LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-PROC-825096

Classifier Comparison for Radionuclide Identification from Gamma-ray Spectra

J. R. Romo, K. T. Nelson, M. Monterial, K. E. Nelson, S. E. Labov, A. Hecht

July 29, 2021

INMM & ESDARSA Joint Virtual Annual Meeting
Vienna, Austria
August 23, 2021 through September 1, 2021

Disclaimer

Proceedings of the INMM & ESDARSA Joint Virtual Annual Meeting
August 23-26 & August 30-September 1, 2021

Classifier Comparison for Radionuclide Identification from Gamma-ray Spectra

John-Ryan Romo[1*], Kai Tyrus Nelson[2,3], Mateusz Monterial[2], Karl E. Nelson[2], Simon E. Labov[2], Adam Hecht[1]

[1]*University of New Mexico, Albuquerque, NM, USA.*
[2]*Lawerence Livermore National Laboratory, Livermore, CA, USA.*
[3]*Las Positas College, Livermore, CA, USA.*
** johnromo04@unm.edu*

Abstract

Machine learning is increasingly applied for gamma-ray spectra analysis, particularly gamma-ray identification. Radionuclide identification is a multiclass multilabel classification problem which can be tackled with methods including artificial neural networks, extreme gradient boosted tress (XGBoost) and Random Forest. In this study, we compare the performance of these classifiers using the Benchmark Algorithm for Radionuclide Identification (BARNI) as a framework for processing raw spectra into pre-selected features extracted using a peak search algorithm. Sampled spectra from a 3"x3" NaI detector with a library of 33 radionuclides under a spanning set of shielding configurations was used for training and testing. The overall performance of each classifier was assessed using the F1-score. We also break down the per nuclide performance, and demonstrate which classifiers are better for identification of particular radionuclides. The experimental results show that XGBoost was the optimal choice when performance and training time was valued. The Multi-Layer Perceptron (MLP) neural network showed potential as a runner up although lacked the proper amount of optimization and tuning in order to surpass the XGBoost classifier. The Random Forest classifier was the weakest perform in both performance and training time.

[2] This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-PROC-825096

Introduction

Combating the illicit trafficking of nuclear materials is necessary to prevent terrorist from obtaining the means to produce weapons of mass destruction. However, not all sources of radiation are a threat, which necessitated the capability of distinguishing between benign and threat sources. Radionuclide identification though gamma-ray spectroscopy requires spectral analysis by qualified experts. However, there are more gamma-ray spectrometers deployed than expert available to interpret the measurements, which necessitates algorithms capable of identifying radionuclides for non-expert users.

There are different classification algorithms each with pros and cons which are used for different applications. Some classifiers may be better with certain data sets than others and this is due to the mathematics or structure of the model, the type of data, etc. The classifiers used for the application of radionuclide identification are Random Forest, multi-layer perceptron (MLP) and extreme gradient boosted trees (XGBoost) [1-4]. Each of these classifiers can handle a multi-label, multi-class problems such as radionuclide identification.

Classification comparison for radionuclide identification has covered neural networks such as Convolutional neural networks (CNNs) and artificial neural networks (ANNs) there has not been a direct comparison for these specific classification algorithms using the models in Scikit Learn [5]. There also has not been a comparison that uses a large library of radionuclides with a various number of shielding configurations. In this study we will be comparing a neural network and two types of tree models. The main focus is to see which classifier can correctly identify nuclides present in the sample.

For this study we used the Benchmark Algorithm for Radio-Nuclide Identification (BARNI) which is an automatic identification tool for working with gamma-ray spectra [6]. BARNI is a tool that is designed to be used with various Radioisotope Identification Devices (RIID), and provide radionuclide identification comparable with vendor provided algorithms.

Methods

Sample and Feature Generation

A generic 3”x3” NaI detector model in GADRAS was used to generate the template library of spectra for each radionuclide of the 33 radionuclides under a spanning set of shielding configurations. These templates were then used to draw sample spectra, with an included Cs-137 intrinsic source and background. Intrinsic sources are often used to calibrate the detector by having a known peak always present in the spectrum, and this was assumed to be a constant 96 counts per second. The background spectrum consists of a mixture of a world average background, and a mixture of individual components such as primordial radionuclides and cosmic radiation. The total counts sampled were exponentially distributed between 1,000 and 100,000 counts, and the source rate was sampled uniformly between 50 and 200 counts per second, which implied a measurement time and provided the number of intrinsic source counts to sample. The background was then added in providing

anywhere between 20 and 80% of total sample counts. Nuisance parameters, such as gain and resolution variation, were incorporated into the sampling process in order to account for real-world changes in calibration and resolution.

