PROCEEDINGS OF SCIENCE

A neural network to classify GRAND radio time traces

Sandra Le Coz,[a,*] **Aurélien Benoit-Lévy**[c] **and Olivier Martineau-Huynh**[a,b]

[a] *Sorbonne Université, CNRS/IN2P3, Laboratoire de Physique Nucléaire et de Hautes Energies, LPNHE, 4 place Jussieu, F-75005 Paris, France*

[b] *Sorbonne Université, CNRS-UMR 7095, Institut d'Astrophysique de Paris, 98 bis boulevard Arago, F-75014 Paris, France*

[c] *Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France*

E-mail: lecozsandra@yahoo.com, aurelien.benoit-levy@cea.fr, martineau@lpnhe.in2p3.fr

GRAND is a Ultra High Energy (UHE) cosmic particles detection project, consisting in a giant, self-triggered, antenna array. Wherever we decide to setup the antennas apart from polar areas, we will generally face a high rate of background signals, orders of magnitude higher than the rate of extensive air showers. To avoid the saturation of the acquisition, we need to reject a significant part of the data at the antenna level, with a more sophisticated method than a basic peak-over-baseline selection. We present here an attempt to discriminate air showers and background radio time traces with a convolutional neural network, using experimental data rather than simulations. These data were produced with TREND, a self-triggered 50-antennas array, which was the seed for the GRAND project. At the antenna-level, for a set of signals that had triggered TREND, it was possible to reject 82% of the ultra-dominant background, while preserving 86% of the air shower signals.

9th International Workshop on Acoustic and Radio EeV Neutrino Detection Activities - ARENA2022
7-10 June 2022
Santiago de Compostela, Spain

[*] Speaker

PoS(ARENA2022)041

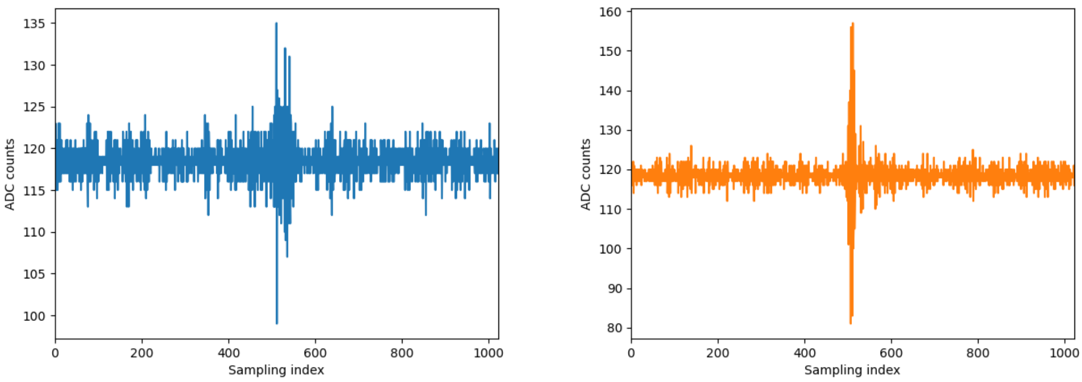

Figure 1: TREND traces, recorded at 5ns sampling. Left was labeled "background", right was labeled "air shower".

1. Introduction

The sources of UHE cosmic rays are still a mystery. But UHE neutrinos are produced in the same sources than cosmic rays are, with the advantage of being undeflected by magnetic fields. Their detection is then relevant to point to these UHE sources. The GRAND project[1] aims at the detection of the earth-skimming UHE tau neutrinos that interact in the rock via charge current, generating taus which then decay into the air, leading to air showers. To get a chance to detect neutrino events despite their low cross-section of interaction with matter, a huge area of detection is necessary. Antennas are a good option because they are cheap, and 1 km^{-2} antenna density is enough thanks to the large footprint of the very inclined air showers induced by neutrinos. Furthermore, for cost efficiency, antennas must be self-triggered, and thus be able to reject background signals to avoid the saturation of the acquisition. Yet, the trigger rate of background signals that were recorded by TREND[2], the seed-project for GRAND, was in average always above 10Hz, despite its remote location in the Tianshan mountains. For the sake of the future GRAND array, we have decided to use the TREND experimental data to study the potential of neural networks to reject background signals at trigger level. Neural networks are indeed theoretically able to find the best combination of relevant features to perform the discrimination. TREND data are 5.120 μs length time traces, digitized at 200MS/s, such as the ones shown in figure 1. A neural network would then be fed with 1024 features per input trace. The target this neural network would have to reach is 1 if the trace corresponds to an "air shower", and 0 otherwise. The output of the neural network would then be between 0 and 1, indicating the probability of the trace to have an air shower origin. In the next section, we will present the TREND experiment in more details, explain how we have selected traces to train and test our neural network, and how we were able to put « 0 » and « 1 » labels to these traces.

