



**Pacific
Northwest**
NATIONAL LABORATORY

PNNL-30149

Neural MUSE Analysis

July 2020

Jonah Cullen
Isidro Garcia
Thomas Grimes
Cheslan Simpson

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, **makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical
Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: <http://www.ntis.gov>

It is of great interest to the US government to be able to identify and locate illicit nuclear material for protecting the public and preventing the proliferation of nuclear weapons. Some forms of detection involve a stationary detector, such as portal monitors at ports and borders or air samplers on top of buildings. Others involve a moving detector attached to a truck or airplane or handhelds used by human operators. The data is read in as listmode counts at various frequencies, so the operator must rely on software to interpret the results and give feedback in real time. The results, then, must have appropriate accuracies for identifying a threat, and for ignoring background noise.

As an exercise toward innovations for vehicle-mounted detectors in an urban setting, a dataset was created by researchers at Oak Ridge National Laboratory (ORNL) as part of the MUSE (Multi-Agency Urban Search Experiment) project. The data was generated using Monte Carlo transport models through a combination of SCALE, MAVRIC, MCNP, and GADRAS on a simulated city street. The street is empty of all pedestrians and cars, neither driving nor parked. The detector moves down the street 1m off the ground at a constant speed that varies each run. Three example runs are shown in Figure 1, with the sources located in different locations for each run. It is important to note however that the building locations and materials were changed on different runs in order to prevent the algorithm from learning specific to a given street layout.

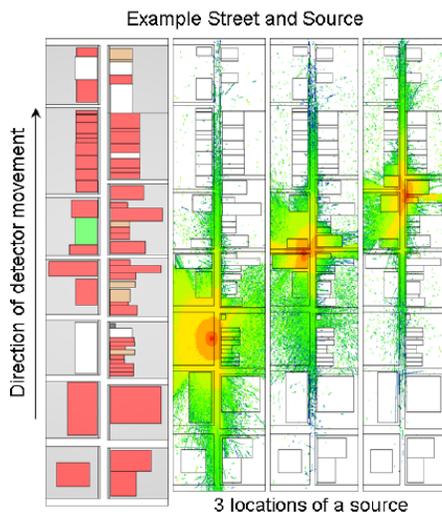


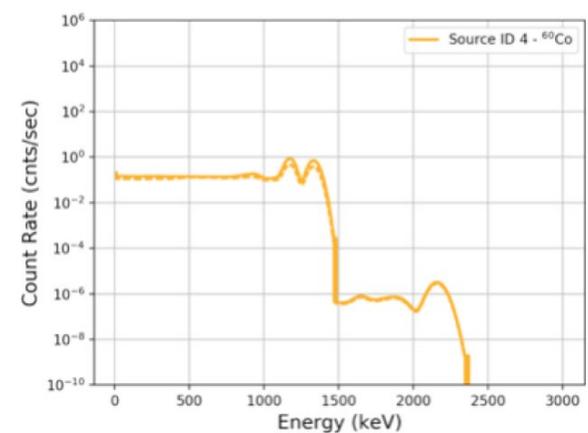
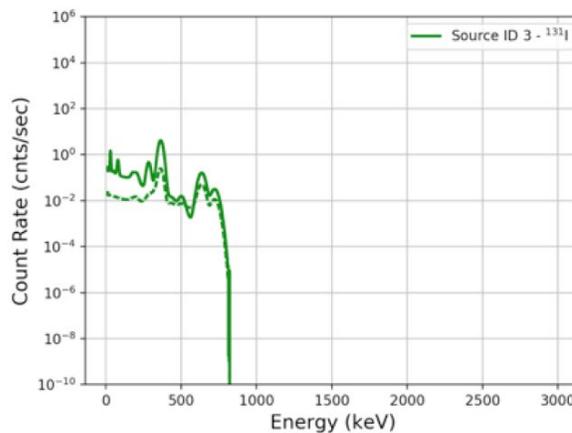
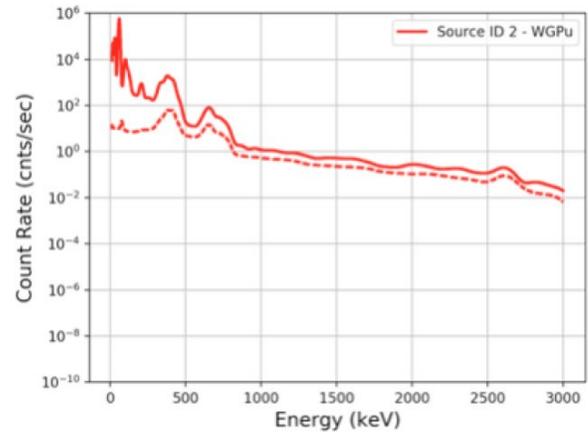
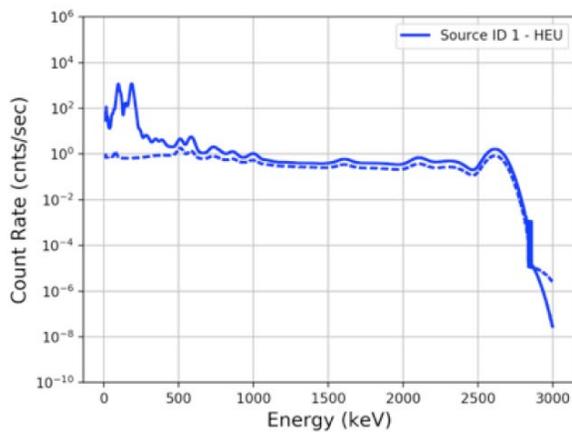
Figure 1: Example city street model and gamma-ray flux from three sources at different locations in the model (Taken from ORNL's *Data for Training and Testing Radiation Detection Algorithms in an Urban Environment*)

