Application of adversarial learning for identification of radionuclides in gamma-ray spectra

Zakariya Chaouai, Geoffrey Daniel, Jean-Marc Martinez, Olivier Limousin, Aurélien Benoit-Lévy

Application of Adversarial Learning for Identification of Radionuclides in Gamma-Ray Spectra

Zakariya Chaouai[1], Geoffrey Daniel [*,1], Jean-Marc Martinez[1], Olivier Limousin[2], and Aurélien Benoit-Lévy[3]

[1]Université Paris-Saclay, CEA, Service de Thermo-hydraulique et de Mécanique des Fluides, 91191, Gif-sur-Yvette, France
[2]AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, F-91191, Gif-sur-Yvette, France
[3]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Abstract

The rapid and accurate identification of radionuclides brings crucial information for nuclear monitoring to diagnose unknown radiological scenes. Recent studies have used a deep learning approach based on neural networks to develop algorithms that perform well in terms of accuracy and computation time and can also identify radionuclides with a limited number of photons. However, it has been shown that conventional neural networks are not necessarily robust, in the sense that a small particular perturbation of the input data can mislead the networks. A specific learning procedure is necessary to overcome this lack of robustness. In this paper, we show that small perturbations intentionally injected into gamma-ray spectra, with respect to the Poisson statistics, are able to fool the network. We propose applying a robust learning procedure, called "adversarial learning". We evaluate this procedure using a CdTe detector, namely Caliste-HD. We train a Convolutional Neural Network (CNN) with a synthetic database composed of simulated spectra and we test its performance on real data acquired with a Caliste detector.

1 Introduction

In the context of nuclear safety and security, the diagnosis and monitoring of radiological scenes represent a major challenge: being able to identify the radionuclides present in the scenes. This identification must be fast, accurate, and reliable, especially in critical applications. The common way to achieve this identification is to detect individual X-ray and gamma-ray photons emitted by radioactive sources and to measure their energies to build a spectrum. The resulting spectrum is a signature of the isotopes mixture and its analysis leads to their identification.

Classic approaches have been developed to perform this analysis automatically, such as peak fitting algorithms, which focus on radionuclide emission lines [1, 2], or model fitting algorithms, which aims to use the entire spectral information by fitting a combination of spectral models [3, 4]. Recently, deep learning approaches based on neural networks have been successfully applied to gamma-ray spectroscopy [5, 6, 7].

*Corresponding author: geoffrey.daniel@cea.fr

Their performance in other fields, such as computer vision [8, 9, 10], or natural language processing [11, 12] have been a motivation to overcome the difficulties of gamma-ray spectra analysis by classic methods, such as low photon counting statistics or complex mixture of radionuclides.

However, studies have shown that neural networks are very sensitive to adversarial attacks: small perturbations of the input data which are specifically designed to mislead the network [13, 14, 15]. This lack of robustness is a major weakness that must be solved for critical applications, such as nuclear safety and security, which requires a certification of the tools used in this context. Procedures have been proposed to improve the robustness of the neural networks and make them less sensitive to small perturbations in the input data [16]. In the present paper, we show that neural networks used in gamma-ray spectroscopy are also sensitive to this problem, and that we can compute small perturbations in the spectrum, according to the Poisson statistics, that are sufficient to fool the neural network. We show that an adversarial learning procedure is helpful to avoid this problem. We evaluate this approach on real data registered with a CdTe detector, Caliste-HD, dedicated to the detection of high-energy photons.

The paper is structured as follows: Section 2 describes the Caliste detection system and presents the deep learning approach we use as a baseline. Section 3 presents the method for computing the specific small perturbations that can mislead our neural network and we describe the adversarial learning procedure to ensure the robustness of the neural network. In Section 4, we evaluate the performance of this method on a real data set.

2 Caliste-HD detector and classic learning

2.1 Caliste-HD

Caliste-HD [17] is a miniature detector for high-energy photons counting and imaging spectroscopy. Its sensitive area is made of a monolithic pixelated CdTe crystal, 1-mm thick, with 16×16 pixels and 625 µm pixel pitch. Its readout electronic, the ASIC IDeF-X HD [18], ensures a low electronic noise and low power consumption, that are advantageous for the spectrometric performances of the detector. Caliste-HD achieves a resolution of 700 eV FWHM at 60 keV and 4.1 keV FWHM at 662 keV, with an energy range from 2 keV to 1 MeV (single pixel events). These spectroscopic features are relevant to perform spectro-identification of radionuclides, and the data of this study are based on this detection system.

