Anomaly detection and classification of sparse gamma-ray spectra using machine learning algorithms for depleted uranium remediation

By

Austin Matthew Finney

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Biological Engineering
in the Department of Agricultural and Biological Engineering

Mississippi State, Mississippi

May 2020
Anomaly detection and classification of sparse gamma-ray spectra using machine learning algorithms for depleted uranium remediation

By

Austin Matthew Finney

Approved:

______________________________
Fei Yu
(Major Professor/Graduate Coordinator)

______________________________
Steven H. Elder
(Committee Member)

______________________________
Andy D. Perkins
(Committee Member)

______________________________
Charles A. Waggoner
(Committee Member)

______________________________
Jason M. Keith
Dean
Bagley College of Engineering
The quality of the spectral data collected by radiological survey systems depends on many factors including the survey environment, configuration of the system and its detectors, and the radionuclides in question. Algorithms in the field of machine learning have the potential to classify data that would be difficult and time-intensive for a human to analyze. Depleted and natural uranium spectra are of particular interest due to known contamination at domestic sites and world-wide. Several machine learning classifiers were developed with data collected from laboratory experiments. This thesis demonstrates the potential of machine learning algorithms to discriminate gamma-ray emitting sources using sparse, or low-count statistic, data. Effectiveness has been demonstrated for discriminating chemical forms of uranium, mixtures with differing uranium isotope distributions, and predicting source masses given certain detector geometries and a known target distribution. All activity has been supported by the U.S. Army Engineering Research and Development Center (ERDC).
DEDICATION

In dedication to my dear mother, Reda Finney, and the rest of my family for all their love and support along the way.
ACKNOWLEDGEMENTS

This work was supported by the U.S. Army’s Corp of Engineers, Research, and Development Center (W912HZ-16-2-15).

I would like to sincerely thank Ron Unz, Dr. Charles Waggoner, and the rest of ICET’s staff for their great magnanimity and guidance as I pursued my Master of Science degree and began my career. I would like to also thank Dr. Steven Larson and John Ballard at the U.S. Army Engineering Research and Development Center for supporting my research and allowing me to contribute to their project. Thanks to Adam Randle for his assistance with collecting radiation data.
TABLE OF CONTENTS

DEDICATION......................................................................................................................... ii

ACKNOWLEDGEMENTS........................................................................................................ iii

LIST OF TABLES.................................................................................................................... vii

LIST OF FIGURES .................................................................................................................. viii

LIST OF ABBREVIATIONS...................................................................................................... xii

CHAPTER

I. INTRODUCTION.................................................................................................................. 1

1.1 General Survey Concepts ............................................................................................... 1

1.2 Gamma-ray Spectroscopy .............................................................................................. 4

1.3 Differences in DU and NU composition ......................................................................... 7

1.4 Machine Learning .......................................................................................................... 10

1.5 Deep Learning ................................................................................................................ 13

1.6 Research Objectives ....................................................................................................... 18

II. LITERATURE REVIEW ..................................................................................................... 19

III. EXPERIMENTAL DESIGN............................................................................................... 23

3.1 Data Collection ............................................................................................................... 23

3.2 Preprocessing ................................................................................................................ 24

3.3 Model Instantiations ..................................................................................................... 27

3.3.1 Logistic Regression .................................................................................................... 27

3.3.2 Fully-connected Neural Network ............................................................................. 29

3.3.3 Convolutional Neural Network ............................................................................... 30

3.3.4 Autoencoder .............................................................................................................. 33

3.4 Model Evaluation .......................................................................................................... 34

IV. RESULTS.......................................................................................................................... 35

4.1 Logistic Regression (LRs) ............................................................................................. 36

4.1.1 Logistic Regression Model Performance on Dataset 1 ............................................ 36

4.1.2 Logistic Regression Model Performance on Dataset 2 ............................................ 37
4.1.3 Logistic Regression Model Performance on Dataset 3 ........................................38
4.1.4 Logistic Regression Model Performance on Dataset 4 ........................................39
4.1.5 Logistic Regression Model Performance on Dataset 5 ........................................40
4.1.6 Logistic Regression Model Performance on Dataset 6 ........................................41
4.1.7 Logistic Regression Model Performance on Dataset 7 ........................................42
4.1.8 Logistic Regression Model Performance on Dataset 8 ........................................43
4.1.9 Logistic Regression Performance Summary ......................................................43

4.2 Fully-connected Neural Network (FCNNs) .............................................................44
4.2.1 Fully-connected Neural Network Model Performance on Dataset 1 ..................44
4.2.2 Fully-connected Neural Network Model Performance on Dataset 2 ..................45
4.2.3 Fully-connected Neural Network Model Performance on Dataset 3 ..................46
4.2.4 Fully-connected Neural Network Model Performance on Dataset 4 ..................47
4.2.5 Fully-connected Neural Network Model Performance on Dataset 5 ..................48
4.2.6 Fully-connected Neural Network Model Performance on Dataset 6 ..................49
4.2.7 Fully-connected Neural Network Model Performance on Dataset 7 ..................50
4.2.8 Fully-connected Neural Network Model Performance on Dataset 8 ..................51
4.2.9 Fully-connected Neural Network Performance Summary ..................................51

4.3 Convolutional Neural Networks (CNNs) .................................................................52
4.3.1 Convolutional Neural Network Model Performance on Dataset 1 ..................52
4.3.2 Convolutional Neural Network Model Performance on Dataset 2 ..................53
4.3.3 Convolutional Neural Network Model Performance on Dataset 3 ..................54
4.3.4 Convolutional Neural Network Model Performance on Dataset 4 ..................55
4.3.5 Convolutional Neural Network Model Performance on Dataset 5 ..................56
4.3.6 Convolutional Neural Network Model Performance on Dataset 6 ..................57
4.3.7 Convolutional Neural Network Model Performance on Dataset 7 ..................58
4.3.8 Convolutional Neural Network Model Performance on Dataset 8 ..................59
4.3.9 Convolutional Neural Network Performance Summary ..................................59

4.4 Comparison of model performance on zero-valence DU and triuranium octoxide (Datasets 4 and 5) .................................................................60

4.5 Autoencoder performance .......................................................................................62

V. DISCUSSION AND CONCLUSION ...........................................................................63

5.1 Performance on Different Radioisotopes ...............................................................63
5.2 Model Performance on One-Second Spectra .........................................................63
5.3 Effect of Activity and Composition ..................................................................64
5.4 Classifier Comparison .........................................................................................64
5.5 Autoencoder Performance ..................................................................................65
5.6 Conclusion ............................................................................................................66

REFERENCES .............................................................................................................67

APPENDIX

A. TERMS AND EQUATIONS ....................................................................................70
A.1 Convolution .................................................................71
A.2 Max pooling.................................................................72
A.3 Flattening.................................................................72
A.4 Stochastic Gradient Descent (SGD).................................72

B. DECAY SCHEMES ................................................................74
B.1 Uranium-238.................................................................75
B.2 Co-60 ...........................................................................76
B.3 Cs-137...........................................................................77
LIST OF TABLES

Table 3.1  Table of final hyperparameters.................................................................30

Table 3.2  Table of final hyperparameters for the CNN...................................................33
LIST OF FIGURES

Figure 1.1  Schematic of a scintillation detector (Stanford, 2009). ...........................................5
Figure 1.2  One-second LaBr spectrum of background radiation.................................6
Figure 1.3  30-second LaBr spectrum of background radiation. ........................................7
Figure 1.4  One-second LaBr spectrum of a 254g Arizona ore sample...........................8
Figure 1.5  30-second LaBr spectrum of a 254g Arizona ore sample. ...........................8
Figure 1.6  One-second LaBr spectrum of a 286g zero-valence DU sample. ...................9
Figure 1.7  30-second LaBr spectrum of a 322g zero-valence DU sample. .......................10

