Training Neural Networks to Classify and Denoise Cosmic-Ray Radio Signals Using Background Measured at the South Pole

Dana Kullgren

February 3, 2022

Co-authors: IceCube Collaboration, Abdul Rehman, Alan Coleman, Frank Schroeder

Motivation

- Unlike A. Rehman[1], I am using measured radio background from the prototype station at the South Pole.
- Using measured noise would make these networks more applicable to actual data.

Photo: IceCube Collaboration

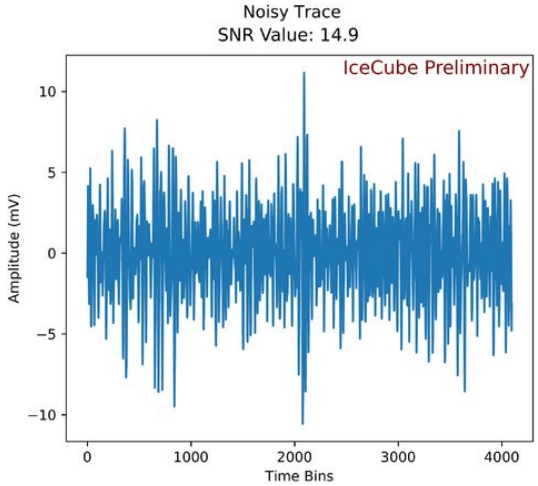
Goals: Use machine learning to…

- Improve the detection threshold of radio experiments.
- Remove background noise from signals.

[1]A. Rehman, A. Coleman, F. G. Schröder, and D. Kostunin *PoS* **ICRC2021** (2021) 417.

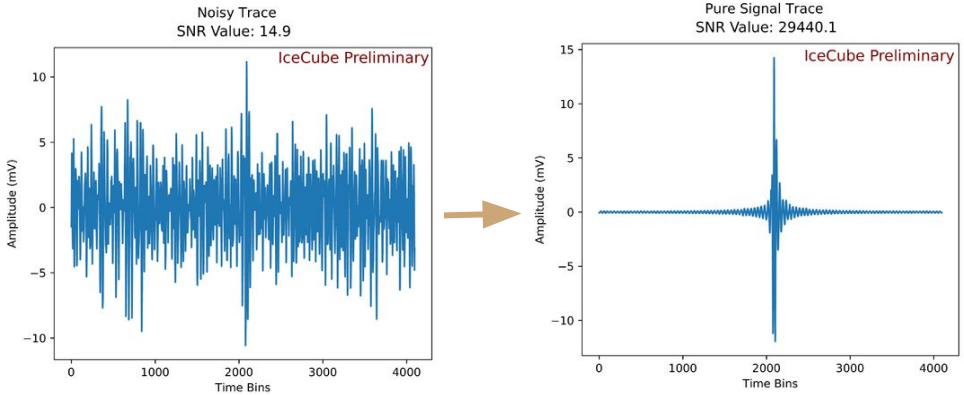
Machine learning can be used to...

Classify: Determine if an antenna has received a radio signal



Does this trace contain a cosmic-ray radio signal?

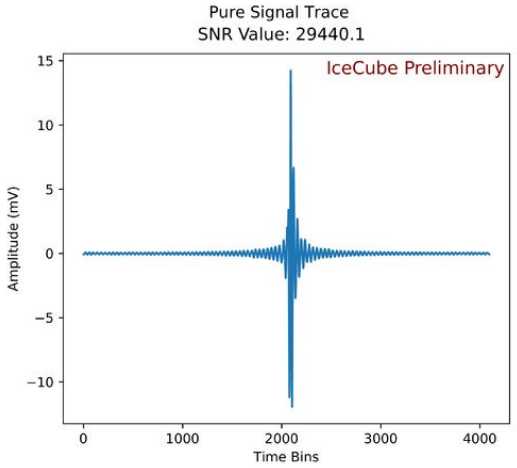
Denoise: Remove the background noise from a noisy signal

