0,-1	1,-1

Fig. 9: Center cell and the first neighborhood of cells

carried out using the ReLU activation function, described in Equation 10,a non-continuous and not centered on zero function that has a reduced computational cost because it does not apply exponential operations, forcing negative values to be canceled out [20]. This feature contributes to correcting an uncertainty present in the data from the Lorenzetti simulation for the outermost cells, which have negative energy values. The loss function used for the network proposed in this study was MSE, shown in Equation 11, where M is the number of observations in the training dataset [15].

$$f(x) = max(0, x) \begin{cases} x \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \tag{10}$$

$$L_{MSE} = MSE = \frac{1}{M} \sum_{i=1}^{M} |\mathbf{X}_i - \tilde{\mathbf{X}}_i|^2 \tag{11}$$

The comparison between the target value and the output predicted by the network was carried out using RMSE values, a metric used to check how closely a prediction made by a neural network comes to the expected value [21]. The RMSE error is calculated according to Equation 12, where the values of e_i are the error calculated on the sample i, over the n samples. The encoder kernel size used was 4×4, where the convolutional layers had 64 filters, while the deconvolutional layers had 86 filters. The validation interval used was 0.2, which means that 20 percent of the data is used for validation while the remaining 80 percent is used for training. The optimizer chosen for training was the Adam optimizer, which combines SGD (Stochastic Gradient Descent) functionalities with adaptive learning rates to accelerate convergence by preventing values from stalling at local minima [18].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} e_i^2} \tag{12}$$

TABLE II: Hyperparameters and Configuration Details

Parameter	**Value**
Activation Function	ReLU
Convolution Filters	64
Deconvolution Filters	86
Kernel Size	4×4
Batch Size	256
Optimizer	Adam
Validation Split	0.2
Loss Function	MSE
Folds	10
Epochs	20
Adam Parameters	**Value**
Learning Rate	0.001
β_1	0.9
β_2	0.9999

The network performance on the predictions was compared using two different scenarios. In scenario **A** the autoencoder predictions were made after using the optimal filter on the samples, while in scenario **B** the predictions were made on the samples before they are applied to the autoencoder, Figure 10 shows the two scenarios executed in the tests. The network was analyzed using the cross-validation process; in this process, the data are separated into folds where training is carried out using different regions of the data set. Validation was performed using 10 folds, where each fold was evaluated over 20 epochs. Figure 11 shows the loss curve for the validation and training set in Scenario A, where the samples used for training came from the data in the Amplitude files, obtained after processing by the optimal filter algorithm. Figure 12 shows the loss curve for the validation and training sets in scenario B, where the data used to train the network came from the Samples files, samples that were not subjected to the optimal filtering process.

The energy values resulting from the prediction processes of the proposed network were compared with the target value and the output of the optimal filter. In both scenarios, the RMSE value is also shown in relation to the target energy value. The results for Scenario **A** are shown in Table III and Table IV shows the results for Scenario **B**.

The histograms compare the energy values and the number of times they were recorded in the cells. With this, it can be seen that the network proposed in the article was able to attenuate the data corrupted by the crosstalk and obtain energy reading values closer to the true energy value. In the center