Figure 1.8  A schematic of a general fully-connected neural network. The I layer is the input layer whose only operation is to pass a unique feature for each neuron. The H layer is a hidden layer (which there may be an arbitrary but certain number of), and the O layer is the output layer. The subscripts N, J, and K refer to the number of neurons in the respective layers........................................14

Figure 1.9  A high-level schematic of a single neuron and its connections. The I layer is the input layer. The arrows are the connections to the next layer with values $\theta_{IJ}$ to $\theta_{NJ}$. The central circle is one of the J hidden neurons represented by the sum of each input times its respective connection value plus a bias term for that layer. The next set of arrows are the outgoing connections with values $\theta_{J1}$ to $\theta_{JK}$. The O layer is the output layer. .................................................................15

Figure 1.10 General schematic of a convolutional neural network. Neurons are only connected to a subset of neurons in the previous layer, referred to as a receptive field. Instead of summing the products of all inputs and respective connections as in fully-connected networks, convolutional neurons only learn a specific set of connections to the previous layer.........................................................16

Figure 1.11 General architecture of a rudimentary autoencoder. The I layer is the input layer of size N. The E layer represents the encoding region of I (N>i>K) hidden neurons. The L layer is the latent layer with K neurons containing the latent representation of the data of interest. The D layer represents the decoding region of j (N>j>K) hidden neurons. The O layer represents the output layer of size N........................................17
Figure 3.1  Detector and laptop setup .................................................................24
Figure 3.2  Depiction of dataset partitions for model optimization ......................26
Figure 3.3  Final FCNN architecture. ................................................................30
Figure 3.4  Initial CNN architecture implementation based on one by Kamuda (2018) 31
Figure 3.5  Architecture of final CNN. .................................................................32
Figure 3.6  Architecture of the autoencoder. Initial configuration was sufficient. 34
Figure 4.1  Confusion matrix of logistic regression model discrimination of background from a small DU fragment .................................................................36
Figure 4.2  Confusion matrix of logistic regression model performance on background, zero-valance depleted uranium, and Arizona ore spectra .................................37
Figure 4.3  Confusion matrix of logistic regression model performance on background, zero-valance depleted uranium, and triuranium octoxide spectra ..................38
Figure 4.4  Confusion matrix of logistic regression model performance on different masses of DU .....................................................................................................39
Figure 4.5  Confusion matrix of logistic regression model performance on various masses of triuranium octoxide ........................................................................40
Figure 4.6  Confusion matrix of logistic regression model performance on background, 1µCi of $^{60}$Co and 0.25 µCi $^{137}$Cs ..................................................................................41
Figure 4.7  Confusion matrix of logistic regression model performance on different masses of DU (30-second spectra) .................................................................42
Figure 4.8  Confusion matrix of logistic regression model performance on various masses of triuranium octoxide (30-second spectra) .............................................43
Figure 4.9  Confusion matrix of fully-connected neural network (fcNN) discrimination of background from a small DU fragment .........................................................44
Figure 4.10 Confusion matrix of fully-connected neural network (fcNN) discrimination of background, zero-valance depleted uranium (ZVDU), and Arizonian uranium ore ..........................................................45
Figure 4.11 Confusion matrix of fully-connected neural network (fcNN) performance on background, zero valance depleted uranium (ZVDU), and triuranium octoxide ($U_3O_8$) ........................................................................46
Figure 4.12 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of zero-valance depleted uranium (ZVDU)...........47

Figure 4.13 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of triuranium octoxide (U₃O₈). .................................48

Figure 4.14 Confusion matrix of fully-connected neural network (fcNN) performance on background, 1 µCi ⁶⁰Co, and 0.25 µCi ¹³⁷Cs.................................................................49

Figure 4.15 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of zero-valance depleted uranium (ZVDU) (30-second spectra).........................................................................................................50

Figure 4.16 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of triuranium octoxide (30-second spectra).........51

Figure 4.17 Confusion matrix of convolutional neural network (CNN) discrimination of background from a small DU fragment.................................................................52

Figure 4.18 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of zero valance depleted uranium and Arizonian uranium ore. ....53

Figure 4.19 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of zero valance depleted uranium, and triuranium octoxide (U₃O₈). ..................................................................................................................54

Figure 4.20 Confusion matrix of convolutional neural network (CNN) performance on one-second spectra of zero-valance depleted uranium (ZVDU) at different masses..................................................................................................................55

Figure 4.21 Confusion matrix of convolutional neural network (CNN) performance on one-second spectra of triuranium oxide (U₃O₈) at different activities. .........................56

Figure 4.22 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of ⁶⁰Co and ¹³⁷Cs.................................................................57

Figure 4.23 Confusion matrix of convolutional neural network (CNN) performance on 30-second spectra of zero-valance depleted uranium (ZVDU) at different activities.................................................................58

Figure 4.24 Confusion matrix of convolutional neural network (CNN) performance on 30-second spectra of triuranium oxide (U₃O₈) at different activities .........................59

Figure 4.25 Bar graph of model accuracies for the zero-valence depleted uranium (ZVDU) dataset. Each model was trained on a random subset of the available data to get a general measure of accuracy. .........................................................60
Figure 4.26 Bar graph of model accuracies for the triuranium octoxide (U₃O₈) dataset. Each model was trained on a random subset of the available data to get a general measure of accuracy.

Figure 4.27 Autoencoder reconstruction error per test sample. Class labels were added after analysis.

Figure A.1 Convolution operation (Jacobs, n.d.) with filter size of three.

Figure A.2 Max pooling with a window size of two.

Figure A.3 Flattening of two-dimensional array into a one-dimensional array.
LIST OF ABBREVIATIONS

DU – Depleted Uranium

NU – Natural Uranium

FWHM – Full Width at Half-Max

ZVDU – Zero-Valence Depleted Uranium

U3O8 – Triuranium Octoxide

MSE – Mean Square Error

LR – Logistic Regression

CNN – Convolutional Neural Network

fcNN – Fully-Connected Neural Network

ReLU – Rectified Linear Unit
CHAPTER I
INTRODUCTION

Radiological surveyors typically travel at a rate 0.5 to 1 m/s, dwelling longer in areas with elevated radioactivity to get more defined spectra for characterization. Sparse gamma-ray spectra present a challenge to surveying efforts as their emission detection distributions tend to lack photopeaks that are characteristic of certain radioisotopes. A reliable means to classify sparse data can facilitate remediation by effectively reducing the false positive rate post-surveying.

There are large areas in the United States, such as Yuma Proving Grounds (YPG), that are widely contaminated with depleted uranium (DU) (Ebinger et al., 1996). Most of this contamination is primarily due to munitions testing. Survey efforts to locate fired DU munitions are complicated by the presence of high concentrations of natural uranium. The cost of remediation efforts relevant to the U.S. Army Engineering Research and Development Center (EDRC) may be reduced by reducing the false positive rate of depleted uranium and natural uranium. The concepts behind this work are also applicable to detecting other kinds of gamma-ray sources.

1.1 General Survey Concepts

Radiological surveying is necessary for determining the locations of radioisotopes that present a threat to public safety. A number of regulatory guidelines have been published to facilitate radiological remediation and decommissioning efforts such as the Multi-Agency
Radiation Survey and Site Investigation Manual (MARSSIM, also known as NUREG-1575) and NUREG-1507.

