PROCEEDINGS OF SCIENCE

ONLINE ICRC 2021
THE ASTROPARTICLE PHYSICS CONFERENCE
Berlin | Germany
37th International Cosmic Ray Conference
12–23 July 2021

Classification and Denoising of Cosmic-Ray Radio Signals using Deep Learning

Abdul Rehman,[a,*] **Alan Coleman,**[a] **Frank G. Schröder**[a,b] **and Dmitriy Kostunin**[c,d]

[a] *Bartol Research Institute, Department of Physics and Astronomy, University of Delaware, Newark, DE 19716, USA*

[b] *Karlsruhe Institute of Technology, Institute for Astroparticle Physics, D-76021 Karlsruhe, Germany*

[c] *DESY, Zeuthen, 15738, Germany*

[d] *JetBrains Research, 194100 St. Petersburg, Russia*

E-mail: arehman@udel.edu, acoleman@udel.edu, fgs@udel.edu, dmitriy.kostunin@desy.de

The radio detection technique, with advantages like inexpensive detector hardware and full year duty cycle, can prove to be a vital player in cosmic-ray detection at the highest energies and can lead us to the discovery of high energy particle accelerators in the universe. However, radio detection has to deal with continuous, irreducible background. The Galactic and thermal backgrounds, which contaminate the radio signal from air showers, lead to a relatively high detection threshold compared to other techniques. For the purpose of reducing the background, we employ a deep learning technique namely, convolutional neural networks (CNN). This technique has already proven to be efficient for radio pulse recognition e.g., in the Tunka-Rex experiment. We train CNNs on the radio signal and background to separate both from each other. The goal is to improve the radio detection threshold on the one hand, and on the other hand, increase the accuracy of the arrival time and amplitude of the radio pulses and consequently improve the reconstruction of the primary cosmic-ray properties. Here we present two different networks: a Classifier, which can be used to distinguish the radio signals from the pure background waveforms, and a Denoiser, which allows us to mitigate the background from the noisy traces and hence recover the underlying radio signal.

37th International Cosmic Ray Conference (ICRC 2021)
July 12th – 23rd, 2021
Online – Berlin, Germany

*Presenter

PoS(ICRC2021)417

1. Introduction

High-energy Cosmic rays (CRs) from extra-terrestrial sources are of great interest to astrophysics and the search of their origin is a long-standing question. Studying the composition and direction of these CRs can lead us to the discovery of highest energy particle accelerators in the universe. At the same time, it would also allow us to better understand the mechanisms by which these cosmic accelerators impart energies in excess of what can be produced on Earth.

Detection of high energy CRs is done using balloon-borne or ground based experiments where the cascade of secondary particles, known as extensive air-showers (EAS), that are produced as a result of the interactions of primary CRs, are detected. One such method of EAS detection includes the use of radio antennas. Radio emission is produced due to the separation of charged particles, mainly electrons and positrons, present in the EAS [1, 2]. Although, radio emission from EAS is known for more than half a century [3], a lot of development of the radio detection technique has been made only recently. Despite the fact that digital radio detection is a fairly new technique, radio experiments, in the past couple of decades, have proven that the radio technique can compete in precision with other detection methods for the determination of arrival direction and composition (mainly through the observation of the depth of shower maximum X_{max}) of the primary CRs [4, 5]. Moreover, the radio technique has its advantages over other techniques like the full year duty cycle and low cost of the detection instruments. The one limiting factor that radio technique has to deal with is the irreducible radio background that continuously contaminates the radio signals from the air showers. The continuous background is both a primary factor in the detection threshold of radio experiments as well as a source of uncertainty in the reconstruction accuracy of air-showers parameters.

In order to mitigate the effect of background on the radio detection and reconstruction, we are using deep learning techniques. Deep learning techniques, like convolutional neural networks (CNNs), have been widely use in many fields and have shown promising results in recognizing different patterns in data. Indeed machine learning techniques have previously been explored for radio waveforms from cosmic ray observations at Tunka-Rex [6] and the Pierre Auger Observatory [7]. These experiments work in the 30-80 MHz band. In this work, we consider a frequency band of 50-350 MHz, that will be used by the IceCube Surface Enhancement, and use the corresponding electronics response [8]. Other experiments at Antarctica use even higher maximum frequencies, such as the balloon-borne ANITA mission [9].

