The Dissertation Committee for Adam William Drescher Certifies that this is the approved version of the following Dissertation:

Leveraging Machine Learning Capabilities for the Characterization of Irradiated Uranium: A Case Study of Prediction Methods for Nuclear Safeguards and Nuclear Forensics

Committee:

_____

Sheldon Landsberger, Supervisor

_____

Ken Dayman, Supervisor

_____

William Charlton

_____

George Biros

_____

Derek Haas

Leveraging Machine Learning Capabilities for the Characterization of Irradiated Uranium: A Case Study of Prediction Methods for Nuclear Safeguards and Nuclear Forensics

by

Adam William Drescher

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2019

Dedication

I dedicate this dissertation to my parents. Without them this would have literally been impossible.

Acknowledgements

I would like to acknowledge the substantial support provided by my advisor Sheldon Landsberger during the development of this dissertation. Throughout this dissertation Sheldon has always been my number one supporter. He continues to go above-and-beyond to help his graduate students in whatever capacity they require.

I would also like to acknowledge my co-supervisor Ken Dayman for providing me with inspiring ideas, constructive criticisms, and the invaluable opportunity to have nuanced discussions of every aspect of this dissertation.

Other people that I would like to acknowledge in no particular order include: Justin Knowles for help with building ORIGEN input files, Larry and the rest of the staff at the UT Austin Nuclear Engineering Teaching Lab for facilitating irradiations on a very particular schedule, Vince Jodoin and the Nuclear Security Modeling Group at Oak Ridge National Lab for providing me a platform to discuss this work and receive valuable feedback, and my dissertation committee for having just enough flexibility for defense scheduling to be possible.

Leveraging Machine Learning Capabilities for the Characterization of Irradiated Uranium: A Case Study of Prediction Methods for Nuclear Safeguards and Nuclear Forensics

Adam William Drescher, PhD

The University of Texas at Austin, 2019

Supervisors: Sheldon Landsberger, Ken Dayman

The characterization of irradiated actinide materials is a complex multi-variate problem that is relevant in both nuclear forensics and nuclear safeguards. Compounding factors such as irradiation history, initial actinide composition, enrichment, decay time, neutron flux, and irradiation energy spectrum, many of which may be either poorly constrained or entirely unknown, make source attribution challenging. This work demonstrates the capabilities of machine learning to advance the state-of-the-art measurements that are essential to nuclear safeguards and forensics. Predictive models were constructed with regularized linear methods, decision trees, and ensembles of decision trees for simultaneous irradiation time (t_{irr}) and decay time (t_{dec}) independent determination of uranium enrichment via gamma emissions with synthetically generated data. The t_{dec} independent predictive enrichment monitor was then validated with real-world experimentally measured data while holding t_{irr} fixed. All models were trained and evaluated on datasets consisting of exactly seven discrete enrichment values (0.02%, 0.71%, 3%, 20%, 50%, 63%, and 97% ^{235}U).

Models were trained to predict the enrichment of randomly selected gamma-ray emission profiles with varying t_{dec} values but with t_{irr} fixed at 1 hour. Additional t_{irr} values

were introduced into the dataset and new models were then trained. Mismatches between the training and testing dataset t_{irr} values were introduced to characterize the generalization performance of models to data which was unrepresented during training. It is shown that for the ranges of t_{irr} considered, there is no statistically significant degradation in model performance, indicating robust generalization performance. The adaptively boosted decision tree ensemble constructed on these simulated data with fixed t_{irr} was able to perform predictions on uranium enrichment with a mean absolute error of 1.7% of the ^{235}U enrichment values for fixed t_{irr}.

Experimental measurements were performed to validate the fixed t_{irr} modeling capability. An adaptively boosted decision tree ensemble constructed on the experimental data achieved a mean absolute error of only 0.3% of the ^{235}U enrichment values, outperforming the models trained on synthetic data.

Next, due to performance limitations in computing power principal component analysis (PCA) was used for dataset feature truncation. It is shown that PCA results in a substantial reduction in computation time. Furthermore, PCA also improves the performance of models by removing unimportant variations in the data. A decision tree evaluated on PCA truncated data achieved a mean absolute error of only 0.05% of the ^{235}U enrichment values.

The prediction capabilities provided by these models can be naturally extended to application-focused measurements in the fields of nuclear safeguards, nuclear forensics, and nuclear nonproliferation. The scenario presented here represents the first step towards building these application-focused systems.

Table of Contents

List of Tables

List of Figures

Chapter 1: General introduction & high-level project summary

General Introduction

The physical sciences are characterized by an ongoing effort to discover the rules which govern the processes, events, and interactions that occur around us. This is true in every field: forces are governed by Newton's three laws of motion, the motion of astronomical bodies is described by Kepler's laws, the rules of classical electromagnetism are succinctly captured in Maxwell's four equations, quantum mechanics by Schrodinger, and much more. These laws of physics enable scientists to, given the initial condition of a system, predict the state of that system after some set of events have occurred. Indeed, given science's current understanding of many of the laws of physics that shape our reality, and given adequate measurements of all relevant parameters in a system, one could, in principle, predict the state of that system at any point in the future. This is the *forward problem*. The ability to predict the future is indeed powerful. However, in many applications, it can be the case that the quantity of interest is the state of a system in the past. That is to say: what happened to this system in the past to cause it to be in this measured state? This is the *inverse problem*.

Furthermore, as the technology which enables these measurements to be performed improves in concert with our theoretical understanding of the underlying laws, the amount of measured data that it is possible to have of any system increases dramatically with time. With the improvements in measurement technology, science has entered the era of big data. It now becomes possible to approach these problems from a new perspective: given the substantial experimental data that has been collected, is it possible to reconstruct the previous condition of this system? Can the inverse problem be solved by leveraging large amounts of experimental data?

But, this approach to inverse problems is now entrenched in real-world experimental data. We are no longer exclusively using the laws of physics to directly predict all parameters of a system. A prediction method using a model based on experimental data is constrained by those aspects of a system which can be directly

measured. This has the important ramification that there will always be unknown aspects of a system. For instance, imagine the scenario of determining the age of a person. A person's age is not a directly measurable quantity. However, you could identify relationships between measurable physiological factors such as sex, height, weight, etc. that correspond with age. This is a statistical method. On the other hand, fundamental biological relationships exist between the length of telomeres in the body and the aging of cells[1]. In principle, if the state of telomeres throughout the body was known, it could theoretically be possible to determine age based off of the observation of telomeres in the body. However, there not currently a reliable way to measure telomere lengths. This is a system parameter that cannot be known. Indeed, every single parameter of a measured system can never be known. The challenge in statistical predictions is thus to take a large number of weakly correlated features (such as sex, height, weight, etc.) and find a set of rules which allows for a strong prediction of the target value (age) because the strongly correlating feature (telomere length) cannot be readily known. All relevant parameters can be broken into three groups: those parameters which are directly measurable, those which are unknown, and of those unknown parameters those which are of interest to learn.

And so, in the formulation of an inverse problem, it is essential to determine what experimental data is available, which unknown is of interest to learn, and how many additional unknowns, and across what ranges, are tolerable. Of course, if absolutely *nothing* is known besides the experimental measurements, it is likely that the problem is *ill-posed*. Ideally, there will be only a few unknown quantities, and ideally those quantities can be constrained within some reasonable ranges of values. A compelling motivation behind the solution of inverse problems is that it is possible to capture the trends in experimental data that correlate with the target value which is to be predicted while remaining invariant to the other unknown values. This allows for the computation of a quantity which, in the laws-of-physics forward sense, would be intractable due to the number of unknowns.

The advantages of leveraging machine learning prediction capabilities for the solution of the inverse problem are as follows: 1) the problem is well-suited to real-world

applications by relying on experimental data, 2) the number of scenarios in which accurate predictions via machine learning are feasible increase dramatically by allowing some amount of unknown values and ranges, and 3) the solution will uncover relationships in the data which are highly correlated with the target variable of interest while remaining invariant to the other unknown values. For the application presented in this work, these results can be easily validated by identifying the fission or activation products that are responsible for each particular gamma-ray emission and comparing the yields of each product from the isotopes of uranium considered (^{235}U, ^{234}U, ^{236}U, and ^{238}U). Therefore, the results are readily interpreted back to physical properties of the system.

This work presents a sample problem which is well-suited to solution via inverse methods as a testbed for the feasibility of machine learning for applications in the realm of nuclear engineering. The questions that this dissertation aims to answer are as follows:

1. Is it possible to determine uranium enrichment via simulated gamma-ray emissions with independence to decay time (t_{dec}), assuming all other properties are equal (neutron flux energy spectrum, sample mass, geometry, etc.)?
2. Can models trained to perform these predictions be generalized to data which is outside of the domain on which they were trained?
3. To what extent is prediction performance deteriorated when increasing the dimensionality of the unconstrained space? That is, how much worse does a model perform if both the decay time (t_{dec}) and the irradiation time (t_{irr}) are unknown as opposed to only t_{dec}?
4. Extending to the additional complications of real-world measurements, such as: detector response functions, uncertainties in precise irradiation conditions, compounded with additional uncertainties such as mass, geometry, etc. is 1) still possible (with fixed t_{irr})?
5. Are data truncation methods effective for reducing computation time without degrading model performance?

6. What steps would be necessary to extend these capabilities to useful real-world measurements, for instance: re-verification of dry cask spent nuclear fuel, characterization of nuclear weapons from fallout debris, etc.?

The purpose of this work is to answer these questions while developing a procedure for constructing models which can accept an experimental measurement as input and return a prediction on the quantity of interest even when other important parameters are unknown, and to demonstrate key aspects of model performance under varying circumstances. For the purpose of demonstrating and validating this capability, a contrived scenario is introduced and explored. To properly address each of the questions listed above, it is necessary to create a "problem in a box" in which the domain of considered variables is restricted. Finally, examples of common problems in nuclear safeguards and nuclear forensics that are well-suited for machine learning solutions are discussed, as well as the unique considerations and additional layers of sophistication associated with each application.

HIGH-LEVEL PROJECT SUMMARY

This next section will provide a brief high-level overview of each major component in the development of a predictive model for uranium enrichment based on gamma-ray spectra. First, I use synthetic data generated in ORIGEN to prove the feasibility of decay time invariant monitors of uranium enrichment and test model performance in several more difficult scenarios. Next, I irradiate uranium standard reference materials, perform gamma-ray spectrometry, and use the resultant gamma-ray spectra for developing models with real-world experimental data. I then experiment with two methods of truncating the experimental data to speed up computation time. The flowchart shows the major elements of this project. The major steps are:

1. Generating a sufficiently broad database of gamma-ray measurements which captures the variations in enrichments, irradiation times, and decay times that we wish to consider.
2. Preparing and formatting the gamma-ray database for use in machine learning.
3. Selecting machine learning algorithms to use for model construction.

4. Optimizing the hyperparameters of the models during training, coupled with cross-validation and grid searches.
5. Extracting fundamental insights to the relationships in the gamma-ray data which most strongly correlate with uranium enrichment.
6. Evaluating the ultimate models generated by each algorithm and characterization of their performances using scoring metrics.
7. Comparing the performance of models in scenarios of single and double unconstrained values: (t_{dec} vs t_{dec} and t_{irr}).

Figure 1 shows representations of each of these steps and how they fit together into a cohesive process for model development and evaluation.

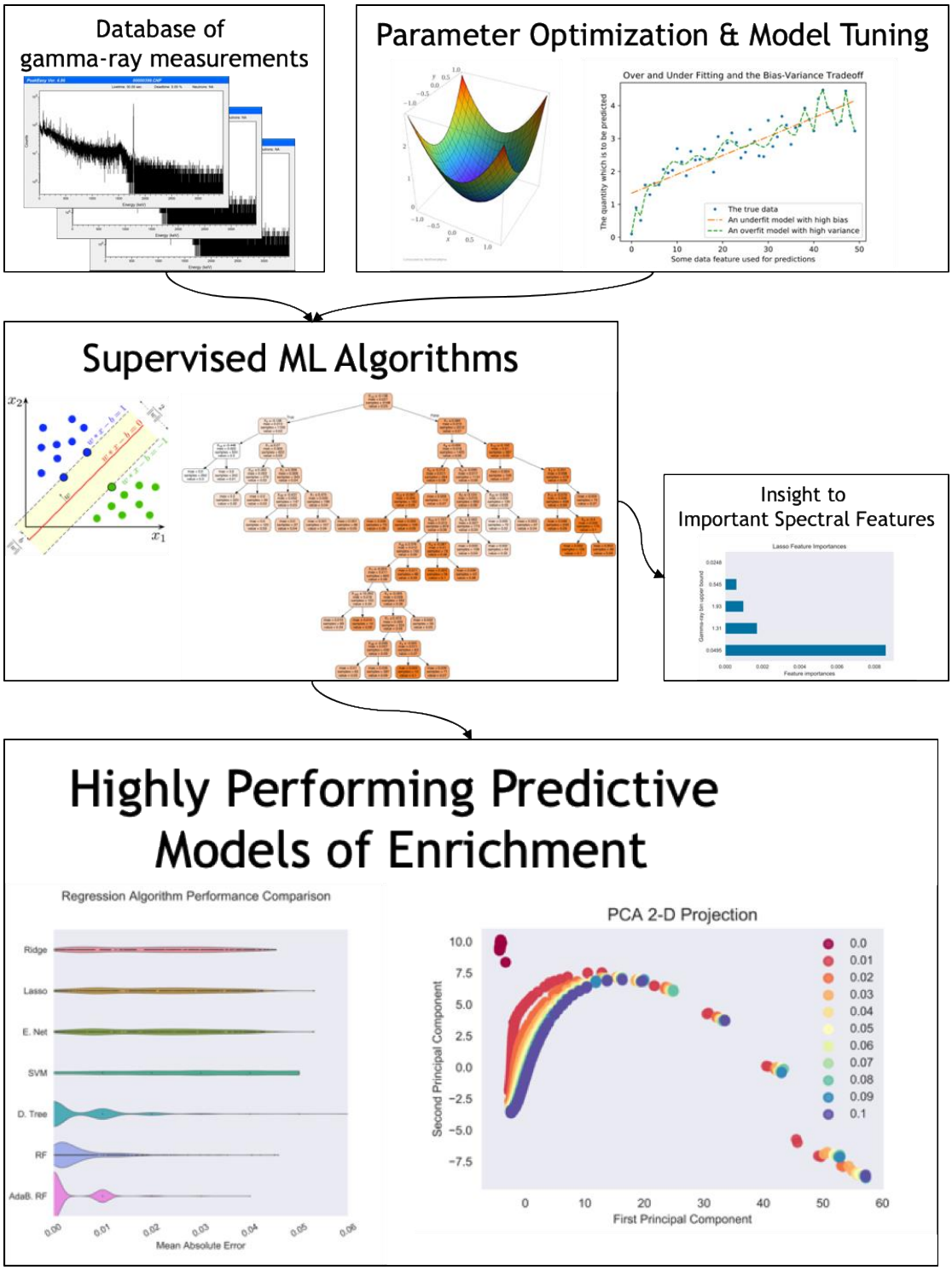


Figure 1: The flowchart which visualizes each major component of the development of a predictive model for uranium enrichment.