The classification of survey units is a measure of the contamination potential and thus determines the extent of the survey effort. Areas are first classified as impacted or nonimpacted after a historical site assessment (HSA). The HSA takes into account both high- and low-level factors that may determine the types of radionuclides present and the extent of contamination, such as leak-test data, records of materials containing radionuclides, or whether the radionuclides were open to the environment or continuously encapsulated.

The minimal detectable concentration (MDC) is the a priori minimum expected activity that the instrument in question may detect true positives 95% and false positives 60% of the time. The MDC should be as low as possible in order to minimize the decision errors coincident with interpreting scanning data. The equation for MDC, in disintegrations per minute/100 cm$^2$, is

$$MDC = \frac{3 + 4.65 \sqrt{C_B}}{KT}$$  \hspace{1cm} (1.1)

where

- $K$ is a proportionality constant that maps the detector response to the activity in a sample for a known set of conditions
- $C_B$ is the background count over time, $T$, for paired observations of the sample and blank.

The derived concentration guideline level is the uniform residual radioactivity concentration level within a survey unit that corresponds to the regulatory limit. The range of realistic MDCs is 10% to 50% of the DCGL for direct measurements and sample analyses.

Machine learning tools can be applied to survey data post-collection when subsets of the data are deemed too poor for direct analysis. Predictive machine learning classifiers of DU and
NU with a true positive rate of at least 95% and false positive rate of less than 60% could improve classification of survey units in scenarios where the MDC is close to the DCGL.
1.2 Gamma-ray Spectroscopy

Gamma-ray spectroscopy is the field of measurement concerning the counts and energies of gamma-rays (Gilmore, 2008). In general, the functions of gamma-ray count and energies are characteristic of their sources, making gamma-ray spectra useful for identifying gamma-ray emitting materials.

Scintillation detectors convert ionizing radiation into visible light, which is then converted into an electrical signal. The number of photons per unit energy is the quantum yield, while the time it takes for a scintillated region to stop luminescing is the decay time. The intensity of visible light and subsequent electrical signal is proportional to the energy of the gamma-ray. Scintillation detectors generally have two main components: a scintillator, which may be organic or inorganic. The other component is a photomultiplier tube (PMT), which amplifies the scintillation and converts it into an electrical signal using the photoelectric effect.

There are a number of known scintillation materials (Lowdon et al., 2019). The most widely used scintillator is sodium iodide. It has a quantum yield of about 38,000-55,000 visible light photons per one-MeV gamma-ray and a decay time of 250 ns (Knoll, 2010). NaI scintillators have a tendency to drift, especially with respect to increasing temperature. While these scintillators are suitable for most applications, there are several other scintillators that are seeing more widespread use, such as high-purity germanium (HPGe) detectors and lanthanum-based detectors such as cerium-doped lanthanum (III) bromide (LaBr₃(Ce) or just LaBr). LaBr is similar to NaI in that it has a high effective atomic number and density and emission wavelengths that are compatible with most photocathodes. LaBr has a quantum yield of 63,000 photons per one-MeV gamma and a scintillation decay time of 16 ns. The following is a list of comparisons to illustrate the favorability of LaBr over NaI:
The shorter decay time of LaBr means that it can distinguish between more incident gamma-rays than NaI over a given period.

The greater quantum yield means that the detector is more sensitive to gamma-rays and has a higher count-rate.

LaBr detectors can be programmed to self-calibrate due to the inherent presence of $^{138}\text{La}$.

Each gamma-ray interacts with an atom via Compton scattering. The recipient electron is ejected from its atom carrying a partial amount of the gamma-ray’s energy and interacts with other atoms in the scintillator, producing electron-hole pairs. The electron/hole pairs collapse with each pair yielding a single visible photon. A number of the photons exit the crystal and interact with the photocathode where electrons are freed in a cascade until they reach the anode of the PMT. The charged anode can then be measured by a capacitor so the signal may be passed to another interface. A schematic of a scintillation detector, with scintillator and PMT components, is depicted in Figure 1.1.

![Figure 1.1 Schematic of a scintillation detector](image)

Several factors add uncertainty to the detection process. The subsequent Compton scattering that occurs after each primary gamma-ray interaction generates a distribution of energies around that initial energy, leading to hill shaped peaks instead of precise energy peaks at
one energy. Gamma-rays may also interact with the materials surrounding the detector, followed by the then Compton-scattered gamma-rays entering the detector at lower energies. These scattered, low-energy gamma-rays manifest as a valley between two photopeaks referred to as the Compton shelf. The Compton shelf is more apparent in spectra collected over a longer livetime, such as the one in Figure 1.3.

Figure 1.2 One-second LaBr spectrum of background radiation.
1.3 Differences in DU and NU composition

Natural uranium (NU) has an isotopic composition of 99.28% $^{238}$U, 0.72% $^{235}$U, and trace amounts of $^{234}$U (~0.0058) (US EPA, n.d.). The decay of $^{238}$U leads to detectable amounts of the following radioisotopes which decay at particular energies: $^{234}$Th (63 keV), $^{214}$Pb (295 keV), $^{214}$Bi (609 keV), and $^{234m}$Pa (1001 keV).
The enrichment of $^{235}\text{U}$ leads to ample amounts of depleted uranium, with an isotopic composition of about 99.7% $^{238}\text{U}$ and 0.3% $^{235}\text{U}$. DU is stripped of its $^{238}\text{U}$ daughters during the enrichment process. DU samples begin to reach secular equilibrium between $^{238}\text{U}$ and the
daughters $^{234}\text{Th}$ and $^{234}\text{mPa}$ after 125 days. The presence of these daughters is indicated by their photopeaks at their respective energies in a gamma-ray spectrum. The $^{238}\text{U}$ decay chain depicted in Appendix B shows the set of radionuclides present in natural uranium ore as parent and daughter nuclides.

![Figure 1.6](image)

Figure 1.6 One-second LaBr spectrum of a 286g zero-valance DU sample.
Depleted uranium exists in various chemical forms, but two forms of interest to this paper include triuranium octoxide (U₃O₈) and zero-valance DU (ZVDU). Triuranium octoxide is an intermediate in the refinement process of uranium ore to enriched uranium, while ZVDU is a reduced, metallic form of DU used for military and industrial purposes such as munitions, radiation shielding, and counter-weighting (Briner, 2006). The isotopic distributions between U₃O₈ and ZVDU are generally the same, but the greater density of ZVDU reduces the number of gamma-rays that escape the material. This attenuation results in a difference in the photopeaks, particularly for lower energy ranges.

1.4 Machine Learning

Machine learning is the subset of artificial intelligence where machines learn to map one distribution of values onto another without being explicitly programmed. Machine learning is a broad field that includes a range of algorithms such as linear and nonlinear regression, decision
trees, support vector machines, and many others. Machine learning can be subdivided into four types of learning; they are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

Supervised learning includes algorithms that learn to predict a discrete or continuous value given a distribution of data composed of input vectors and coincident target vectors. The target vectors are generally human-defined labels (e.g., “positive” or “negative”) that are assumed to be true. The labeling process may be particularly time intensive depending on the data. Supervised learning methods tend to be extremely powerful, but their utility is bounded by the quantity of training data available and how well the training data captures the real distribution of the data. Supervised learning methods include linear/logistic regression, many neural network designs such as fully-connectedly connected neural networks and convolutional networks. Fully-connected and convolutional neural networks are frequently used for classification tasks and have been applied to gamma-ray spectroscopy in the literature (M. Kamuda, Stinnett, & Sullivan, 2017; Mark Kamuda & Sullivan, 2019; Mark Kamuda, Zhao, & Huff, 2018).