The detector has an energy resolution of 7.5% at 661 keV and is a 2 by 4 by 16 inch NaI(Tl) gamma detector. Each run also includes natural background radiation from the concrete and other materials in the model. The dataset was originally created for a competition hosted on Topcoder, of which the winning algorithm scored an accuracy of 76.4% for source identification. This work seeks to build upon this work and improve the results through the application of novel machine learning techniques.

The provided data was separated into two sets: one training set and one testing set. The training set includes a file containing the corresponding correct source labels and source location (given by time at which the detector passes the source) for each run to be used for training a machine learning model. The goals of the created algorithm were to output the correct source or correctly identify no source present, and to give the time in seconds since the beginning of the run at which the truck passed the source, thus providing the location of the source. The dataset includes six source types plus no source present, totaling seven possible labels. The six sources included are:

1. HEU: Highly enriched uranium
2. WGPu: Weapons-grade plutonium
3. ^{131}I : Iodine, a medical isotope
4. ^{60}Co : Cobalt, an industrial isotope
5. $^{99\text{m}}\text{Tc}$: Technetium, a medical isotope
6. A combination of $^{99\text{m}}\text{Tc}$ and HEU

It is common to use counts vs time spectra to identify nuclides. The spectra for these isotopes are shown in Figure 2, and the locations of their peaks act as an identifier for each.



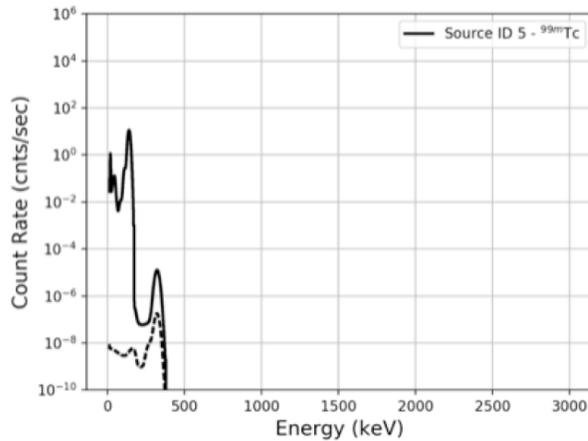


Figure 2: Bare (solid) and shield (dashed) example spectra from all five sources. (Taken from ORNL’s *Data for Training and Testing Radiation Detection Algorithms in an Urban Environment*)

To begin, the data was binned into a 224 by 224 matrix for each run, resulting in multiple counts per energy at various times. These matrices were plotted as “waterfall plots” to give a visual representation of the data. Figure 3 shows these waterfall plots for three different bin spacings: linear, logarithmic, and square root spacing. Each plot is from the same run to demonstrate the differences between the bin spacings.

