SnT 2019
CTBT: SCIENCE AND TECHNOLOGY CONFERENCE

Detecting Low Magnitude Seismic Events using Convolutional Neural Networks

SAND2019-5278C

Robert Forrest [1], Jaideep Ray[1] , Chris Young[1]

Sandia National Laboratories[1]

T3.5-P22

Sandia National Laboratories

National Nuclear Security Administration

Objective

Currently traditional methods are used to detect arrivals in three-component seismic waveform data collected at various distances. Accurately establishing the identity and arrival of these waves in adverse signal-to-noise environments is vital in detecting and locating the seismic events. Autocorrelation and template matching techniques are just a few of the various methods that may be used, each with their own benefits and drawbacks. For example, template matching methods have been shown to be effective, but are restricted to repeating signals and become computationally intensive with increasing numbers of templates.

In this work, we move to convolutional neural networks (CNNs). CNNs have been shown to significantly improve performance at local distances under certain conditions. In this work we expand the use of CNNs to increasingly remote distances and lower magnitudes. We explore the advantages and limits of a particular approach and begin to understand requirements for expanding this technique to different types, distances and magnitudes of events in the future.

We describe in detail performance results of this method tuned on a new dataset with expert defined arrival picks. The dataset used is from the Dynamic Network Experiment 2018 (DNE18) and comes from sensors in Utah. We demonstrate the ability to train the CNN on events from the dataset and achieve significantly higher test set performance than standard methods. Furthermore, we validate performance on streaming data, including very low magnitude expert picked arrivals.

Goal:

Use established CNN methods on a new expert picked arrival dataset to understand advantages and limitations of current CNN methods.

CNNs

CNNs are a particular type of feedforward artificial neural networks that use a linear convolution in some layers which has several advantages over fully connected layers. Convolutions act as filters choosing spatially invariant features; sharing these parameters throughout the data dramatically increases generalizability and reduces the parameters that need to be trained, improving efficiency.

Architecture:

In this work, we follow the lead of the architecture of **ConvNetQuake** (ref 1). The architecture takes raw seismic waveforms from three component 100 Hz ground motion sensors and divides the input into 10 second windows (figure 1). The network consists of eight convolutional layers that down sample the signal by a factor of two and contain 32 filters.

Output: 128 features then a fully connected layer containing the class scores. Class scores are either 'No Event' or 8 other classes corresponding to rough geographical location of event.