Generating a sufficient database of gamma-ray measurements

The first step in developing a predictive model is the collection of a sufficiently representative dataset of gamma-ray spectrometry measurements labeled according to the uranium enrichment levels to which each individual spectrum corresponds. The phrase "sufficiently representative" means that the dataset captures all possible variations in the parameters of the data. Here, this includes variations in the parameters of enrichment, decay time (t_{dec}), and irradiation time (t_{irr}). I generated datasets via simulations and via experimental measurements. I reserve the details of how these data were obtained for the Methods section.

Preparing and formatting of the gamma-ray database for use in machine learning

Once a representative collection of gamma-ray measurements is acquired (either synthetically or experimentally), I formatted these data.

For the synthetic data, I generated gamma-ray emissions at 180 decay times for each irradiated uranium isotope and blended these results into intermediate enrichments. I then labelled the data instances according to enrichment, split the data into training cross-validating and testing sets, normalized the data by feature, and introduced gaussian noise.

For the experimental data, I irradiated each separate enrichment of uranium separately in UT's reactor. Therefore, no synthetic blending of results was necessary to obtain intermediate enrichment values. With individual count times as low as three minutes, there will already be a high degree of uncertainty and statistical variations in each data instance. I also labelled these data according to enrichment, split them into training cross-validating and testing sets, and normalized each observation by feature. For the experimental data, I also had to truncate the data to reduce its dimensionality to allow for reasonable computation times. I utilized and compared two methods for accomplishing this by: 1) decreasing the time and energy resolution of the full dataset by combining adjacent observations and gamma-ray bins, and 2) high-pass principal component analysis, in which only the top few components which maximally explain the variance of the dataset are preserved. Once all of these steps have been completed, the data are stored as `NumPy` arrays suitable for machine learning via the python package scikit-learn[2].

Selecting machine learning algorithms to use for model construction

The next step is to choose an initial selection of machine learning algorithms that have properties that make them appropriate for solving this type of problem. Several different categories of machine learning algorithms were considered for this problem.

The first category of algorithms considered are three types of regularized linear models. Regularized linear models have a simple architecture that makes them easy to interpret and explain. The different regularization techniques that each algorithm employs allow them to achieve different types of solutions. They will each be described in more detail in the Theory section.

The next class of algorithms considered are decision trees. A decision tree is a collection of rules which evaluate values of various features in the data of a given data instance to place that instance into a category which has an associated prediction value. The main issue with decision trees is that they can become very dependent on highly specific features of the data. This can cause decision tree-based models to have high prediction errors when presented with data that is slightly different than the data on which they were trained. This behavior is referred to as overfitting.

A solution to this problem is to place restrictions on how much an individual decision tree is allowed to grow, and instead of growing a single highly complex decision tree, grow a large number of simple restricted trees. These are the third class of algorithms: ensemble methods based on decision trees. The motivation behind ensemble methods is the notion that "a large number of weak learners forms one strong learner". It has been shown[3] that an ensemble predictor can in many situations perform more strongly than a single strong learner due to favorable exchanges of bias for variance.

Optimizing the hyperparameters of the model during training, coupled with cross-validation and grid searches

Machine learning algorithms have two types of parameters: those which are learned via training, and those which are chosen by the user prior to training (called hyperparameters). The parameters that are learned via training are the values which minimize the objective function of the algorithm. The hyperparameters and help govern the form of the model to

be estimated. For instance, in regularized linear models, the hyperparameter is the regularization parameter, which determines how strongly regularized those models will be. Or in the case of random forests, the number of trees to construct. Decision trees also have many hyperparameters which also place restrictions on their growth, including maximum number of features to consider when forming a node, maximum depth of branches of the tree, minimum number of instances to constitute a leaf, etc. These parameters will be described in greater detail later.

I selected the optimal hyperparameters via a grid search over a range of potential values for each hyperparameter of interest. A grid search is a methodical procedure for evaluating a collection of models, each of which is built with a unique set of hyperparameter values. Every combination of hyperparameter values is used to train and evaluate models through the process of cross-validation. The optimum set of hyperparameters is the set which yields the highest score on the cross-validation tests. Cross validation is a methodical procedure for effectively utilizing the training set alone to tune the hyperparameters of a model. When evaluating a model, it is critical to score the model's prediction capabilities on data with which it was not trained. Cross-validation splits the training set into k groups. The model is trained on k-1 groups within the training set, and then its predictions on the kth group (holdout set) are scored. This process is repeated k times, switching the holdout set each time. The end result is five performance metrics which demonstrate the generalization performance of the models generated by an algorithm with that set of hyperparameters. This is the procedure that is iterated for every combination of hyperparameter values, and the set of values associated with the best results from the cross-validation testing are selected and used for model development. Figure 2 visualizes the splitting of the training set into five subsets and performing the five-fold cross-validation procedure.

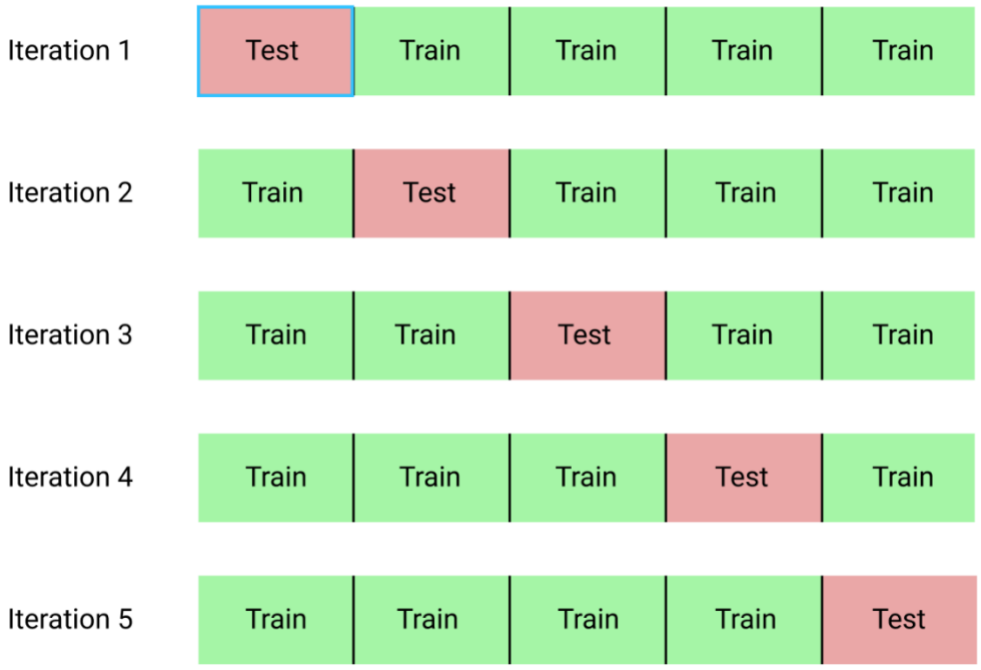


Figure 2: A visualization of the cross-validation procedure with k=5.

Extracting fundamental insights into the gamma-ray data which most strongly correlate with uranium enrichment

With predictive models of uranium enrichment constructed, we can now study the characteristics of the model to discover trends in the gamma-ray data which strongly correlate with enrichment while remaining invariant to decay time. The ways in which each model is deconstructed in order to "look under the hood" and understand which features (gamma-ray energy bins) are informing on the predictions will necessarily vary between algorithms. Generally, for the regularized linear models, the absolute magnitude of the coefficient weights will serve as a gauge for the importance of each particular gamma-ray energy. For decision tree-based models, feature importance is determined according to the features which most strongly decrease the impurity of the data. That is to say: which features when present at a node on a decision tree have a threshold value which most cleanly splits the dataset into groups according to enrichment? The features that are selected by each model can then be related back to the corresponding gamma-ray energies,

and potentially identification of the source radioactive fission products which are responsible for each emission. It will be particularly promising if models created by very different algorithms have common features that they each select as this will indicate a high predictive power for those features. Furthermore, I compared the gamma-ray energies selected by the models and those conventionally used in the literature for uranium assay.

Evaluating the ultimate models generated by each algorithm and characterization of their performances using scoring metrics

Finally, I used scoring metrics to evaluate the performance of each generated model on uranium enrichment predictions with the testing data, which is unique data that the models have not seen during training. The scoring metric that has been utilized here is "mean absolute error". The equation is given below:

$$MAE = \frac{\sum_{i=1}^{n} |y_{pred}^{i} - y^{i}|}{n} \quad \text{(eq. 1)}$$

Comparing the performance of models in scenarios of single and double unconstrained values (t_{dec} vs t_{irr} and t_{dec}).

This dissertation in total considers model construction in three different scenarios: 1) models built and evaluated on synthetic data with a fixed t_{irr} and varying t_{dec}, 2) models built and evaluated on synthetic data with both parameters varying along with mismatches between the training and testing sets, and 3) models built and evaluated on real-world experimental data with a fixed t_{irr} and varying t_{dec}. How do models based on synthetic data perform compared to models based on real-world experimental data? What is the amount of performance degradation that occurs when a second parameter is allowed to be unconstrained?

This concludes the high-level summary of the major components of this dissertation. The next chapter will present a motivating scenario for the solution that is developed in this dissertation: the determination of the enrichment of a given irradiated uranium sample via gamma-ray spectrometry, and the associated errors that are introduced as the constraint on its decay time is loosened.

Chapter 2: Motivating Scenario

This section motivates the development of the decay-time invariant enrichment monitor presented in this work. Here, I replicate a standard method for determining irradiated uranium enrichment developed by Mandel et al. based on measurements of activation and fission product gamma-rays. Next, I demonstrate the challenges associated with measuring the enrichment level of irradiated uranium using this method if the decay time of the material is unknown. For this measurement, I use the gamma-ray peak net areas associated with ^{239}Np and select ^{235}U fission products to determine net area ratios which are linearly correlated with the ^{238}U/^{235}U ratio. This method developed by Mandel et al.[4] relies on several assumptions: 1) irradiation in a well-thermalized neutron flux, 2) uranium enrichment values of 20% or less (at higher enrichment levels, the ^{239}Np peaks become almost completely obscured by much stronger fission product activities), and 3) a 24-hour decay time. This section will study the degradation in performance of this measurement if the decay time is no longer fixed at 24 hours, but rather, ranges from 8 hours to 58 hours post-irradiation.

The method for isotopic analysis of uranium enrichment is based on measuring three photopeaks from ^{239}Np (an activation product of ^{238}U through the ^{238}U(n, γ)^{239}U$\rightarrow_{\beta}$ ^{239}Np reaction), and photopeaks from each of the fission products ^{99}Mo, ^{133}I, ^{91}Y, ^{97}Nb, and ^{132}I. These fission products are produced by ^{235}U fissions and thus act as a signature of the ^{235}U content. With three photons representative of the ^{238}U content and six photons representative of the ^{235}U content, there are a total of 18 replicate peak area ratios that can be measured to determine the uranium isotope ratio. That is, the ratio of observed photons from ^{239}Np and each of the fission products (^{99}Mo, ^{133}I, ^{91}Y, ^{97}Nb [two photons], and ^{132}I) should be proportional to the ^{238}U/^{235}U fraction of the uranium sample post-irradiation. Mandel et al. developed calibration curves with experimental data for each of these eighteen ratios as a function of the ^{238}U/^{235}U fraction. Table 1 shows the specific photon energies and parent nuclides that are considered. We define each of the 18 relevant isotopic ratios as: $r_i = a_{En}/a_{Ef}$, where a_{En} is the

net area of one of the three photons from ^{239}Np and a_{Ef} is the net area of one of the six photons from fission products, and i ranges from 1 to 18 so as to index each of the photopeak ratios. I will re-create each of these eighteen calibration curves for two scenarios: data collected at precisely 24 hours of decay, and data across the full range of decay times that I have measured (8 hours to 58 hours post-irradiation).

Table 1: The photopeaks and nuclides used for photopeak ratios in Mandel et al.

Nuclide	**Energy [keV]**
^{239}Np	105
^{239}Np	210
^{239}Np	278
^{99}Mo	140
^{133}I	530
^{91}Y	556
^{97}Nb	658
^{97}Nb	743
^{132}I	773

The experimental dataset I collected is used for this demonstration. This dataset consists of 7,000 gamma-ray spectra representing 7 enrichment levels post-irradiation. Starting 8 hours after the end of irradiation, spectra were saved every 3 minutes until 58 hours total post-irradiation. However, the method utilized in Mandel et al. is restricted to low-enriched uranium below 20 weight percent ^{235}U content: this is because at enrichment levels greater than 20% the ^{239}Np peaks are indistinguishable from the much more prominent fission product activities. Therefore, I will only use the three low-enriched uranium samples from my experimental dataset: 0.71%, 3%, and 20%. I also combine every 20 individual 3-minute gamma ray spectra to obtain 50 one-hour spectra for each enrichment value (150 total spectra). I then show how the indicative photopeak ratios vary if the constraint of fixed decay time is relaxed. First, I present in Figure 3 the photopeak

count ratio results obtained from the original work by Mandel et al. as a function of the $^{238}U/^{235}U$ ratio. (Note that this figure only shows six of the calibration curves they developed.) We recreate all 18 curves with our own experimental data, and then see how the prediction performance of the ensemble varies as the decay time of the measurements is varied.