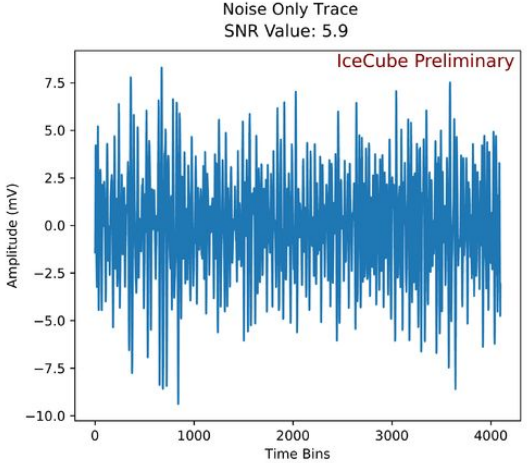


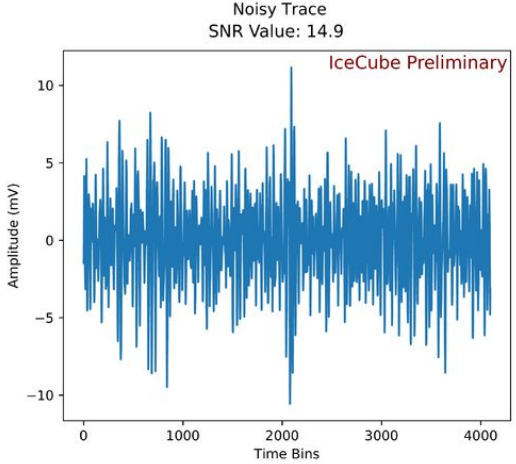
Noisy trace (signal plus noise)

Clean trace (signal only)

Data Collection and Processing







Cosmic-Ray Signal (simulation) + Background Noise (measured) = Noisy Traces

Data Used

Simulation data: created using CORSIKA and CoREAS

- Proton
- Star pattern
- Energy bin: 10^{17} eV
- Zenith angles: 0° to 72°

Measured data: background noise from the South Pole prototype station

- Using forced triggers for the background
- Dec 2020
- Jan-Feb, Dec 2021

} Traces taken from dates under good operating conditions

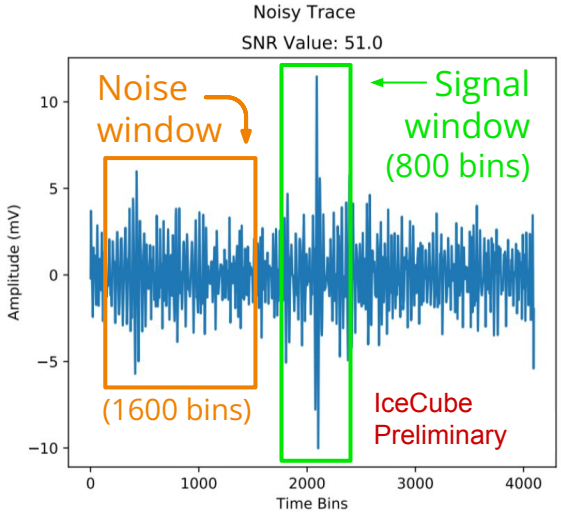
Data Processing Details

Modifying simulation data for air showers by:

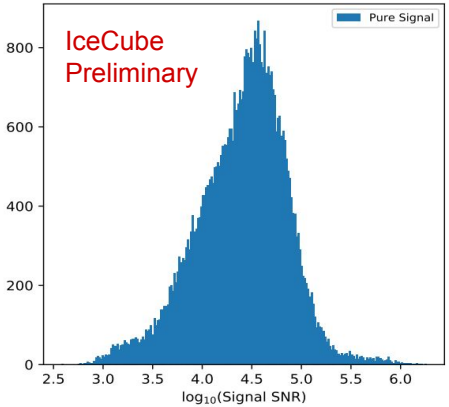
- Adding antennas/electronics responses
- Filtering frequencies (100-350 MHz)
- Only choosing clean signals
- Rescaling signal amplitudes
- Shifting the location of the signal peak within the trace
- Adding measured background noise
- Cleaning up artifacts in noise data
- Normalizing the signal [-1, 1]