2. Training and test datasets

TREND was a self-triggered antennas array, consisting in 50 single-polarized, 50-100MHz, butterfly antennas, setup in the XinJiang province in China, and running between 2011 and 2014.

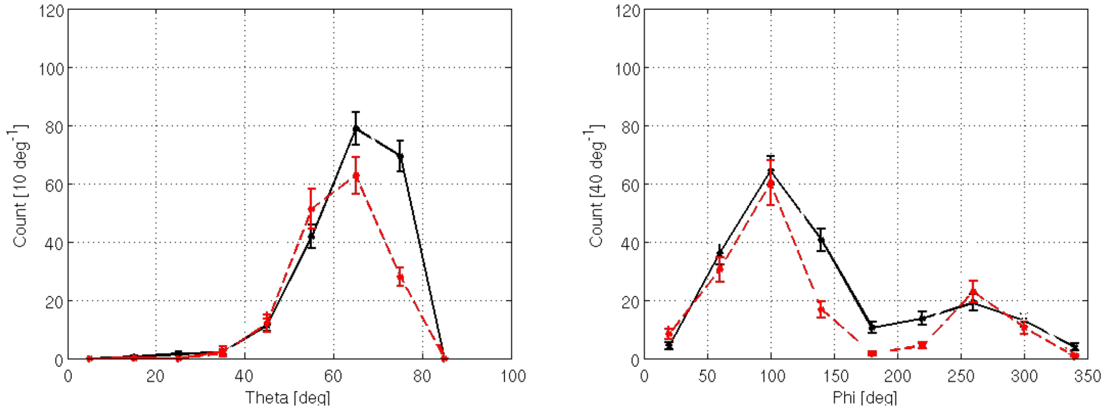

Figure 2: Zenith (left) and azimuth (right) distributions for the 200 selected experimental coincidences (black, plain line), and for the 160 air showers that would have been selected in that time according to simulations (red, dotted line). East is at $\phi = 0°$, North is at $\phi = 90°$.

TREND was triggered at the antenna level when the maximum absolute amplitude of a radio transient was at least 6 standard deviations above the baseline. TREND was then triggered at the array level if there was 5 or more antenna-level triggers in space-time coincidence. For the neural network trigger study, we used a set of 1.25×10^8 experimental coincidences, consisting of 9×10^8 radio time traces, corresponding to 80 days of acquisition. The main reason for which we decided to use experimental data rather than simulations is that neural networks might be very sensitive to differences between simulations and reality, that are sometimes hard to apprehend. Therefore, a neural network might work on simulations, but completely fail when put it into production. Furthermore, a lot of data from TREND were already available, for which the only critical point was to add labels. Actually, to add these "air shower" or "background" labels on the recorded time traces, we used the offline cuts that were defined in the final analysis of TREND, which was already published[2]. In this analysis, cuts were applied to reject : coincidences in time clusters, time traces with too long antenna signals, coincidences with a high χ^2 of reconstruction, coincidences for which the source position is below 3 km, zenith angles above 80°, discontinuous trigger pattern at ground, and direction-time coincidences clusters. These cuts allowed to select 200 coincidences, consisting of 1495 radio time traces. Then, the aperture of TREND was computed from simulations of air showers and of the TREND detector response. About 370 recorded air showers were expected, out of which 160 would have been selected by the cuts. Figure 2 shows the distribution in zenith and azimuth of the selected experimental (black) and simulated (red) coincidences. The consistency between the two distributions indicates that the 200 experimental coincidences are mostly air showers, with a $\sim 20\%$ contamination by background. Because contamination is higher for zenith angles above 70° and azimuth angles between 160° and 240°, we decided to reject these directions from the labelisation process. The 984 radio time traces consisting the remaining 139 coincidences were then labeled "air shower" while we labeled "background" the other 9×10^8 traces. Yet, using such an unbalanced dataset would have led the neural network to just ignore the "air shower" instances and to ouput « 0 » whatever the input. To obtain a class-balanced dataset while taking advantages of the almost unlimited pool of "background", we decided to put a weight of 2 for the 984 "air shower" instances and to randomly pick $984 \times 2 = 1968$ "background" instances. This dataset was split into a training set and a test set, with 80% and 20% fractions respectively.