The sampled spectra were then used to generate features using the BARNI algorithm. The BARNI algorithm works by using peak finder which defines features to be used for training the classifier model. This sample and feature generation process was repeated twice to generate a set of independent training and testing data sets. The training data sets were used to build each classifier model, and the testing set was used to determine the performance of each classifier. The dimensions for each of our training sets consisted of 66,000 features where we decimated our data down to 13,200 or approximately 20% of the original. The purpose of this reduction is for the sake of training times

Optimizing Hyper Parameters

Each classifier has several input parameters for specifying the construction of each model, these are called the hyperparameters. In order to have a fair adjudication of the performance of each classifier, it is necessary that the optimal parameters are chosen for every classifier. It is not enough that every classifier is fed the same set of training and testing data.

In order to find the best parameters, we conducted a data science competition among four of summer interns at Lawrence Livermore National Laboratory in 2020. Each student was provided with an identical training and testing data set, and then tasked with finding the hyperparameters that gave the higher F1-score, the performance metric used to judge radionuclide identification performance. A separate validation data set, which was hidden from the participants, was then used to verify the performance of each classifier.

Scoring Metrics

In order to quantify the performance of the classifiers the F1-score was used which is the harmonic mean of precision and recall. Each classification model needed to perform well with this metric to ensure that it was identify the correct nuclide in the gamma spectra. The F1-score is very useful when it comes to multi-class multi-label classification. Another metric used to visualize the prediction fraction was a confusion matrix. This metric was used to show where each individual nuclide was getting confused with another nuclides or predicting nothing. The prediction fraction metric is the fraction of correctly identified radionuclides over the total number of samples present in each data set. The x-axis displays the actual nuclides while the y-axis shows the predicted nuclides. The diagonal of the confusion matrix contains the prediction fraction and is the fraction of correctly predicted nuclides. While the off diagonals are where the nuclides got confused with a another nuclide. The none category is where there was a false negative or where none of the nuclides were detected.

Classifiers

We used the "one vs all" strategy of classifier model building in order to deal with the multi-label and multi-class problem of radionuclide identification. The strategy behind the one-vs-all is to fit one classifier per class so that way a single classifier can be able to predict a single nuclide. This means that for every single nuclide in our library a single classifier was trained to identify a specific nuclide (multi-class). A nuclide is identified when its classifier returns a value above a pre-determined threshold, this allows for multiple nuclides to be identified per sample (multi-label).

Random Forest

The Random Forest classifier is one of the most popular machine learning algorithms and is the default classifier used in BARNI. The Random Forest classifier consists of many decision trees and uses ensemble methods which can be used for classification or regression problems. [4] Each tree has a structure that contains a root node, internal nodes, and leaf nodes where at each split at the nodes represents a layer of depth. Each decision tree represents an estimator in the Random Forest and is one of the parameters of the classifier. The random aspect in Random Forest is a vector that each tree depends on that is sampled individually but is still part of the same distribution of the trees in the forest. Once each tree is grown and gives a classification the final Random Forest will predict the majority of the voted class. The hyperparameters used for our application were a max depth of 20 and a number of estimators of 100.

The nonlinear nature of the Random Forest can give it an advantage over other linear classifiers. However, Random Forests are highly prone to overtraining which can lead to inaccurate predictions therefore the hyper parameters are important.

Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) neural network is a traditional artificial neural network (ANN) which uses feed forward methods. The MLP model is composed of multiple layers an input layer, output layer, and the hidden layers which contain the perceptrons/neurons. The MLP neural network works by using gradient descent techniques which fits the data to the model. Through these nonlinear techniques the model is trained to then classify our data. The hyperparameters that we chose to implement into the model were the ADAM solver, the ReLU activation function, and as for the hidden layers we chose to use 8 layers with 135 perceptrons/neurons per layer.

Extreme Gradient Boosted Trees (XGBoost)

Extreme Gradient Boosting (XGBoost) is a type of gradient descent ensemble method similar to Random Forest and decision trees. While Random Forest grows each tree and combines the results in parallel during the voting segment XGBoost uses a method called boosting. This boosting method combines many single split trees also known as weak learners in subsequent order to correct the performance of the previous tree based on the residual error. XGBoost then takes the predictions from the previous tree and uses them to update the new tree for improved performance. The

hyperparameters found in the study were a max depth of 16 and a learning rate of 0.02. The subsample ratio of training instances was set to 75% and a version of a regression tree was used as the weak learner ("gbtree").