We present the development of two CNNs for the purpose of identifying and removing background from observed waveforms. The first is used to distinguish waveforms with an air shower pulse, called a *Classifier*. The second network is designed to recover the underlying radio signals from the noisy traces, called a *Denoiser*.

We will start by describing the data set we have used for the training and testing of networks in section 2. Next, we will explain the structure of our two networks and the results of both networks in section 4 and section 5. Finally, we will provide a summary and give an outlook of this work.

2. Data Preparation

To prepare the data set we used the CORSIKA [10] software to simulate air-showers induced by CRs and used CoREAS [11] to produce the radio signals from these air showers. The simulation library includes zenith angles ranging from 0 to 65°, in steps of 5°, and energies of 100, 500 and 1000 PeV. For each energy and zenith bin, we produced 10 simulations with azimuth angles picked randomly. Proton and iron were used as primary particles, with Sibyll 2.3d [12] as a high-energy hadronic interaction model. The simulations of the radio emission were calculated on a grid, shaped in the form of an eight-legged star with its center located where the shower axis would intersect the ground. The lengths of the star arms were twice that of the Cherenkov ring radius.

The noise considered in this work was generated using the Cane Model for average Galactic noise [13], with an additional thermal component corresponding to a temperature of 30 Kelvin.

The definition of signal-to-noise ratio (SNR) that will be used to quantify the strength of the signal with respect to the background is given by:

$$\mathrm{SNR} = \left(\frac{\mathrm{Signal}_{\mathrm{Peak}}}{\mathrm{Noise}_{\mathrm{RMS}}}\right)^2, \tag{1}$$

where $\mathrm{Signal}_{\mathrm{Peak}}$ is the maximum voltage in the signal window (a 50 ns wide window around the expected signal peak) and $\mathrm{Noise}_{\mathrm{RMS}}$ is the root-mean-squared of a noise window (a non-overlapping window consisting of about 800 ns).

The data produced for the analysis is processed through the following steps: Firstly, The raw signal traces (output from CoREAS) of length 5120 bins sampled at 5GHz, are convoluted with the SKALA v2 antenna response [14], and are then down-sampled at 1 GHz. The SNR of the signal traces is computed before adding noise and only traces with SNR > 10,000 are kept for further processing. This selects signals which have coherent emission in the considered band and, thus, antennas far outside the Cherenkov cone are removed. Secondly, the background traces using the Cane Model are generated with the same length and sampling rate as our signal traces. The amplitude of the signal traces from the first step are scaled such that the true SNR[1] is in the range of 10^{-1} to 10^{3}. The scaled signals are then added to the background traces to create noisy traces. Thirdly, the traces are folded with the response of the readout hardware (for more details see [15]) and are digitized at 14 bits with a dynamic range of 1 V peak-to-peak (discrete amplitude bins of size 2^{-14} V). Lastly, the traces are filtered in the frequency band of 50-350 MHz and up-sampled at 0.25 ns using zero-padding in the frequency domain. The resulting traces are 1024 ns or 4096 bins long.

A total of 103k noisy (signal + noise) traces were produced for training the Classifier along with 135k background traces (noise-only). The SNR distribution of the given data set is shown in fig. 1 left. The blue distribution in the figure represents the noisy traces and the orange distribution represent the noise-only traces.

The total data set was then divided in to two sets with 80% used for the training of the networks and 20% as a test set. For the validation of the networks, an additional small independent data set containing 11k signal and 15k background traces was produced using the same method as

[1]The true SNR is defined similar to eq. (1), only the peak is computed from the signal trace and the rms is computed from the corresponding background trace which will be added later to create a noisy trace.

