Development of Machine Learning Algorithms for Directional Gamma Ray Detection

Matthew Durbin[1], Ryan Sheatsley[2], Christopher Balbier[1], Tristan Grieve[1], Patrick McDaniel[2], Azaree Lintereur[1]

[1]Department of Nuclear Engineering
The Pennsylvania State University
137 Reber Building, State College, PA 16801

[2]Department of Electrical Engineering and Computer Science
The Pennsylvania State University
207 Electrical Engineering West, State College, PA 16801

Abstract

The search for rouge radioactive materials can be accomplished through directional detection, but success is often burdened by scatter, attenuation, and Poisson statistics. Thus, it is necessary to utilize algorithms that can maximize the confidence of conclusions drawn from data associated with high noise and variability. A typical method of directional detection data processing is to use a prepopulated database of detector responses with known source locations. An unknown detector response is then compared to this database by preforming a least squares assessment to estimate the angle. While this method is effective when the prepopulated database is representative of the environment of the unknown measurements, it is limited when searching for sources at distances or in environments considerably different than those available in the database. Thus, a method capable of analyzing data with large amounts of variability will advance directional detection capabilities. To that end, various machine learning algorithms were implemented on a series of simulated unknown source location scenarios to develop an algorithm for search applications less dependent on pre-generated data. Monte Carlo Neutral Particle (MCNP6) was used to simulate detector responses for an array of eight 5 cm x 10 cm x 41 cm NaI detectors in a cylindrical configuration. A large dataset was simulated with distances ranging from 1-15 meters at random angles on a plane. Simulated source settings were selected to represent a 10 second count of a 1.4 micro-Curie ^{60}Co source. These conditions yielded various uncertainties, some over 10%, emulating the limited statistics of real-world scenarios and greatly complicating the task at hand. Initial studies using a k-Nearest Neighbors algorithm have yielded a correct location within 2 degrees for approximately 49% of trials with an overall average angular error of 4.3 degrees, compared to the least squared method which yielded a correct location within 2 degrees for approximately 34% of trials with an overall average angular error of 4.7 degrees. This work serves to investigate various machine learning architectures for the use of directional detection problems, in order to create an algorithm effective over a reasonably large search area. Various algorithms are explored to investigate their differences in performance while computing an increasingly complicated source search problem. Presented here are results obtained with simulations that include different scenarios of scatter and obstructions as well as investigations of various algorithm implementations.

I. Introduction

The possibility of rogue or illicit radioactive material in the public sphere poses a threat to international security and public safety. This is especially the case for large gatherings such as the Superbowl or the New York City New Year's Eve ball drop, and for densely populated urban areas. Currently deployed methods to locate these sources typically involve two phases of detection [1, 2]. The first phase utilizes a number of large volume detectors to determine if radioactive material is present in a general area based on elevated count rates but provides no directional information. The second phase tends to rely on personal manually searching for the source on foot, utilizing handheld or portable detectors to determine the source location. This two-phase method is time intensive, and the lack of localization in the first phase can cause the overall method to be inefficient.

Directional gamma ray detection attempts to improve the overall efficiency of the source search scenario, by extracting the source's angular location based on small differences in the counts received in individual detectors within an array of a fixed geometry. Traditional algorithms to process the detector array data for these purposes employ prepopulated databases, or reference tables of known source locations [3]. This method breaks down over large distances, or when the database is not representative of the new measurement's environment. A more robust algorithm, capable of better determining the angular component of a source location in various environments, would improve directional gamma ray detection capabilities. A method which has shown promise in various detection applications, including isotope identification [4] and radon mapping [5] is the use of machine learning algorithms. In these cases, and others, the machine learning algorithms are able to draw more accurate conclusions given statistically burdened data compared to analytical methods. To investigate the potential of machine learning algorithms for directional gamma ray detection purposes, this work tests the performance of various machine learning architectures compared to the traditional analytical method on datasets of source search scenarios. Datasets are simulated using the radiation transport code Monte Carlo N-Particle (MCNP) to model a detector array of eight 5 cm x 10 cm x 41 cm NaI detectors and a ^{60}Co point source at various angles and distances up to 15 m on a plane from the center of the detector array [6]. Three datasets were used, which included trials with no obstruction, trials with an obstruction of fixed location, and trials where the location of an obstruction randomly varies. For each dataset, algorithms were assessed in terms of average angular error, the percent of trials within a specific degree of accuracy, and performance as a function of detector array to source distance.