Results

Radionuclide Identification Performance Comparison

Table 1 contains the best and worst performing nuclides for each classifier along with the overall F1-scores for each classifier. The majority of all the radionuclides performed very well however, there were some outliers that stood out only for specific classifiers. For example, Pu-239 performed very poorly for all three classifiers although it performed the worst for the Random Forest with a 0.377 F1-score. For both of tree models, XGBoost and Random Forest Pu-239 was the worst performer out of all the nuclides which may an issue with how the trees were generated for Pu-239. Both of the models predicted mostly in the "none" category for Pu-239.

Table 1: Classifier comparison using F1-Scores.

Random Forest				**MLP**				**XGBoost**			
Best		Worst		Best		Worst		Best		Worst	
Nuclide	F1-score	Nuclide	F1-score	Nuclide	F1-Score	Nuclide	F1-Score	Nuclide	F1-Score	Nuclide	F1-Score
^{111}In	0.985	^{99}Mo	0.778	^{111}In	0.995	^{166}Ho	0.813	^{40}K	0.987	^{235}U	0.870
^{54}Mn	0.980	^{57}Co	0.777	^{54}Mn	0.990	^{92}Ir	0.801	^{111}In	0.986	^{133}Xe	0.865
^{233}U	0.977	^{235}U	0.762	^{233}U	0.987	^{232}Th	0.772	^{233}U	0.984	^{99}Mo	0.863
^{40}K	0.974	^{226}Ra	0.761	^{40}K	0.984	^{226}Ra	0.741	^{54}Mn	0.980	^{166}Ho	0.859
^{123}I	0.971	^{238}U	0.727	^{51}Cr	0.979	^{239}Pu	0.738	^{207}Bi	0.970	^{241}Am	0.853
^{232}U	0.961	^{192}Ir	0.624	^{123}I	0.975	^{238}U	0.739	^{60}Co	0.969	^{137}Cs	0.803
^{51}Cr	0.952	^{137}Cs	0.609	^{201}Tl	0.969	^{137}Cs	0.707	^{123}I	0.968	^{192}Ir	0.795
^{22}Na	0.950	^{239}Pu	0.337	^{109}Cd	0.968	^{99}Mo	0.649	^{232}U	0.967	^{239}Pu	0.694
Overall F1-Score				**Overall F1-Score**				**Overall F1-Score**			
0.859				**0.888**				**0.910**			

Cs-137 is our intrinsic source in the detector and this is part of the reason it is performing poorly for all of our classifiers. With its strong peak at 662 keV it could be the reason that nuclides with peaks around 662 keV are performing poorly. As for nuclides such as uranium, thorium, and radium these elements are part of the background spectra, and their radionuclides are more likely to be confused when classified. However, K-40 is a strong component of the background spectrum and is performing very well and this is particularly due to its strong high energy peak at 1460 keV.

The overall F1-scores are a direct result of the performance of the classifier. XGBoost comes in first place then MLP and then Random Forest. XGBoost is a popular method with data scientists, where out of 29 machine learning challenges XGBoost was used 17 times while the second most popular method was deep neural networks [2]. This is because XGBoost right out of the box does a

great job dealing with large sets of data and the parameters do not need to be optimized quite as precise. While the MLP classifier has potential to outperform XGBoost the architecture of the model needs be much more optimized for the data.

Confusion matrices are shown in Figures 1-3, and provide a visualization of how each nuclide is performing against the other nuclides. The lighter the shade of block is a higher performance of that specific nuclide. As displayed in the figures XGBoost has the lightest diagonal out of the three figures. You can see where each classifier managed to struggle with each individual nuclide. Random Forest categorized its incorrect predications as none instead of other nuclides where as MLP and XGBoost did the same.

A note for Figure 2 the MLP confusion matrix there is confusion between Mo-99 and Tc-99m and this is due to the parent daughter relationship that these two nuclides share. The classifier is seeing the parent and then is seeing the decay into the daughter and getting confused when counting the nuclide. This is also true for the other classifiers as well there is an issue with the labels.