2.2 Datasets

2.2.1 Training and validation set

Our training set is constructed by a GEANT4 Monte-Carlo simulation associated to the computed detector response. The details of the simulation can be found in [7]. We simulated the spectra of six radionuclides that are available in our laboratory: ^{241}Am, ^{133}Ba, ^{57}Co, ^{137}Cs, ^{152}Eu, ^{22}Na. The advantage of this synthetic database is the possibility to simulate other radionuclides that are not available in the laboratory. The training database contains 200 000 examples of gamma-ray spectra with 2000 channels, sampled uniformly from 0 keV to 1000 keV with 0.5 keV channel widths. Each spectrum contains a mixture of radionuclides among the six simulated ones, with a random number of photons N_{photons}, ranging from a few tens to several million photons, according to Equation 1. Thanks to this method, the training database contains spectra with low counting statistics of photons as well as very high counting statistics.

$$N_{\text{photons}} = 10^{7x+1}, \quad x \sim \text{Unif}([0,1]) \tag{1}$$

Before starting the learning of models, it is necessary to normalize each spectrum. A normalization by the total number of photons can be considered. However, since the photons with the highest energies have much lower probabilities of interacting in the detector than those with lower energies, it creates an imbalance in the spectrum where the photons with lower energy are very visible, unlike high energy ones. Hence, it is preferable to use a logarithmic normalization. For a spectrum s, we note s_i the photons number in the bin number i. The first step is to add one count for each bin, so that the logarithmic function will be well defined, then we apply the logarithmic function given by Equation 2:

$$s_i^{'} = \log\left(\frac{s_i + 1}{\sum_j (s_j + 1)}\right) \tag{2}$$

Since all components of the vector $s_i^{'}$ are negative, they should ideally be readjusted between zero and one, which can be done following Equation 3:

$$\hat{s_i} = -\frac{s_i^{'}}{\min(s^{'})} + 1 \tag{3}$$

where $\hat{s}$ is the normalized spectrum that we will use as the input of our neural networks.

The expected output of the neural network is represented by a vector Y_c with six components, corresponding to the six radionuclides. Each component is a binary variable, which is equal to 1 if the corresponding radionuclide is present in the spectrum and 0 if not.

We use also a validation dataset, which is a part of our training set. More precisely, we take 20% of our training set and we use it to give us an estimate of model performance, while adjusting the model's hyperparameters such as the number of layers and the number of neurons in our model.

2.2.2 Test set

In order to evaluate the performances of the neural networks in real conditions, we built a test set from real data acquired with the Caliste-HD detector. We took a long, well-calibrated acquisition for the sources available in the laboratory, in order to obtain large photon count statistics. The available sources correspond to the six simulated radionuclides. The photon counts are recorded in a list so that we can construct spectra by selecting any statistic of interest to evaluate sensitivity and to arbitrarily create a virtual mixture by selecting events from the photon lists. We constructed six sets of test spectra each with a fixed count statistic: 100, 1 000, 10 000, 100 000, 1 000 000, 10 000 000 photons. Each set contains 10 000 example spectra. We apply the same normalization to the test set spectra before applying the neural networks.

2.3 Classic Convolutional Neural Network approach

In this study, we use a convolutional neural network (CNN) [19] with multiple convolutional hidden layers in one dimension (Conv1D). Every convolution layer is followed by a spatial dropout [20] in one dimension (SpatialDropout1D), a batch normalization [21] and a max-pooling layer [22] in one dimension (MaxPooling1D) with size equal to two. The motivation for using a convolution structure is that gamma spectra possess local structures, such as photoelectric peaks or Compton continua, that can be extracted by the convolution operations, as described in [7]. We add two fully-connected layers after the convolutional blocks, in order to perform the identification of the radionuclides. The activation function used in all hidden layers is the Rectified Linear Unit (ReLU) function. For the output layer, we use the sigmoid function, which is relevant for our multi-class classification problem, and the corresponding loss function is the binary cross-entropy given in Equation 4:

$$\ell(x, y; \theta) = -\sum_{j=1}^{L} [y_j \log(f_\theta(x)_j) + (1 - y_j) \log(1 - f_\theta(x)_j)] \tag{4}$$

where x represents the input data, y the expected ground truth; j is the index for each class, L the number of classes and f_θ is the output of the neural network with some parametrization θ.

We implement our model with TensorFlow version 2.2.0 [30]. In the learning part, Adam [23] was used as the optimization algorithm with a learning rate equal to 10^{-4} and a rate decay equal to 10^{-5}. We trained on mini-batches with a size of 2 000 inputs and we use early stopping to stop the learning process: we evaluate the loss function on the validation set at each iteration, and we stop the learning if the cost function does not decrease for 5 successive iterations.

To evaluate the performances of our neural network, we choose the binary accuracy metric, which measures how often the model gets correct predictions. It is defined in Equation 5:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

where TP represents the number of true-positive identifications, TN for true negative, FP for false positive, and FN for false negative. We consider each radionuclide independently, which means that, for one example, if the neural network fails to classify correctly only one radionuclide among the 6 radionuclides, we do not consider it as a totally wrong answer, since the accuracy for this example is 5/6. The performances of this neural network in terms of accuracies are described in section 4.