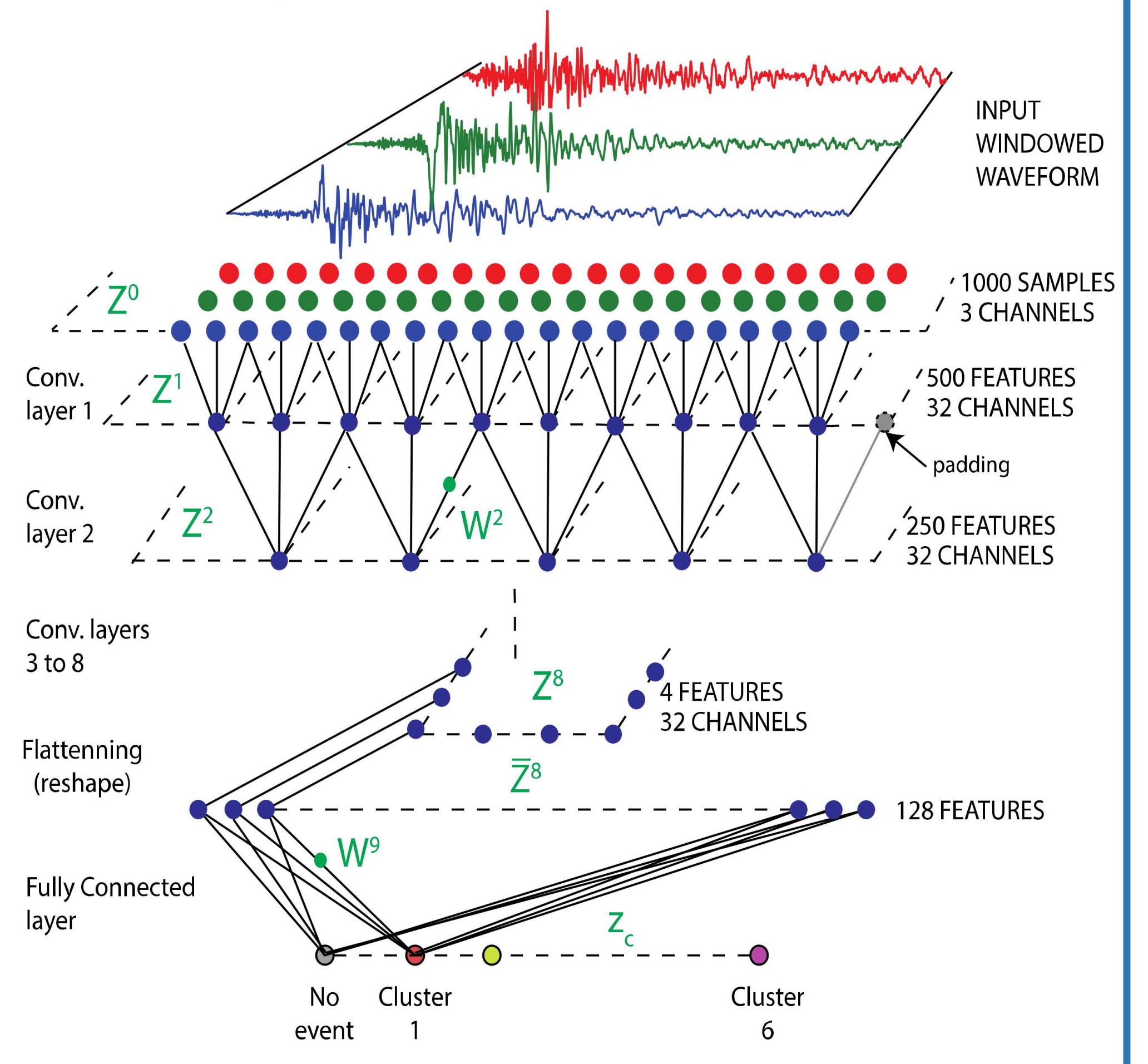


Figure 1: ConvNetQuake Architecture (Reference 1)

Seismic Data

The data used in this study is the 2018 Dynamic Networks Data Processing and Analysis Experiment (DNE18) dataset. A more detailed description can be found elsewhere. It consists of University of Utah Seismograph Stations (UUSS) waveform data spanning from 01/01/2011 through 01/14/11. Stations sample three component data (N, E, Z) at 100 Hz. Although there are many stations to choose from, we only use a select one for this study.

This catalog consists of a high quality, hand-picked seismic events derived from analyst Chip Brogan's picks with the Analyst Review Station (ARS) software as well as other automated event input sources. The event must be sensed on 3 or more stations in order to build an event. All events (man-made or natural) generated match this criteria. The advantage of this dataset is the low magnitude and expert curated events we will use as ground truth.