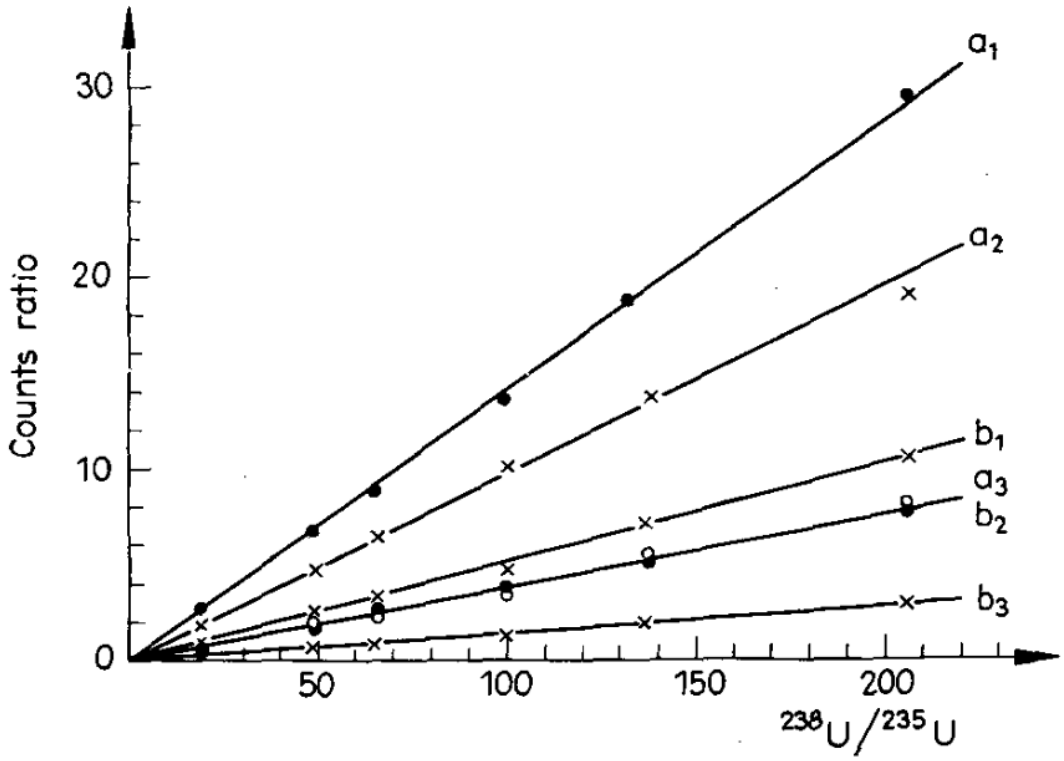


Figure 3: A few of the count ratios indicative of uranium enrichment as determined by Mandel et al. (1969). a_1=^{239}Np 105 keV/^{99}Mo, b_1=^{239}Np 105 keV/^{133}I, a_2=^{239}Np 278 keV/^{99}Mo, b_2=^{239}Np 278 keV/^{133}I, a_3=^{239}Np 210 keV/^{99}Mo, b_3=^{239}Np 210 keV/^{133}I.

Figure 4 shows the photopeak ratio curves recreated with my own experimental dataset for the one-hour gamma-ray spectra collected at 24 hours of decay time. Each curve has linear behavior with only slight deviations. The shaded regions surrounding each curve represent the uncertainty in linear fits. When I calculate the $^{238}U/^{235}U$ ratios of new specimens based on these curves I will take the average prediction of all 18 curves.

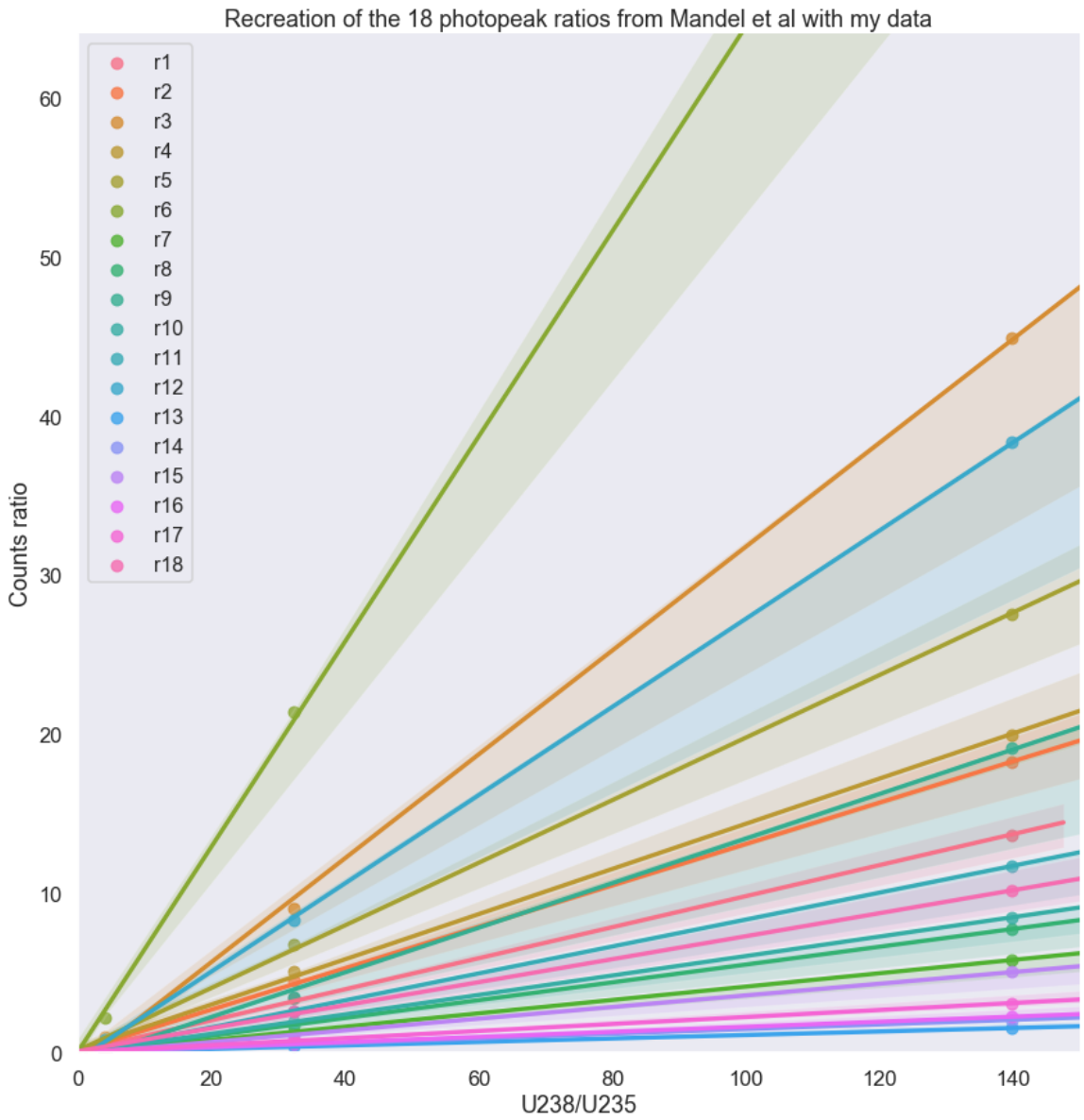


Figure 4: The 18 photopeak ratios from Mandel et al. using my experimental dataset at a fixed decay time of 24 hours.

I then used each of these curves to predict the $^{238}U/^{235}U$ fraction corresponding to measurements of 3% ^{235}U enriched uranium ($^{238}U/^{235}U$ value of 32.33) using the following protocol:

1. Determine the net areas of all nine photopeaks for one-hour gamma-ray spectrometry measurements of 24 hour decayed specimens of: 0.71%, 3%, and 20% enriched uranium.
2. Calculate each of the 18 photopeak ratios based on these net areas.
3. Perform linear regression using the photopeak ratio data from the 0.71% and 20% enriched uranium measurements, with the constraint that the y-intercept must be zero (a sample with no ^{238}U content will necessarily have a count ratio of zero because there will be no ^{239}Np counts in the numerator of the photopeak ratio).
4. Use the eighteen linear regression curves to perform eighteen independent predictions on the $^{238}U/^{235}U$ fraction corresponding to the 3% ^{235}U data based on its photopeak ratio values.
5. Use statistical analysis of the results to determine the mean prediction, mean absolute error in prediction, and standard deviation in the absolute error in prediction.

The mean prediction using this method was 33.22±1.81, which is in agreement with the true $^{238}U/^{235}U$ value of 32.33 corresponding to 3% ^{235}U enriched uranium. Figure 5 shows a histogram plot of the mean absolute error in prediction for each of the eighteen photopeak ratios, as well as summary statistics of the dataset. These results prove that the method developed by Mandel et al. works well when the decay time of the specimens is fixed.

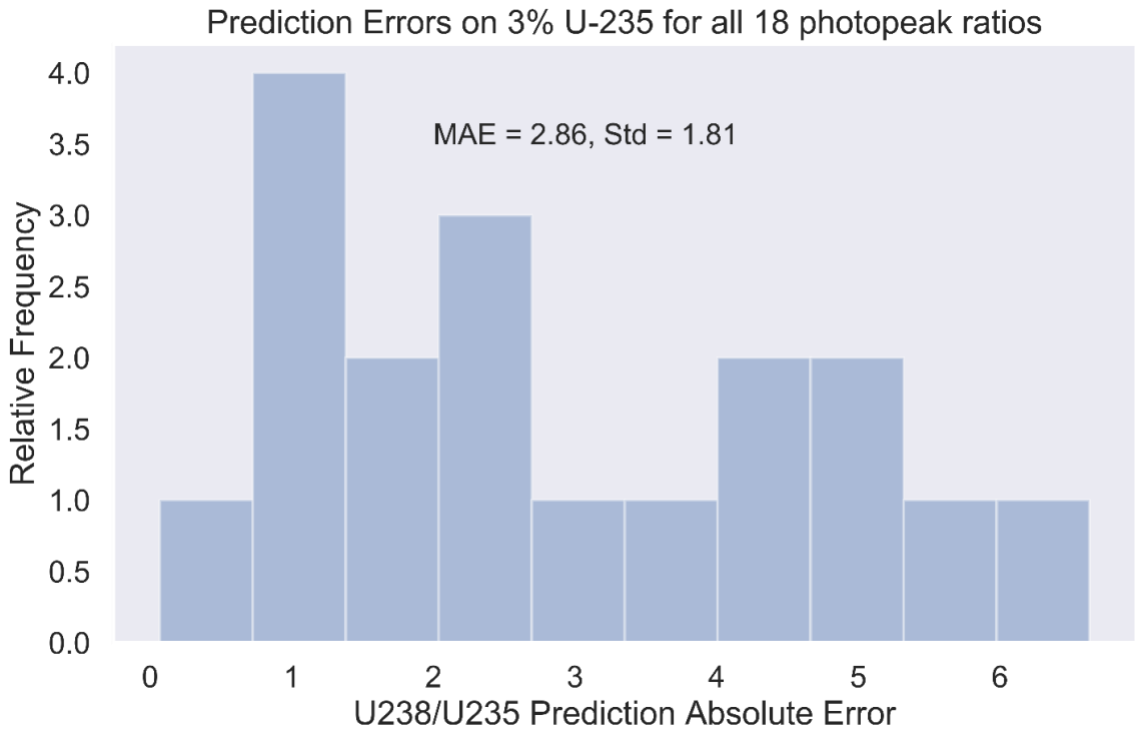


Figure 5: Absolute errors in prediction for all 18 photopeak ratios for linear regression fitted to and tested on 24 hour decayed specimens.

However, the scenario that I am particularly interested in is the determination of uranium enrichment when the decay time of the sample is loosely constrained to a range of values (8 to 58 hours in this case). Therefore, I repeat the procedure outlined above with the full experimental dataset across the decay time range of 8 to 58 hours post-irradiation. The linear regression and uranium isotope fraction predictions are both performed across all 50 decay times. The mean prediction using this method was 22.5±12.7, which, although still technically in agreement with the true $^{238}U/^{235}U$ value of 32.33 corresponding to 3% ^{235}U enriched uranium, has much larger absolute errors in prediction than the results previously obtained with a fixed decay time dataset. Figure 6 shows the histogram plot of the mean absolute error in prediction for each of the eighteen photopeak ratios, as well as summary statistics of the dataset. The results corresponding to predictions across all t_{dec} values are much worse than those where t_{dec} is fixed at 24 hours, as in Figure 5, which demonstrates that the method of Mandel et al. for uranium enrichment predictions suffers significant

degradation in performance as the constraint on decay time is relaxed. The goal of my work therefore is to develop a new method for enrichment determination which has performance comparable to Mandel et al. at fixed decay time but suffers minimal degradation in performance as the decay time constraint is relaxed.