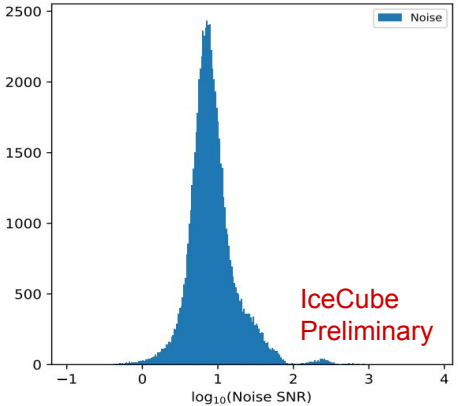
SNR of Dataset

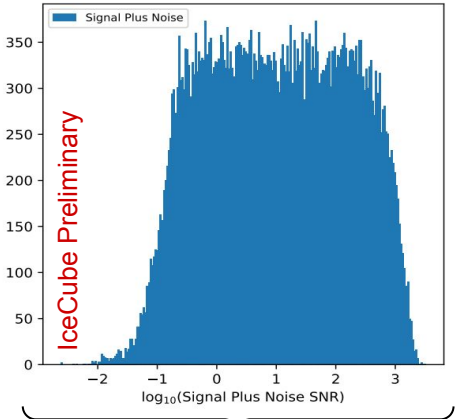
- SNR = (signal window peak / noise window RMS)^2
- True SNR = (Pure signal signal window peak / noise window RMS)^2
- Mean SNR of background noise traces: 12.6



Signal-to-noise ratio (SNR) distributions of the training/testing dataset:







True SNR distribution

Splitting Data

The training and testing dataset has 71k traces each for pure signals, noise only, and noisy signals.

- 80% = training
- 20% = testing

Inputs and Outputs of Neural Networks

These networks use supervised machine learning, so the training data is labeled and the neural network is given pairs of inputs and desired outputs.

	Input	Desired Output
Classifier	Traces (signal plus noise and noise only)	Label (1 and 0)
Denoiser	Noisy trace (signal plus noise)	Pure signal (signal only)

Note: The denoiser does not receive noisy traces from the classifier, it is trained using all of the noisy traces in the dataset.

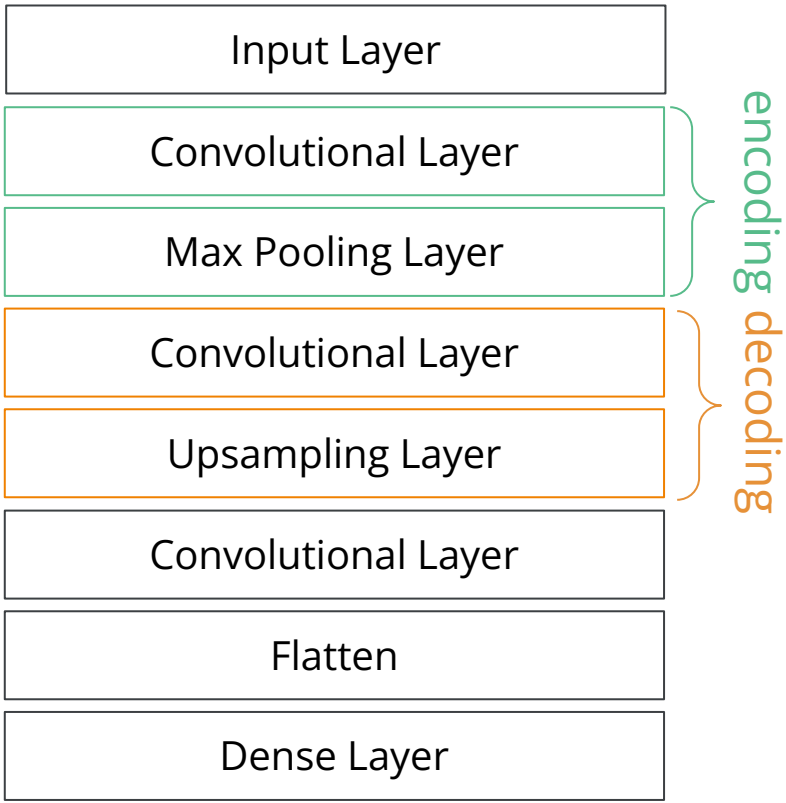
Network Structure

Loss function:
MSE (mean squared error)