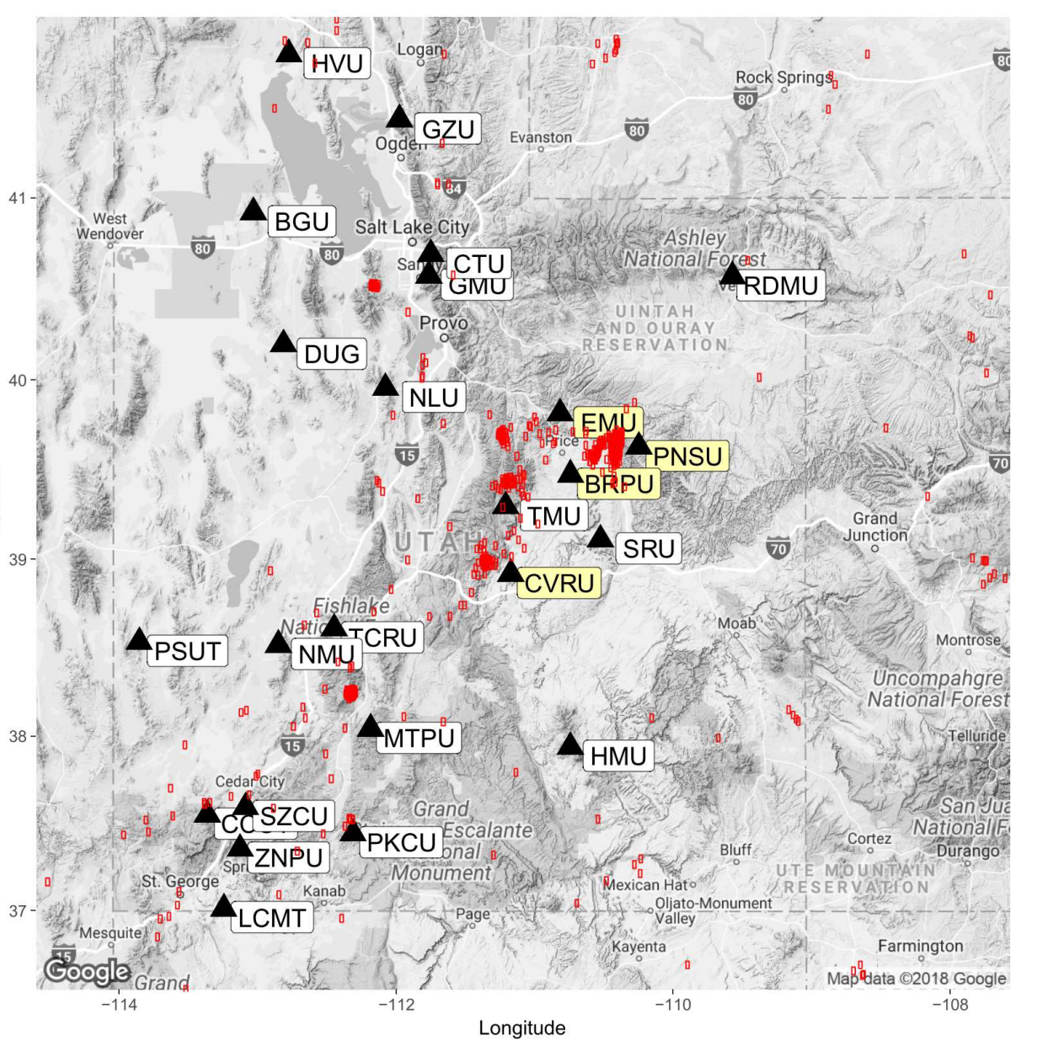


Figure 2: Map of DNE18 stations (black triangles) and seismic events (red circles)

Training and Validation

Data Preparation:

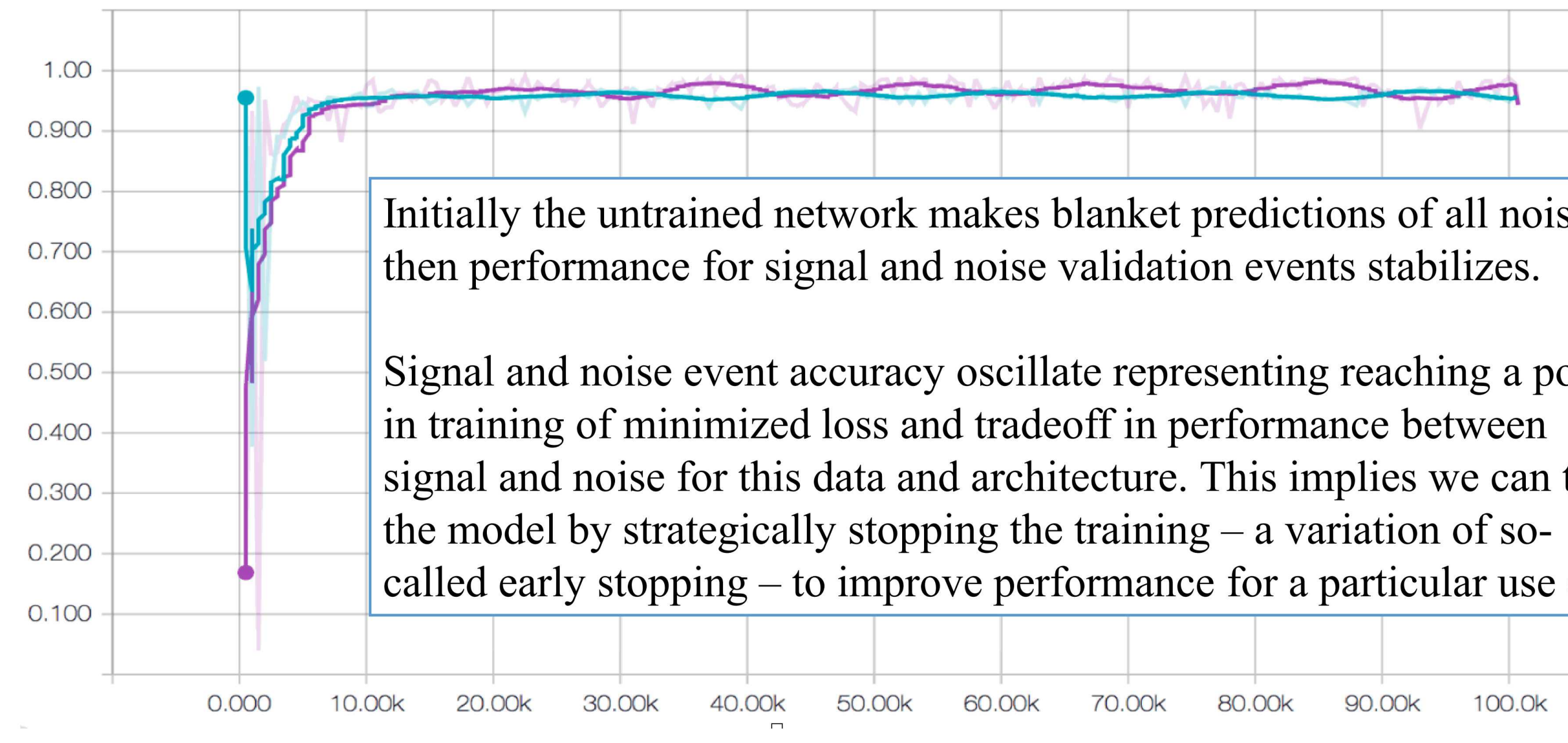
Streams of waveform data are normalized and divided into 10 second windows. Each window is classified as having either seismic noise or signal based on labeled arrivals in the event catalog. Because we have exact arrival times, yet want to train the network to recognize events anywhere in the window, we introduce a time buffer into some events between the start of the window and event onset.

Training:

We train the CNN by minimizing with backpropagation the L2-regularized cross-entropy loss function which measures the difference between the predicted and actual class probability. We form batches of 128 windows consisting of 64 noise and 64 signal events per batch. We use the ADAM optimizer and a learning rate of 10E-4. We use a batch of 128 signal and noise events. We train until the accuracy saturates without any improvement which is at about 96%.

Validation:

We separate the dataset into a primary training set, and a distinct validation dataset. We use data from 01/13/11 00:00 to 01/15/11 00:00 for validation. This gives sufficient signal events to accurately measure performance on a distinct set of data. We have both a signal and a noise validation set. Detection accuracy is the metric we use to gauge performance. We track this metric as the neural network is trained, see figure 3 below.