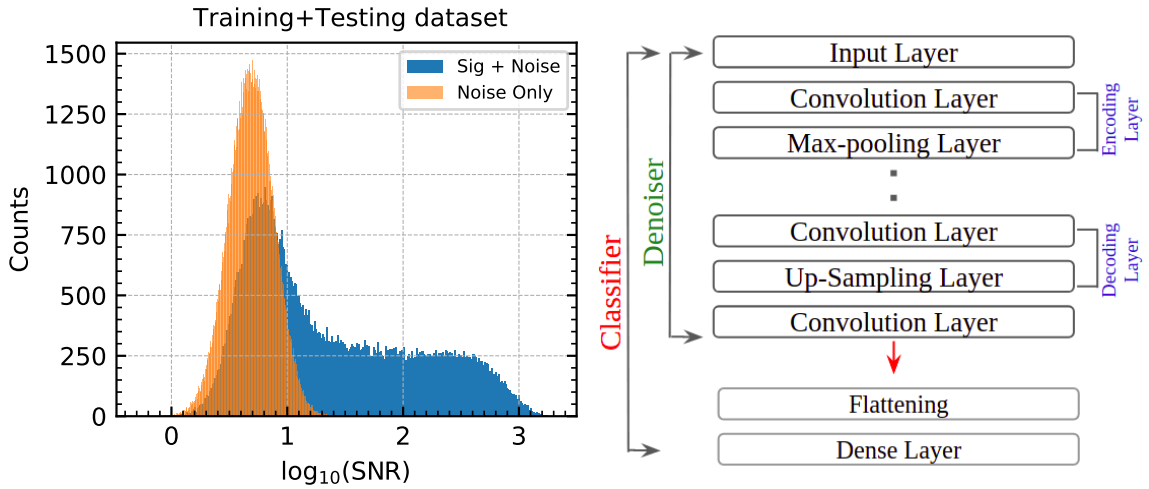

Figure 1: Left: Signal-to-noise ratio (see eq. (1)) distribution for the data set used for training and testing. The signals in the traces are scaled to have more values in the low SNR region for training. Right: Outline of the model architecture. The denoising network consists of only encoding and decoding layers with the output layer being the convolutional layer. The Classifier has the same structure as the Denoiser with additional flattening and dense layer at the end.

the training and testing set. Before feeding the prepared data to the networks, the traces are also normalized in the range [-1, 1], which scales all the features and avoids any biases in the data.

3. Model Architecture

The network architecture used in this analysis is shown in fig. 1, right. The networks are based on the technique called an auto-encoder, wherein the information is first encoded into a lower dimensional space and then decoded back to the original dimension. The first layer, the *input* layer, defines the dimensionality of our input data array, i.e., the length of the traces. The input layer is followed by several encoding and decoding layers respectively. These layers are added symmetrically to the front and back, respectively, of the auto-encoder. Also, all the layers considered here are one dimensional, since we are only dealing with time series data.

The encoding layer further consists of two layers: a convolution layer and a max-pooling layer. Convolution layer takes **n** fixed-length (but non-identical) sliding windows (also known as kernels) and perform convolution operations on the input data to create **n** outputs. The **n** different outputs of the convolution layer are also called filters. The max-pooling layer reduces the dimensions of the output of the previous layer by dividing it into pools of two neighboring bins and selecting the maximum value from each pool. The pooling operation keeps the important features and reduces the number of parameters for the network to learn.

The decoding layer, similarly, also consists of two layers: a convolution layer followed by the up-sampling layers. The up-sampling layer maps the extracted features into higher dimensions. The decoding layer is followed by a final convolution layer, this layer also serves as the output layer for the Deniosing network. The Rectified Linear Unit (ReLU) [16] is used as activation function in all the layers except the final convolution layer which uses a liner activation function. The output

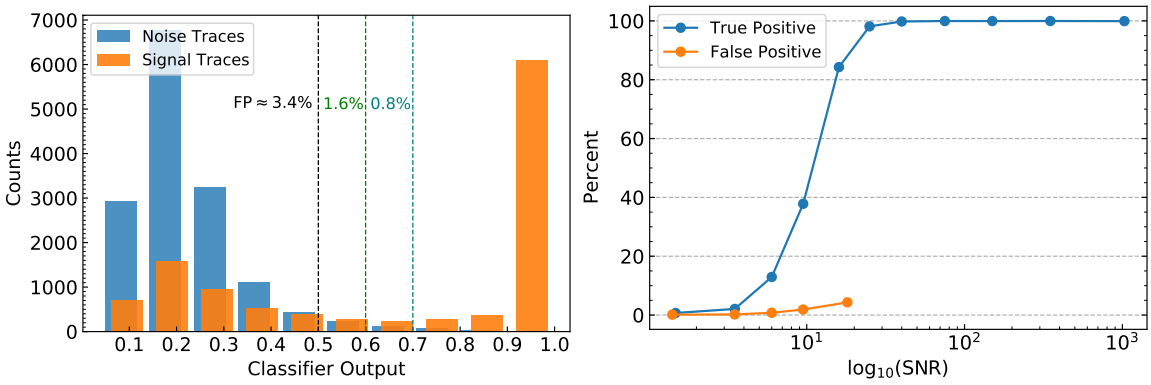

Figure 2: Left: Output of the Classifier network for the validation data set. The sigmoid activation function, which is used in the final layer, gives an output between 0 and 1 which is shown here for all the traces in our data set. The orange distribution shows the output for the background traces and the blue distribution is for the signal traces. The blue, green and tale vertical dashed lines represent the threshold levels of 0.5, 0.6 and 0.7 respectively to achieve different level of False Positive rates (indicated on the plot for the used data set). Right: the True and False positive rates (in percent) with the chosen threshold of 0.6 as a function of the SNR. The False positive rate is only meaningful up to SNR values of about 20 because this is about the maximum value of SNR of pure background traces in our validation set.