Parameters:

- DL: # dense layers
- Fil: # filters
- KS: kernel size (length of kernel)
- Layers: determines depth of network

Layers of an example CNN:

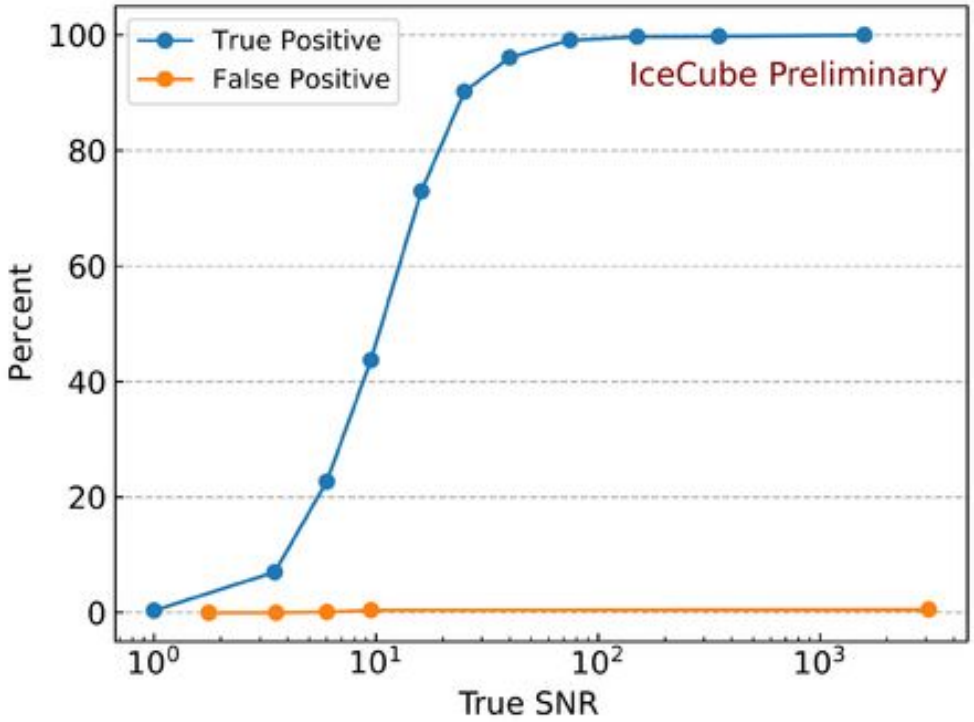


Classifier Accuracy (Best Model: Layers=7, DL=1, Fil=16, KS=512)

This model is very unlikely to give false positives.

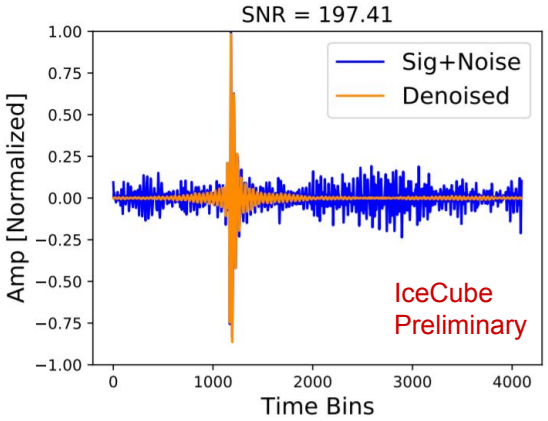
True positive: the network correctly identifies a trace that contains a signal

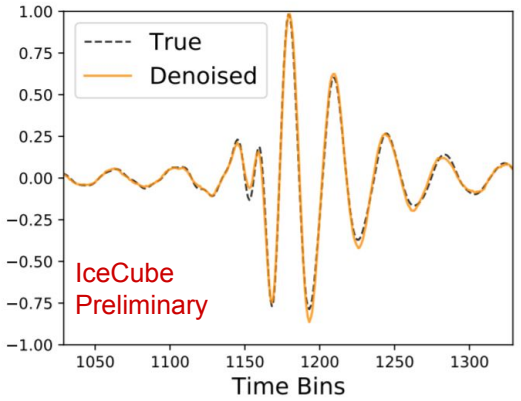
False positive: the network identifies a noise only trace as a trace that contains a signal