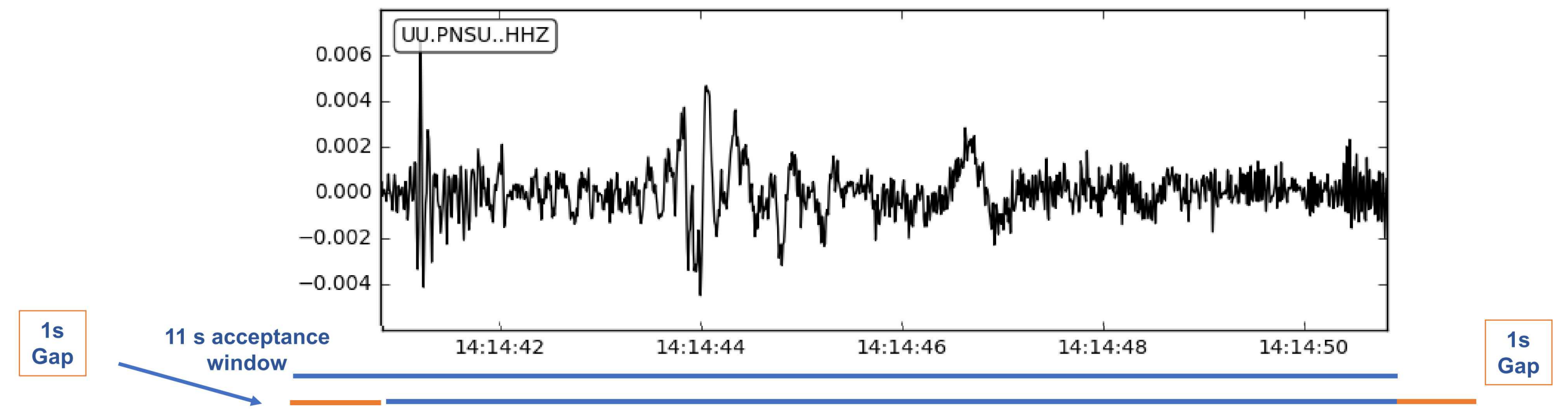
Initially the untrained network makes blanket predictions of all noise, then performance for signal and noise validation events stabilizes.

Signal and noise event accuracy oscillate representing reaching a point in training of minimized loss and tradeoff in performance between signal and noise for this data and architecture. This implies we can tune the model by strategically stopping the training – a variation of so-called early stopping – to improve performance for a particular use case.

Figure 3: Detection accuracy of noise (purple) and signal events (blue).

Results

To measure performance, we run the trained CNN on streaming data and compare the results to expert picked event catalog. Our network requires 10 second windows, and we employ two methods to prevent potential overlap of signal events between windows. First, we choose windows every 11 seconds. Second, we count as a successful picks any 10 second windows in which there is an event from t = -1 s to t = 10s. This is illustrated below.



Although we do have this curated event catalog with expert picked arrivals, it still may be challenging to understand results other than true positive arrivals labeled by the network. To address this in the future, we plan to compare with other established detection methods. Additionally we plan on examining potential false positive and false negative events to expert analysts to verify a lack of signal in the data.

We examine CNN picked arrivals and determine at least a majority are true arrivals that were not previously picked, implying a precision of about 79%. We also examine missed picks and determine they often fall at the end of the signal window.