array at the end has the same dimensions as the input array with ideally the noise features removed from all the traces.

The Classifier on the other hand, has a similar structure as the Denoiser but with additional Flattening and fully connected (Dense) layers which are added at the end to get the desired output, a value between 0 (background-like) and 1 (signal-like) to identify the signal and background traces. This is achieved by using a Sigmoid activation function in the last dense layer.

Several models with different configuration have been tested. Our best performing network model consists of two pairs of encoding, decoding layers. And each convolution layer in our network has 8 filters per layer and a kernel size of 256. Also, for the classifier, models with one dense layer at the end performed better than models with more dense layers. Other models with similar size were also tested, but no statistical significant improvements were seen.

Both networks are implemented and trained using the python libraries Keras [17] and Tensorflow [18]. We use the Adam optimizer [19] and mean squared error (MSE) as a loss function to train both networks. The early stopping function of Keras was used as a regularization technique to avoid over-fitting the training data.

4. Classifier

The Classifier serves the purpose of distinguishing waveforms which contain air shower pulses from pure background waveforms. We trained the Classifier on the data sets defined in section 2. Since, we use supervised learning, we also provide labels (1 and 0 for signal and background traces respectively) to the network for training. The output values we get from the network for the validation set are shown in fig. 2, left.

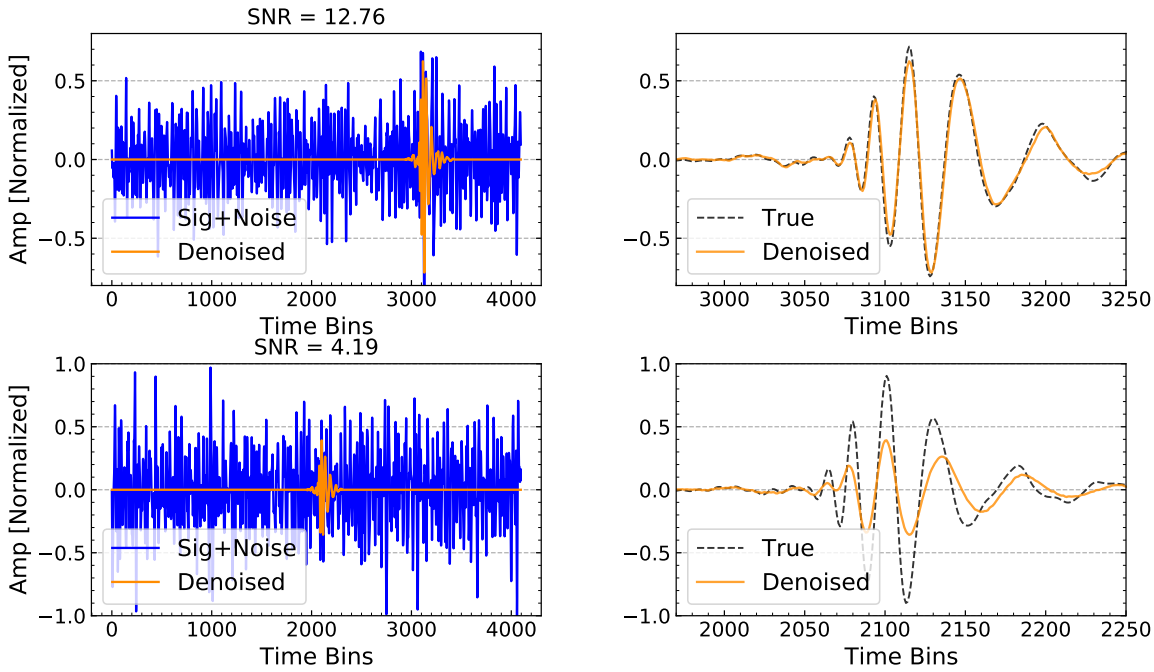

Figure 3: Example of the output of the denoising network. (Top) Best case example where the background was completely removed from the trace and radio signal was also fully reconstructed. (Bottom) Example where the background was again completely removed from the trace, but the network was unable to reconstruct the shape of the signal properly. The right plots are zoomed in along the x-axis in order to show more detail of the pulse. The SNR values of the input noisy traces are given in the title of left plots.