II. Directional Gamma Ray Detection

For a source incident on an array of detectors in a set geometry, a different solid angle will be subtended by each detector to the source. In addition, the radiation from a source will have a different path to each detector, which may include another detector partially blocking the source's line of sight. These factors will lead to slight differences in each detectors' response, from which the angular component of a source's location should be abstractable. The eight NaI detectors used in this work are arranged in a cylindrical configuration, as shown in Figure 1.

Figure 1. Detector array of eight NaI detectors in a cylindrical geometry.

Traditional reference table algorithms essentially encode the detectors' response as a function of angle, typically for a circle of fixed radius. In this work, a reference table taken at a distance of 3 m is used. The detector array response to an unknown source location is compared against each point in the database, typically using a least squares scheme, and the angular prediction is made by assigning the angle to the point within the database that produces the minimum of the least squares. Given the change in solid angle and corresponding decrease in statistics with increased source to detector array distance, this method breaks down when there are large discrepancies between the reference table and measured distance. With increased distance, the differences in detector response will reduce, and statistical noise will become increasingly dominant. These issues degrade the angular uniqueness [7], making an accurate angular prediction harder as the distances get larger. In addition to distance, the traditional algorithms will break down when the environment of the database is no long representative of the environment of the unknown source measurement. Environmental factors that contribute to this break down include scatter and attenuation associated with obstructions, as well as changing background radiation. For all environments and distances, detection data will be burdened by statistical noise given the limited count times of these applications. The machine learning algorithms were investigated to determine their robustness to distance and various sources of noise in the search source scenario, in hopes of improving overall accuracy and sensitivity of directional detection.

III. Machine Learning Algorithms

Machine learning refers to a class of algorithms that compute a task for which they have not been explicitly programed. This is typically done by exposing the algorithm to a large set of training examples consisting of inputs and corresponding outputs. Upon training, the algorithm updates itself to provide a mapping function that best correlates the training inputs to the correct outputs. Performance is measured by exposing the algorithm to a new set of testing example inputs and determining the accuracy in predicting the correct output. For this work, the counts received in each detector serve as an input, and the angle of the source with respect to the detector array serves as the output. Various machine learning architectures are used in this work, utilizing the python library Sci-Kit Learn[8]. Algorithms used include Logistic Regression, Decision trees, k-Nearest Neighbors, and a simple Neural Network briefly described below.

Logistic Regression (also called *Logit Regression*) stems from one of the most common machine learning algorithms used for regression-based problems (that is, instead of mapping inputs to classes, regression problems are those that call for mappings between inputs and continuous values; for example, predicting housing prices), namely *Linear Regression*. The main difference is that Logistic Regression is designed for classification problems by estimating the probability that a particular input belongs to a particular class. Like Linear Regression, Logistic Regression also computes a weighted sum of input features (with a bias). However, as opposed to outputting this weighted sum directly, Logistic Regression outputs the *logistic* of the result: a sigmoid function that maps any arbitrary input to a number between 0 and 1 (i.e., a probability). Then, it is simple to establish rules for class predictions based on the probability that an input belongs in a given class.

Decision Trees create, as the name implies, a tree-like model of decisions. In general, these models are trained with the *Classification and Regression Tree* (CART) algorithm. Conceptually, the algorithm iterates through each feature k with a threshold t_k. The pair (k, t_k) which minimizes the cost (i.e., the branches produce groups with labels that are homogenous) is chosen. Decision Trees are often favorable in classification problems due to their simplicity, indifference towards nonlinear relationships between features (a common hurdle for many machine learning algorithms), and applicability to many problem domains.

k-Nearest Neighbors (kNN) assigns a class to a point based on the class of the majority of that point's "nearest neighbors." The neighbors are the training data points and are considered nearest based on the Euclidean distance between the training point and the testing point in the feature space defined by the inputs. Among the simplest of machine learning algorithms, kNN is unlike other machine learning algorithms in that it does not explicitly attempt to *generalize* from seen data. Instead, it compares new inputs with all other inputs used for training. This style of learning (often called *instance-based learning*) adapts particularly well to unseen data, as hypotheses are generated on each new input individually, as opposed to building a static hypothesis from training data.