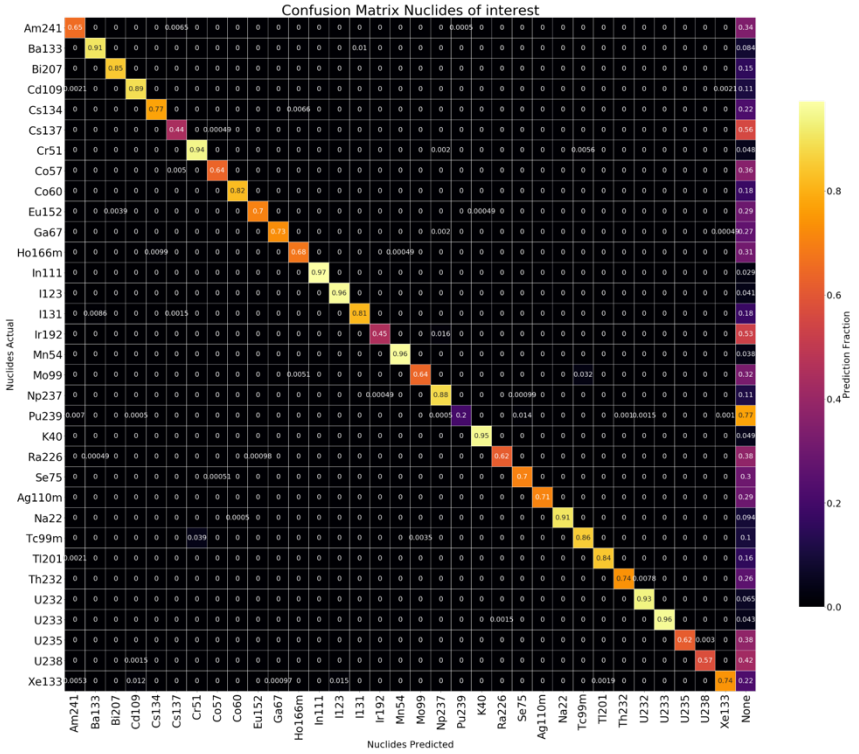

Figure 1**. Random Forest Confusion Matrix**

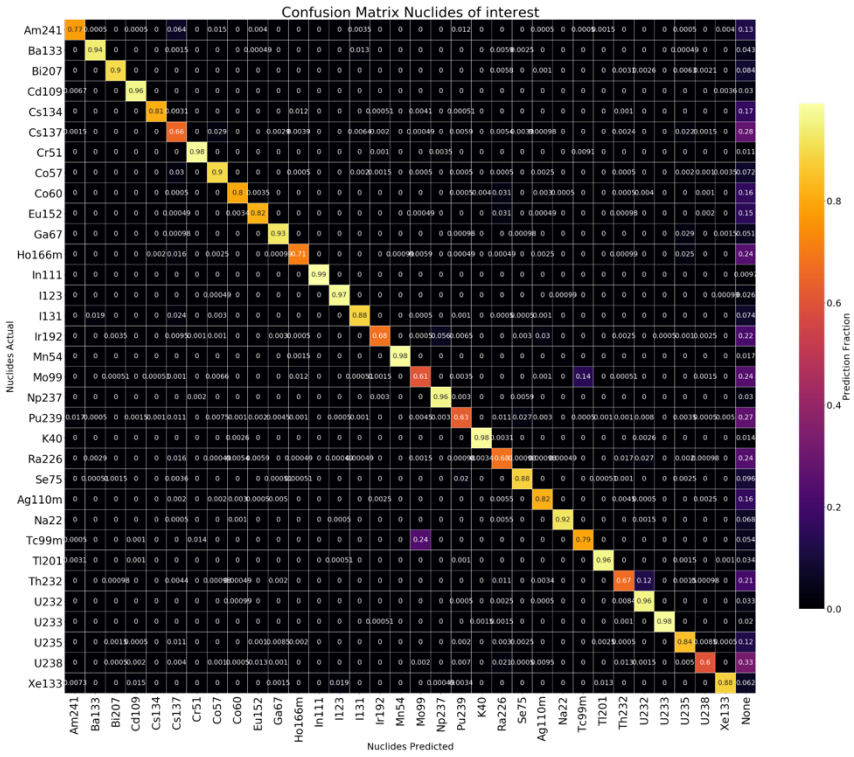

Figure 2. MLP Confusion Matrix

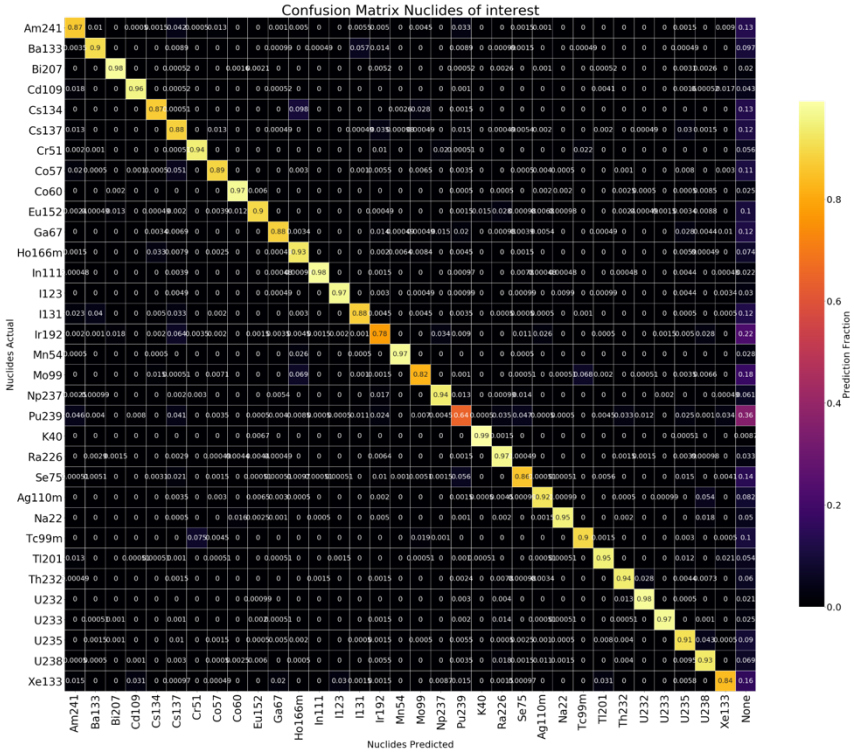

Figure 3. XGBoost Confusion Matrix

Training Times

As for the training times of the classifiers XGBoost once again comes in first place with MLP second and Random Forest third. The training times are a good representation of how the data is trained and what type of operations are being performed. XGBoost trains approximately 4 times faster than MLP and approximately 7 times faster than Random Forest.

Table 2 Training Times for Each Classifier

Classifier	Training Time (seconds)
Random Forest	42.28
MLP	24.63
XGBoost	5.97

Conclusion

In this study, we have compared three different machine learning classifiers on a library of 33 radionuclides with a number of shielding configurations. The data was also decimated down from 66,000 to 13,200 features or to 20% of the original set for the sake of training times. The classifiers were then trained on the data and timed. The study showed promising results for boosted tree models and there was potential for the neural network to achieve promising results as well.

Overall XGBoost performed the best in the classification portion using the F1-score and to train the classifier. XGBoost right out the box is optimized for speed and performance, so the results were predictable. The MLP classifier has potential to outperform XGBoost however, it needs to be much more optimized to fit the data set in order to achieve better performance. MLP also has a better performance predicting Pu-239 which is a crucial nuclide that should be identified. The training time for MLP takes some time to train on the architecture of