Unsupervised learning methods are those that find meaningful patterns after training on data without target vectors. These methods are useful when it is impossible or otherwise too expensive to label data for supervised methods. Such methods include principle component analysis, K-nearest neighbors, and autoencoders.

Semi-supervised learning refers to machine learning algorithms that utilize aspects of both supervised and unsupervised learning. These methods are useful in cases when only some instances in the data of interest have known target vectors. An example would be that for an anomaly detection model, the training data may consist of an unlabeled “baseline" distribution, the test data would consist of one or more different types of “anomalous” and baseline data. The
evaluation process would be measuring how well the model differentiated between baseline and anomalies. Those seeking further reference on the domains of supervised and unsupervised learning are directed to *Hands-on Machine Learning with Scikit-Learn and TensorFlow* by Aurélien Géron.

Reinforcement learning methods are those where a given rule is sought after that maximizes some reward. Reinforcement learning methods are beyond the scope of this work but are included to cover the breadth of machine learning. The interested reader is recommended *Reinforcement Learning: An Introduction* by Richard S. Sutton and Andrew G. Barto.

One the simplest machine learning algorithms is logistic regression. Logistic regression is a statistical model that uses a logistic function to map a probability or set of probabilities onto a set of parameters. The logistic regression equation is

\[
P = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_i x_i)}}
\]  

(1.2)

In the above equation, each \( \theta \) value is a parameter that weighs the significance of each feature \( x \), which would be a gamma-ray count from a particular MCA channel. Logistic regression is equivalent to a one-layer neural network. The universal approximation theorem implies that a single-layer neural network can approximate any function (Baker & Patil, 1998). The universal approximation theorem is often used as an argument for using neural networks that contain only one hidden-layer. One-layer neural networks may indeed converge to a solution for many problems but they can fail to generalize to the broader real-world distribution (Goodfellow, 2016). This paper and many other endeavors resort to deep learning methods to generate the most generalizable models for a given task.
1.5 Deep Learning

Deep learning is the subset of machine learning where models learn $2^{nd}$ order parameters or higher that capture more obscure features of the data beyond any $1^{st}$ order parameters. Neural networks are algorithmic abstractions of biological neural networks (Rosenblatt, 1958). They are composed of neurons and their connections which are referred to as weights or parameters. A layer is composed of its constituent neurons and their immediate connections. In feedforward networks, information propagates through three general types of layers: input layers, hidden layers, and output layers. The input layer passes a value for a certain variable for each neuron to the following layer. Hidden layers are those between the input and output layers and may perform various types of operations. The output layer takes the result of the last hidden layer and permutes it into a final value based on the objective of the neural network. For a classification task, the output layer may be configured to predict a class based on the processing of the preceding layers. During training, the output is used to calculate error using a predetermined error function, such as mean-squared error or categorical cross-entropy. This error is used during backpropagation to compute an error gradient as a function of the model’s parameters, which are optimized using a learning function such as stochastic gradient descent (Appendix A.4).

The number of input neurons depends on the number of features, or predictor variables, of the given data. The number of output neurons is equal to the number of class labels for a classification task. The number of hidden neurons depends on the particular problem and is optimized during the training process by the experimenter. Other hyperparameters are randomly initialized then optimized experimentally by some trial and error process.
A schematic of a general fully-connected neural network. The $I$ layer is the input layer whose only operation is to pass a unique feature for each neuron. The $H$ layer is a hidden layer (which there may be an arbitrary but certain number of), and the $O$ layer is the output layer. The subscripts $N$, $J$, and $K$ refer to the number of neurons in the respective layers.

An individual neuron is represented by an activation function, which is a function that determines what type of value will be passed based on its argument. A common activation function that is used in the hidden layers of a neural network is the rectified linear unit (ReLU):

$$relu(x) = \max (0, x)$$ (1.3)

The ReLU function is a decision function that passes the value 0 if its argument is negative and passes the argument itself if that argument is positive. Fully-connected hidden layers simply sum the products of the connection weights and their inputs (plus an additional bias term) and pass the result to the next layer based on the activation function. Figure 1.9 is a graphical depiction of the relationship between an individual neuron and its connections.
Figure 1.9  A high-level schematic of a single neuron and its connections. The I layer is the input layer. The arrows are the connections to the next layer with values $\theta_{1J}$ to $\theta_{NJ}$. The central circle is one of the J hidden neurons represented by the sum of each input times its respective connection value plus a bias term for that layer. The next set of arrows are the outgoing connections with values $\theta_{J1}$ to $\theta_{JK}$. The O layer is the output layer.

The value of the output layer is returned from the model itself after a full forward pass through the model. The softmax function normalizes the output of the model to unity, yielding probabilities of 0 to 1 for each class per sample using the equation

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}}$$  \hspace{1cm} (1.4)$$

where $z_j$ is a vector of the $j$ parameter values of the previous layer, and $K$ is equal to the number of output neurons.

A convolutional neural network is a deep neural network that contains convolutional hidden layers. A convolutional layer is one in which the neurons are not fully connected to the neurons in the preceding layer. Each neuron only has connections to a certain group of neurons
in the previous layer (a receptive field) as depicted in Figure 1.10. Neurons in the convolutional layers perform an operation in which filter matrices of parameters are convolved or cross-correlated with the respective receptive field. These filter matrices of parameters are learned during training in the case of CNNs. Additional details about convolution are available in Appendix A.1.

CNNs may also have fully-connected layers, pooling layers, and dropout layers in addition to convolution layers. Fully-connected layers are layers where each neuron has a connection to all neurons in the previous layer. Pooling layers are dimension reducing layers that may pass a set of average or max values from their previous layers based on the pool size.
Dropout layers are a form of regularization that prune neurons from the previous layer during the model optimization process.

An autoencoder is a neural network with the objective to learn the feature distribution of its training data such that it can reconstruct its input from a compressed representation. Autoencoders have three parts: an encoding region composed of one or more hidden layers, a latent region composed of one hidden layer that contains the fewest hidden neurons and the most compressed representation of the training data, and the decoding region composed of one or more hidden layers. The more dissimilar new input is from the training data, the greater the error when the network tries to reconstruct the training data. The general architecture of feedforward autoencoder is depicted in Figure 1.11.

Figure 1.11  General architecture of a rudimentary autoencoder. The $I$ layer is the input layer of size $N$. The $E$ layer represents the encoding region of $I$ ($N > i > K$) hidden neurons. The $L$ layer is the latent layer with $K$ neurons containing the latent representation of the data of interest. The $D$ layer represents the decoding region of $j$ ($N > j > K$) hidden neurons. The $O$ layer represents the output layer of size $N$. 