A source is considered to be detected if the predicted sigmoid output given by our model is higher than a threshold of 50 %. However, the study of the effect of the threshold [24] is useful to tune the correct threshold according to the detector, which is not further investigated in the present paper.

3 Adversarial perturbations and adversarial learning

Recently, a severe weakness has been discovered by the research community in deep learning. It has been demonstrated that deep neural networks are vulnerable to adversarial examples [16, 25]. An adversarial example corresponds to an example, originally correctly classified by the network, which has been modified by a small[1] perturbation and it is not recognized by the network. It is important to mention that adversarial examples are unavoidable and universal by definition: one can always build an additive noise at input to make the model misclassify an example. The vulnerability to adversarial inputs can be problematic and even prevent the application of deep learning methods in security and safety critical applications. In our application, we show that we can build adversarial examples whose perturbation follows the Poisson statistics of the photon counting.

3.1 Computation of the perturbations

An adversarial example can be represented as follows: given an original input x, x_{adv} is an adversarial example of x if $x_{\text{adv}} = x + \delta$, such that δ is a small value, which means that x_{adv} is slightly different from x, and the prediction of x_{adv} by a considered model is different from the prediction of x.

Since their discovery by Szegedy et al. [25], several methods have been proposed to generate adversarial examples to fool a trained model. Most of those adversarial examples are referred to as Adversarial Attacks.

[1]The notion of "small" perturbation is relative to the data and the application.

In the literature, there are several types of attacks such as Fast Gradient Sign Method (FGSM) [16], Basic Iteration Method (BIM) [13], Projected Gradient Descent (PGD) [14]. The survey [26] describes these algorithms of attack and introduces also other algorithms.

In this study, we use an adaptation of the BIM algorithm. We denote $\ell(x, y; \theta)$ the loss function between the ground truth y associated with the example x and the associated prediction $f_\theta(x)$ of the neural network with some parametrization θ. The loss function is minimal when the prediction of the neural network allows the best reconstruction of the ground truth by the model f_θ. The computation of the adversarial example consists of finding a perturbation δ of the input x, which maximizes the loss function, so that the prediction of the neural network is far from the ground truth. Mathematically, it consists in solving the problem given by Equation 6:

$$\delta^* \in \underset{\delta \in \mathcal{B}(x)}{\operatorname{argmax}}\, \ell(x + \delta, y; \theta) \tag{6}$$

where $\mathcal{B}(x)$ represents some constraints on the perturbation δ, so that this perturbation can be considered small, relative to the input x. δ^* is the optimal perturbation that affects the neural network's prediction. In our application, we consider a small perturbation if it can be considered as noise measurement in the acquisition, given by the Poisson fluctuation of the photons detection. It consists in three constraints:

1. A first local constraint linked to the Poisson's statistics for each bin i independently:

$$\forall i \in [\![1\ ;\ N_{\text{bin}}]\!],\ \ |\delta_i| < \sqrt{x_i} \tag{7}$$

2. A global constraint linked to the Poisson's statistics of the complete measurement:

$$|\sum_{i=1}^{N_{\text{bin}}} \delta_i| \leq \sqrt{\|x\|_1} \tag{8}$$

   $\|x\|_1$ corresponds to the total number of photons in the spectrum and this constraints means that the total variation of the number of photons must be bounded. We point out, that the constraint (7) does not imply the constraint (8) since $\sqrt{\sum_{i=1}^{N_{\text{bin}}} x_i} \leq \sum_{i=1}^{N_{\text{bin}}} \sqrt{x_i}$.

3. A final constraint is related to the fact that each spectrum is corresponding to the number of photons, each component must be an integer number. Consequently, the perturbation δ follows the same constraint:

$$\delta \in \mathbb{Z}^{N_{\text{bin}}} \tag{9}$$

The first two constraints correspond to 1-σ Poisson noise deviation on the perturbation amplitude. We have also tested 2 and 3-σ deviations, but 1-σ was already sufficient to show a significant degradation of the neural network's performances on the adversarial examples and we focus on this constraint in our study. Further works will focus on harder attacks (2 and 3-σ and more).

The BIM algorithm approach consists of using a gradient ascent algorithm to solve the problem given by Equation 6. It is based on the computation of the gradient $\nabla_x \ell(x, y; \theta)$ of the loss function with respect to the inputs[2] of the neural network. This gradient ascent is iterative and it uses at each iteration the sign of the gradient, so that it identifies the direction to disturb for each component. We adapted this algorithm in Algorithm 1 in order to respect the constraints 7, 8 and 9. We apply $T = 15$ iterations of gradient ascent