Neural networks map the inputs to the output through a series of hidden layers, consisting of neurons. Each neuron takes an input vector consisting of either the initial inputs, or the outputs of the previous layer. The input vector is multiplied by a weight vector, and a bias vector is added to compute an output. The outputs of each layer are then fed into the inputs of the next layer, until the final output is reached. Through training, these weights are iteratively updated until the best mapping is achieved. The neural network in this work is called a multi-layer perceptron, which is one of the earliest forms of the neural network. The algorithm is set up as a classification problem, with each class corresponding to an angle. All of these algorithms, including the reference table analytical method, were investigated for use in directional detection. The performance parameters considered with the average angular error, percent of trials within a certain degree of accuracy, and accuracy as a function of distance.

IV. Simulations and Datasets

MCNP simulations were utilized to create three datasets to test the performance of the various algorithms at determining the angle of an incident source. The detector array of Figure 1 was modeled and run with a ^{60}Co point source located on a two dimensional plane orthogonal to the long axis of the detectors, intersecting at the crystal center. Figure 2 shows an example of these simulations.

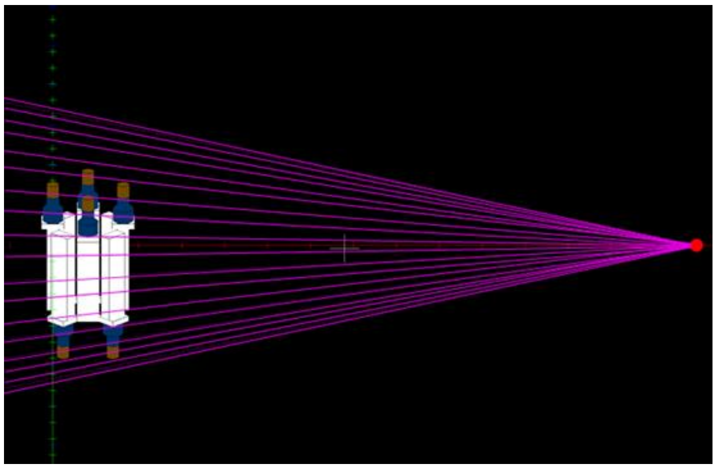

Figure 2. MCNP simulated run of a ^{60}Co point source and detector array.

Dataset-1 had no obstructions and consisted of 72,000 individual MCNP runs such that each angle between 0° and 360° had 200 trials. Exact angles were continuous, but assigned labels corresponding to integer values. Distances between the source and array were randomly varied from 1-15 meters for each trial. A million particles per run were simulated, corresponding to a 14 second count of a 1 µCi ^{60}Co point source. Parameters were set in this way for simulation efficiency and feasibility of future experimental works. An F8 pulse height tally was collected for each of the eight detectors, serving as the eight inputs for the various algorithms.

Dataset-2 was identical to Dataset-1, but with the introduction of a solid concrete obstruction with a density of 2.3 g/cm^3 7.5 m away from the array an angle of 90°. The obstruction was investigated to mimic the effects of a concrete building in an urban environment. The geometry of Dataset-2 is shown in Figure 3. For Dataset-3, trials were run with source angles ranging from 0° to 90°, to reduce the number of simulations which needed to be performed, over the same 1-15 meter range. A smaller concrete obstruction with the same density was used to increase the number of unattenuated particles and was randomly varied between ten locations on the quarter plane. Each location was run with 30 trials for each angle, for a total of 27,000 trials. In addition to one million particle runs for all datasets, the obstruction datasets trials were also run with ten million particles, corresponding to a 14 second count of a 10 µCi ^{60}Co point source. A summary of the datasets is provided in Table 1

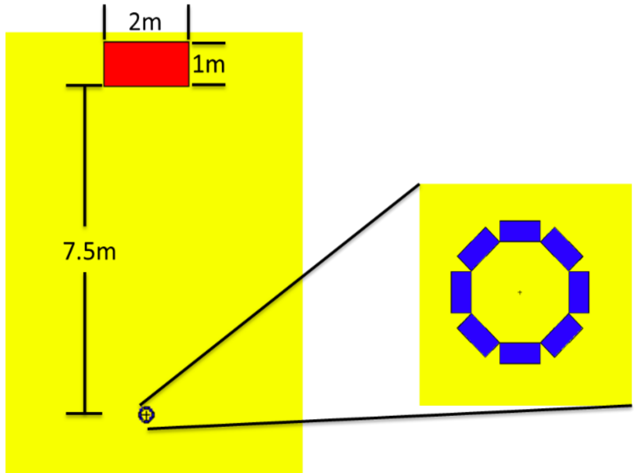

Figure 3. Dataset-2 geometry with concrete obstruction (red) and detector array (blue).