As indicated by the dashed lines, one can then choose a threshold for the output values of the network to select a preferred level of true positive (TP) and false positive (FP) rates. We chose a threshold value of 0.6 meaning that if the output value of the network is ≥ 0.6, the trace is classified as a signal trace, otherwise, it is classified as a background trace. With this threshold, we achieve a FP rate of 1.6% for all the background traces in the validation set. The TP and FP rates as a function of SNR are shown in fig. 2, right. We achieve a TP rate greater than 80% for SNR value > 15 and the FP rate at these SNR values is below 5%.

5. Denoiser

The second network we work on is called the Denoiser which is trained in order to eliminate the background from the noisy traces and it ideally would only leave the radio signals in the traces. To train the network, we use the same number of noisy traces as described in section 2 and also 50k background only traces. To label the data, for training, we use the underlying pure signal traces (CoREAS traces after all steps explained in section 2, but without adding noise) as labels for the noisy traces and arrays of zeros for the background traces.

The classified signal traces in the validation data set, i.e., those traces exceeding the threshold from the Classifier, are fed to the trained Denoiser for cleaning. Two example plots of waveforms after being processed by the denoising network are shown in fig. 3. The left plots are the output traces from the Denoiser (in orange) in comparison with the input noisy traces (in blue). On the

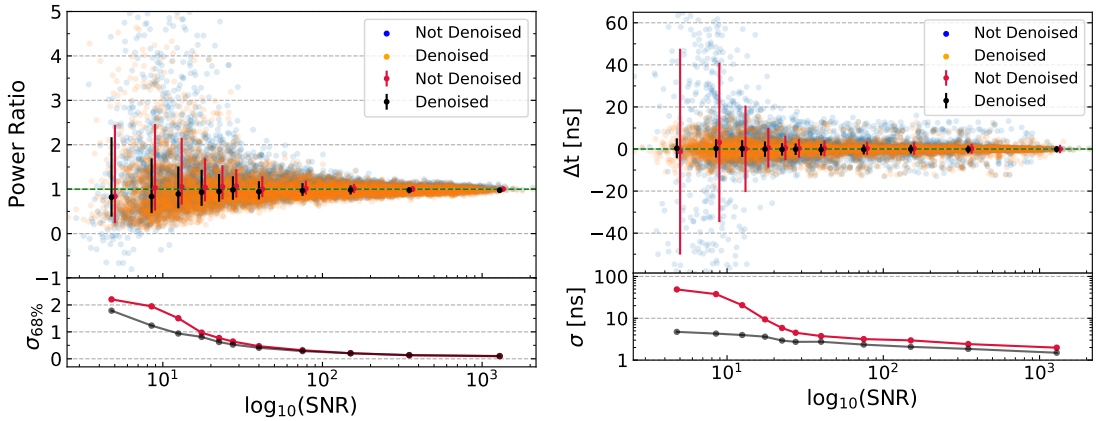

Figure 4: Accuracy metrics for the denoising network: Power ratio (left) and peak time difference (right) plotted before and after the Denoiser network as a function of SNR (see text for definition). Blue and red dots in both plots shows the given quantities respectively for traditional and Denoiser methods. Only validation data that have already passed the Classifier network are included in the plots.

right, the denoised traces are compared with the true traces (the labels used for training). The top row in the figure shows one of the best example where the network was able to identify and recover the radio signal within the traces while removing all the background. The right plot on the other hand also shows that the shape of the signal was also nicely recovered by the network. The bottom plot in the figure shows a waveform where, although the signal was correctly identified, the amplitude and shape of the signal were not completely recovered by the network.