Algorithm 1 Adversarial attack

Input: A pattern $x \in \mathbb{N}^m$ associated to the target variable $y = (y_1, ..., y_L) \in \{0,1\}^L$. Initial model $f_\theta(x) = (f_\theta(x)_1, ..., f_\theta(x)_L) \in [0,1]^L$.
Hyperparameter: the maximum number of iterations T, the prediction threshold α (0.5 for our case).
Initialization:
$x_{\text{norm}} \leftarrow \text{normalization}(x)$ (given by Equation 3)
$x_{\text{adv}} \leftarrow x$
$t \leftarrow 0$
while $t < T$ and $\prod_{i=1}^{L}(y_i - \texttt{boolean}(f_\theta(x_{\text{norm}})_i > \alpha)) == 1$ **do**
  $\delta \leftarrow \frac{\sqrt{x}}{T} \odot \text{ sign } (\nabla_{x_{\text{norm}}} \ell(x_{\text{norm}}, y; \theta))$ (the factor $\frac{\sqrt{x}}{T}$ ensures the local constraint 7)
  $x_{\text{new}} \leftarrow x_{\text{adv}} + \delta$
  $x_{\text{int}} \leftarrow \text{int}(x_{\text{new}}) + (x_{\text{new}} > x)$
  if $(|\sum_{i=0}^{m} x_{\text{int}_i} - x_i| \leq \sqrt{\sum_i^m x_i})$ **then**
    $x_{\text{adv}} \leftarrow x_{\text{new}}$
  else
    break (the iterations stop if the global constraint 8 is no longer respected)
  end if
  $x_{\text{norm}} \leftarrow \text{normalization}(x_{\text{adv}})$
  $t \leftarrow t + 1$
end while
Return: $x_{\text{int}} = \text{int}(x_{\text{adv}}) + (x_{\text{adv}} > x)$ (this last operation ensures the constraint 9)

and we stop the iterations if the neural network gives a wrong prediction for one of the radionuclides or if the constraints are not ensured.

Algorithm 1 operates a non-targeted attack, which means that we do not aim to fool the neural network on a specific radionuclide. However, it can be adapted to target a specific radionuclide and force the neural network to make a false prediction about that radionuclide, regardless of its prediction for other radionuclides. In the frame on this study, all the results are obtained by non-targeted attacks.

3.2 Example

In Figure 1, we attack a spectrum from the test database, from real acquisitions with Caliste, using our attack algorithm (1) and use our classic model to compare the prediction results on the original spectrum and the attacked spectrum. The predictions obtained on the original spectrum are correct. Specifically, the classic model gives probabilities that exceed 80 % for all five radionuclides existing in the spectrum, although the one radionuclide that does not exist, ^{152}Eu, is associated with a probability of about 35 %, well below the 50 % threshold. The predictions obtained from the adversarial spectrum by the classic model have changed radically. The probability of the existence of ^{57}Co decreased from 90 % to 5 %, which is a radionuclide that actually exists in the spectrum, and the probability of the existence of ^{152}Eu increased from 35 % to almost 90 %, which is a radionuclide that does not exist in the spectrum. Because of this drastic change in predictions, this result shows that we have successfully fooled our classic model.

Figure (2) is an illustration of the adversarial noise added to the spectrum in figure (1) by our adversarial

[2]In a classic learning process, the parameters of the neural network are updated by using the gradient $\nabla_\theta \ell(x, y; \theta)$ of the loss function with respect to the parameters.

attack. As we can see, the type of perturbation added to the spectrum is revealed by the addition or extraction of one or two photons from the spectrum in certain energy bins. This perturbation respects the constraints of the Poisson statistics. We can say that this perturbation is soft and could have been randomly obtained by some misfortune in the same acquisition. A full analysis is performed in Section 4 on the impact of these adversarial perturbations on the accuracy of the network.

3.3 Adversarial learning

This sensitivity to adversarial perturbation requires the introduction of methods to prevent this effect. Until now, two methods exist to harden neural networks against adversarial perturbations. The first one is regularization methods that penalize noise expansion throughout the network [27, 28, 29]. The second method is the adversarial learning [16, 13, 14] which consists of using adversarial examples generated from the training data set to increase robustness locally around the training samples. In this paper, we use this method to create a robust model.

Mathematically, the objective of a classic learning (i.e. without specific defense against adversarial perturbations) is to find, among all the possible parameters Θ, the optimal parameters θ^* that minimizes a total loss function, which is the mean of the individual loss functions $\ell(x, y; \theta)$ for each example (x, y) of the dataset $\mathcal{D}$ containing N examples, as given in Equation 10:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{D}} \ell(x^{(i)}, y^{(i)}; \theta). \tag{10}$$

Adversarial learning consists of learning on the adversarial examples computed through the adversarial perturbation, as described in section 3.1. This learning procedure is typically presented as a robust min-max optimization problem, given by Equation 11. The adversarial perturbations must satisfy certain constraints given by $\mathcal{B}(x^i)$.

$$\theta^*_{\text{adv}} = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{D}} \max_{\delta \in \mathcal{B}(x^{(i)})} \ell(x^{(i)} + \delta, y^{(i)}; \theta). \tag{11}$$

The learning is usually processed using an optimization algorithm based on gradient descent on mini-batches. It is important to note that at each iteration of the optimization process, the neural network parameters are updated, and it is necessary to compute the adversarial perturbations with respect to these new parameters at each iteration. This step represents additional computation time, which can be non-negligible depending on the application.