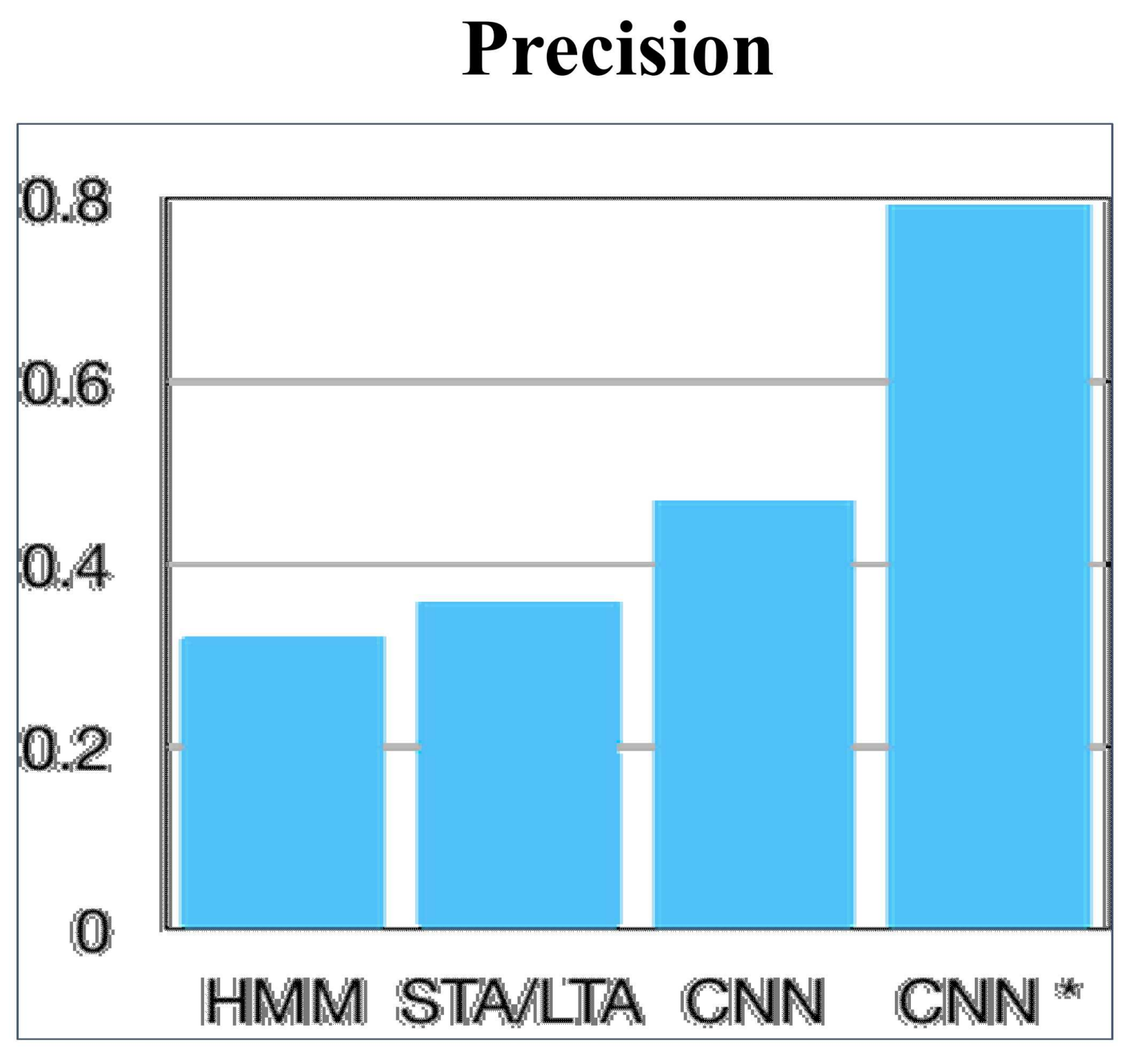
$$Precision = \mathbf{47\%}\ *$$

$$Recall = \ \mathbf{78\%}$$

Confusion matrix for CNN detections for the UU.PNSU station compared to expert picked arrivals

Detections		CNN	
		True	Missed
Expert-picked	True	231	65
	False	263 *	

* We examine CNN picked arrivals and determine at least a majority are true arrivals that were not previously picked, implying a precision of about 79%



Precision of various detection algorithms on this dataset.

HMM – Hidden Markov Model detection algorithm. See Reference 3.
CNN – ConvNetQuake
CNN * - ConvNetQuake with potential newly discovered arrivals

Conclusions

1. CNN provide a fast and scalable way to detect earthquakes from raw 3 component waveform data.
2. When applied to an expert picked dataset, the ConvNetQuake architecture has comparable performance.
3. Further training on large datasets is likely to improve performance, especially at lower magnitudes.

Future Work:

4. Validate arrival picks with various other arrival detection techniques to better understand performance.
5. Examine false positive and false negative events to understand CNN outputs.
6. Move to spectrogram image inputs.
7. Explore various neural network architectures and training methods for increased performance.

Acknowledgements:

We would like to especially acknowledge the original work done by Thibaut Perol, Michael Gharbi and Marine A. Denolle who created the original ConvNetQuake on which this CNN is based.

References:

1. Convolutional Neural Network for Earthquake Detection and Location, Science Advances 14 Feb 2018: Vol. 4, no. 2, e1700578
2. The 2018 Dynamic Networks Data Processing and Analysis Experiment (DNE18), AGU 2018 S53E-0469
3. Detecting Seismic Events Using a Supervised Hidden Markov Model , Lynne Burks, AGU 2017