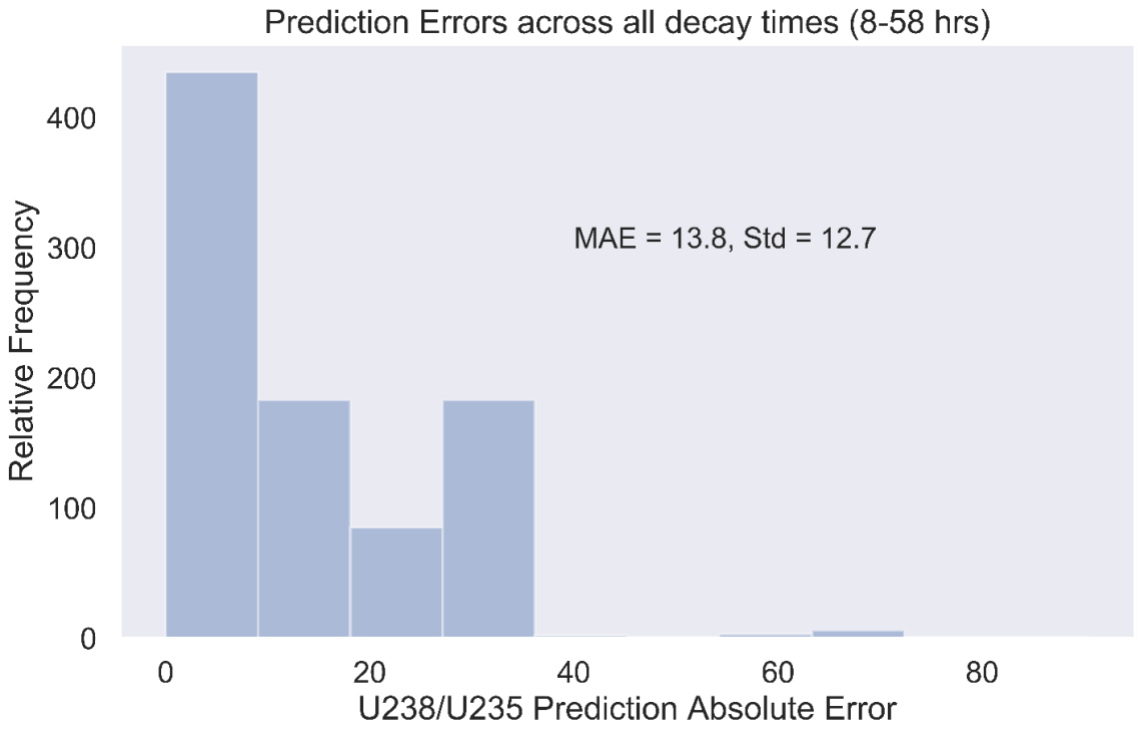


Figure 6: Absolute errors in prediction for all 18 photopeak ratios for linear regression fitted to and tested on specimens representing the full range of decay times (8-58 hours).

I will present one additional demonstration of the effect of varying decay time on the performance of Mandel et al.'s method. I have calculated two of the peak ratio curves shown in Figure 5 across the full 50-hour decay time range. Calculating the ratios b_2 and b_3 in Figure 5 with my experimental data across the full range of decay times results in the following spreads shown in Figure 7 and Figure 8 respectively. Note the strong dependence of the counts ratios for each function on the decay time.

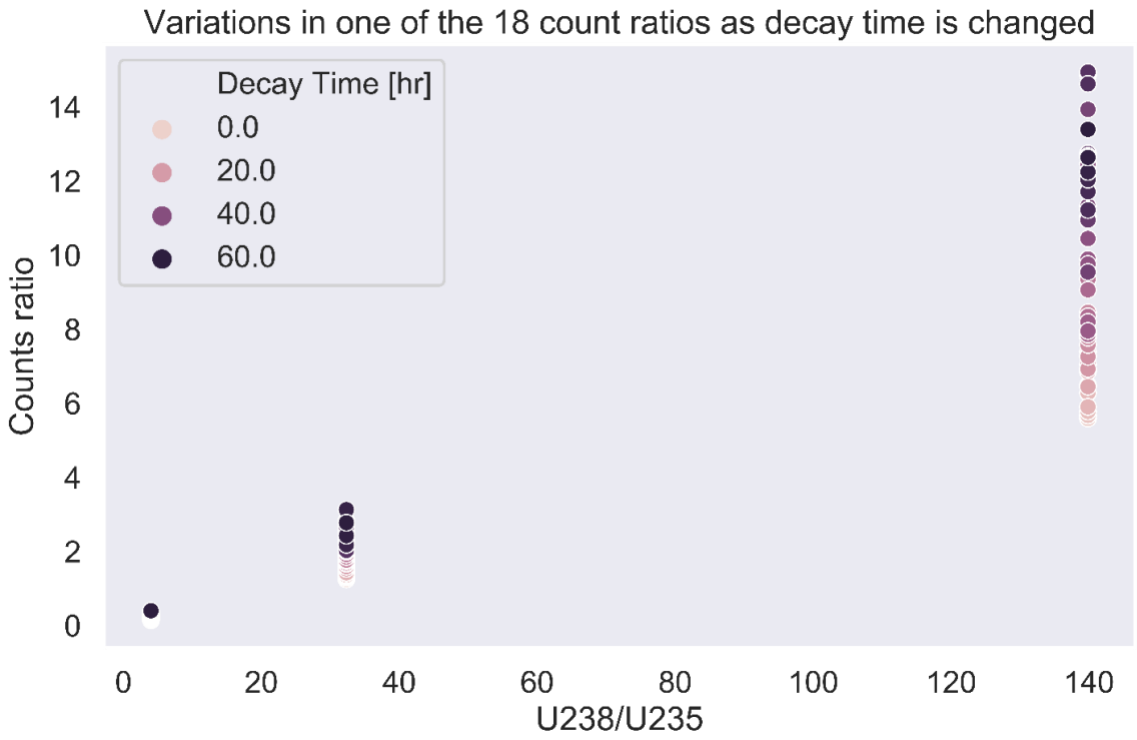


Figure 7: The photopeak ratio $b_2 = 278_{\text{net area}}/530_{\text{net area}}$ as a function of uranium isotope ratio across a range of decay times.

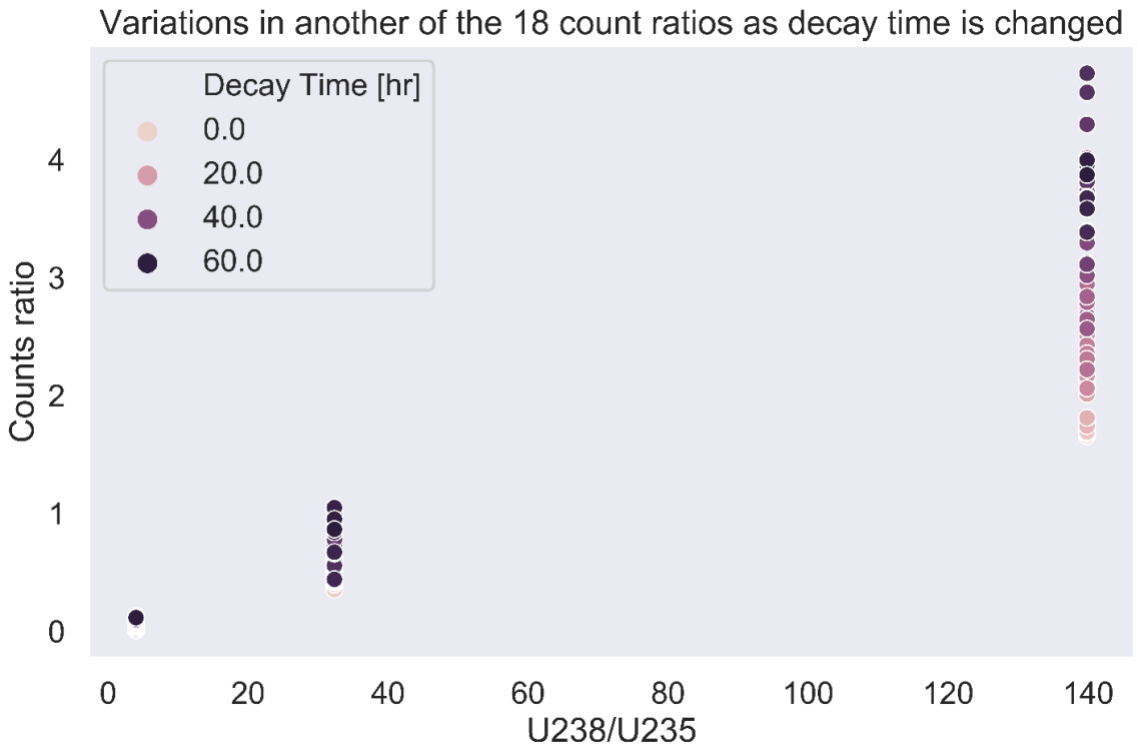


Figure 8: The photopeak ratio $b_3 = 210_{\text{net area}}/530_{\text{net area}}$ as a function of uranium isotope ratio across a range of decay times.

The decay time dependence of the count ratio value explains the poor performance shown in Figure 6, and is the essential motivating factor for my work: Mandel et al.'s method of utilizing a set of count ratios to determine uranium isotope ratios is dependent on precise knowledge of the decay times of every sample used in the construction of the model, and the sample that needs to be tested. The secondary motivating factor is that the old method is only valid at enrichment values less than 20% due to the relatively low intensity of the ^{239}Np emissions as compared to fission product activities at higher enrichment values. The machine learning method that I develop will improve on both of these shortcomings by developing decay time invariant models which perform well across all enrichment values.

Chapter 3: Literature Review

Nuclear Forensics and Nuclear Safeguards Multi-Variate Statistical Measurements

This section summarizes current methods for nuclear safeguards and nuclear forensics related measurements, with an emphasis on predictive analytic methods. It will also provide an overview of other assay methods applicable to spent nuclear fuel.

Nuclear safeguard objectives necessitate continuous and high throughput monitoring of the characteristics of special nuclear materials in spent fuel repositories, particularly for sites that implement spent fuel reprocessing. The high throughput of measurements and large amounts of data generated make this an ideal scenario in which to employ machine learning algorithms. Using machine learning to autonomously process and analyze data collected from special nuclear materials assay will significantly decrease the time, money, and labor burden of a robust nuclear safeguards infrastructure. Furthermore, such a system may perform more accurate and sensitive materials characterizations than is possible with conventional analytical techniques. This section will touch on a few of the current problems in nuclear forensics and nuclear safeguards which can or have already benefitted from statistical analysis and machine learning.

There is a history of experiments which have validated the utility of machine learning for performing meaningful measurements in nuclear safeguards contexts. For instance, an effective binary classification algorithm for discriminating between natural uranium and reprocessed uranium waste drums was developed with support vector machines and gamma-ray spectra obtained with thallium-doped sodium iodide scintillation detectors (NaI:Tl)[5]. Figure 9 shows typical gamma-ray spectra for natural uranium (NU) and reprocessed uranium (RU).

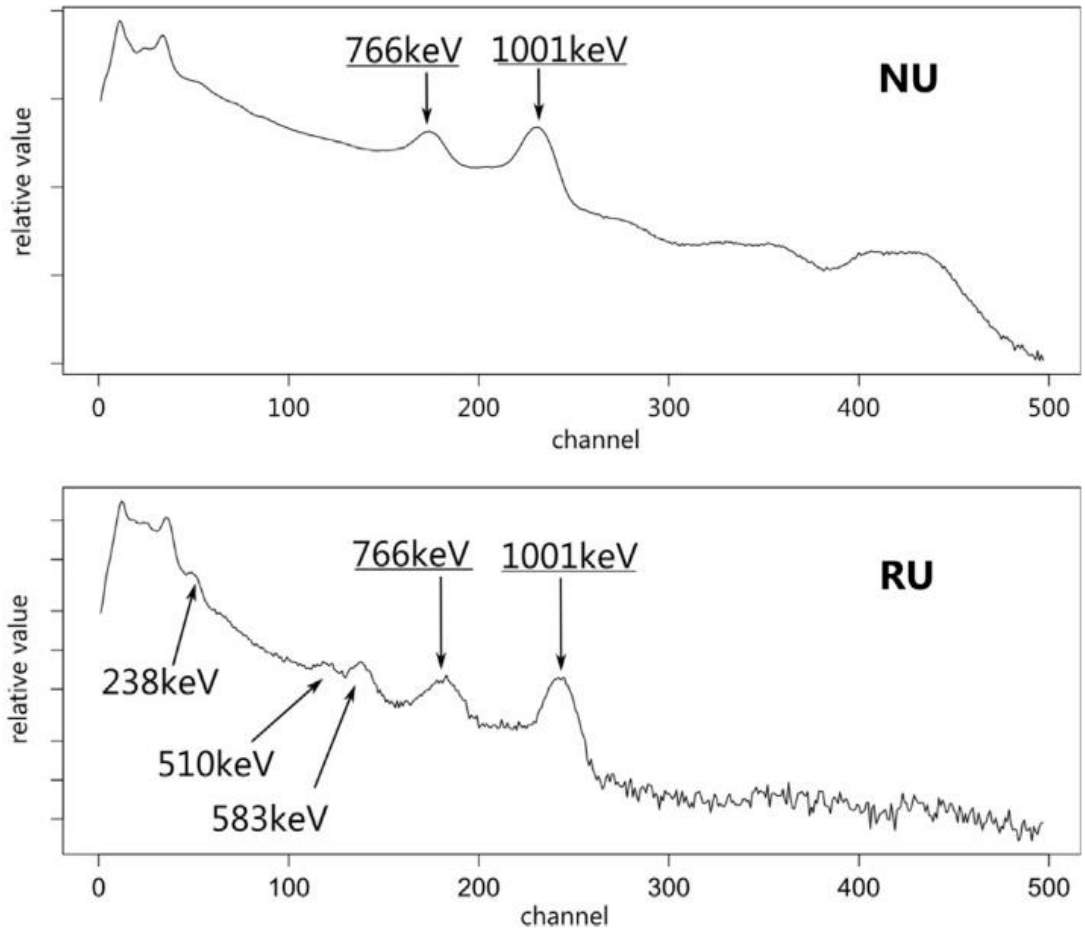


Figure 9: Typical NU and RU spectra with NaI(Tl) detectors [5].

The model hyperparameters were optimized via a logarithmically spaced grid search over a range of values. This support vector machine was trained on only 12 instances of low-resolution sodium-iodide measurements and was capable of successfully classifying 951 out of 955 test instances[5].

There are several studies which have focused on the characterization of spent nuclear fuel via nondestructive assay. In particular, gamma-ray spectrometry via scintillation detectors is a favorable nondestructive assay technique due to the simplicity of measurement, minimal labor involved, and potential to automate all steps in the process including data acquisition, analysis, and prediction. One study characterized the performance of cerium-doped lanthanum bromide ($LaBr_3$:Ce) detectors in a gamma-

gamma coincidence configuration for the measurement of very high-count rate spent nuclear fuel. In particular, this detection system was compared to the state-of-the-art gamma-ray spectrometry technology: high-purity germanium (HPGe) semiconductors. It was shown that the coincident $LaBr_3$:Ce detection system had a greatly improved signal-to-noise ratio (defined as the ratio of the true coincidence photopeak counts to the surrounding random coincidence count rate) than the HPGe system as the count-rate of the measured samples increased to levels commensurate with spent nuclear fuel[6]. The capability to perform automated nondestructive assay with such a system is high, and ongoing research efforts at The University of Texas at Austin are coupling the measurement capabilities of $LaBr_3$:Ce with automated survey robots for fully autonomous materials assay.