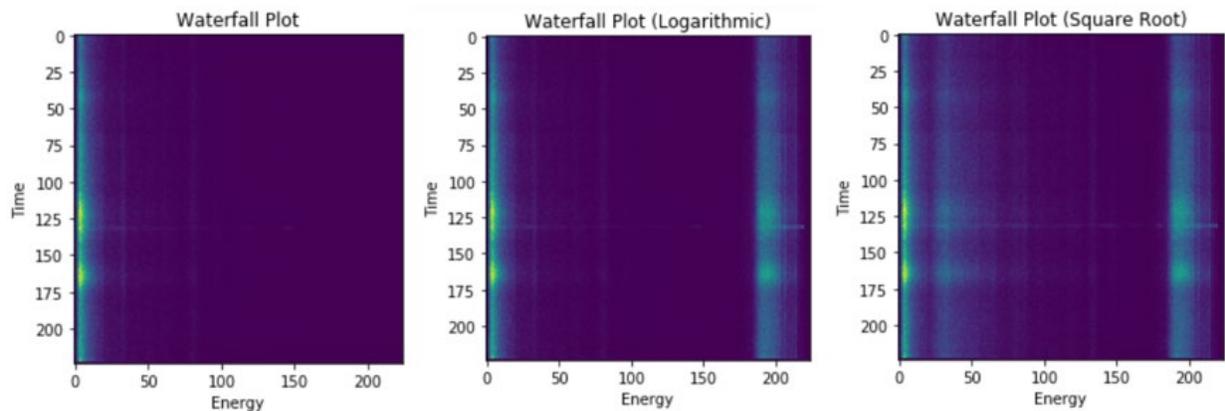


Figure 3: Waterfall plots for a single run with linear, logarithmic, and square root bin spacing. Each horizontal slice is a spectrum with energy on the horizontal axis in keV and number of counts represented in color. These spectra are stacked vertically in time (in microseconds).

The training data provided was then split into training and validation data (80% and 20% of the original training data respectively) to train and evaluate the model with. Each waterfall plot in both set were then mirrored on the time axis to create an identical run that happened in reverse, since the same run should be able to be performed by the truck starting at the other end of the street. A mirrored waterfall plot is shown next to its original in Figure 4.

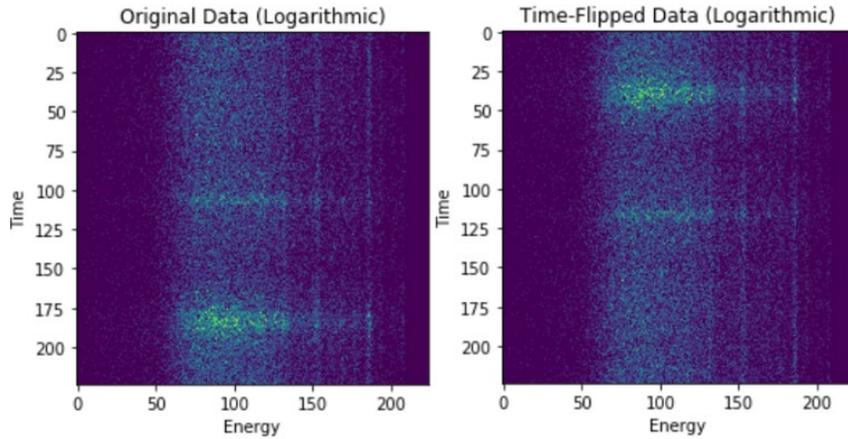


Figure 4: Original and time-flipped waterfall plots for a single run

After each plot was doubled via mirroring, each plot and its mirrored clone were used to create nine more plots each by using Poisson random sampling. The probability of some number of counts being detected far away from the source can be approximated using a Poisson distribution. Each bin of the original data is treated as an independent Poisson Distribution for all instances for which there are sufficient counts (defined as greater than or equal to 5 counts). For all instances of insufficient counts (fewer than 5 counts), the number of counts will vary as $1/r^2$ plus background radiation. Using this knowledge, smoothing was applied along the time axis to generate an estimate of the distribution mean. This was done by taking the average of the current bin and three bins both forward and back in time from that bin for the same energy. Once all the distribution parameters have been determined, the distribution can be resampled to generate new inputs. Figure 5 shows a Poisson copy of a run next to its original.