We apply this learning procedure to our convolutional neural network with our training data set, based on simulated spectra. As in classic learning, we use the early stopping method to stop the learning iterations. The results are presented in section 4 for the original test data set and the corresponding adversarial examples.

4 Results

The performance evaluation of the network is computed using the binary accuracies of the neural network as defined by Equation 5. We use the test data set, constructed from real measurements, with a controlled number of photons to analyze the performance based on the counting statistics. We evaluate four configurations:

1. the classic model tested on the original test data set, this configuration is considered as a baseline with classic learning and no perturbations added to the test examples;

2. the classic model tested on an adversarial test data set, the adversarial examples are specifically created to fool the classic model;

3. the robust model, trained by adversarial learning, tested on the original test data set;

4. the robust model tested on an adversarial test data set, with the adversarial examples specifically created to fool the robust model.

Figure 3 shows the overall accuracy of the neural network for all four configurations. The baseline accuracy of the classic model on the original examples increases from over 90 % with 100 photons to over 98 % with 10 000 000 photons. As described in Ref. [7], the misclassified radionuclides are likely due to radionuclides in low proportions in the spectra, whose signal is masked by the other radionuclides. Predicting on the adversarial examples, the accuracy drops down to between approximately 80 % and approximately 88 %, which is an absolute difference of 10 % in accuracy. This result means that a small perturbation, which can be considered as Poisson noise in the measurement, fools the neural network for 10 % of the examples. Our robust model performs similarly to the classic model on the original examples. This result should be verified: our robust learning does not degrade the identification performance of the network. For some counting statistics, 100 photons and 1 000 000 photons, we can slightly distinguish an improvement, but it is not significant. On the adversarial examples, the improvement is very clear, the accuracy increases from 83 % for 100 photons to almost 97 % for 10 000 000 photons. The adversarial learning process makes the model less sensitive to adversarial perturbations. The problem is not fully corrected, but with a spectrum containing at least 1 000 photons, the number of adversarial examples that can fool the network is reduced by a factor of approximately 5 to 10.

Finally, we have studied the model performances by analyzing the classification accuracy for each radionuclide independently. The results are given in Figure 4. As we mentioned above, the classic and the adversarial models have similar performances. From 10 000 photons, the accuracies for all radionuclides obtained from both models are above 95 % except for ^{137}Cs, where both models obtain accuracies above 95 % from 1 000 000 photons. This effect can be explained by the emission lines of the ^{137}Cs, especially its discriminant line which is at high energy, 662 keV, where our CdTe cristal is less efficient for the photoelectric effect. Furthermore, we observe that the performance of the adversarial model exceeds the classic model with respect to ^{137}Cs, which shows that the regularization due to the adversarial learning can have a positive effect on the model performances on the original examples. There is a small reduction of the performances of the robust network for the ^{241}Am but it is not significant, and also for ^{133}Ba with 1 000 photons. For the adversarial examples, as mentioned above, the robust model is less sensitive to the adversarial perturbations than the classic model, for all the radionuclides. We also observe that some curves are not monotonic with respect to the number of photons in the spectrum. This effect is due to the creation of the adversarial examples, which consists of non-targeted perturbations, i.e. we do not try to specifically mislead the prediction for a particular radionuclide. Consequently, it appears for instance that the perturbations of the neural network bring to wrong prediction for ^{152}Eu more likely at high counting statistics than low counting statistics. This can be explained by the fact that at high count statistics, the ^{152}Eu spectra have more noise-sensitive structures, with many emission lines at high energies, whereas at low count statistics, only the most dominant peaks appear, and the rest of the spectrum is less easy to perturb to fool the neural network.

5 Conclusion and outlooks

The sensitivity of neural network based on deep learning models to adversarial perturbations represents a strong weakness of these approaches. Radionuclide identification using gamma-ray spectroscopy and deep

learning algorithms also faces this type of problem with perturbations that can be so small compared to the original acquired spectrum that they can be mistaken for Poisson noise. These specific adversarial perturbations result in a non-negligible reduction in accuracy, on the order of 10%, which can have important consequences for critical applications, such as nuclear safety and security. Even if the probability of randomly acquiring such a spectrum is low, it is relevant to have a method to mitigate this problem in order to have a reliable algorithm. It is important to notice that the tests in this study are done in a controlled environment of a laboratory. In real nuclear safety situations, the environment would be far more complex with shielding or scattering materials around the radioactive sources and would affect the acquired spectra. This complexity could possibly imply a higher sensitivity to specific adversarial attacks.