Another study by Shuryak (2017) demonstrates that machine learning on relatively small radiological datasets can benefit from the introduction of synthetic noise variables for constructing models which can distinguish between important and unimportant data variations. Several numerical experiments are carried out through the introduction of synthetic noise and variation to different elements of the modeling process: noise to the predictor data, adding noise to the target values, adding artificial effects to some predictors, varying the random number seed to see inherent variations in random model construction, and testing the performance of competing machine learning algorithms. The datasets used in this paper include synthetically generated data and published data: fungal taxa in samples of soil contaminated by the Chernobyl nuclear power plant accident, and bacterial abundance in soil samples under a ruptured nuclear waste storage tank. This work demonstrates improved analysis of these data compared to previous works. In particular, a negative effect of radioactive contamination was identified, and chromium was found to be a potentially more limiting factor for bacterial abundances than the radionuclides ^{137}Cs and ^{99}Tc [7].

The Korean Atomic Energy Research Institute developed a combined gamma and neutron measurement system for determining burnup and plutonium/uranium actinide ratios[8]. This system used a shielded collimator to expose the HPGe detector to only a small

discretized portion of the fuel element for each measurement. In addition to the experimental measurements for determination of Pu/U, the ORIGEN computer code was utilized to compute the expected Pu/U value, and destructive assay via mass spectrometry was performed. All three measurement methods were found to agree within ~4.5%. The ORIGEN results were used to find a relationship between $^{134}Cs/^{137}Cs$ ratio and burnup. This relationship was applied to HPGe experimental measurements of the $^{134}Cs/^{137}Cs$ ratio to determine the burnup. A similar relationship was found between the $^{134}Cs/^{137}Cs$ to the Pu/U ratio.

Frequently, nuclear safeguard measurements are complicated by the presence of heavy shielding material surrounding special nuclear material. Therefore, analysts must develop measurement methods that are insensitive to heavy shielding. High energy neutrons are a particularly effective probe of heavily shielded SNM due to their low total interaction cross-sections with the typical high Z shielding materials, such as Pb. One study by Tian et al. (2017) used active neutron interrogation with subsequent time-correlated neutron and gamma-ray measurements to determine mass and enrichment of uranium in the presence of heavy shielding material [9]. 14 MeV neutrons probe the total uranium mass because the fission cross-sections for ^{235}U and ^{238}U are similar at this energy. Therefore, measurements of the total fission rate can be directly correlated to the total uranium content while remaining invariant to the enrichment of the material. After the total uranium content is determined, the 14 MeV neutrons are moderated to obtain fission spectrum neutrons. The fission rates from the fission spectrum neutrons determine the uranium enrichment in ^{235}U. This is because the fission cross-section for ^{235}U at this energy is much higher than ^{238}U. Therefore, fissions from ^{238}U can be neglected while determining the enrichment of ^{235}U. Additionally, functional relationships between the coincidence signal response and the uranium mass and enrichment are both determined.

Jones et al. (2014) used dimensionality reduction techniques to analyze large datasets of nuclide concentration information in spent nuclear fuel and create classification models which determine the reactor type of origin[10]. The reactor types considered included Advanced Gas Reactors (AGR), Boiling Water Reactors (BWR), Canada Deuterium

Uranium (CANDU), Magnesium Non-Oxidizing (Magnox), and Pressurized Water Reactors (PWR). Variance and dimensionality reduction were achieved with Laplacian Eigenmaps (LE). This is an unsupervised method where samples are projected non-linearly onto a lower dimensional space. Figure 10 shows the LE representation of the nuclide compositions of the initial dataset into three dimensions corresponding to the first three principal components. These results demonstrate that there are clear, distinct differences between the isotopic compositions of spent nuclear fuel from each different reactor type and that a three-dimensional principal component truncation of the data is sufficient to differentiate between reactor types.

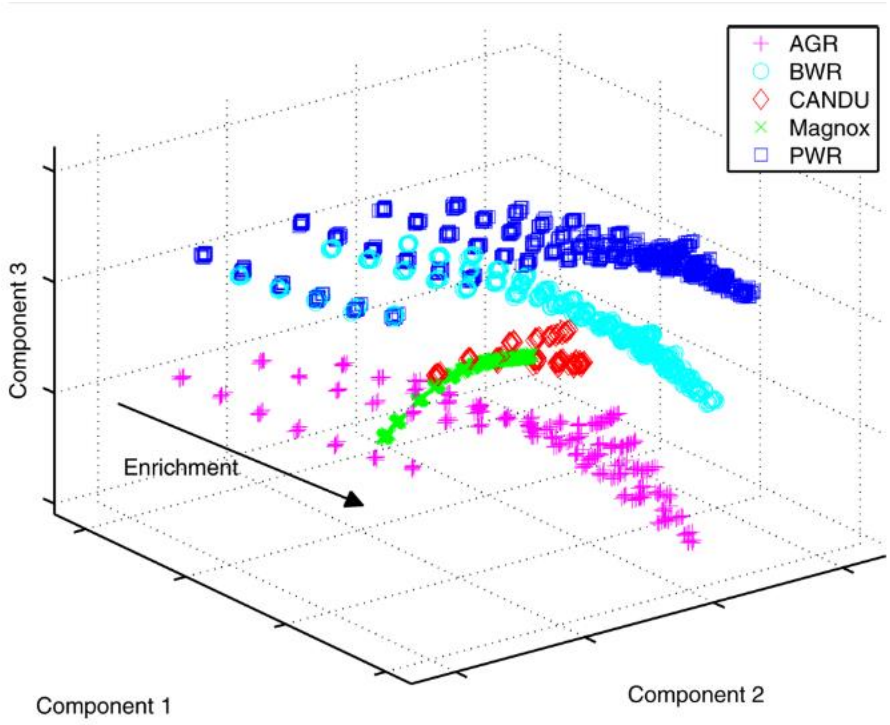


Figure 10: The 3-dimensional LE projection of the initial dataset[10].

Classifiers used included Random forest classifiers, linear discriminant analysis, quadratic discriminant analysis, and the Parzen window, which is a probabilistic neural network for classification. Classification accuracy was observed to increase with increasing number of trees in the forest. The most accurate classification model tested was the Parzen window. The dataset of spent nuclear fuel compositions was constructed with the Fission Product

Inventory (FISPIN) depletion code. This work exclusively studied synthetic analytical chemistry data. The author suggests extending the classification techniques here to real-world measurements.

There have also been experiments which considered unknown multi-variate parameters spaces for nuclear forensic measurements such as this next paper, which develops nuclear forensics measurement capabilities with unknown multi-variate parameters[11]. Robel et al. developed a position independent monitor (PIM) measurement technique for the determination of irradiation history of light water reactors via measurement of intercepted environmental samples that are presumed to have come from an unknown position in the reactor core. Combinations of Pu, Cs, and Ba isotopes are shown to have a consistent functional form that varies with irradiation history and is independent of irradiation position. This result is validated computationally, experimentally, and analytically. However, this measurement technique required mass spectrometry for the determination of isotope ratios, which is a labor intensive and time-consuming process. Furthermore, it is constrained to the analysis of light water reactor irradiation history and has no capability to discriminate between other materials sources. The resulting trend between Ba isotopic ratios and irradiation time is shown in Figure 11 and Figure 12 along with its axial position independence.

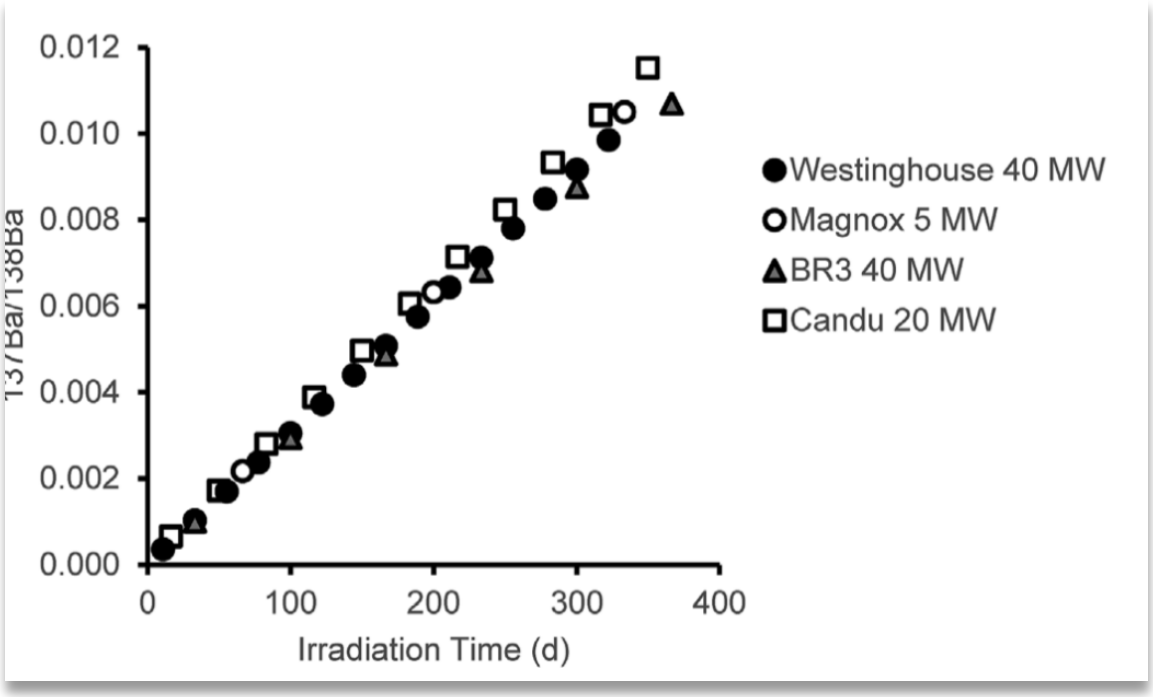


Figure 11: The reactor burnup monitor results across four different reactor designs.

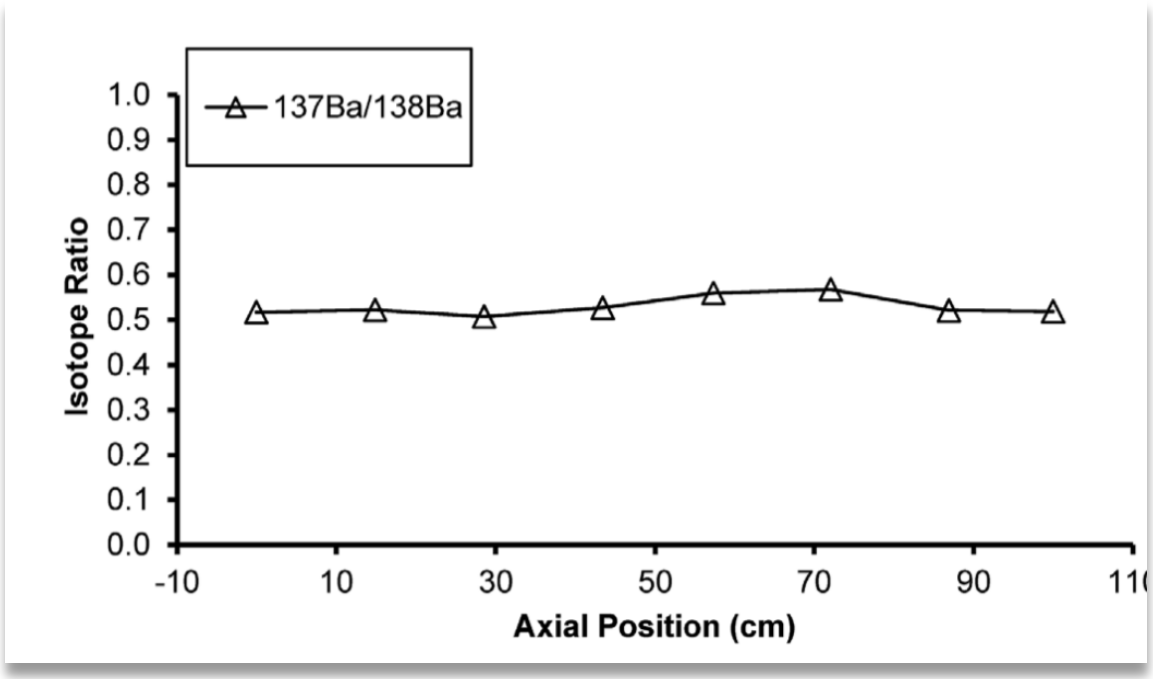


Figure 12: The axial position sampling invariance of the reactor burnup monitor [11].

The multi-isotope process (MIP) monitor is a non-destructive assay online monitoring program for the nuclear safeguards measurements of materials control and accounting at reprocessing facilities [12]. This monitor tracks material flows in real-time while collecting gamma-ray data. The gamma-ray data is subject to principal components analysis (PCA) and partial least squares regression to detect any changes to the material composition in near real-time. PCA is used to cluster the data instances in a lower-dimensional space which is useful for outlier detection. Research is ongoing to incorporate these analyses into a deployable system.

Dayman et al. (2014) created the reactor-type classifier with embedded used nuclear fuel burnup estimator [13]. This work is an extension of the MIP monitor. It includes a binary classification model based on partial least squares for discrimination between pressurized and boiling water reactors. This work was limited to artificial data generated in ORIGEN, and no data noise was introduced. This work also relied on direct nuclide activities output from ORIGEN, rather than a nondestructive assay-based, more experimentally accessible dataset such as gamma-ray spectra. The developed models therefore had artificially low errors due to lack of data noise and use of exact nuclide activities. Multi-variate parameters were included in the training and testing sets including uranium enrichment, cooling time, and burnup values spanning the expected ranges of commercial used fuel.