Table 1. Descriptions of Datasets

Dataset	Obstruction Size (m)	Obstruction Location	Angle (°)	Distance (m)	Number of Trails	Particles per Run
Dataset-1	-	-	0-360	1-15	72,000	$1x10^6$
Dataset-2	1x2x5	Fixed	0-360	1-15	72,000	$1x10^6$, $1x10^7$
Dataset-3	0.5x1x5	Varied	0-90	1-15	27,000	$1x10^6$, $1x10^7$

For all datasets, the data was split into 80% training and 20% testing. This is done so that the algorithm is not tested over data it has already been exposed to. The same split of each dataset was used for all algorithms, including the reference table, for consistency in comparison. In addition, a stratified shuffle was used to ensure that each angle was equally represented for both the training and testing sets.

V. Results

The average angular error for each of the investigated algorithms is shown in Table 2 for all trials. It is shown that machine learning can produce a smaller average angular error than the reference table for all the tested datasets. For Dataset-1 and 2, the k-Nearest neighbor algorithm performed the best out of all algorithms, including the reference table. While the introduction of a fixed obstruction reduced the reference table performance by approximately 15%, the k-Nearest Neighbors saw no notable difference. For Dataset-3, the decision trees algorithm performed the best, and was the only algorithm to experience a decrease in error when moving from a fixed obstruction to a random obstruction. Figure 4 shows the accuracy within different degrees of tolerance for all datasets, and Figure 5 shows these metrics as a function of source to detector distance.

Table 2. Average Angular Error of for all Algorithms

Learning Technique	Dataset-1	Dataset-2 10^6	Dataset-2 10^7	Dataset-3 10^6	Dataset-3 10^7
Reference Table	4.74°	5.43°	2.88°	5.79°	3.37°
Logistic Regression	5.73°	7.11°	4.56°	10.61°	9.95°
Decision Trees	5.90°	5.92°	2.76°	5.07°	2.05°
k-Nearest Neighbors	4.31°	4.26°	1.69°	6.83°	3.55°
Multi-layer Perceptron	35.22°	35.92°	35.22°	38.96°	38.96°

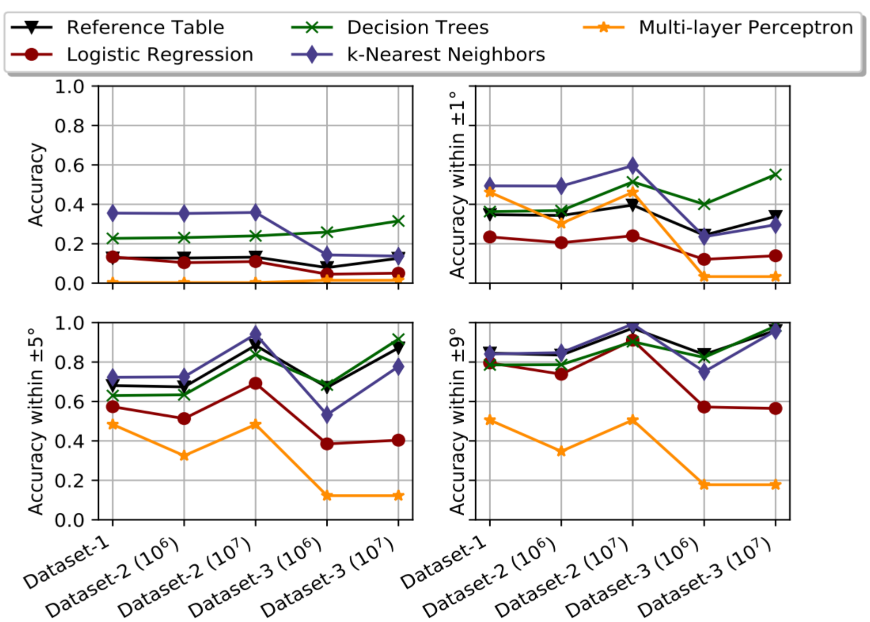

Figure 4. Algorithm accuracy within different tolerances for all datasets.