3. Model

3.1 Convolutional Neural Network

A convolutional architecture has been chosen because it is far more efficient then a regular fully connected neural network for images recognition and related tasks. For each layer, there are ranges of neurons that share the same weights, because they are receptive to the same convolution pattern, but localized in different overlapping areas, so that they are not connected to all the inputs. The outputs of the neurons that share the same weights constitute a feature map. The neurons of the next layer are then receptive to patterns on the former-layer set of feature maps, also sharing weights, and being connected to limited scale inputs.

3.2 Training

Once the architecture is chosen, a neural network has to be trained to be able to make predictions. Training is the process of updating weights to minimize the loss function, which measures the discrepancy between the neural network output (the predictions) and the target to reach (the labels). The minimization method is the gradient descent: iteratively, the partial derivative of the loss function with respect to each weight is computed, and each weight is then updated in order to make the loss decrease (if the partial derivative is positive, weight is reduced, if it is negative, weight is increased). We chose "adam" for the gradient descent algorithm and a "cross entropy" loss function. For 2-classes classifications, cross entropy is just equivalent to $-(1/n)\sum \ln(1 - error)$, where error is the absolute difference between the neural network output and the target. As the error tends towards 1, cross entropy therefore tends towards infinity, making it more relevant for classification tasks, compared to a "mean absolute error" or a "mean squared error" function.

3.3 Hyperparameters tuning and results

We used the Keras/TensorFlow libraries. We tried different first layer inputs: time traces, Fourier transforms, and combinations of both. Best inputs were FFT real and imaginary parts, put in a succession alongside frequency. Also, many hyperparameters combinations were tested: convolutional kernel sizes within [3,5,7,9,11,13,21,31,41,51...81], and number of filters per convolutional layer within [4,8,16,32]. We finally got the best performances with a neural network made of 3 conv-layers, with kernel size of 51, 8 filters for the first conv-layer and 16 filters for the second and third ones. Each of the conv-layer is followed by a maxpooling of size 2. A dense layer of 64 neurons bounds the last conv-layer to the final output, which is compelled within [0,1]. Dropout (60%) and L2 regularizations (2e-3) were adjusted to keep the lowest loss while avoiding overfitting over 80 training epochs (overfitting is when the loss on the training set is lower than the loss on the test set, and one epoch is when all the training instances has been used once). We chose the ReLU activation function because it is generally a good default, and padding was set to "same" not to lose information. We finally got a 83% accuracy for the training and test sets, the accuracy being defined as the number of well classified data divided by the total number of data in a set. The accuracy

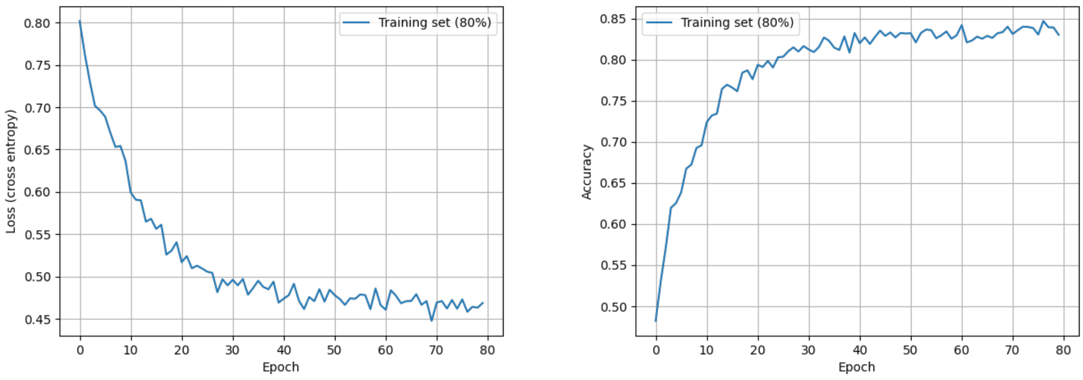

Figure 3: Loss (left) and accuracy (right) as a function of the number of training epochs (see text for definitions), for a training set composed of 1574 distinct background traces and 787 distinct air shower traces (with a weight of 2 for the latter).