the hidden layers so a more optimized neural network with a larger data set may be able to get faster results than XGBoost. As for the Random Forest the hyperparameters were not enough to achieve a higher performance than the two other classifiers nor achieve faster training times. Future work consists of working with more classifiers and designing architecture for neural networks. There is also a need to test the classifiers against real world data, which will include more variability than was included in the differences between testing and training data sets.

References

[1] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–94. *arXiv.org*, doi:10.1145/2939672.2939785.

[2] Kingma, Diederik P., and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Dec. 2014. *arxiv.org*, https://arxiv.org/abs/1412.6980v9.

[3] Glorot, Xavier, and Yoshua Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks." *AISTATS*, 2010.

[4] Raedt, Luc de, and Peter Flach, editors. Machine Learning: ECML 2001: 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001: Proceedings. Springer-Verlag, 2001.

[5] S.M. Galib, P.K. Bhowmid, A.V. Avachat, and H.K. Lee. "A Comparative Study of Machine Learning Methods for Automated Identification of Radioisotopes Using NaI Gamma-Ray Spectra." *Nuclear Engineering and Technology*, June 2021. *www.sciencedirect.com*, doi:10.1016/j.net.2021.06.020.

[6] M. Monterial, K.E. Nelson, S.E. Labov, S. Sangiorgio Benchmarking Algorithm for Radio Nuclide Identification (BARNI) Literature Review (No. LLNL-SR-767242) Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States) (2019)

[7] Mitchell, Dean, and & Mattingly, John. *Gamma Detector Response and Analysis Software (GADRAS) v. 16.0*. Computer software. Vers. 01. USDOE. 24 Dec. 2009. Web.