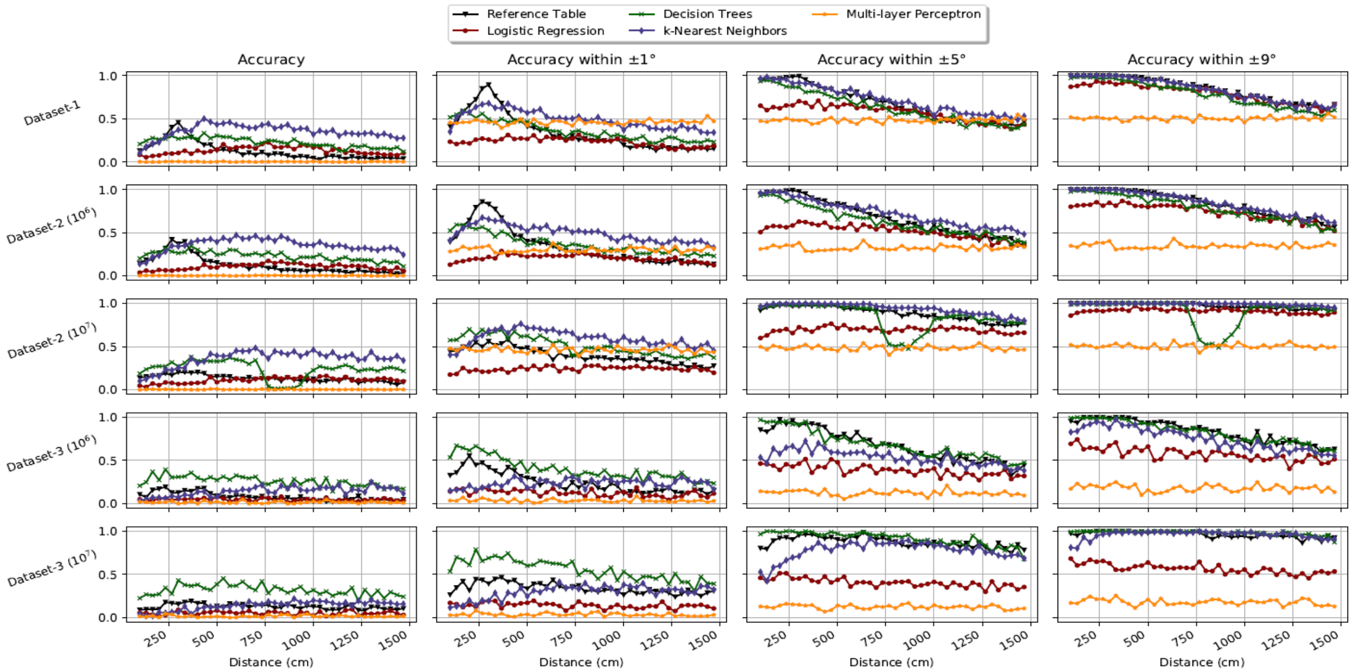

Figure 5. Algorithm accuracy within different tolerances for all datasets as a function of source to detector distance.

In addition to a decreased average angular error, Figure 4 shows how machine learning can also greatly improve the accuracy of directional detection. For Dataset-1 and 2, the accuracy (meaning percent of trials in which the predicted angle matched the true angle) of the reference table method was approximately 12%, while the k-nearest neighbor has an accuracy of close to a factor of three higher at 35%. Within the various degrees of tolerance, a machine learning algorithm was the highest performer across the board for all datasets. In regard to accuracy, the Decision Trees algorithm performed the best for dataset-3. As expected, all algorithms improved performance when exposed to a dataset consisting of a larger number of simulated particles per trial, due to the associated increase of statistics. In addition to the overall performance, a machine learning algorithm was the highest performer for all datasets over the largest range of distances. In many of the subfigures of Figure 5, it is seen that the refence table peaks at around 3 m, which was the distance at which the reference table was created. Other distances were investigated for this method, but the 3 m distance yielded the best results [7]. Unlike the reference table method, the machine learning algorithms do not experience this peak. They have reasonably consistent performance across distances, with some slight dropping with distance due to the associated decreases in statistics. This suggests that while reference table methods will do best when the environments of the measurements match those of the table, machine learning is able to better generalize the relationship between detector response and source angle over a larger range of distances and environments. For the Decision Trees algorithm, there is a dip in accuracy in dataset-2 at around 750 cm. This distance corresponds to the distance the obstruction is located at and may have experienced this dip due to an unlucky shuffle of the training and testing data, that only affected this algorithm type. To mitigate effects like these, future works will include a larger dataset, and will include a stratified shuffle on distance as well as angle. This will ensure that there is equal representation of all distances, in addition to angles, in the training and testing data set.