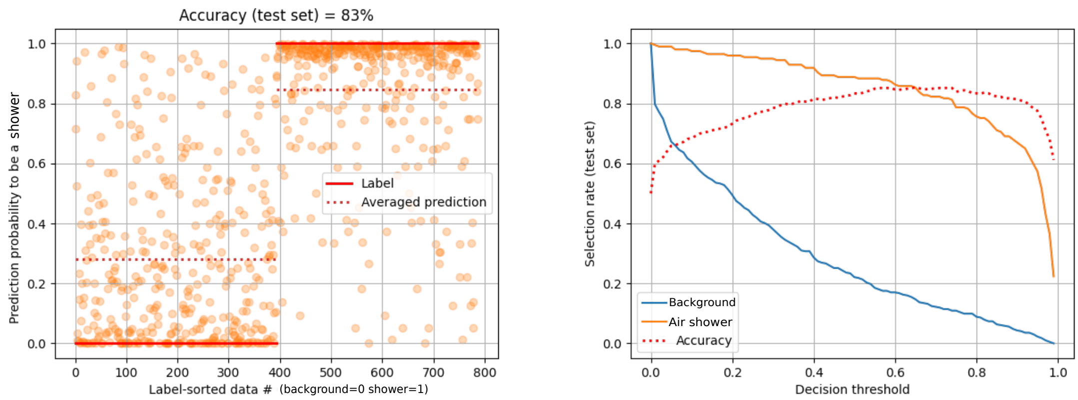

Figure 4: Left: Output of the neural network (prediction probability to be a shower) for all the traces of the test set, that are sorted by label on the x-axis ("background" first, "air shower" then). Right: Ratio of "air shower" traces (orange) and "background" traces (blue) that are predicted as "air shower", as a function of the prediction decision threshold, for the test set. Accuracy (dotted red) is 83% for a default decision threshold of 0.5. The test set is composed of 394 distinct background traces and 197 distinct air shower traces (with a weight of 2 for the latter).

is here computed with the default value of 0.5 for the decision threshold (threshold relative to the neural network output to decide if a trace is predicted to belong to class « 0 » or « 1 »). These results are shown on figures 3 and 4.

4. Inferences

Our aim was to simulate the discrimination performance of our neural network in production mode. To do that, we used our neural network to classify the whole set of 9×10^8 radio time traces consisting the 1.25×10^8 recorded coincidences. When the neural network output was above the decision threshold, we got a "Neural Network trigger" at the antenna level. If we had 5 or more antenna-level triggers in space-time coincidence, we got a "Neural Network trigger" at the array level, and the coincidence was considered as recorded by the neural network. The number of coincidences that may be recorded that way depends on how the traces of each coincidence are related to the others. If each individual trace in a coincidence has an identical probability p to be classified as "air shower", then the probability of this air shower coincidence to be recorded is equal to p. If, on the contrary, the probabilities to be classified as "air shower" are completely independent for all the traces in a coincidence, the probability of an air shower coincidence to be recorded is $\sim 1 -$ binom.cdf(successes = 5, trials = 8, p) (because 8 is the average number of traces per recorded coincidence and we ask for 5 traces at least). Same consideration applies for background coincidences. Therefore, with a decision threshold set to 0.6 (where the accuracy is the highest), we expect the neural network trigger to preserve between 0.07% (87 500) and 18% (2.25×10^7) of the 1.25×10^8 coincidences that were recorded with the standard TREND trigger. As we have shown earlier, according to simulations, ~ 370 of these 1.25×10^8 coincidences might be air showers. The neural network trigger is expected to preserve between 86% (318) and 91% (337) of the 370. It finally turned out that our "Neural Network trigger" recorded 3×10^6 coincidences (2.4%), meaning that traces in each background coincidences are correlated to some extent, from our neural network point of view. The intersection between that 3×10^6 coincidences and the 200 selected with the standard TREND analysis consist of 159 coincidences. One can note that this intersection was expected to contain at least 120 coincidences, because 86% of the traces consisting 139 coincidences amongst the 200 have already been predicted as "air shower" by the neural network while training and test stages. For these 159 coincidences, the number of recorded traces per coincidence (multiplicity) has decreased by 7% because of the neural network trigger. That loss of multiplicity involves an average direction misreconstruction under 3% in zenith and under 1% in azimuth (see figure 5).