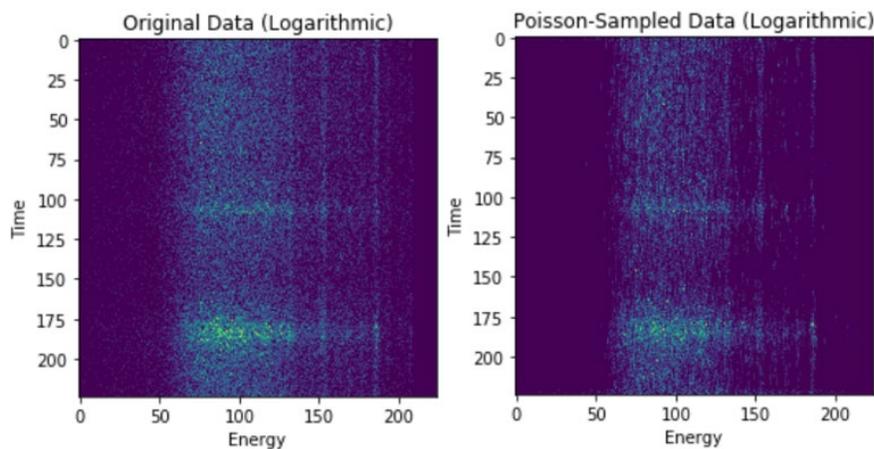


Figure 5: Original and Poisson-sample cloned waterfall plots for a single run.

A residual convolutional neural network was chosen as the algorithm to handle the data for this project. Specifically, a pretrained network called ResNet50 from the Keras module in Python. This network attempts to take advantage of the massively successful capability of convolutional networks for image identification and allow it to go deeper by creating “skip connections” that preserve the backpropagating gradient calculations that can become weakened in deep-layered conventional networks. The output layer of ResNet50 was removed and replaced with a custom layer containing the seven output labels for the corresponding isotopes after being processed through a global average pooling layer. Early stopping was also implemented with a patience of 20 epochs, meaning if there is no improvement on the accuracy for 20 epochs, the model will stop training. The best model’s weights are also saved and can be loaded at any time.

A data generator was also created to load the data into the network in a specific way. To save memory, all of the data was stored in a different file for each run, so the data generator was designed to only load the necessary runs into memory for any given batch. There were also an unbalanced number of runs for different labels, with “no source” dominating almost half of all the data. The data generator was written to choose a label and then randomly select a run corresponding to that label to give each label the same probability of getting selected and therefore a more equal spread of data for each source to prevent the network from choosing any one over others more often than it should.

In training the network, the maximum validation accuracy was 87.97% with a minimum validation loss of 0.4574. The graphs for the training loss and accuracies are shown in Figure 6 for 30 epochs due to the best result being at epoch 10 and early stopping having a patience of 20.

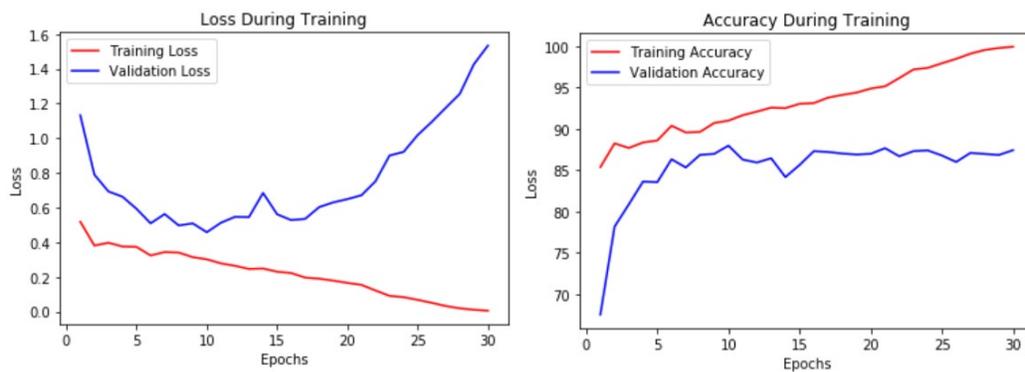


Figure 6: Loss and accuracy during training for both training values (red) and validation values (blue)

The validation accuracies were also investigated deeper using confusion matrices, which show exactly what the model is predicting compared to the correct labels. This was performed on the original data as well as the Poisson clones. These and the total validation confusion matrices are shown in Figure 7.