In some cases, Lindell et al. (2018) show that non-linear classification algorithms are shown to have the highest success rate[14]. For example, algorithms including k nearest neighbor and gaussian kernel SVM were used for binary discrimination between uranium oxide and mixed-oxide fuel gamma-ray emissions. This was accomplished with simulated gamma-ray activities of these two fuel types for decay times ranging up to 20 years. Classification scores of 100% success were obtained. The most important isotopes for the classifications were identified as ^{134}Cs, ^{137}Cs, and ^{154}Eu.

The characterization of irradiated uranium materials is a ubiquitous problem throughout the fields of nuclear safeguards and nuclear forensics. One of the primary challenges faced in nuclear safeguards is the re-verification of spent nuclear fuel casks

during dry storage. The IAEA has set forth standards regarding the detection expectations for proliferation-sensitive activities including "the ability to detect the diversion of 75 kg of ^{235}U in the form of low-enriched spent nuclear fuel within one year of potential diversion"[15]. The specific IAEA nuclear safeguards detection goals are to:

- Detect the diversion of 8 kg of plutonium in the form of spent fuel within three months of possible diversion.
- Detect the diversion of 8 kg of ^{233}U in the form of spent fuel within three months of possible diversion.
- Detect the diversion of 75 kg of ^{235}U in the form of spent low-enriched fuel within one year of possible diversion.
- Detect the diversion of 20 tonnes of thorium in the form of spent fuel within one year of possible diversion.
- Detect possible misuse of the facility for undeclared nuclear activities (e.g. use of an ISFSI as a temporary store for undeclared movements of nuclear fuel.)

The present work builds the foundations for a method that could be applied to the enforcement of any of these goals, but with a current emphasis on detection of ^{235}U in unshielded conditions. However, the assay of complex material such as spent nuclear fuel is difficult due to the wide ranges of multi-variate parameters and inherent variations that exist between individual samples including (but not limited to): burnup, core location, neutron flux profile, decay time, initial enrichment, amount of shielding materials, uncertainties in fission product yields, etc. These compounding sources of uncertainty make traditional nondestructive assay via, for instance, gamma-ray spectrometry through conventional analytic techniques for isotopic uranium content intractable. Therefore, there is need for more robust analytic methodologies which can perform accurate predictions for uranium assay in the presence of unknown parameters and inherent sample-to-sample variations. Currently, a Compton Dry-Cask Imaging System (CDCIS) re-verifies nuclear fuel in vertical spent fuel casks by measuring the ratio of scattered to un-scattered gamma-rays to determine if individual fuel rods are missing. Figure 13 shows a top-down view of

a representative spent fuel canister. The CDCIS is positioned above each individual fuel rod to perform this measurement and takes many hours to re-verify a single spent nuclear fuel canister[16].

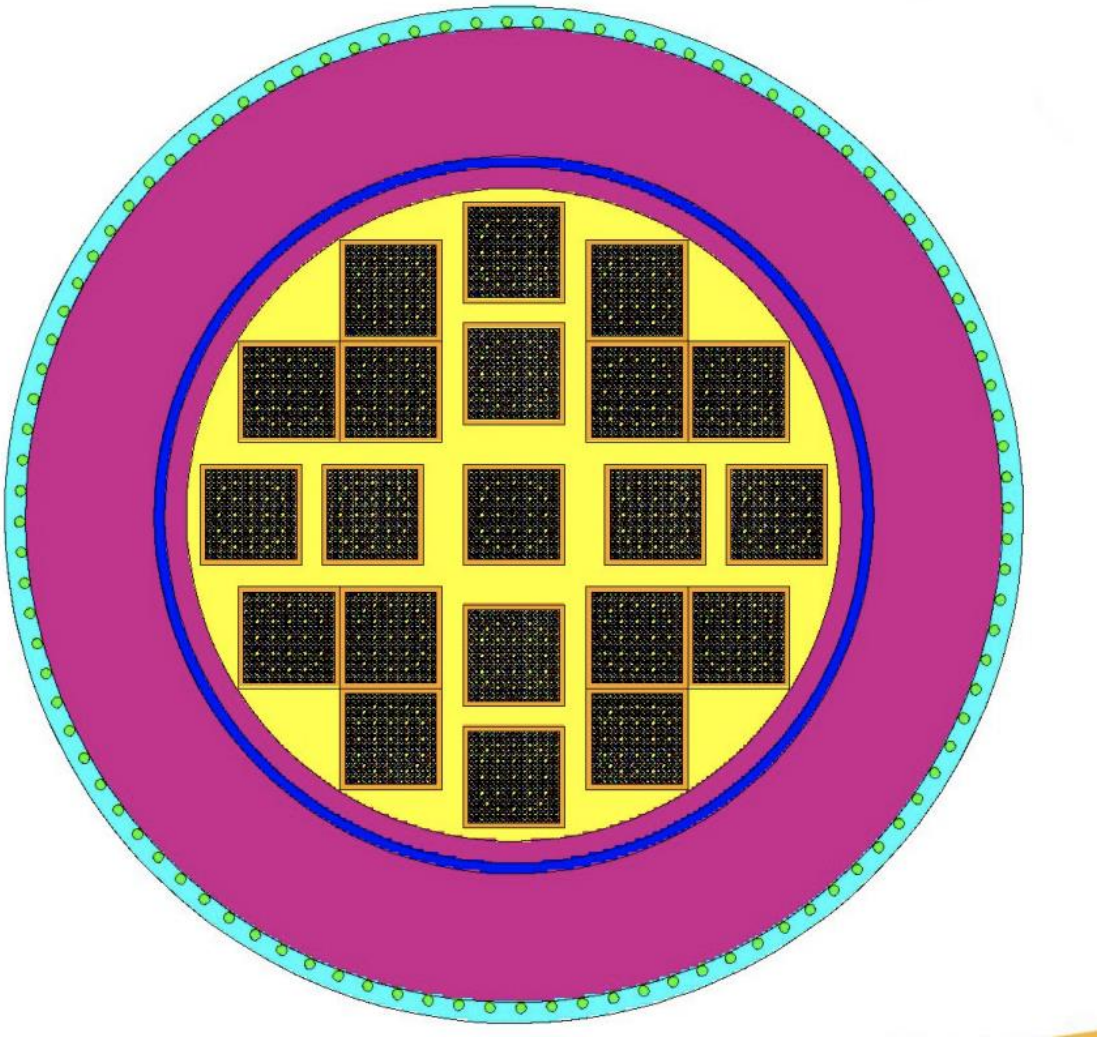


Figure 13: A top-down view of a spent nuclear fuel canister measured by the Compton Dry-Cask Imaging System[16].

In nuclear forensics, the collection of environmental samples is used as a proxy to gain as much information as possible about an inaccessible reactor or recently detonated

nuclear weapon. Again, this is a problem which has a host of unknown parameters and significant sample-to-sample variations.

The use of artificial neural networks and convolutional neural networks is investigated by Kamuda et al., for automatic identification of gamma-ray spectra. These models were shown to be very sensitive to the background radiation field. Neural nets were shown to be insensitive to changes in the detector energy calibrations, and could therefore prove useful for analysis of gamma-ray spectra collected with scintillation detectors such as NaI(*Tl*), which are prone to gain shifts depending on the counting conditions[17]. However, neural networks may not be the optimal tool for analysis of gamma-ray spectra. In general, it is difficult to uncover the features of the data which neural networks use to make decisions. This makes it hard to interpret the results of a neural network model for identification of gamma-ray spectra. My work differentiates itself from Kamuda et al. by using "white box" machine learning models such as decision trees for the analysis of gamma-ray spectra as opposed to "black box" models such as neural networks, which allows for analysis of the important features of the data which are used for making prediction decisions.

OTHER APPLICATIONS OF MACHINE LEARNING ALGORITHMS

Zhang et al. (2014) used decision trees to solve classification problems such as spam detection in email. In particular, the goal was to use decision trees to reduce the false positive error of mislabeling genuine emails as spam. This was achieved by defining a cost matrix for the objective function which increased the cost weight of false positive as opposed to false negative errors. K-fold cross-validation was also employed to reduce generalization errors. The total experimental dataset consisted of 6000 emails. Models were characterized using sensitivity, specificity, and accuracy with achieved scores of 91.02%, 97.51%, and 94.27%[18].

Bordari (2019) demonstrated that decision trees are useful for identifying the most influential variables in a dataset for making predictions. Decision trees were used to predict compliance with posted speed limits. Study participants filled out a questionnaire which included self-reported habit index (SRHI) ranking of intentions to comply with posted

speed limits as well as additional characteristics of driving behavior based on a scale of 1-6 representing level of agreement with the statement. The self-reported intentions were compared with the Theory of Planned Behavior (TPB). The decision trees allowed for comparison of the self-reported intentions and expected outcomes of driving behavior based on the factors reported in the questionnaires. The most important question for determining drivers' intentions to comply with the posted speed limit was found to be: "Driving within the 30 km/h speed limit is important." Other questions still impact the ultimate decision tree prediction to lesser extents. The predictions made on TPB by the decision tree matched the self-reported intentions exactly in 58.1% of cases, and were close in 84.5%[19].

Li (2018) used random forest regression for simultaneous lithium-ion battery capacity estimation and battery health monitoring. The capacity predictions use measurable signals as features such as current, voltage, and time, which are available during battery operation. Incremental capacity analysis was used as a complementary method for selecting the input features because incremental capacity curves can indicate relationships between voltage-capacity data and battery capacity fade. This technique was able to successfully evaluate the health of different batteries under various charge-discharge conditions with a root-mean-square error of less than 1.3%[20].

Chen (2019) compared random forest (RF) models to logistic regression (LR) and traditional bivariate weights of evidence (WoE) for landslide susceptibility mapping. Traditional WoE calculated results were used as inputs for the RF and LR models. Model performances were evaluated using receiver operating characteristic (ROC) curves and area under the curve (AUC). The resulting AUC scores for WoE, LR, and RF were 0.720, 0.773, and 0.802 for the training dataset, and 0.695, 0.763, and 0.782 for the validation set, respectively. The random forest therefore had the highest performing model in this study[21].

Chen et al. (2019) applied random forest regression to the monitoring of gross primary productivity (GPP) in forests as a proxy measurement for carbon cycle analysis and global climate change. Remote sensing data, meteorological data from Google Earth, and GPP estimates from eddy covariance measurements were used to develop a model to

predict site-scale GPP. The models were trained and tuned on 16 global deciduous broadleaf forest (DBF) flux sites and then evaluated on the remaining eight DBF sites. The RMSE performance metric was 1.93 $gCm^{-2}d^{-1}$. Feature analysis shows that the most important feature for these predictions is the enhanced vegetation index[22].

In the financial sector, Chihab et al. (2019) used random forests for foreign exchange speculations and algorithmic trading strategy development. A Probit regression model is used in conjunction with Random forest driven rule discovery. The random forest developed rules for entering and exiting the market. The training dataset consisted of 650 days of EUR/USD currency exchange rates with 7 past days and the moving average of the last week and last month for each of the 650 days. Coupled with this data was a collection of technical indicators including: relative strength index, stochastic oscillator, average directional index, commodity channel index, high-low, directional movement index, and moving average[23].

It can be useful to look to other problem domains to identify best practices in the development of a machine learning solution to new problems. Chattopadhyay (2018) considers the problem of securing machines from cyberattacks and detecting intrusions. Different machine learning techniques are compared on various datasets and using various performance metrics[24].

Ng (2004) studied regularization methods to prevent overfitting of logistic regression models when the underlying dataset has many irrelevant features. The number of training examples required to learn "well" is shown to grow logarithmically to the number of irrelevant features using L_1 regularization. Further, rotationally invariant algorithms (including logistic regression with L_2, SVMs, and neural nets trained by backpropagation) have sample complexity which at worst grows at supra-linearly in the number of irrelevant features. L_1 regularization is shown to have nearly optimum performance on datasets that fundamentally have a small number of relevant features that are used for prediction[25].

Chapter 4: Theory

This section will describe in detail the essential technical information that is necessary to understand each component of this dissertation. There are two main subsections. 1) The first subsection will describe the machine learning algorithms that were utilized: the fundamental concepts behind each, the objective functions that they optimize, the optimization routines they employ, and the hyperparameters that place restrictions on their construction. 2) The second subsection will cover the nuclear engineering aspects of this work including: operation of nuclear reactors, neutron activation analysis, irradiation of uranium, the resulting material composition and radiation field, and the mechanisms by which detection via gamma-ray spectrometry is performed.

TECHNICAL DESCRIPTION: MATHEMATICAL NOTATION & TERMINOLOGY

Before descriptions of the mathematical underpinnings of the learning algorithms and methods in this dissertation are presented, the *terminology* and mathematical notation must be clarified, due to conflicting standards in the academic community. I will remain consistent with the following terminology for the rest of the document.

- There are three separate *experiments* that are presented in this dissertation: the algorithm performance comparison on simulated data for t_{dec} invariant enrichment prediction with fixed t_{irr}, the performance comparison on simulated data as t_{irr} is incrementally increased to 2, 3, 4, and 5 discrete values, and the performance comparison on experimental data and fixed t_{irr}.
- Each of these experiments has its own associated *dataset* which is either experimentally collected or computationally simulated. The dataset is denoted: D.
- Each dataset consists of n *samples*, which are each of the unique measurements and associated labels. A sample is thus denoted by a measurement vector and a label scalar, so for n samples, the dataset is: $D = \{(\boldsymbol{X}_1, y_1), \dots, (\boldsymbol{X}_n, y_n)\}$.
- The *measurements* are gamma-ray emission values discretized into energy bins for the simulated data, and gamma-ray spectra as measured with a high-purity germanium detector for the experimental data. Each measurement is a vector $\mathbf{x}$

containing j features. We thus have, for measurement n: $\boldsymbol{X}_n = [X_n^1, \ldots, X_n^j]$. The complete set of measurements of all of the samples is thus the matrix $\mathtt{X}$.