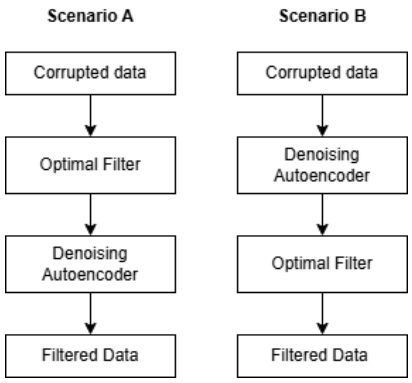


Fig. 10: Evaluated autoenconders approaches

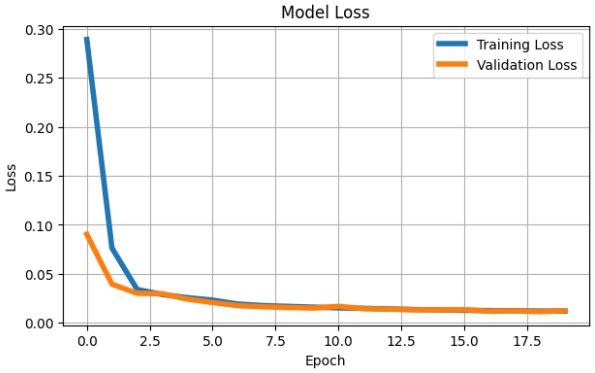


Fig. 11: Training and Validation Loss × Number of Epochs for Scenario **A**

cell, where the greatest amount of energy is deposited, it can be seen that the network was able to remove the influence of crosstalk, bringing the predicted energy values closer to the true energy, as can be seen in the overlapping graphs in Figure 13 for scenario **A** and Figure 14 for Scenario **B**. The RMSE value shows that there is a greater loss with the use of the optimal filter, while the use of the Autoencoder reduced this error value, especially in scenario **B**.

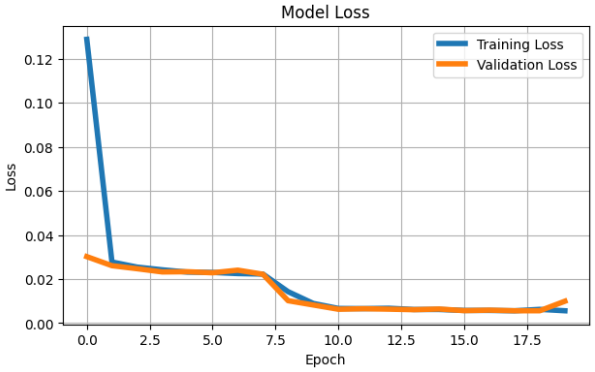


Fig. 12: Training and Validation Loss × Number of Epochs for Scenario **B**

TABLE III: Comparative results between Optimal Filter and Denoising Autoencoder in Scenario A.

Cell	Target Energy (GeV)	OF Energy (GeV)	DAE (Post OF) Energy (GeV)	RMSE DAE (GeV)	RMSE OF (GeV)
0,0	13.30 ± 2.02	13.80 ± 2.06	13.40 ± 2.00	0.056	0.072
-1,1	1.26 ± 1.04	1.75 ± 1.22	1.16 ± 1.18	0.356	0.285
0,1	6.12 ± 5.94	6.63 ± 6.00	6.21 ± 6.02	0.115	0.298
1,1	0.60 ± 0.42	1.05± 0.60	0.52 ± 0.53	0.317	0.328
-1,0	2.35 ± 0.34	3.05 ± 0.34	2.37 ± 0.33	0.046	0.230
1,0	1.04 ± 0.17	1.69 ± 0.18	1.04 ± 0.17	0.047	0.325
-1,-1	1.26 ± 1.04	1.76 ± 1.22	1.17 ± 1.18	0.356	0.285
0,-1	6.15 ± 5.93	6.66 ± 5.98	6.07 ± 6.18	0.413	0.299
1,-1	0.60 ± 0.43	1.06 ± 0.60	0.52 ± 0.53	0.316	0.327

TABLE IV: Comparative results between Optimal Filter and Denoising Autoencoder in Scenario B.

Cell	Target Energy (GeV)	OF Energy (GeV)	DAE (Before OF) Energy (GeV)	RMSE DAE (GeV)	RMSE OF (GeV)
0,0	13.30 ± 2.02	13.80 ± 2.06	13.50 ± 2.02	0.014	0.072
-1,1	1.26 ± 1.04	1.75 ± 1.22	1.28 ± 1.05	0.037	0.285
0,1	6.12 ± 5.94	6.63 ± 6.00	6.17 ± 5.97	0.049	0.298
1,1	0.60 ± 0.42	1.05 ± 0.60	0.57 ± 0.40	0.034	0.328
-1,0	2.35 ± 0.34	3.05 ± 0.34	2.40 ± 0.34	0.016	0.230
1,0	1.04 ± 0.17	1.69 ± 0.18	1.05 ± 0.17	0.013	0.325
-1,-1	1.26 ± 1.04	1.76 ± 1.22	1.28 ± 1.05	0.033	0.285
0,-1	6.15 ± 5.93	6.66 ± 5.98	6.21 ± 5.99	0.046	0.299
1,-1	0.60 ± 0.43	1.06 ± 0.60	0.44 ± 0.31	0.106	0.327