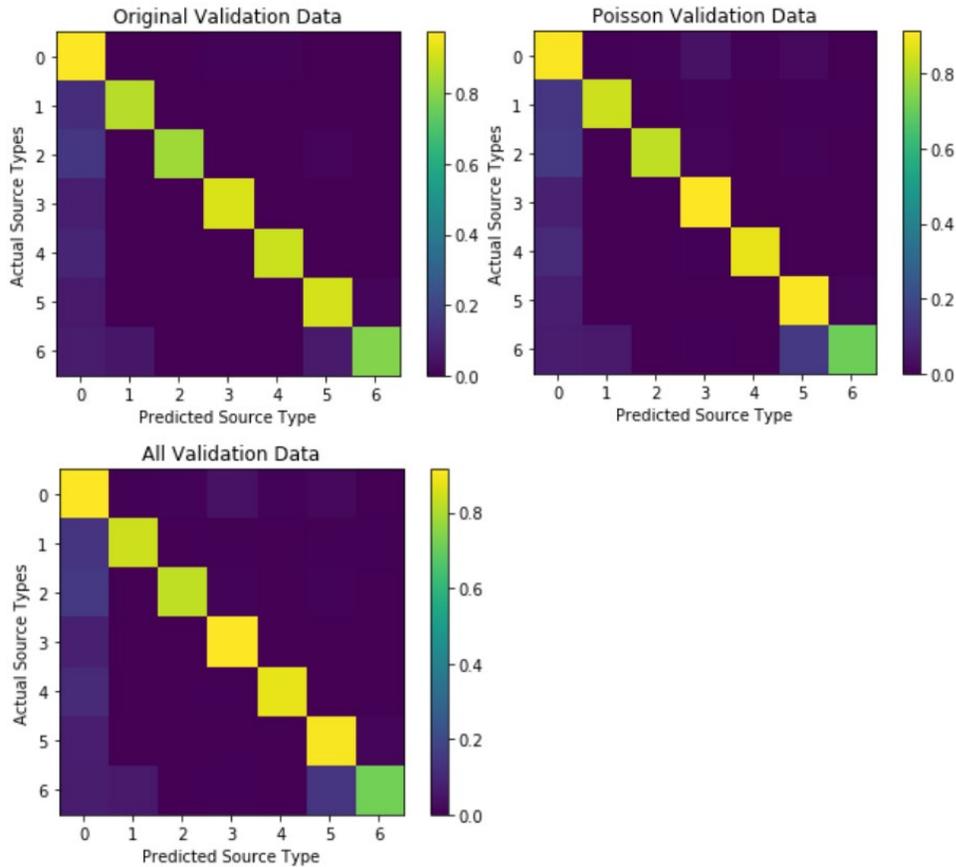


Figure 7: Confusion matrices for original data, Poisson clones, and total validation data.

After training the model, the best model (saved at epoch 10) was loaded and tested with the testing data provided with no labels. It performed with an accuracy of 60.14%, making it much lower than the validation accuracy of 87.97%. This disparity is the subject of investigation for the rest of this project, and several methods have been identified that may be solutions. One method to investigate would be L1 and L2 regularization, which places limits on how much weights can grow to prevent overfitting. Dropout will also be investigated, where certain nodes are removed entirely from the network in order to train the model as an ensemble of skilled subnetworks rather than all of the decisions being made by a single subnetwork. Providing the total elapsed time to the network may also improve it, effectively providing it with the speed the detector was moving at for the run. This data would be fed directly to the output layer to add an extra piece of data with which to make the decision rather than just relying on the activations of the last layer of the neural network. Finally, an adversarial network approach may be taken, where two networks work against each other: one trying to generate new data from the given data, and one trying to determine if the data is real or fake. This would, in theory, create very realistic artificial data that could then be used by the classification network to better train.

References

[1] Peplow, A., et al. Modelling Urban Scenarios and Experiments (MUSE) Dataset. Oak Ridge National Laboratory.

[2] Nicholson, A. D., et al. Data for Training and Testing Radiation Detection Algorithms in an Urban Environment. Pg. 2. <https://doi.ccd.ornl.gov/ui/doi/74>.

[3] Topcoder. Detecting radiological threats in urban areas (2020). URL <https://www.topcoder.com/challenges/30085346>.

[4] Chollet, F., et al. Keras, 2015. URL <https://keras.io>