URSI AP-RASC 2019, New Delhi, India; 09 - 15 March 2019

Using Machine Learning for the detection of Radio Frequency Interference

Kevin Vinsen*(1), Samuel Foster(1), and Richard Dodson(1)
(1) M468, ICRAR, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

Abstract

Radio Astronomy, by its very nature, detects extremely faint cosmic radio signals and is therefore very susceptible to Radio Frequency Interference (RFI). We present some initial results of our work to identify RFI using a Machine Learning (ML) based approach. The data is taken from VLBI observations data from three well separated observatories in Australia: ATCA, Parkes and Mopra; and we work with the 2-bit data directly from the telescopes. Our approach uses a Generative Adversarial Network (GAN) and an autoencoder to perform unsupervised machine learning on the data.

1 Introduction

In this paper we present our ongoing efforts towards developing new and robust methods to identify Radio Frequency Interference for the next generation of Radio Telescopes, using unsupervised and supervised Machine Learning. The sensitivity of the new instruments in development will introduce many new sources of RFI that have, to date, been unobserved. The Square Kilometre Array (SKA) [1] will require advances of several orders of magnitude in the processing of data, pushing RFI identification, classification and excising into the 'Big-Data challenge' [2].

Current generation radio telescopes such as: the Murchison Widefield Array (MWA) [3], the Australian SKA Pathfinder (ASKAP) [4], the Upgraded Giant Metrewave Radio Telescope (GMRT) [5], the Low Frequency Array (LOFAR) [6], MeerKAT [7] and the Very Large Array [8] are seeing interesting new sources of RFI. These include, but are not limited to: FM radio stations, cellular phone towers (directly and reflected), electricity grid equipment, office and kitchen appliances and digital TV stations. Some of the interference appears through tropospheric ducting [9] from sources 100-800 km away.

As an example of the impact of RFI on existing instruments and surveys, figure 1 shows the flagging of channels for phases 1 and 2 of the CHILES project [10], which uses the VLA, with some bands being 100% flagged due to RFI.

Mosiane *et al.* [11] used statistical machine learning (K-Nearest Neighbour, Random Forest Classifier, and Naive Bayesian) on KAT-7 data and achieved impressive results.

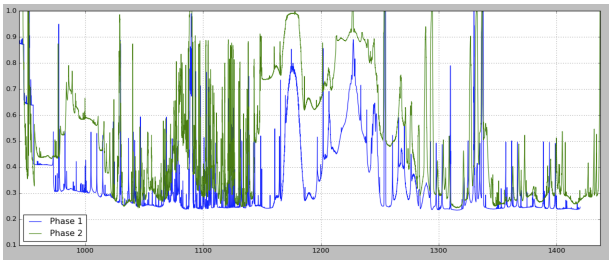

Figure 1. Flagging of Phase 1 and Phase 2 of CHILES.

Their work shows the potential of using statistical machine learning approaches. Our work moves into the domain of non-linear analysis using Deep Learning Neural Networks.

2 The Dataset

The dataset used for this study is from VLBI observations that provided contemporary data from three well separated observatories: ATCA, Parkes and Mopra. The VLBI was 2-bit (4 level) data, divided into 4 adjacent spectral windows of 16MHz in dual polarisation (right handed and left handed circular (RHC and LHC)). The lowest frequency was 6.3 GHz.

The VLBI dataset consisted of 1 minute of continuously sampled data every hour over the experiment (V255ae), which provided 8 files from each of the three sites. The RFI conditions from each of these sites, being hundreds of kilometres apart, are independent; Parkes is considered to have the more challenging RFI contamination, but ATCA and Mopra can also be poor at times. Experiment V255 is a maser VLBI project, so at times the data will contain a strong spectrally pure line feature. It is important that the RFI excision does not remove this signal.

As VLBI data analysis consists of the correlated signals from the independent telescopes where the 2-bit data is collected. This data is recorded and then transported to a common site for correlation. When the voltage signals are due to Gaussian noise, as it would be when dominated by the amplifier noise at the telescope, the losses associated with using low quantisation data are very low: only 22% for 2-bit data.

3 Machine Learning

3.1 Data Preparation

Each of the three VLBI datasets obtained from ATCA, Mopra, and Parkes are contained in VLBI files of roughly 4GiB. The samples from each dataset cannot be directly used and must be transformed into 32-bit floating point values. Each sample is inflated by a factor of 16 during transformation, resulting in 64GiB of output for each dataset. Due to their size, these transformed datasets are very challenging to use directly for GAN training. They require sophisticated loading and caching techniques to process them efficiently within the ML environment.