17
1.6 Research Objectives

One-second gamma-ray spectra are not defined well enough to make classification decisions with the naked eye, especially between different permutations of a single radionuclide (DU). The primary goal of this project was to find a classification model that can reliably make a decisive prediction on experimentally collected low count-statistic (sparse) gamma-ray spectra of DU samples at low concentrations. The classification algorithms that were investigated were logistic regression, fully-connected neural networks, and convolutional neural networks. The autoencoder algorithm was also investigated in order to find a method that may be effective when only background spectra are available but unknown, illicit radionuclides are expected. The secondary goal of this project was to figure out how to optimize the hyperparameters of each candidate algorithm in order to get the best fit with each model.
CHAPTER II
LITERATURE REVIEW

There is a growing body of research with the aim to automate gamma-ray spectroscopy for applications such as soil characterization and portal monitoring, especially with machine learning methods. The following is a brief overview of past work relevant to the methodology and objectives of this paper. The authors of one paper (Kangas, Keller, Siciliano, Kouzes, & Ely, 2008) developed a neural network as an anomaly detection system for radiation portal monitoring. A radiation portal is a checkpoint where objects are scanned for hazardous radioisotopes. The authors were seeking a replacement for an algorithm that yielded an alarm when the change in gamma-ray detections over time reached a certain threshold. One of the goals of the anomaly detection system was to minimize the false positive rate while maximizing the true positive rate of illicit radionuclides. They collected their data using a plastic scintillator, polyvinyl toluene (PVT), which has relatively poor energy resolution. However, this detector was capable of measuring gamma-ray activity and elucidating spectral shape after 512 channel MCA was added. The authors note that they did not control for the natural-background radiation shielding properties of the truck. Despite these limitations, the authors were able to reach the desired false alarm probability target through the combined use of change in gamma-ray counts and spectral shape as predictive variables.

More recently, Sharma et al. (2012) explored machine learning algorithms and other statistical methods as spectral anomaly detectors at a radiation portal in a public space. The
authors used NaI detectors to collect background data and simulated $^{57}$Co spectra. They made separate alarm systems for rain and non-rain days based on each algorithm. The authors acknowledged the performance of neural networks in the literature but also sought more “dynamic” methods. The neural network architecture that was evaluated was an autoassociator (or autoencoder), which was compared to a support vector machine, Mahalanobis distance, and Variance in Angle Spectrum. The authors found the training periods of the autoassociator and the Variance in Angle Spectrum to be prohibitive. The authors concluded that Mahalanobis distance was the most preferable method due to the inherently Gaussian characteristics of the spectra. However, the assumption of a multivariate Gaussian distribution for gamma-ray spectra data does not always hold (Varley, Tyler, Smith, & Dale, 2015).

Several experiments have shown the feasibility of applying neural networks to the analysis of natural and manmade gamma-ray sources (Chen & Wei, 2009; Dragovic & Onjia, 2007; Varley, Tyler, Smith, Dale, & Davies, 2016). Dragovic (2007) compared statistical and machine learning models as regional classifiers of soil samples. The gamma-ray spectra that they used to characterize the soil samples were collected by a high-purity germanium (HPGe) detector. HPGe detectors have higher resolution than the LaBr$_3$(Ce) scintillation detector used in this study, but their operation requires a sophisticated cooling system (Marian et al., 2015). Their relatively simple neural network (8 neurons x 7 neurons x 15 neurons) demonstrated the arguable advantages of deep learning models over conventional statistical models in the domain of gamma-ray spectroscopy. Medhat (2012) performed a feasibility study similar to Dragovic et al. using soil samples with curated radioisotopes and a fully-connected neural network architecture.

Varley et al. (2015) demonstrated a consistent performance advantage of using neural networks over support vector machines (another class of machine learning algorithm). Varley et
al. (2016) deployed neural networks for the prediction of depth and activity of manmade $^{226}$Ra deposits. They used both Monte Carlo simulated data and data collected from handheld LaBr detectors. The inclusion of simulated data allowed them to better control the uncertainty across their training and validation sets. Their training set consisted of data with low statistical uncertainty (with their measured data being collected over 60 seconds) while the validation set consisted of data with higher statistical uncertainty consistent with walkover data collection.

Wei (2009) sought to further develop the application of neural networks to gamma-ray spectroscopy by using the Karhunen-Loeve (K-L) transform of the spectra as input features. They used a sodium iodide (NaI) detector to measure the activity of their isotopes of interest. This method greatly reduced the number of spectral features (about 1200 with their detectors), which reduced the computational complexity of the neural network. Such a compact architecture is more likely to be usable on portable devices than those with the full range of channels.

The works by Kamuda et al. (2017 and 2018) were of particular importance to the development of this paper. Their earliest paper (M. Kamuda et al., 2017) explored the efficacy of fully-connected neural networks as classifiers of NaI spectra and appeared to be the first in the literature to develop models for sparse gamma-ray spectra. The spectra with the lowest count statistics were those simulated at a 10-second detector live-time. All of their spectral data were simulated using Monte Carlo N-Particle 6 (MCNP6), which simulates many aspects of the data collection process including detectors, radiation sources, geometries, and various low-level variables. Kamuda et al. demonstrated that a fully-connected neural network could accurately predict mixing coefficients, the percent contribution of gamma-ray counts by particular radioisotopes within a mixture.
The 2018 paper (Mark Kamuda et al., 2018) then compared the performance of a fully-connected neural network against that of a convolutional neural network. Comparisons were made based on classification performance and predictions of mixing coefficients. The CNN had less variance in its output compared to the FCNN. This was attributed to the ability of a CNN to learn relationships about features in close proximity to each other, such as the Compton continuum along with photopeaks. This paper suggested that a CNN would be a prudent follow-up to the FCNN, as well as provided a proven CNN architecture to start from. It also showed that neural networks could be trained to be invariant to gain shift, a calibration problem that plagues NaI detectors in particular. Another paper (Chatzidakis & Botton, 2019) corroborates this capability for gain shift invariance with a fully-convolutional neural network.
CHAPTER III
EXPERIMENTAL DESIGN

3.1 Data Collection

A 7.6 cm by 7.6 cm Saint-Gobain BrilLanCe 380 cerium-doped lanthanum bromide (LaBr$_3$(Ce)) scintillation detector as shown in Figure 3.1 was used to create a spectrum library of natural uranium and DU samples. LaBr$_3$(Ce) scintillators have a relatively high energy resolution, a high ratio of emitted photons to detection energy, and a short decay time compared to sodium iodide scintillators (Mouhti, Elanique, Messous, Belhorma, & Benahmed, 2018). The LaBr$_3$(Ce) detector was connected with 14-pin Ortec® digiBASE™ which includes a power supply, preamplifier, and multi-channel analyzer (MCA). The digiBASE was interfaced with a Panasonic Toughbook® by USB. Ortec® GammaVision® software was used for operating the digiBASE and collecting spectral data. This detector is also the same type used on the ICET mobile surveying units. Background and source data were collected on the same days. Sources were placed directly beneath the detector at the same distance as the mobile survey units’ distance to the ground.
Data collection was automated using an Ortec® Job Control script. The script controls the detector live-time, resets the MCA, collects and saves data, and reiterates the process for each spectrum. The detector performed ~993 measurements with one-second live-times to approximate short resident times of mobile platforms.

3.2 Preprocessing

A script was written in Python and some of its supported data science libraries that handled the machine learning workflow including preprocessing, neural network architecture, model predictions, and model saving. Scikit-Learn and Pandas are Python software libraries. Scikit-Learn is a library for data science and machine learning that has many data preprocessing tools and Pandas is a library developed for working with multidimensional datasets in science and greatly facilitates the process of formatting data for other libraries. TensorFlow is a machine
learning framework developed by Google primarily aimed at deep learning applications that can be used to design and train neural networks and other deep-learning models.