Adversarial learning is an interesting solution that reduces the probability of finding an adversarial perturbation that can mislead the neural network by a factor of 5 to 10 in our application. In this work, we focus on non-targeted perturbations, but in future work we will study this effect with targeted perturbations, which aim to deceive a specific radionuclide. This approach would be of interest for applications that require high precision on specific radionuclides. In addition, we impose constraints that the perturbations are statistically equivalent to Poisson noise, so that they can originate from the stochastic processes of the acquisition. However, we can imagine some intentional attacks, if someone is able to add specific materials to the environment, such as a specific shielding or scattering material, in order to fool the network. This could be a breach of security systems for monitoring radioactive sources. This approach would require the ability to design constraints on disturbances that could be physically applicable by a human to fool the network.

References

[1] G. W. Phillips and K. W. Marlow, Automatic analysis of gamma-ray spectra from germanium detectors, Nucl. Instrum. Methods 137, 525, 1976.

[2] R. Gunnink and R. Arlt, Methods for evaluating and analyzing CdTe and CdZnTe spectra, Nucl. Instrum. Methods Phys. Res. A, vol. A458, pp. 196-205, 2001.

[3] C. J. Sullivan and J. Stinnett, Validation of a Bayesian-based isotope identification algorithm, Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 784, pp. 298-305, Jun. 2014.

[4] C. Bobin, O. Bichler, V. Lourenco et al., Real-time radionuclide identification in c-emitter mixtures based on spiking neural network. Appl. Radiat. Isot. 109, pp. 405–409, 2016.

[5] M. Kamuda, J. Stinett and C. Sullivan, Automated isotope identification algorithm using artificial neural networks, IEEE Trans. Nucl. Sci., vol. 64, no. 7, pp. 1858—1864, Apr. 2017.

[6] M. Kamuda, J. Zhao and K. Huff, A comparison of machine learning methods for automated gamma-ray spectroscopy, Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 954, Art. no. 161385, Feb. 2020.

[7] G. Daniel, F. Ceraudo, O. Limousin, D. Maier and A. Meuris, Automatic and real-time identification of radioisotopes in gamma-ray spectra: A new method base on convolutional neural network trained with synthetic data set. IEEE Trans. Nucl. Sci., vol. 4, pp. 644–653, 2020.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.

[9] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (ICLR), pp. 1-14, 2015.

[10] J. Redmon and A. Farhadi, Yolo9000: Better, faster, stronger, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263-7271, 2017.

[11] G. Saon, H. K. J. Kuo, S. Rennie, and M. Picheny, The ibm 2015 english conversational telephone speech recognition system, Seventeenth Annual Conference of the International Speech Communication Association, 2016.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems, pp. 3104–311, 2014.

[13] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world.arXiv preprint arXiv:1607.02533, 2016.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu,Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083, 2017.

[15] S.M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582, 2016.

[16] I. Goodfellow, J. Shlens and C. Szegedy. Explainingand harnessing adversarial examples.arXiv preprint arXiv:1412.6572. 2014.

[17] A. Meuris, O. Limousin, O. Gevin, F. Lugiez, I. Le Mer, and F. Pinsard, Caliste HD: A new fine pitch Cd(Zn)Te imaging spectrometer from 2 keV up to 1 MeV, in Proc. IEEE Nuclear Sci. Symp. Conf. Rec., pp. 4485—4488, 2011.

[18] O. Gevin, O. Lemaire, F. Lugiez, A. Michalowska, P. Baron, O. Limousin and E. Delagnes, Imaging X-ray detector front-end with high dynamic range: IDeF-X HD, Nuclear Instruments and Methods in Physics Research A 695, pp. 415–419, Dec. 2012.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324 , 1998.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[21] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, Int. Conf. on Mach. Learn., vol. 37 2015.

[22] A. Giusti, D. C. Cireșan, J. Masci, L. M. Gambardella, and J. Schmidhuber, Fast image scanning with deep max-pooling convolutional neural networks, ICIP, pp. 4034-4038, 2013.

[23] D.P. Kingma and J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR), 2015.

[24] M. Kamuda, J. Stinett and C. Sullivan, Automated isotope identification algorithm using artificial neural networks, IEEE Trans. Nucl. Sci., vol. 64, no. 7, pp. 1858-1864, 2017.

[25] Ch. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow,and R. Fergus. Intriguing properties of neural networks, Int. Conf. on Learn. Repres., 2014.

[26] K. Ren, T. Zheng, Z. Qin, and X. Liu, Adversarial attacks and defenses in deep learning, Engineering, vol. 6, no. 3, pp. 346–360, 2020.

[27] D. Jakubovitz, R. Giryes. Improving DNN robustness to adversarial attacks using Jacobian regularization, Proceedings of the European Conference on Computer Vision (ECCV), pp. 514–529, 2018.

[28] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] D. Varga, A. Csiszárik, and Z. Zombori, Gradient regularization improves accuracy of discriminative models, arXiv:1712.09936, 2017.