As part of the analysis of the VLBI data, a number of plots were generated. Figures 2 and 3 show the amplitude spectral density, and spectrogram of the first 30ms of samples from the ATCA dataset.

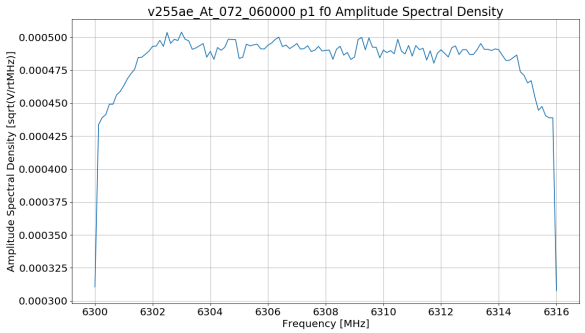

Figure 2. Amplitude Spectral Density of Polarisation 1, Frequency 0 of the first 30ms of the ATCA dataset.

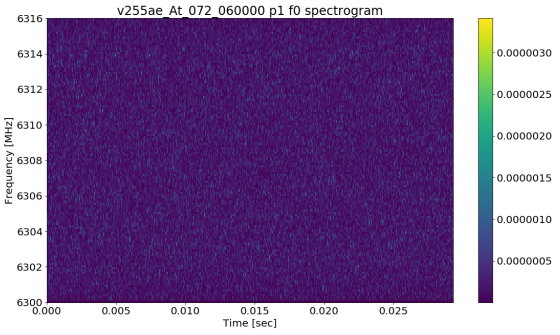

Figure 3. Spectrogram of Polarisation 1, Frequency 0 of the first 30ms of the ATCA dataset.

3.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were originally described by Goodfellow[12]. A GAN consists of a discriminator network and a generator network that are trained in unison. The discriminator is a classifier network that receives input from both a real data source, and from the generator. The goal of the discriminator is to determine whether its input is from the real data source, or whether it was created by the generator. The generator accepts a noise vector as input, and attempts to produce an output that the discriminator will classify as being from the real data source. Together, the discriminator and generator operate as an actor-critic model, in which the generator tries to fool the discriminator into classifying its output as real, and the discriminator tries to classify the generator's output as fake.

Once a GAN is fully trained, the discriminator and generator can be used in isolation. The generator can be used to generate data that looks like it could be from the real data source, and the discriminator can be used to determine if data is from the real data source. The discriminator is now working, effectively, as an anomaly detector; when it flags a sample as being fake, that sample has some "unknown" anomaly which can be analysed and classified.

3.2.1 Network Design

The discriminator uses a dense network structure as shown in figure 4, in which we take a Fast Fourier Transform (FFT) over each small time-step and the resulting real and imaginary components of the complex result are concatenated and processed by dense layers. This approach analyses the sample in the frequency domain, and is therefore time invariant.

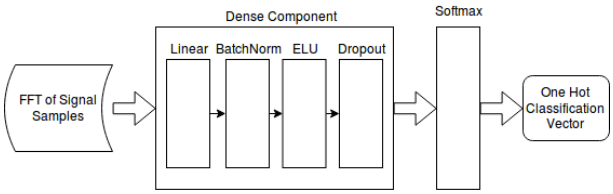

Figure 4. Dense structure for discriminator. The dense components are repeated multiple times.

The generator is modelled as an autoencoder that can be split into encoder and decoder components. The encoder accepts a vector of RFI data and produces an encoded representation of the data, and the decoder accepts an encoded representation and produces a vector of RFI. The output of the decoder is then compared with the input to the encoder to calculate the losses than need to be propagated back into the dense layers. Figure 5 illustrates this design. As a preliminary training step, the generator is trained on the RFI data as a full autoencoder, in which RFI data is passed through the encoder and decoder components. Once trained to a suitable level, the decoder component is separated and is used as the generator for GAN training. The noise vector is used as input into the trained decoder to produce a vector of RFI.

The losses of both the discriminator and generator are calculated using the Binary Cross Entropy loss function that is applied to the one hot classification vector output by the discriminator. Optimisation is done using an Adam optimiser[13].