GammaVision outputs each individual spectrum as an SPE file in a comma-separated value format. The pandas (derived from “panel data”) library was used to convert the spectral data from comma-separated value format into pandas dataframes and label each spectrum according to the source type for the neural network model. The following datasets were defined from the spectrum library:

- 1-second detector live-time, (background class also included)
  
  Dataset 1: 75 g zero-valance DU  
  Dataset 2: 322 g zero-valance DU, 245 g Arizonian uranium ore  
  Dataset 3: 93.3 g zero-valance DU, 118.75 g U₃O₈  
  Dataset 4: 75 g, 86.2 g, 93.3 g, 322 g, and 842.8 g of zero-valance DU  
  Dataset 5: 1.17 g, 2.36 g, 6.14 g, 11.84 g, 23.65 g, 59.05 g, 118.75 g of U₃O₈  
  Dataset 6: 1 µCi Co-60, 0.25 µCi Cs-137

- 30-second detector live-time, (background also included)
  
  Dataset 7: 75 g, 86.2 g, 93.3 g, 322 g, and 842.8 g of zero-valance DU  
  Dataset 8: 1.17 g, 2.36 g, 6.14 g, 11.84 g, 23.65 g, 59.05 g, 118.75 g U₃O₈

where each class has approximately 993 samples.

Scikit-Learn was used for delineating training, test, and validation sets. The training set is the labeled data the supervised machine learning models use to optimize the parameters in their hidden layers. The test set is unlabeled data used to measure the model’s performance as its hyperparameters are optimized. The validation set consists of unlabeled data used to evaluate the neural network’s performance after training and hyperparameter optimization.
The Scikit-Learn Python package was used to delineate training, validation, and test sets. The train/validation/test split was 60/20/20, meaning 60 percent was used for training, 20 percent was used for hyperparameter optimization, and 20 percent was set aside as a holdout to validate the final model. This splitting process is necessary to have a way to tweak models after training, and then evaluate on unseen data. The test set is necessary to ensure the hyperparameters are not overfitted to the validation set and the model generalizes. The final result of preprocessing is depicted in the following figure:

![Diagram of dataset partitions for model optimization.](image)

Figure 3.2 Depiction of dataset partitions for model optimization.

Data were normalized using min-max normalization. The equation for min-max normalization for each spectral feature is

\[
    x_{\text{normalized}} = \frac{x - x_{\text{minimum}}}{(x_{\text{maximum}} - x_{\text{minimum}})}
\]

(3.1)
3.3 Model Instantiations

The machine learning models were implemented using Sci-kit Learn, Keras, and TensorFlow (Abadi et al., 2016; Géron, 2019; Pedregosa et al., 2011). Each algorithm was implemented separately.

3.3.1 Logistic Regression

The logistic regression model was implemented using Sci-kit Learn’s logistic regression class object. The model was trained straightforwardly using a pandas DataFrame of spectral counts as the set of input vectors and the labels as the set of target vectors. The training phase is essentially the optimization of the model’s internal parameters such that, for each forward pass through the model, the error between the model’s output and the true value is minimal. This optimization was carried out using stochastic gradient descent as the optimizer on the categorical cross-entropy cost function until the model reached perfect accuracy on the training data set or at least converged to a local minimum in the error space. The equation for categorical cross-entropy is

\[ E = - \sum_{j=0}^{M} \sum_{i=1}^{N} (y_{ij} \log (\hat{y}_{ij})) \]  

(3.2)

where

- \( E \) is error
- \( M \) is the number of possible classes
- \( N \) is the number of samples
- \( y_{ij} \) is the true probability of a class label
- \( \hat{y}_{ij} \) is the predicted probability of a class label
Model predictions and accuracy on the training, test, and validation sets were the primary outputs of the model, which were passed to a python module to be presented as confusion matrices.
3.3.2 Fully-connected Neural Network

Rudimentary neural network designs were explored to reduce the risk of developing overfit 1st order models. The first neural network architecture that was explored was the fully-connectedly-connected neural network (FCNN). The algorithm was implemented using the Keras backend of TensorFlow 2 along with some of TensorFlow’s utilities such as tensorflow.distribute, which was used to parallelize the training process across the two Nvidia Titan Vs GPUs. Distribution of the training processes across multiple processing units is a common means of expediting training sessions. FCNNs were trained using stochastic gradient descent as the optimizer and categorical cross-entropy as the loss function. Parallel training decreased the time it took to train each FCNN instantiation nearly by a factor of two.

The architecture of the machine learning model and other hyperparameters were revised until the model’s loss function converges on the validation data. The FCNN architecture was revised this way to see how the depth (number of hidden-layers) and width (number of hidden neurons per hidden layer) affected FCNN performance. The first instantiation consisted of a single-hidden layer FCNN with 1000 hidden units. The number of hidden units was iteratively reduced until model performance began to decline. Once the model has converged on a set of hyperparameters and the loss can no longer be reduced, the model was then evaluated on a test set to predict model performance in general. The final FCNN architecture is depicted in Figure 3.3.
Figure 3.3  Final FCNN architecture.

Table 3.1  Table of final hyperparameters.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Final Hyperparameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hidden units per hidden layer</td>
<td>200</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.00001</td>
</tr>
<tr>
<td>Batch size</td>
<td>40</td>
</tr>
</tbody>
</table>

### 3.3.3 Convolutional Neural Network

A neural network design based on that implemented by Kamuda et al, 2018 was reimplemented and evaluated. The design was composed of layers in the following sequences:

- one 1-dimensional (1D) convolution layer,
- three max pooling layers,
- three 1D convolution layers,
- five max pooling layers,
- five flattening layers,
• two fully-connected + dropout pairs
• a fully-connected + softmax layer.

The convolutional layers passed their output using the ReLU function. The measure of error, $E$, within the model was measured using categorical cross-entropy.

![Diagram of CNN architecture](image)

Figure 3.4 Initial CNN architecture implementation based on one by Kamuda (2018).

The gamma-ray spectra produced by the LaBr$_3$(Ce) detector has 1024 features (MCA channels), so there are 1024 neurons in the input layer. The parameters of the model were optimized for the lowest possible values of the error function using Adam, short for adaptive
moment estimation, which is a stochastic optimization method that uses only the 1st order
gradients of the error function (Appendix A.5).

The former architecture introduces considerable translational invariance, which means
that the model may learn similar shapes in different spectral regions, increasing the confusion
between classes (Kauderer-Abrams, 2017). Implementing a shallower and more purely
convolutional architecture is one way of reducing translational invariance as there are fewer
pooling layers that reduce the association between adjacent features. Fewer layers in general is
associated with less translational invariance (Dai, Li, He, & Sun, 2016; Kauderer-Abrams, 2017).
The final architecture consisted of

- an input layer,
- two convolutional layers with the ReLU activation function
- a flattening layer
- a fully-connected + softmax output layer.

![Figure 3.5](image)

Figure 3.5 Architecture of final CNN.
Table 3.2  Table of final hyperparameters for the CNN.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Final Hyperparameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter size</td>
<td>2</td>
</tr>
<tr>
<td>Stride length</td>
<td>2</td>
</tr>
<tr>
<td>Number of filters per convolutional layer</td>
<td>600</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.00001</td>
</tr>
<tr>
<td>Batch size</td>
<td>40</td>
</tr>
</tbody>
</table>

3.3.4  Autoencoder

The autoencoder was trained on background data and evaluated on a mix of background and several other sources. Optimization was done using mean squared error (MSE) as the cost function. The equation for MSE is

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2. \]  (3.3)

MSE is also the error function used to measure the reconstruction error, the difference between the input and the reconstruction from the output. The difference in RE between samples is the metric by which anomalous spectra can be distinguished from background. From beginning to end, the autoencoder had the following dimensionality of neurons per layer: 1024, 512, 256, 200, 256, 1024.
3.4 Model Evaluation

The model accuracies and predictions on individual spectra in the test data were the primary metrics used for evaluation. The predictions of each trained model on the respective test data were organized into confusion matrices using matplotlib. Confusion matrices (Figures 4.1-4.24) are used to illustrate model accuracy in several scenarios by visually comparing the class predictions with the actual class identities.
CHAPTER IV