In order to measure the overall accuracy of the trained network, we devised two accuracy metrics, the power ratio and peak time difference, which are compared to the case of using the raw waveforms (not denoised). We compute the power in the signal window, P_S, and power in the noise window, P_N, and then the power ratio is given as,

$$\text{Power Ratio} \;=\; \frac{[P_S - P_N]_{\text{Measured}}}{[P_S - P_N]_{\text{True}}} \;. \tag{2}$$

The peak time, T, is computed by taking the Hilbert transform of the waveform and finding the location of its peak. The difference is then,

$$\Delta t \;=\; T_{\text{measured}} - T_{\text{true}} \;. \tag{3}$$

We compute both metrics for all the signal traces that pass the Classifier in the validation data set as a function of SNR (fig. 4). The orange and blue dots in both plots represent the given quantities computed with and without the use of the Denoiser network. The means of the data sets are shown by the red and black dots for the denoised and raw waveforms, respectively. The bars represent 68% containment in the left plot and the standard deviation in the right plot. We see from the plots that by using the CNNs, a significant improvement is made in computing the power ratio for pulses at low SNR values of around 10, i.e., those pulses for which noise otherwise has a significant impact on the accuracy. The mean power ratio at the lowest SNR values is about 0.85 which shows a small bias and is due to the fact that the network is under-predicting the signal amplitudes at these values (see fig. 3). The accuracy of the peak time is improved for all SNR values, and a particularly large improvement is achieved for small signals.

6. Summary and Outlook

Deep learning models, especially CNNs, have made a lot of progress in pattern recognition and reducing the noise from the data for example from the images. This utility of neural networks have been utilized in this work in order to identify the radio signals emitted by CR air-showers. We presented the results of two networks, a Classifier and a Denoiser. Both networks are based on the auto-encoder technique where the network first compresses, then decompresses the data in dimensionality. We use these network for the classification of radio signals, hidden in modeled background noise, and also to directly recover the true waveforms. We achieved a true positive rate of about 80% for SNR values of above 15 with a false positive rate of 5% at these values with the selected threshold of 0.6 on the output of our Classifier. However, these values can also be tuned to the preferred levels by changing the threshold. The Denoiser network was able remove the noise and correctly identify and recover the radio signals within the traces. In some cases, especially for the small signal traces, the network was unable to completely recover the amplitude of the signal.

The results presented here make use of only the time-series information of radio signals, other aspects of the radio waveform such as frequency domain information can also be used in to the training of network to try to further improve the results. Also, we only used simulated data for the present analysis but these techniques can be used on real data in the future for the better reconstruction of the air-shower properties.

Acknowledgement: This work was supported by NASA EPSCoR project, Grant 80NSSC20M01. We thank the Tunka-Rex collaboration for sharing their code, which became the starting point of this work (url: https://gitlab.ikp.kit.edu/tunkarex/denoiser).

References

[1] T. Huege *Phys. Rept.* **620** (2016) 1–52.
[2] F. G. Schröder *Progress in Particle and Nuclear Physics* **93** (2017) 1–68.
[3] J. Jelley *et al. Nature* **205** no. 4969, (1965) 327–328.
[4] S. Buitink *et al. Physical Review D* **90** no. 8, (2014) 082003.
[5] P. Bezyazeekov *et al. Journal of Cosmology and Astroparticle Physics* **2016** no. 01, (2016) 052.
[6] D. Shipilov *et al. EPJ Web of Conferences* **216** (2019) 02003.
[7] M. Erdmann, F. Schlüter, and R. Šmída *Journal of Instrumentation* **14** no. 04, (2019) P04005.
[8] **IceCube** Collaboration, M. Oehler and R. Turcotte *PoS* **ICRC2021** (2021) 225.
[9] **ANITA** Collaboration, P. Miocinovic *et al. eConf* **C041213** (2004) 2516.
[10] D. Heck *et al. Report fzka* **6019** no. 11, (1998) .
[11] T. Huege, M. Ludwig, and C. W. James *AIP Conference Proceedings* **1535** (2013) 128–132.
[12] F. Riehn *et al. Physical Review D* **102** no. 6, (2020) 063002.
[13] H. Cane *Mon. Not. R. Astron. Soc.* **189** no. 3, (1979) 465–478.
[14] E. de Lera Acedo *et al. Experimental Astronomy* **39** no. 3, (2015) 567–594.
[15] **IceCube** Collaboration, A. Coleman *PoS* **ICRC2021** (2021) 317.
[16] A. F. Agarap *arXiv preprint arXiv:1803.08375* (2018) .
[17] F. Chollet *et al.*, "Keras." https://keras.io, 2015.
[18] M. Abadi *et al. arXiv preprint arXiv:1603.04467* (2016) .
[19] D. P. Kingma and J. Ba *arXiv preprint arXiv:1412.6980* (2014) .