These results demonstrate that machine learning has the potential to be beneficial for directional detection. Decision Trees and kNN produced the best results, even in the presence of fixed and randomly located obstructions. kNN's success may be in part due to the similarities between how the algorithm operates (measuring "points" that are close together) and the problem domain (measuring particle concentrations that collide with certain points, i.e. detectors). Given how Decision Trees optimize branches by maximizing the number of homogeneous classes on either branch, these algorithms are robust to errors in the training data. Conceptually, the advantage of Decision Trees can be related to the alterations of the physical phenomena that obstructions induce – even though obstructions can drastically change paths of certain particles or photons (thus taking paths unexpected of other particles or photons who did not collide with the obstruction), Decision Trees are capable of largely ignoring such alterations. Recall, however, that Decision Trees are a greedy algorithm: as the impact of obstructions on the path of the radiation increases, the likelihood that the algorithm performs an incorrect split also increases. Thus, while Decision Trees demonstrated impressive results for some of the initial experiments, it is hypothesized that they may struggle as the complexity of the experiments increase future works. Logistic regression, while producing reasonable results for some datasets, may largely be unsuitable for this task. There is a difficulty converging given the lack of good data separation, likely related to the non-linearity of the problem. The multi-layer perceptron gave the poorest results. Attempts to manipulate various hyperparameters (e.g., number of hidden layers, number of neurons per layer, learning rate, etc.) did not have significant impacts of accuracy. While multi-layer perceptrons are a form of neural networks, they are one of the simplest, and lack some of the generalization capabilities often seen in state-of-the-art deep neural networks. To this end, it is expected migration of the neural network architecture to deep neural networks may result in improved performance. Thus, the use of deep neural networks is differed to future work. From the experiments studied in this paper, the results suggest that machine learning is an attractive approach for this problem domain.

VI. Conclusions

This work investigated the use of various machine learning algorithms for directional detection. Datasets consisted of MCNP simulated point source search scenarios on a two dimensional plane, and included a dataset with no obstructions, an obstruction with a fixed location, and an obstruction with a randomly varied location. For each of the tested datasets, a machine learning algorithm outperformed the traditional reference table analytical algorithm in terms of average angular error and accuracy over the range of tested distances. In particular, a k-nearest neighbor algorithm was more accurate than the reference table algorithm by a factor of 3 for the no obstruction and fixed obstruction data set, and the Decision Trees algorithm yielded a similar increase in accuracy for the obstruction with randomly varied location dataset. These results suggest that even with the attenuation and scatter associated with obstructions, machine learning algorithms can outperform reference table methods regarding both error and accuracy over a large range of distances. In addition, with an increasingly complex problem (i.e. obstructions with random locations), different machine learning algorithms can yield the best results. Future works will largely focus on including more complex obstructions more representative of urban environments, incorporating background radiation into measurements, and optimizing a machine learning algorithm to handle the noise associated with both sources of noise. In addition, experimental measurements will be conducted to validate these methods on real world data.

VII. References

[1] A. L. Remick, J. L. Crapo, and C. R. Woodruff, “U.S. national response assets for radiological incidents,” *Health Phys.*, vol. 89, no. 5, pp. 471–484, 2005.

[2] “Radiological assistance program (sixty years),” *National Nuclear Security Administration*, 2018.

[3] C. Schrage, N. Schemm, S. Balkir, M. W. Hoffman, and M. Bauer, “A low-power directional gamma-ray sensor system for long-term radiation monitoring,” *IEEE Sens. J.*, vol. 13, no. 7, pp. 2610–2618, 2013.

[4] L. J. Kangas, P. E. Keller, E. R. Siciliano, R. T. Kouzes, and J. H. Ely, “The use of artificial neural networks in PVT-based radiation portal monitors,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 587, no. 2–3, pp. 398–412, 2008.

[5] A. Varley, A. Tyler, L. Smith, P. Dale, and M. Davies, “Remediating radium contaminated legacy sites: Advances made through machine learning in routine monitoring of ‘hot’ particles,” *Sci. Total Environ.*, vol. 521–522, pp. 270–279, 2015.

[6] J. T. Goorley *et al.*, “No. LA-UR-13-22934,” pp. 0–42, 2013.2019. *Submitted*

[7] M. Durbin *et al.,* "Development of a Fully Connected Residual Network for Direcitonal Gamma Ray Detection." *Int. Con. App. Nuc. Tech*,

[8] F. Pedregosa, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. 1, pp. 2825–2830, 2011.