RESULTS

The confusion matrices in sections 4.1 - 4.3 are measures of model classification performance. The sum of the numbers in a row is the total number of samples of that particular class (set of labels). The sum of the numbers in a column is the total number of predictions of that class. The overlap of a column and row for a particular class represents an accurate prediction. Comparisons of the three classification algorithms on the most challenging datasets are made in section 4.4.
4.1 Logistic Regression (LRs)

4.1.1 Logistic Regression Model Performance on Dataset 1

Figure 4.1 Confusion matrix of logistic regression model discrimination of background from a small DU fragment
4.1.2 Logistic Regression Model Performance on Dataset 2

Figure 4.2 Confusion matrix of logistic regression model performance on background, zero-valence depleted uranium, and Arizona ore spectra.
4.1.3 Logistic Regression Model Performance on Dataset 3

Figure 4.3  Confusion matrix of logistic regression model performance on background, zero-valance depleted uranium, and triuranium octoxide spectra.
4.1.4 Logistic Regression Model Performance on Dataset 4

Figure 4.4 Confusion matrix of logistic regression model performance on different masses of DU.
4.1.5 Logistic Regression Model Performance on Dataset 5

Figure 4.5 Confusion matrix of logistic regression model performance on various masses of triuranium octoxide.
4.1.6 Logistic Regression Model Performance on Dataset 6

Figure 4.6 Confusion matrix of logistic regression model performance on background, 1µCi of $^{60}$Co and 0.25 µCi $^{137}$Cs.
4.1.7 Logistic Regression Model Performance on Dataset 7

Figure 4.7 Confusion matrix of logistic regression model performance on different masses of DU (30-second spectra).
4.1.8 Logistic Regression Model Performance on Dataset 8

Figure 4.8 Confusion matrix of logistic regression model performance on various masses of triuranium octoxide (30-second spectra).

4.1.9 Logistic Regression Performance Summary

The logistic regression models trained on Datasets 1-4 and 7-8 all performed with 100% accuracy. The models trained on the one-second ZVDU and U₃O₈ datasets performed with near perfect accuracy for all classes with the exception of the lowest activity classes.
4.2 Fully-connected Neural Network (FCNNs)

4.2.1 Fully-connected Neural Network Model Performance on Dataset 1

Figure 4.9  Confusion matrix of fully-connected neural network (fcNN) discrimination of background from a small DU fragment.
4.2.2 Fully-connected Neural Network Model Performance on Dataset 2

Figure 4.10 Confusion matrix of fully-connected neural network (fcNN) discrimination of background, zero-valance depleted uranium (ZVDU), and Arizonian uranium ore.
4.2.3  Fully-connected Neural Network Model Performance on Dataset 3

Figure 4.11  Confusion matrix of fully-connected neural network (fcNN) performance on background, zero valence depleted uranium (ZVDU), and triuranium octoxide (U₃O₈).
4.2.4 Fully-connected Neural Network Model Performance on Dataset 4

Figure 4.12 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of zero-valence depleted uranium (ZVDU).
4.2.5 Fully-connected Neural Network Model Performance on Dataset 5

![Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of triuranium octoxide (U₃O₈).](image)

Figure 4.13 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of triuranium octoxide (U₃O₈).
4.2.6  Fully-connected Neural Network Model Performance on Dataset 6

Figure 4.14  Confusion matrix of fully-connected neural network (fcNN) performance on background, 1 µCi $^{60}$Co, and 0.25 µCi $^{137}$Cs.
4.2.7 Fully-connected Neural Network Model Performance on Dataset 7

Figure 4.15 Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of zero-valance depleted uranium (ZVDU) (30-second spectra).
### 4.2.8 Fully-connected Neural Network Model Performance on Dataset 8

**Figure 4.16** Confusion matrix of fully-connected neural network (fcNN) performance on background and various masses of triuranium octoxide (30-second spectra).

<table>
<thead>
<tr>
<th>True label</th>
<th>Predicted label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>SU308-1 1.17 g U3O8</td>
</tr>
<tr>
<td>Background</td>
<td>215</td>
</tr>
<tr>
<td>SU308-1 1.17 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-2 2.36 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-3 6.14 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-4 11.84 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-5 23.65 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-6 59.05 g U3O8</td>
<td>0</td>
</tr>
<tr>
<td>SU308-7 118.75 g U3O8</td>
<td>0</td>
</tr>
</tbody>
</table>

### 4.2.9 Fully-connected Neural Network Performance Summary

The FCNN model performed well on datasets 1-3 and 6 as can be seen in figures 4.9-11 and 4.14. Model performance was less adequate for datasets where the illicit radionuclides only differed in amount of activity.
4.3 Convolutional Neural Networks (CNNs)

4.3.1 Convolutional Neural Network Model Performance on Dataset 1

Figure 4.17 Confusion matrix of convolutional neural network (CNN) discrimination of background from a small DU fragment.
4.3.2 Convolutional Neural Network Model Performance on Dataset 2

Figure 4.18 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of zero valance depleted uranium and Arizonian uranium ore.
4.3.3 Convolutional Neural Network Model Performance on Dataset 3

Figure 4.19 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of zero valence depleted uranium, and triuranium octoxide (U$_3$O$_8$).
4.3.4  Convolutional Neural Network Model Performance on Dataset 4

Figure 4.20  Confusion matrix of convolutional neural network (CNN) performance on one-second spectra of zero-valance depleted uranium (ZVDU) at different masses.
4.3.5 Convolutional Neural Network Model Performance on Dataset 5

![Confusion matrix of convolutional neural network (CNN) performance on one-second spectra of triuranium oxide ($U_3O_8$) at different activities.](image)

Figure 4.21 Confusion matrix of convolutional neural network (CNN) performance on one-second spectra of triuranium oxide ($U_3O_8$) at different activities.
4.3.6 Convolutional Neural Network Model Performance on Dataset 6

Figure 4.22 Confusion matrix of convolutional neural network (CNN) performance on 1-second spectra of $^{60}$Co and $^{137}$Cs.
4.3.7 Convolutional Neural Network Model Performance on Dataset 7

Figure 4.23 Confusion matrix of convolutional neural network (CNN) performance on 30-second spectra of zero-valance depleted uranium (ZVDU) at different activities.
4.3.8 Convolutional Neural Network Model Performance on Dataset 8

Figure 4.24 Confusion matrix of convolutional neural network (CNN) performance on 30-second spectra of triuranium oxide (U₃O₈) at different activities.

4.3.9 Convolutional Neural Network Performance Summary

CNNs performed with virtually perfect accuracy on datasets 1-3 and 6-8, as can be seen in figures 4.15-20 and 4.22-24. The performance of this algorithm also deteriorated at least moderately.
4.4 Comparison of model performance on zero-valence DU and triuranium octoxide (Datasets 4 and 5)

Figure 4.25 Bar graph of model accuracies for the zero-valence depleted uranium (ZVDU) dataset. Each model was trained on a random subset of the available data to get a general measure of accuracy.
Figure 4.26  Bar graph of model accuracies for the triuranium octoxide (U₃O₈) dataset. Each model was trained on a random subset of the available data to get a general measure of accuracy.
4.5 Autoencoder performance

![Reconstruction Error for Different Classes](image)

**Figure 4.27** Autoencoder reconstruction error per test sample. Class labels were added after analysis.

The autoencoder algorithm outputted consistent reconstruction error for each class. All illicit classes are distinguishable from background which had near-zero reconstruction error.
CHAPTER V
DISCUSSION AND CONCLUSION

5.1 Performance on Different Radioisotopes

The models were expected to be able to discriminate between gamma-ray sources aside from uranium isotopes. In the case of a detonated radiological dispersal device, the environment may be contaminated with radionuclides such as Co-60 and Cs-137 (Bushberg et al., 2007). As depicted by Figures 4.6, 4.14, and 4.22, the models were able to make accurate predictions on all samples in the test set. These radioisotopes have fundamentally different decay schemes and their spectra have distinct photopeaks, unlike the other classes which were varying permutations of uranium and its decay daughters.