VII. Conclusion

Based on the results obtained with the network proposed in the article, it can be seen that the attenuation process carried out in the predictions made by the denoising autoencoder showed results close to the true energy values, significantly eliminating the influence of crosstalk. The energy values in the central cell and its surroundings after the denoising process carried out by the autoencoder proposed in this work show values that are close to the target values, and the analyzes carried out using the RMSE values show good performance in reducing the crosstalk contributions in the samples. The

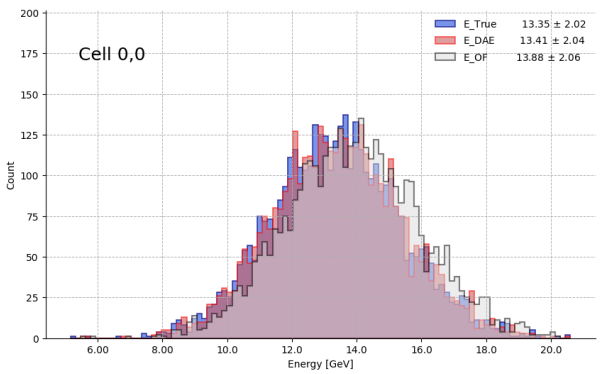


Fig. 13: Histogram with energy level for the center cell for scenario **A**

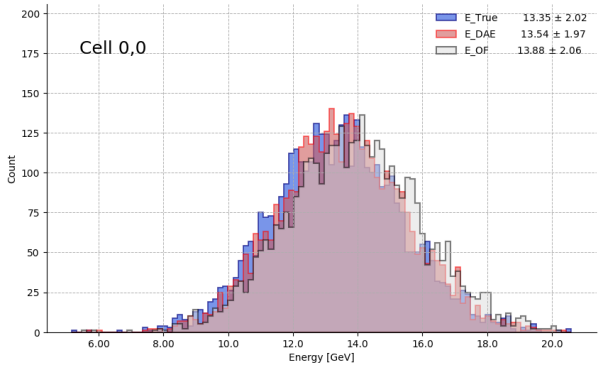


Fig. 14: Histogram with energy level for the center cell for scenario **B**

great advantage of this neural network model for the crosstalk mitigation process is that a real reference value is not needed for the denoising process to be carried out, and it is possible to train with simulated samples only. Some factors such as the uncertainties in the simulation process carried out by the Lorenzetti framework can influence the learning process, negative energy values, for example, can affect the way the network carries out the training process and subsequently makes predictions.

VIII. Acknowledgments

The authors thanks CNPQ and CAPES for the financial support.

References

[1] M. A. Moreira, "O Modelo Padrão da Física de Partículas," Rev. Bras. Ens. Fís., vol. 31, no. 1, pp. 1306.1–1306.11, Apr. 2009. [Online]. Available: https://doi.org/10.1590/S1806-11172009000100006

[2] M. S. dos Santos, "Computational Intelligence applied to Crosstalk Mitigation in a Liquid Argon Calorimeter," Ph.D. dissertation, Universidade Federal da Bahia, Salvador, Brazil, 2024. [Online]. Available: https://repositorio.ufba.br/handle/ri/41431

[3] A. Lopez Solis, "Contribution to the energy and time measurement of electrons and photons in the ATLAS experiment and search for dark matter production in association to a Higgs boson," Ph.D. dissertation, UPMC, Paris, France, Sep. 2017. [Online]. Available: https://cds.cern.ch/record/2291698?ln=pt

[4] M. Stricker, "Crosstalk measurement in the ATLAS electromagnetic calorimeter using muons," M.S. thesis, Physikalisches Institut, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany, Jun. 2023. [Online]. Available: https://www.pi.uni-bonn.de/brock/en/results/data/t00000059.pdf

[5] M. V. Araújo, M. Begalli, W. S. Freund, G. I. Gonçalves, M. Khandoga, B. Laforge, A. Leopold, J. L. Marin, B. S.-M. Peralva, J. V. F. Pinto, M. S. Santos, J. M. Seixas, E. F. Simas Filho, and E. E. P. Souza, "Lorenzetti Showers – A general-purpose framework for supporting signal reconstruction and triggering with calorimeters," Computer Physics Communications, vol. 286, p. 108671, 2023. https://www.sciencedirect.com/science/article/pii/S0010465523000164