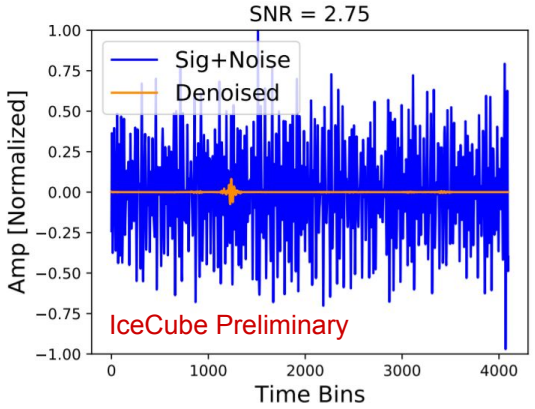
Denoiser Accuracy (Best Model: FIL=64, KS=128, Layers=2)

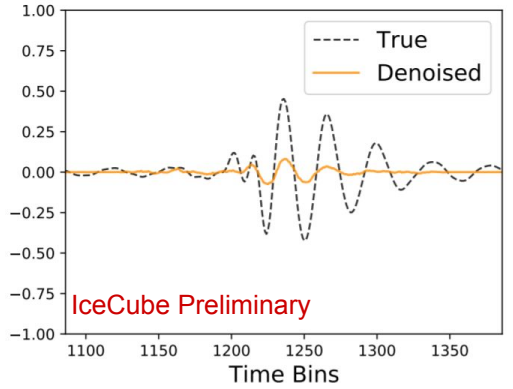
Better Denoising





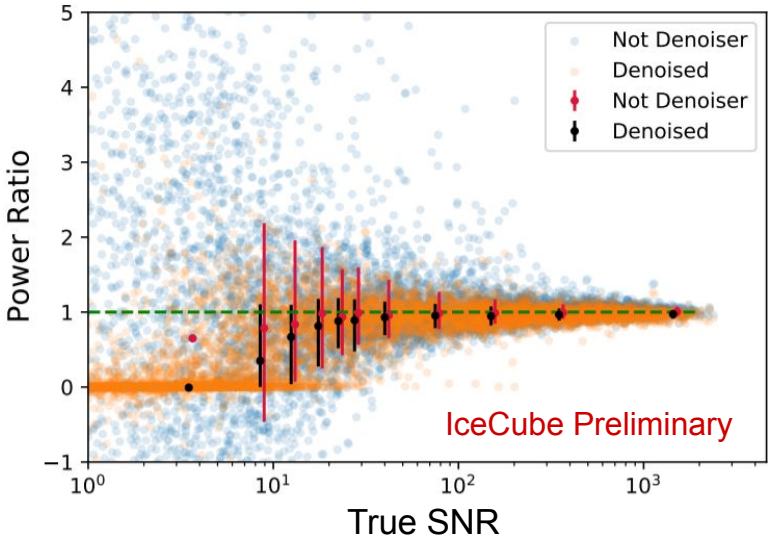
Worse Denoising





Denoiser Accuracy

(Best Model: FIL=64, KS=128, Layers=2)