- The *features* are the j discretized energy bins within the entire gamma-ray measurement (emission levels for simulated data and spectrum bin counts for the experimental data). Each feature has an associated feature value. The set of features is the set of values which compose the measurement vector $\mathbf{x}$.
- The *feature value* of each feature is the relative gamma-ray emission strength for the simulated data, or the raw number of gamma-rays present in that particular energy bin for the experimental data. For feature j the feature value is given as $\mathbf{x}_\mathtt{j}$.
- The *labels* are the uranium enrichment values associated with each particular sample. These are denoted by $\mathtt{y_n}$. The scenario studied in this work is the determination of uranium enrichment.
- In total, the dataset is now represented as:

$$D = \{([X_1^1, \ldots, X_1^j], y_1), \ldots, ([X_n^1, \ldots, X_n^j], y_n)\}$$

- The *model*, denoted as m, is the function that selects features within the measurement set $\mathtt{X}$ to make predictions on the labels $\mathtt{y}$.
- The *training/testing split* is defined as the fraction of the total samples which are used in developing the model. The remaining fraction is then used for performing predictions with the model and characterizing the model performance. As the total dataset size is varied, comparisons will be made on the performance of models trained with a fixed training/testing split versus a fixed number of training samples.
- The *objective function* is the underlying optimization problem that each algorithm solves while determining the model parameters. The objective function will sometimes include regularization parameters and will always include the scoring metric.
- The *scoring metric* is the equation that is used to evaluate the performance of models during grid searches and to compare between algorithms. This work uses *mean absolute error* as the scoring metric. Another very popular scoring metric is

the root-mean squared error. However, it will be explained later that mean absolute error is a more appropriate scoring metric for this dissertation due to varying testing sample sizes.

- The *algorithm* refers to the specific mathematical model used to make predictions. The algorithms considered in this work include: three regularized linear models (ridge regression, lasso regression, and elastic net regression), decision tree regression, and two ensemble methods based on decision trees (random forest regression and adaptive boosting regression).

TECHNICAL DESCRIPTION OF MACHINE LEARNING ALGORITHMS

There are three categories of machine learning algorithms that were considered in this work. The first category is regularized linear models. The next category is decision tree regression. The final category is ensemble methods based on decision trees.

Regularized Linear Models

Regularized linear models are a class of algorithms which are all based on ordinary least-squares with an added layer of sophistication called *regularization*. Regularization is the process of penalizing models for being excessively complicated. That is, having coefficients of large magnitude, or having very many nonzero coefficients. Regularization is accomplished via the introduction of an additional term to the ordinary least-squares objective function. A description of regularized linear models begins with a description of ordinary least squares.

Ordinary least squares (OLS) is a method for estimating the unknown parameters in a linear regression problem. This is accomplished by finding the model parameters w which act as coefficients to the data X to predict the target values y:

$$Xw = y \quad \text{(eq. 2)}$$

The goal of OLS is to minimize the objective function phi, which is a metric that quantifies the difference between the model predictions and the true values:

$$\Phi = \min_{m} ||Xw - y||^2 \quad \text{(eq. 3)}$$

The set of model parameters *w* that minimizes the objective function is typically found via the optimization routine **gradient descent**[26]. The gradient descent step is given by:

$$w^{(next)} = w - \eta \nabla_w MSE(w) \quad \text{(eq. 4)}$$

The procedure for gradient descent is as follows:

1. Choose a set of initial guesses for the model parameters *w*.
2. Evaluate the negative gradient of the objective function with respect to the model parameters at their chosen values.
3. Choose a step size η and adjust each model parameter by the product of the negative gradient and the step size.
4. Repeat this procedure until the objective function value is smaller than a chosen tolerance value.

Now, when building a model on real-world data, it is important to consider the possible effects of noise, random and systemic errors, and other nuisance variations in the data. OLS has a tendency to overfit the data to the target values and build complex models that attempt to predict this noise as well. **Regularization** is a technique to avoid this overfitting and force the construction of simpler, more elegant models that are insensitive to unimportant variations in the data. The three types of regularized linear models considered are **ridge**, **lasso**, and **elastic net**. The only difference between these three is the form of the regularization parameter which manifests in the objective function. Here are the forms of the regularization term in the objective function that each regularized linear model utilizes:

Ridge regression:

$$\Phi = \min_m ||Xw - y||_2^2 + 0.5\alpha||w||_2^2 \quad \text{(eq. 5)}$$

Lasso:

$$\Phi = \min_m ||Xw - y||_2^2 + \alpha||w||_1 \quad \text{(eq. 6)}$$

Elastic net:

$$\Phi = \min_m ||Xw - y||_2^2 + \alpha L_1||w||_1 + 0.5\alpha(1 - L_1)||w||_2^2 \quad \text{(eq. 7)}$$

The regularization parameter alpha is a hyperparameter which is selected by the user, as opposed to learned model parameters *w* which are calculated via gradient descent during model fitting. The strength of regularization is proportional to the magnitude of the selected regularization hyperparameter. For a regularization parameter of zero, for instance, the ridge regression algorithm is equivalent to OLS. In this scenario there is no penalty for model complexity or coefficient magnitude, so it is possible to build a model which heavily weights *all* of the features of the data, and likely captures more variations in the data than are strictly necessary to predict the behavior of interest. On the other hand, as the regularization parameter is increased, the penalty for complexity and therefore the penalty for coefficient magnitudes also increases, until eventually all coefficient weights are driven towards zero.

It is also important to note that, in these three regularized linear model algorithms, there are two different forms of regularization: using the L_1 and L_2 norms of the feature coefficient weights. They are each given by:

$$||\vec{w}||_1 = |w_1| + |w_2| + \ldots + |w_N| \quad \text{(eq. 8) and}$$

$$||\vec{w}||_2 = \sqrt{w_1^2 + w_2^2 + \ldots + w_N^2} \quad \text{(eq. 9).}$$

The L_1 and L_2 norms change the types of solutions that models generate. In particular, the L_1 norm has the property that it tends to drive coefficient weights towards zero. This results in *sparse* models. This is highly desirable if the goal is to produce an elegant, simple model that uses only a few features in the entire feature space. Therefore, Lasso regression, which uses the L_1 norm, will tend to produce simpler models with fewer features than ridge regression, which uses the L_2 norm. This is shown in Figure 14 and Figure 15, which plot the feature coefficient weights as determined by: a ridge regression model and a lasso model. Note that there are only a few non-zero coefficient weights in the Lasso model: the implementation of the L_1 norm in the regularization effectively drives most feature

coefficient weights to zero, whereas the ridge model has very many small but nonzero feature coefficient weights.

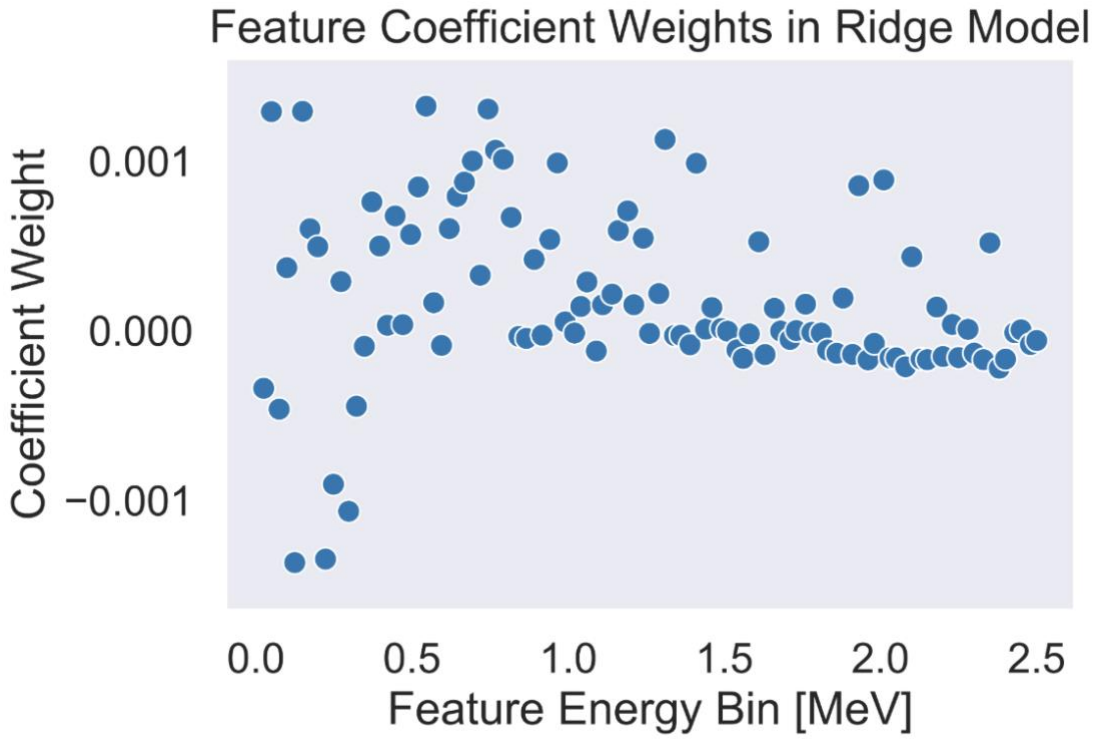


Figure 14: Feature coefficient weights as determined by a ridge regression model.

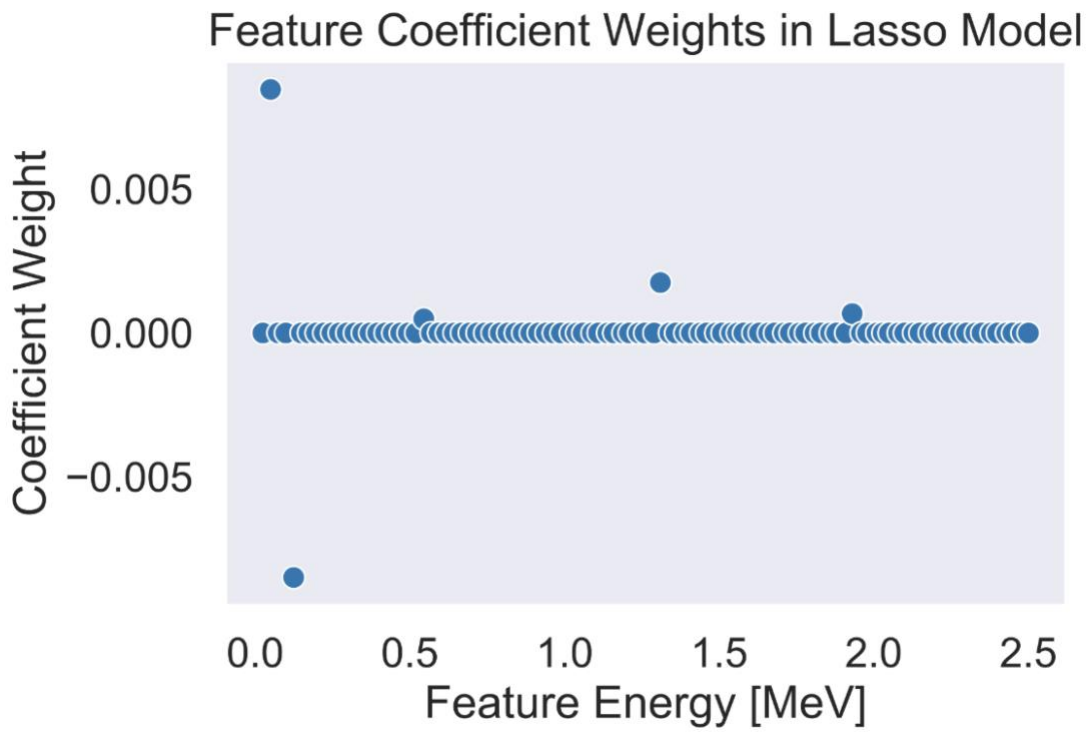


Figure 15: Feature coefficient weights as determined by a Lasso model.

Elastic net is a combination of the two algorithms in that it utilizes both the L_1 and L_2 norms, which is controlled through the inclusion of an additional hyperparameter L_1, which defines the strength ratio of L_1 to L_2 norms in the elastic net objective function.

Over and under fitting and the bias-variance tradeoff

When training a machine learning model, it is important to be aware of the inherent tradeoffs that exist with regards to model complexity. One prime example is the bias-variance tradeoff[27]. Bias is the difference between the average prediction of a model and the true value[28]. Models with high bias are typically underfit and have not learned enough to sufficiently capture the behavior of the training data. On the other hand, variance is the distribution of model predictions between different instances of data. Models with high variance are typically overfit and have captured random elements of the irreducible noise of the data into their formulation and therefore are highly sensitive to small variations.

Figure 16 visualizes the concepts of bias and variance with a set of noisy data that has been fit to two models: the orange model has a high bias - on average, the difference

between the model predictions and the true values of the data is high. It has not sufficiently captured the underlying behavior of the data. On the other hand, the green model has high variance - it is highly sensitive to the random noise fluctuations in the data and attempts to predict these features. This causes it to very accurately predict some quantities, while having huge errors with other. The goal in the construction of a good machine learning model is between these two extremes. It is important to create a model with just enough complexity to capture the important trends in the data while remaining insensitive to irreducible noise or unimportant variations in the data.