[6] D. Bank, N. Koenigstein e R. Giryes, "Autoencoders," arXiv preprint arXiv:2003.05991, 2021. [Online]. Available: https://arxiv.org/pdf/2003.05991

[7] A. A. Abdelalim, "Study of the impact of cross-talk in the ATLAS electromagnetic calorimeter in the prediction of signals in the strip layer," M.S. thesis, Université de Genève, Geneva, Switzerland, 2008. [Online]. Available: https://cds.cern.ch/record/1374681/files/MASTER_ABDELALIM.pdf

[8] M. Delmastro, "Energy reconstruction and calibration algorithms for the ATLAS electromagnetic calorimeter," Ph.D. dissertation, CERN, 2003. [Online]. Available: https://cds.cern.ch/record/953119/files/thesis\protect\discretionary{\char\hyphenchar\font}{}{}2003\protect\discretionary{\char\hyphenchar\font}{}{}033.pdf

[9] M. S. dos Santos, P. C. M. A. Farias, E. F. Simas Filho, B. Laforge e J. M. Seixas, "Redes Neurais como um método de redução do Crosstalk entre sensores no calorímetro eletromagnético do experimento ATLAS," XV Congresso Brasileiro de Inteligência Computacional (CBIC 2021), Joinville, SC, Brasil, 2021. [Online]. Available em: https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC_2021_paper_147.pdf

[10] F. Hubaut, P. Pralavorio, B. Dekhissi, J. E. Derkaoui, A. El-Kharrim, and F. Maaroufi, "Crosstalk in production modules of the Electromagnetic Endcap Calorimeter," ATLAS Note ATL-LARG-2003-012, Oct. 20, 2003. [Online]. Available: https://cds.cern.ch/record/685566

[11] J. Colas, C. de La Taille, R. Lafaye, N. Massol, P. Pralavorio, D. Sauvage, and L. Serin, "Crosstalk in the ATLAS Electromagnetic Calorimeter"; preliminary version, CERN, Geneva, Switzerland, Rep. ATL-LARG-2000-004, 1999. [Online]. Available: https://cds.cern.ch/record/683952

[12] Python Software Foundation, "pickle — Python object serialization," Python 3.9.7 documentation, [Online]. Available:https://docs.python.org/3/library/pickle.html

[13] T. Perumal, N. Mustapha, R. Mohamed, and F. M. Shiri, "A comprehensive overview and comparative analysis on deep learning models," Journal on Artificial Intelligence, vol. 6, no. 1, pp. 301–360, 2024. [Online]. Available: http://dx.doi.org/10.32604/jai.2024.054314

[14] F. Hubaut, Crosstalk Measurements in the EM Barrel Module 0 from 99', May 00' and July 00' Beam Tests, CERN, Tech. Rep. ATL-LARG-2000-009, 2000. [Online]. Available: https://cds.cern.ch/record/684134

[15] U. Michelucci, "An Introduction to Autoencoders," arXiv preprint arXiv:2201.03898, 2022. [Online]. Available: https://arxiv.org/abs/2201.03898

[16] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". Journal of Machine Learning Research, 11(110), 3371–3408. Available: https://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf

[17] Berahmand, K., Daneshfar, F., Salehi, E.S. et al. "Autoencoders and their applications in machine learning: a survey". Artif Intell Rev 57, 28 (2024). Available: https://doi.org/10.1007/s10462-023-10662-6

[18] Yamashita, R., Nishio, M., Do, R.K.G. et al. "Convolutional neural networks: an overview and application in radiology". Insights Imaging 9, 611–629 (2018). Available: https://doi.org/10.1007/s13244-018-0639-9

[19] Feng, D.; Wang, X.; Wang, X.; Ding, S.; Zhang, H. "Deep Convolutional Denoising Autoencoders with Network Structure Optimization for the High-Fidelity Attenuation of Random GPR Noise." Remote Sens. 2021, 13, 1761. Available: https://doi.org/10.3390/rs13091761

[20] M. Gustineli, "A survey on recently proposed activation functions for deep learning," arXiv preprint, arXiv:2204.02921, Apr. 2022. [Online]. Available: https://arxiv.org/abs/2204.02921

[21] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature", Geoscientific Model Development, vol. 7, no. 3, pp. 1247–1250, 2014. [Online]. Available: https://gmd.copernicus.org/articles/7/1247/2014/