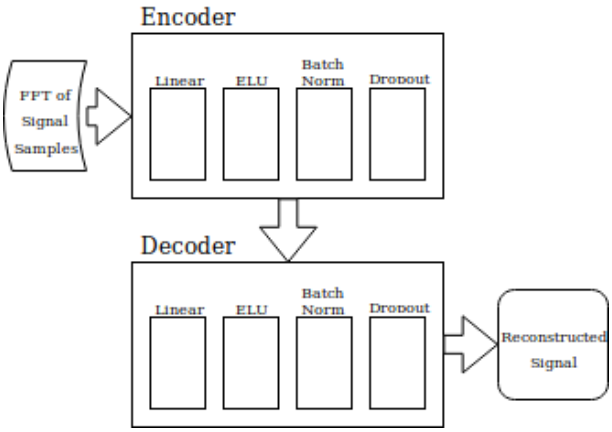

Figure 5. Autoencoder structure for generator. The encoder and decoder layers are repeated multiple times.

Implementation of the GAN and autoencoder is done using PyTorch 0.4.1[14] and Python 3.6, to take advantage of its multi-gpu support, rich machine learning feature set, and easy to use API.

3.2.2 Training Loop

Our GAN training loop is similar to the training algorithm described by Goodfellow[12], with the primary difference being we train the discriminator and generator once per step, as opposed to training the discriminator once per step, and the generator once per epoch. Algorithm 1 illustrates the training loop using pseudocode.

ALGORITHM 1
Pseudocode for training the discriminator and generator.

```
rfi_lbls ← [0,1]
not_rfi_lbls ← [1,0]
while epoch < max_epoch do
    while step < max_step do
        g ← gaussian_noise_minibatch
        r ← rfi_minibatch

        dr ← discriminator output r
        gg ← generator output g
        dgg ← discriminator output gg

        loss ←loss(not_rfi_lbls,dgg) + loss(rfi_lbls,dr)
        discriminator backpropagate loss

        g1 ← gaussian_noise_minibatch
        gg1 ← generator output g1
        class ← discriminator output gg1

        generator backpropagate loss(rfi_lbls, class)
    end while
end while
```

4 Preliminary Results

At the time of writing we have prototypes of the GAN operating correctly using the generator as a full autoencoder. The ML models for each of the telescopes are trained independently, as the response of each instrument is different.

Figure 6 shows the generator's loss over about 50 epochs and 8000 training batches. The loss decreases quickly and asymptotes to around 0.015 after the first 500 training batches. At this point the autoencoder has learnt how to decode the input sample and encode it in a more concise form, and to rebuild the original sample from the concise form.

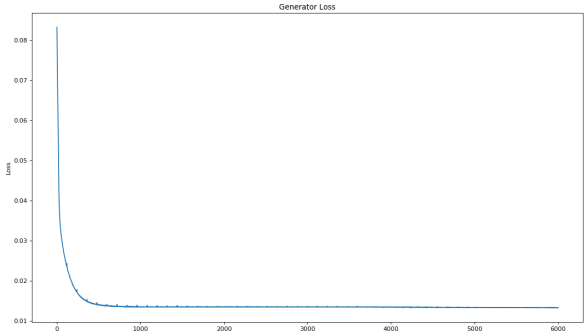

Figure 6. Experimental Autoencoder Loss over 8000 training batches.

Figure 7 shows an example of the output from the generator after 50 epochs. As can be seen, the output of the generator shows the symmetry across the real values of the FFT, but the magnitude of the output values are off by a factor of 10. This is because, at the moment, the final activation function is a Tanh function which constrains the output to -1 to 1. These issues will be resolved by using a deeper network with more training data and modifying the final activation function.

5 Conclusions and Future Work

Using unsupervised machine learning to identify RFI is a new approach for the use of ML in Radio Astronomy. Our early results from the autoencoder are extremely promising and highlight that this approach has merit. We are able to learn to generate realistic simulations of "normal" samples for the various instruments and the GAN is starting to be able to detect fake samples from real samples. This gives us a self learning anomaly detection system.