To complete the analysis of our neural network trigger performances, we applied a set of simplified offline cuts (rejection of time clusters, high χ^2, near source position, zenith>80°, direction-time clusters) to the 3×10^6 coincidences that have triggered with the neural network, resulting in a selection of 225 coincidences. The angular distribution of these 225 coincidences is shown on figure 6. It exhibits a North-South pattern in azimuth, as expected for air showers at first order. For comparison, the figure also shows the distribution of the 200 coincidences that where selected by the TREND standard analysis, and the intersection between the two selected sets (108 coincidences). A summary of all the sets is shown on figure 7. As a conclusion to this section, our neural network has been able to improve trigger purity and to replace some offline cuts.

5. Conclusion

We studied the potential of deep learning techniques to reject significant parts of the massive amount of background signals that trigger antennas. For this study, we used a set of experimental data, consisting of 1.25×10^8 coincidences (9×10^8 radio time traces) that were recorded in 2012 by TREND, a self-triggered array of 50 single-polarized antennas, running in a remote valley of China. The standard analysis of this data had led to a selection of 200 coincidences. Simulations had shown that these 200 coincidences are mostly air showers, with a contamination by background events estimated around 20%. For the purpose of this study, an angular selection was performed

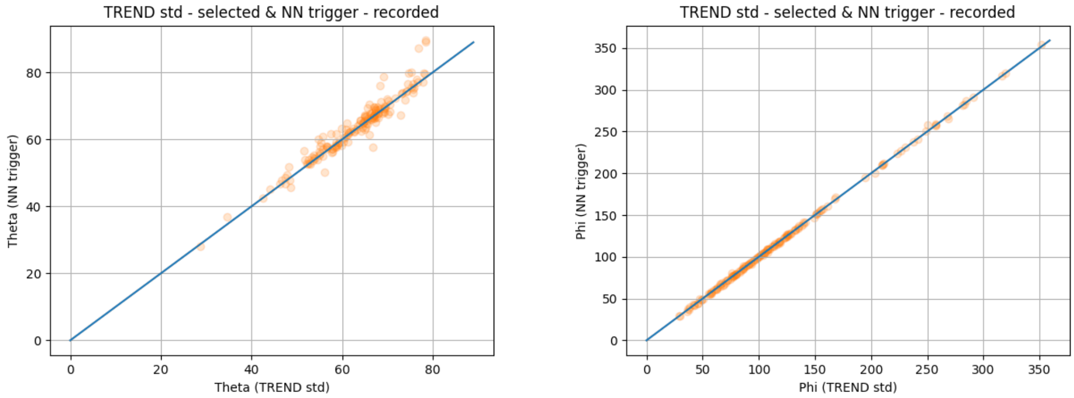

Figure 5: Zenith (left) and azimuth (right) angles for the 159 coincidences that were both selected by the TREND standard analysis and recorded by the Neural Network trigger. Angular reconstruction differs because of a loss of multiplicity due to the Neural Network veto on traces.