[30] Abadi et. al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

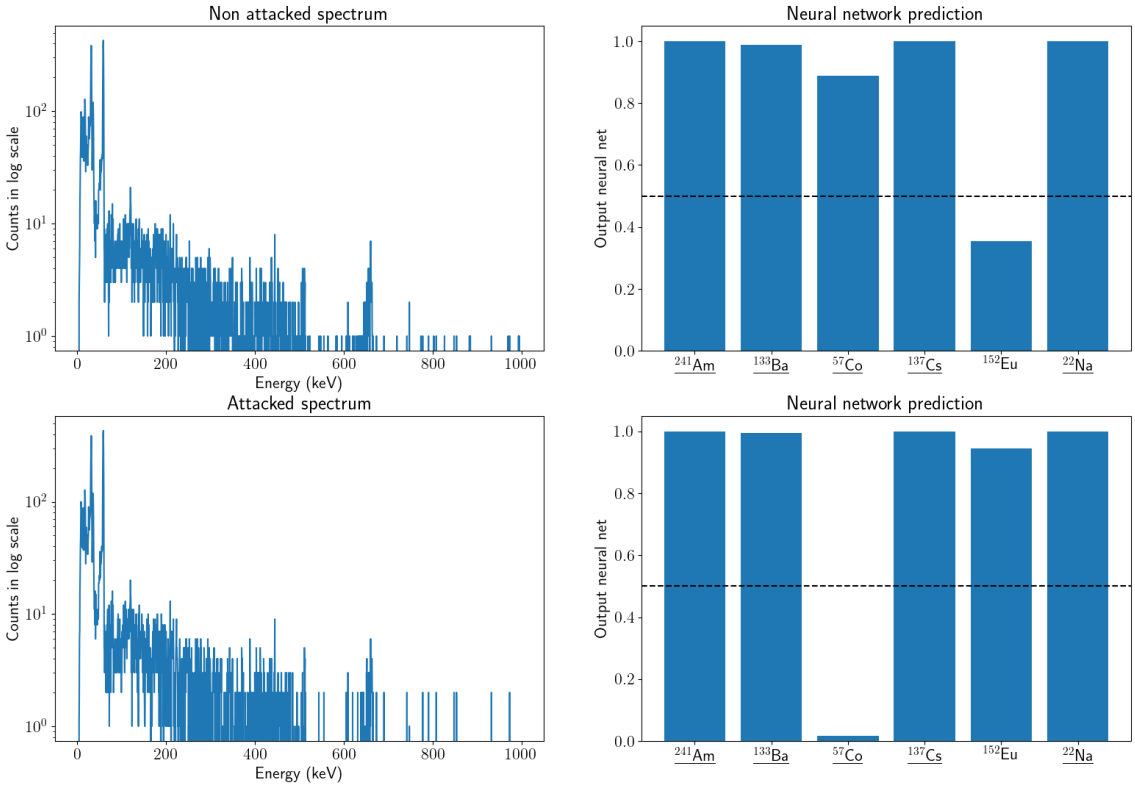

Figure 1: Example of a non-attacked and attacked spectrum and their predictions results gave by a classic model. The dashed lines represent the detection threshold at 50%. The radionuclides that really exist in the spectrum are underlined.