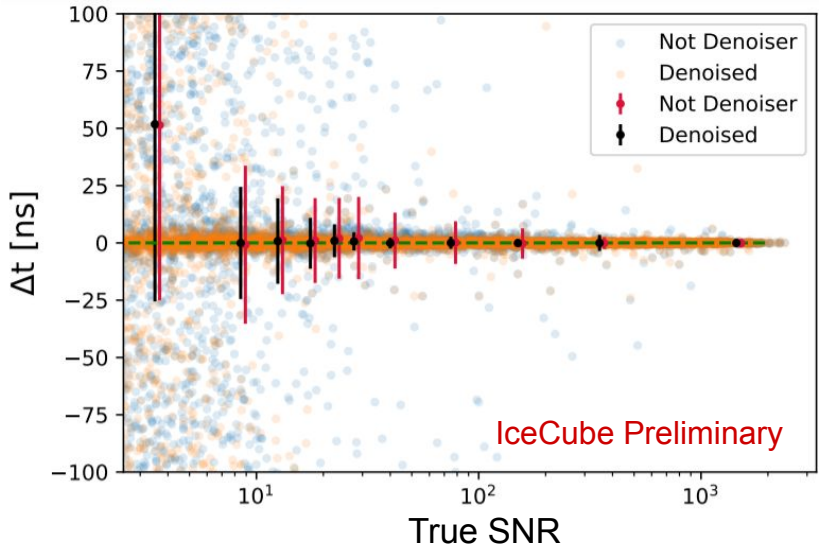


$$\text{Power ratio} = \frac{\text{Predicted power}}{\text{Expected power}}$$

Power = (power in signal window) - (power in noise window)

"Predicted" = trace outputted by network

"Expected" = pure trace (desired outcome)



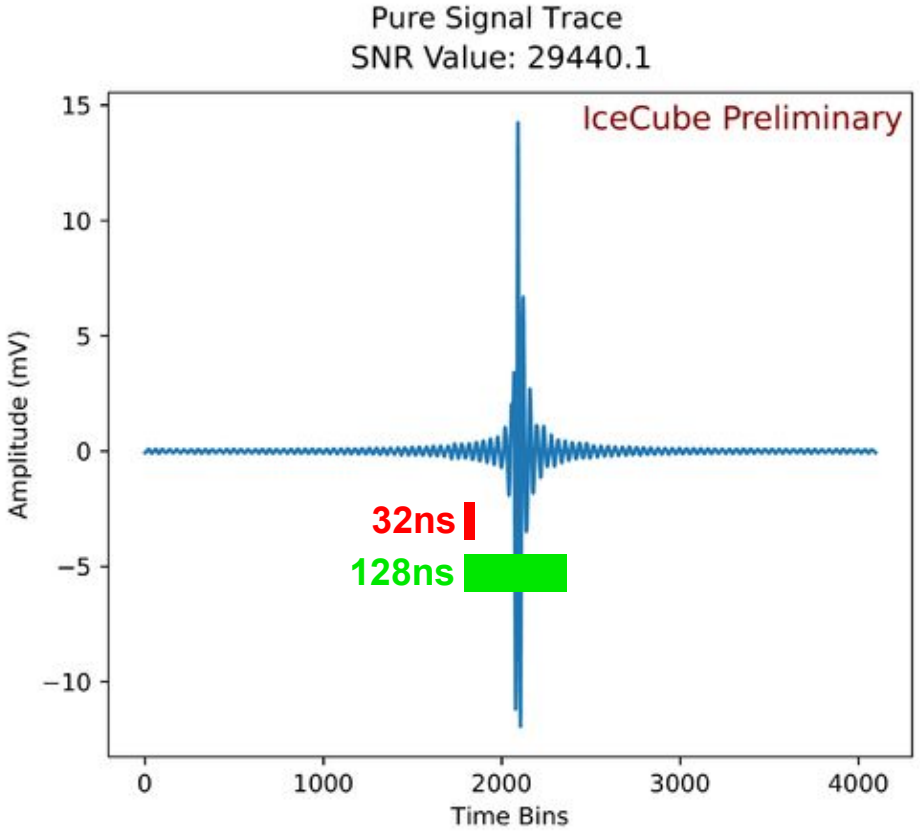
Δt = Predicted arrival time of peak - Expected arrival time of peak

Kernel Size

1 time bin = 0.25 ns

Optimal kernel size:

- **Denoiser**: 128
- **Classifier**: 512



Conclusions and Next Steps

Networks perform better with modeled noise than with measured noise (but they still work and the results can be improved)

- The classifier needs to be improved to do well with small SNR values
- The denoiser is promising, but has room for improvement

Next steps

- Don't include pure signals that aren't clean enough
- Only give signal traces which have been identified by the classifier to the denoiser
- Use channel pairs for each antenna to train models
- Use data from the frequency domain as well as the time domain
- Create a validation dataset

Goal: Apply these networks to data from the prototype station