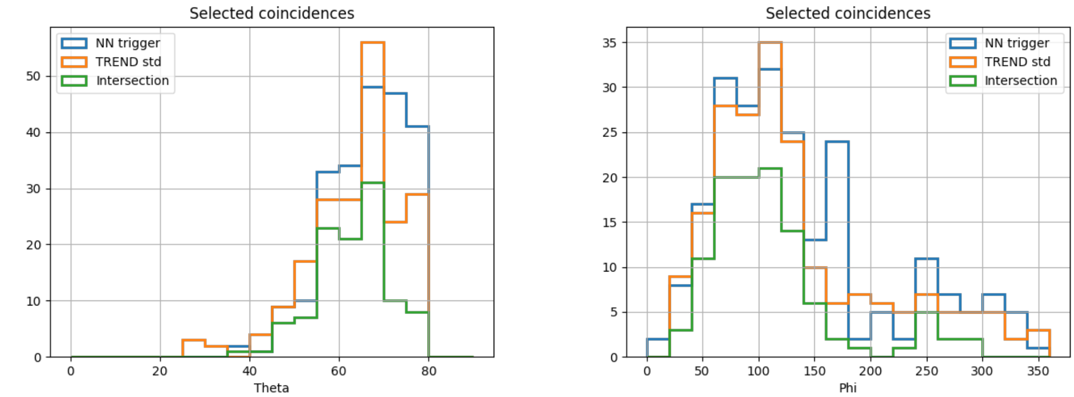

Figure 6: Zenith (left) and azimuth (right) distributions for selected coincidences (in degrees). In blue with the Neural Network trigger and a set of offline cuts (225), in orange with the TREND standard trigger and selection (200). The intersection of the two selections is in green (108). East is at $\phi = 0°$, North is at $\phi = 90°$.

to increase the purity of this selection, leading to 139 coincidences, consisting of 984 radio times traces. These 984 radio time traces and a random selection amongst the much larger amount of background signals were used to feed a convolutional neural network. This neural network was able to reject 82% of the background radio times traces while preserving 86% of the air shower radio time traces. At the array level (when combining traces in space-time correlations), the neural network allowed to reject 98% of the recorded coincidences, while preserving between 86 and 91% of the air shower coincidences. Because the training dataset was quite small, biased and not totally pure, the potential of the deep learning techniques was not fully probed. Furthermore, the antennas of the soon-to-come GRANDproto300 array[3] will for sure carry more information thanks to their 3 arms. There is thus plenty of room for improvements in the near future.

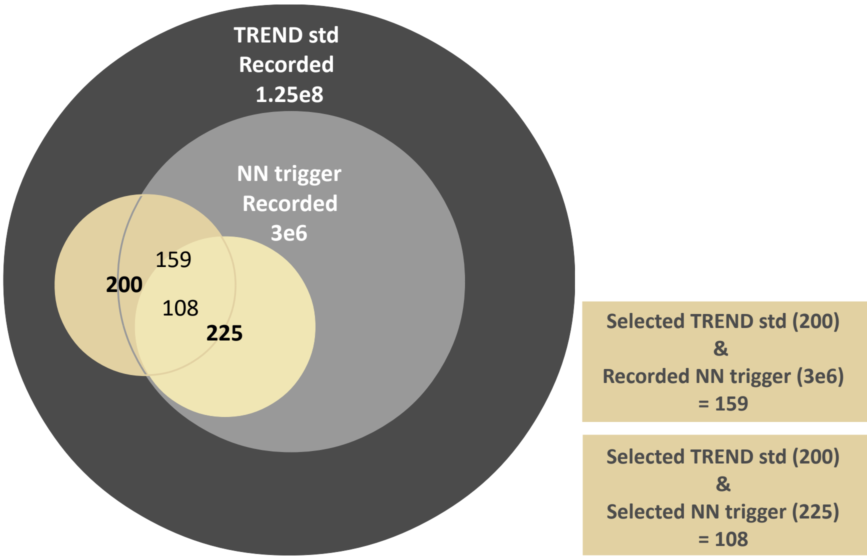

Figure 7: Diagram of coincidences that have been recorded (grey circles) and that have been selected (sand circles) with the TREND standard analysis (darker) and with our Neural Network trigger (lighter).

References

[1] Jaime Álvarez-Muñiz et al. The Giant Radio Array for Neutrino Detection (GRAND): Science and Design. *Sci. China Phys. Mech. Astron.*, 63(1):219501, 2020.

[2] D. Charrier et al. Autonomous radio detection of air showers with the TREND50 antenna array. *Astropart. Phys.*, 110:15–29, 2019.

[3] V. Decoene. The GRANDProto300 experiment: a pathfinder with rich astroparticle and radio-astronomy science case. In *36th International Cosmic Ray Conference (ICRC2019)*, volume 36 of *International Cosmic Ray Conference*, page 233, July 2019.