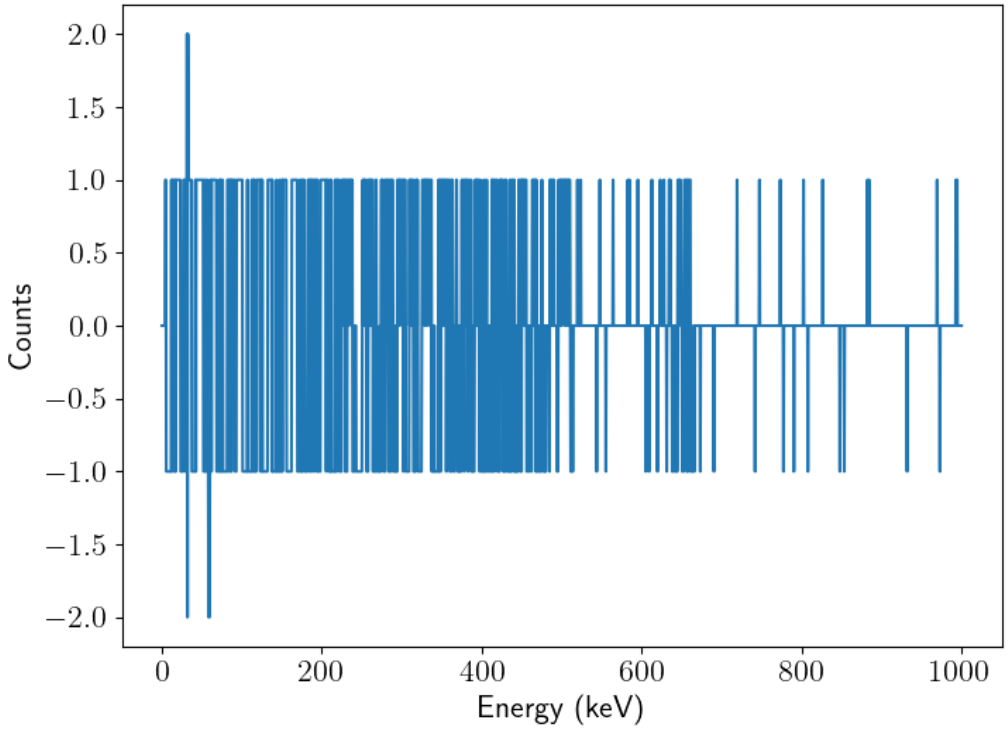

Figure 2: Adversarial perturbation corresponding to the example in Figure 1

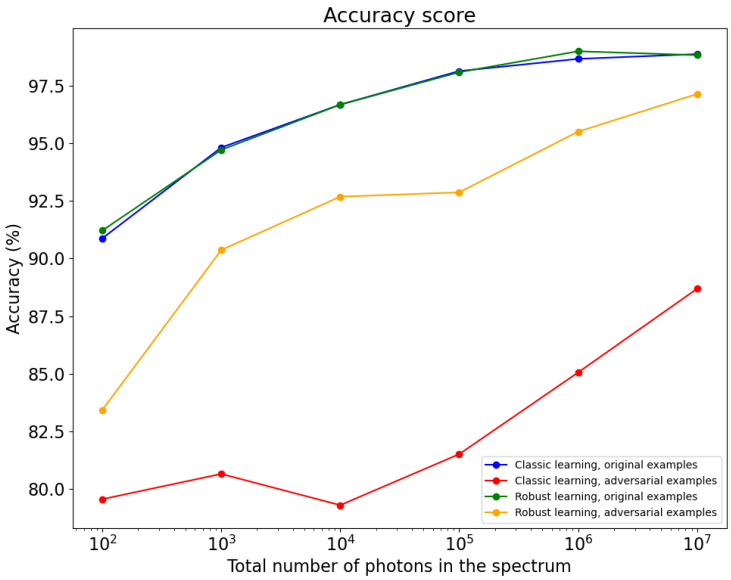

Figure 3: Global accuracies of two models with respect to number of photons, on original and adversarial examples.

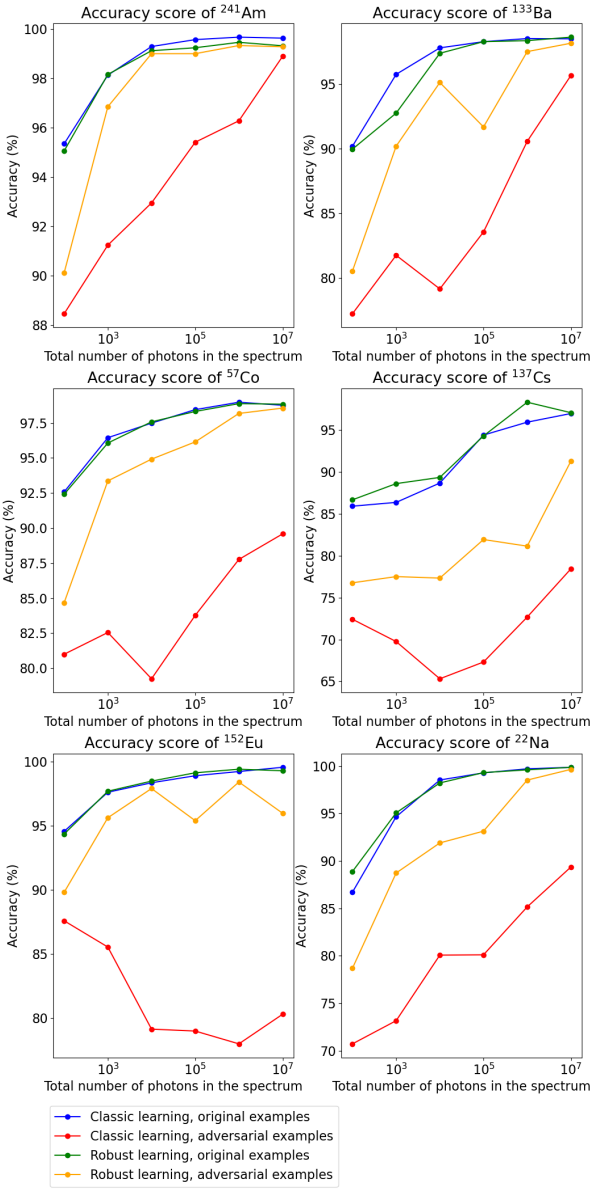

Figure 4: Independent radionuclide accuracies of the models with respect to the number of photons.