5.2 Model Performance on One-Second Spectra

Earlier work in the literature involves data collected at live-times usually longer than one second (Mark Kamuda & Sullivan, 2019; Shoji et al., 2001). One-second live-time data were the focus of this study due to the short resident times of unmanned survey platforms, where it is hoped that these models can be applied. All classifiers were able to achieve 100% accuracy on the tasks of discriminating background, natural, and depleted uranium using only the one-second spectra. These results demonstrate machine learning models can discriminate between depleted and natural uranium. This is likely of interest to those seeking to remediate DU contaminated sites with naturally high background concentrations of uranium. The ability to discriminate between natural and depleted uranium could reduce the cost of efforts substantially.
5.3 Effect of Activity and Composition

Models also performed well on mass differentiation tasks down to the three lowest activity classes. Scintillation detectors measure the activity of radiological samples, which is correlated with mass. Models were trained on the spectra of different masses for a given type of material (DU, U₃O₈, etc.) to assess the ability of the CNN to discriminate between different masses. Low mass spectra (4g) had the highest false-negative rate. Higher activity samples had more clearly defined spectra and were thus more easily distinguished from background, independent of livetime.

However, the closer the spectral classes were in activity the more difficult they were to differentiate between. Classes with very similar activity in the autoencoder models had considerable overlap. This overlap would make this difference in activity fairly inscrutable in a realistic set of survey data where different masses of materials are encountered unpredictably. In addition, many variables are likely to affect the apparent activity of samples in the field. The unpredictable geometries of the gamma-ray sources and detector speed are likely to be the greatest confounders of true activity, diminishing the efficacy of these models as mass discriminators.

5.4 Classifier Comparison

It was of some interest to compare the general performance of each algorithm. All classifiers performed with virtually perfect accuracy when discriminating background from NU, DU, ¹³⁷Cs, and ⁶⁰Co. The models also had high predictive performance when discriminating zero-valent metal DU vs oxidized DU (U₃O₈). Models also performed well between classes of sufficiently disparate activity. For some classifiers, the classes with the lowest activity were occasionally misclassified as background. The most error prone models were those trained on
data of the same radioisotope source. The most frequent, and usually only, source of confusion in these models were the two lowest activity spectra in addition to background. Intuitively, the confused classes also had the most similar activities. The general accuracy of the ZVDU and U₃O₈ classifiers was thus evaluated.

The average accuracy of ten iterations of each classification algorithm was used to generate bar graphs for the zero-valence DU and triuranium octoxide datasets as can be seen in section 4.4. Performance of the logistic regression classifier for both sets of data was higher than expected, as it was believed that logistic regression, being akin to a shallow neural network, would be more likely to overfit the training data. fcNNs on both datasets appeared to underperform relative to logistic regression, while CNNs also underperformed on the U₃O₈ dataset. CNNs performed quite similarly to logistic regression on the zero-valence DU dataset. While not fully demonstrated in this paper, it is believed that CNNs would be the most appropriate algorithm for real-world radionuclide discrimination.

5.5 Autoencoder Performance

Autoencoders were of some interest because they have utility in scenarios when only training data on the background is known. The autoencoder implementation was able to reproduce the background test samples with a RE at or close to zero. RE appeared to be directly proportional to activity and thus mass, suggesting some mass differentiation capability. All known anomaly classes were distant enough from background to be clearly differentiated as “other” but the 75g, 86.2g, and 93.3g ZVDU classes had significant overlap in their RE ranges. This overlap reveals the limit of this implementation as a mass discriminator.
5.6 Conclusion

As a high-volume product of the $^{235}$U enrichment process, DU has been dispersed in various regions by industrial and military entities. DU contamination now presents a radiological and chemical hazard in these areas. Contaminated sites may have hazards in addition to DU such as unexploded ordinance and an intrinsically dangerous environment. Machine learning algorithms were explored to facilitate the classification of large amounts of sparse spectral data, particularly for discriminating the spectra of depleted and natural uranium. This work demonstrates that deep learning methods can discriminate between authentic but sparse spectra. Specifically, these methods can discriminate between background, NU, and DU gamma-ray spectra. These algorithms can also predict source masses with some limitations, which can likely be overcome with additional representative data. Follow-up implementations of these algorithms would incorporate a body of labeled field data to improve generalizability.
References


APPENDIX A

TERMS AND EQUATIONS
A.1 Convolution

The Keras deep learning library actually implements cross-correlation instead of convolution in its convolution layers. The cross-correlation function for one-dimensional values is

\[ F \ast I(x) = \sum_{i=-N}^{N} F(i)I(x+i) \] (A.1)

Where \( x \) is the position on the feature map, \( F \) is the filter, \( I \) is a subset of the features, and \( N \) is the number of features minus one divided by two. The following figure is a depiction of a single step of the correlation operation.

![Convolution operation](image)

Figure A.1 Convolution operation (Jacobs, n.d.) with filter size of three.
A.2 Max pooling

Max pooling is a down-sampling method that reduces the dimensionality of the data. This can be useful for reducing the complexity of the model and reducing translational invariance, if either are needed. The max pooling operation can be abstracted as a sliding window of a certain size in which the maximum value is the output. Figure A.2 is a basic example.

![Max pooling with a window size of two.](image)

Figure A.2 Max pooling with a window size of two.

A.3 Flattening

Flattening is the reduction of a multi-dimensional array to one (flat) dimension. Figure A.3 is a simple example.

![Flattening of two-dimensional array into a one-dimensional array.](image)

Figure A.3 Flattening of two-dimensional array into a one-dimensional array.

A.4 Stochastic Gradient Descent (SGD)

SGD is a learning algorithm that uses a gradient of a function to optimize a set of values for where that function is minimal. SGD is most frequently used in deep learning to optimize the parameters of a neural network with respect to an error function. SGD can also be used for other
types of models and functions other than error functions. The general SGD algorithm is defined below:

\[ W_{t+1} = W_t - \gamma \nabla_W E(z_i, W_t) \]  

(A.2)

where

- \( W_{t+1} \) is the updated set of parameters
- \( W_t \) is the set of parameters used in the previous forward-pass
- \( \gamma \) is a scalar value known as the learning rate
- \( z_i \) is the input vector from the previous forward-step
- \( \nabla_W E(z_i, W_t) \) is the error gradient computed during backpropagation as a function of \( W_t \) and \( z_i \)
APPENDIX B

DECAY SCHEMES
B.1 Uranium-238

(U.S. Geological Survey (USGS), n.d.)
B.2  Co-60

(Nave, 2016)
B.3 Cs-137

55 Cs 137 (30.08 a) → 56 Ba 137m → 56 Ba 137 (stable)

- β⁻: 0.514 MeV, 94.7%
- 1.176 MeV, 5.3%
- 0.892 MeV, 0.0006%

56 Ba 137m → 56 Ba 137

- ce: 2.552 m
- α: 662 keV, 85.1%
- γ: 284 keV, 0.0006%

("Radioactive Decay | Oncology Medical Physics," n.d.)