Our future work is to continue developing the system so that it can classify the RFI, and then, ultimately, excise it from the signal. We will be introducing an attention model [15], so the autoencoder and GAN can learn which frequencies are most significant when developing a characterising of the default samples. Initially ML will not be able to identify the source of the RFI. This will still require a human RFI expert. Once the expert has classified the RFI and we have examples of it, we will develop formal ML classification

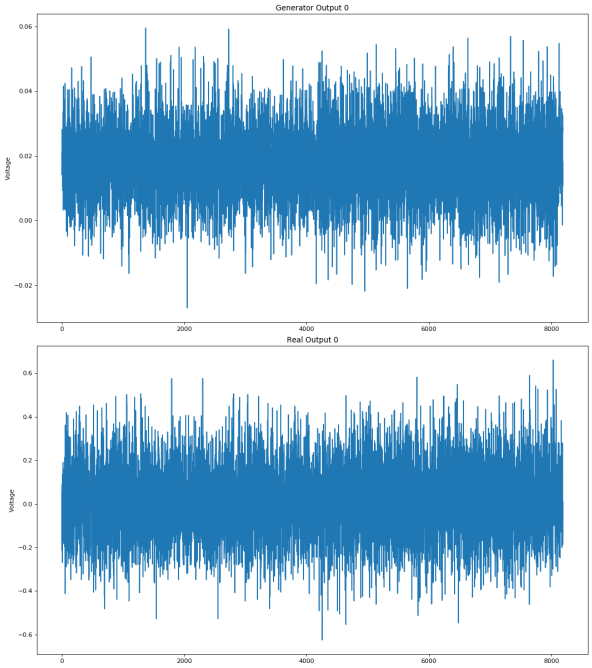

Figure 7. Generator output after 50 epochs compared to Real Data.

systems that will be able to automatically identify the type of RFI after the GAN has identified its presence.

We are currently beginning work with colleagues in Thailand, and China with access to other instruments. We plan to characterise each instrument and develop a database of known RFI sources for each instrument. We expect some common RFI sources will be able to be classified based on identification on different instruments.

References

[1] P. E. Dewdney, P. J. Hall, R. T. Schilizzi, and T. J. L. W. Lazio, "The Square Kilometre Array," *IEEE Proceedings*, vol. 97, pp. 1482–1496, Aug. 2009. DOI: 10.1109/JPROC.2009.2021005.

[2] P. Quinn, T. Axelrod, I. Bird, *et al.*, "Delivering SKA Science," *ArXiv e-prints*, Jan. 2015. arXiv: 1501.05367 [astro-ph.IM].

[3] S. J. Tingay, R. Goeke, J. D. Bowman, *et al.*, "The murchison widefield array: The square kilometre array precursor at low radio frequencies," *Publications of the Astronomical Society of Australia*, vol. 30, e007, 2013. DOI: 10.1017/pasa.2012.007.

[4] S. Johnston, I. J. Feain, and N. Gupta, "Science with the Australian Square Kilometre Array Pathfinder (ASKAP)," in *The Low-Frequency Radio Universe*, D. J. Saikia, D. A. Green, Y. Gupta, and T. Venturi, Eds., ser. Astronomical Society of the Pacific Conference Series, vol. 407, Sep. 2009, p. 446. arXiv: 0903.4011 [astro-ph.CO].

[5] Y. Gupta, "The GMRT: current status and upgrade plans," in *Astronomical Society of India Conference Series*, ser. Astronomical Society of India Conference Series, vol. 13, 2014, pp. 441–447.

[6] M. P. van Haarlem, M. W. Wise, A. W. Gunst, *et al.*, "LOFAR: The LOw-Frequency ARray," *Astronomy & Astrophysics*, vol. 556, A2, A2, Aug. 2013. DOI: 10.1051/0004-6361/201220873.

[7] J. L. Jonas, "MeerKAT - The South African Array With Composite Dishes and Wide-Band Single Pixel Feeds," *IEEE Proceedings*, vol. 97, pp. 1522–1530, Aug. 2009. DOI: 10.1109/JPROC.2009.2020713.

[8] R. A. Perley, C. J. Chandler, B. J. Butler, and J. M. Wrobel, "The expanded very large array: A new telescope for new science," *The Astrophysical Journal Letters*, vol. 739, no. 1, p. L1, 2011. [Online]. Available: http://stacks.iop.org/2041-8205/739/i=1/a=L1.

[9] B. T. Indermuehle, L. Harvey-Smith, C. Wilson, and K. Chow, "The askap rfi environment as seen through beta," in *2016 Radio Frequency Interference (RFI)*, Oct. 2016, pp. 43–48. DOI: 10.1109/RFINT.2016.7833529.

[10] CHILES, *CHILES Cosmos HI Large Extragalactic Survey*, http://chiles.astro.columbia.edu/index.html.,

[11] O. Mosiane, N. Oozeer, A. Aniyan, and B. A. Bassett, "Radio frequency interference detection using machine learning.," *IOP Conference Series: Materials Science and Engineering*, vol. 198, no. 1, p. 012 012, 2017.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980.

[14] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[15] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 2048–2057.