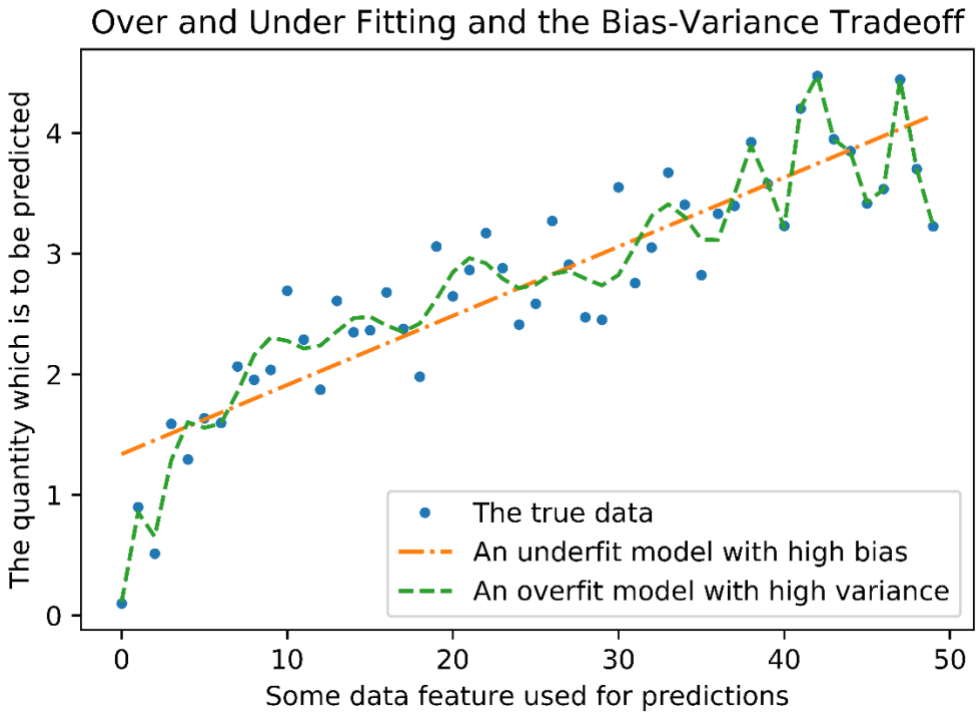
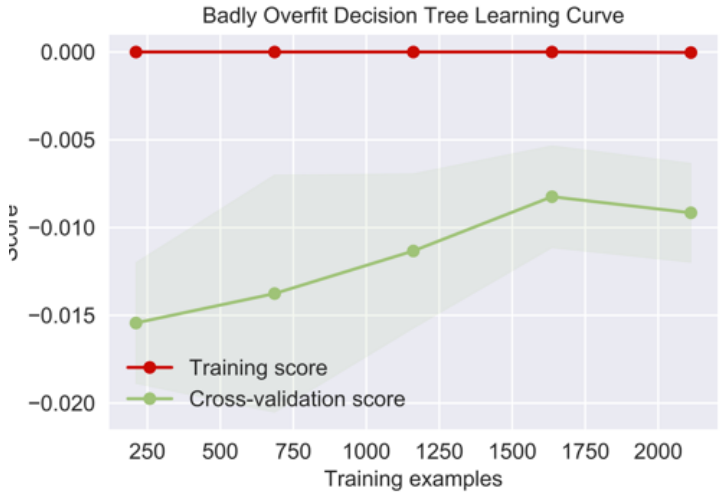


Figure 16: A conceptual representation of models which are underfit and overfit to demonstrate the difference between bias and variance.

Figure 17 visualizes the concept of overfitting with learning curves. Learning curves plot the performance of a given model on predictions on the training set and an evaluation set as a function of the number of training points that the model is given. The top-left learning curve shows a badly overfit decision tree's learning curve. Overfit models

have high variance and have the property that the difference in prediction performance between the training set and the evaluation set is high. Furthermore, as the training examples pass 1750 instances, the performance on the evaluation set continues to decrease, suggesting that the overfitting of this model is getting worse. By contrast, the lower right learning curve shows a well-fit decision tree. As the number of training instances increases, performance on the training set slightly decreases, but generalization to the evaluation set improves, and both curves tend to converge on the same score. This means that the model performs just as well on new data as it does on its own training data, suggesting that it has captured the true behavior of the data rather than random features that only exist in the training data.



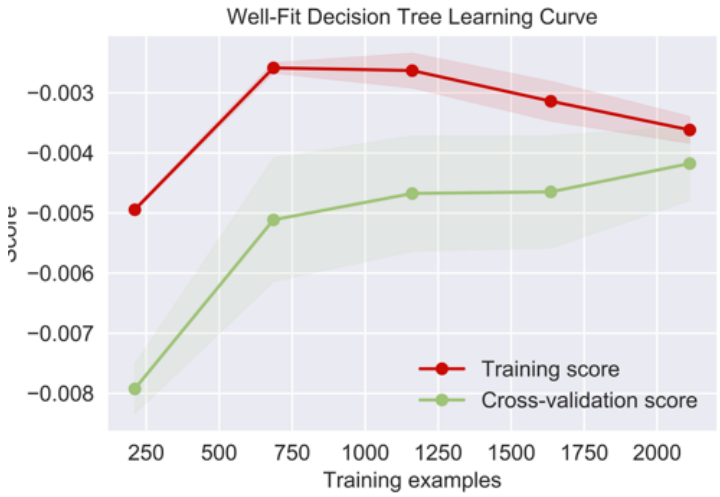


Figure 17: The learning curves of: a) a model which is highly overfit to the training data, and b) an appropriately fit model.

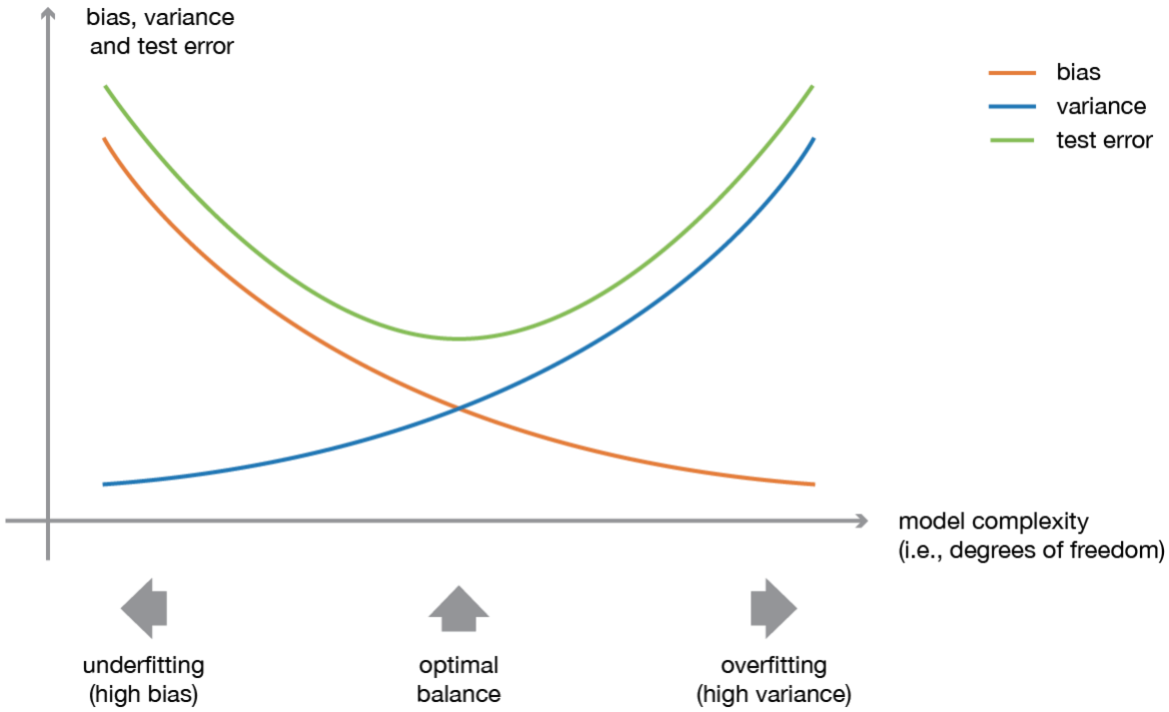


Figure 18: Another illustration of the bias-variance tradeoff[29].

Finally, Figure 18 visualizes the relationship between model complexity and prediction error, showing the concept of an intermediate optimum point where the model balances bias and variance to reach a minimum test error value.

Decision Trees

The next algorithm that was used in this work is the decision tree[30,31]. Decision trees will first be described conceptually and then mathematically. A decision tree is a collection of nodes, branches, and leaves which describe a procedure and order for evaluating individual feature values of a particular data instance to determine which prediction value to assign. The procedure for constructing a decision tree is as follows:

1. Identify the feature that has a threshold value that most cleanly separates the training instances into sets with the maximum "purity". For the case of regression trees, purity refers to the mean-squared error between data instance values and the average value belonging to the set. This identified feature and threshold value forms

the first node of the decision tree. The two groups that are created by this split are the first two branches of the tree.

2. Along each branch, additional nodes are created to maximally split the data into groups with the highest achievable purity. This creates more branches.
3. Continue generating branches until a particular node achieves an acceptably high level of purity. This node becomes a leaf node and is the terminal prediction point for data instances that reach this node.

When unrestricted, decision trees can quickly become extremely complicated as they attempt to generate increasingly long sets of rules that perfectly split the data into many leaf nodes. It is therefore important to place restrictions on the growth of decision trees. The user can place restrictions on the number of leaf nodes that are generated, the minimum number of data instances that must arrive at each leaf node, the maximum depth of the tree itself, the maximum number of features to consider when constructing a particular node, and more. All of these restrictions have the effect of regularizing the growth of the decision tree. They restrict how much the tree is allowed to learn from the training data, which prevents overfitting.

In `scikit-learn`[2], the implementation of decision tree regression is based on the Classification and Regression Trees (CART) algorithm. The mathematical formulation of the CART algorithm is as follows: the algorithm searches for a single feature k and a threshold value t_k which produces the purest subsets of data split by this feature and threshold value pair. The dataset consisting of m instances is then split into m_{left} instances on the left and m_{right} instances on the right. Purest subsets are defined to be the data split which minimizes the mean-absolute error (*MAE*) between each data instance label and the average values on each side of the split, weighted by the fraction of data which ends up on each side of the split. Therefore, the selection J of a feature and threshold value pair (k, t_k) at a given node for the splitting of m data instances is given by:

$$J(k, t_k) = \frac{m_{left}}{m} MAE_{left} + \frac{m_{right}}{m} MAE_{right} \quad \text{(eq. 10).}$$

The mean absolute error metric at each side of the node is given by:

$$MAE_{node} = \sum_{i \in node} |\hat{y}_{node} - y^{(i)}|$$ (eq. 11).

The prediction value on each side of the node is calculated as:

$$\hat{y} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)}$$ (eq. 12).

This process continues recursively, creating sub splits of each previous split, until either a tolerance *MAE* is reached, or until the maximum depth hyperparameter of the decision tree is reached. Figure 19 shows a sample decision tree regressor constructed using this method.

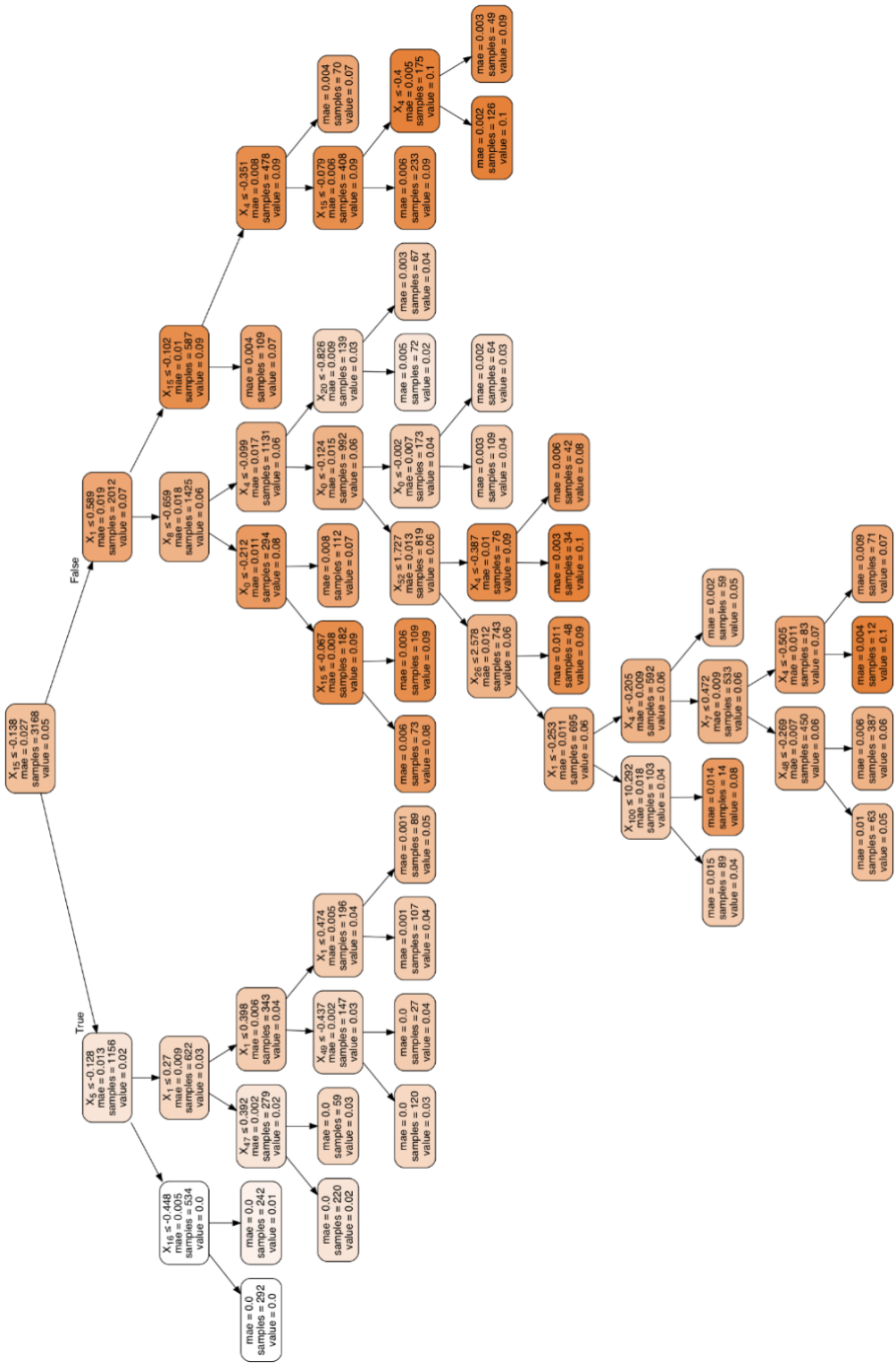


Figure 19: Sample structure of a decision tree.

Ensemble methods: Random Forests

The first ensemble method which was considered is the random forest. A random forest is a collection of decision trees, each trained in parallel on a small subset of the data, which collectively vote on the final prediction that is returned by the model[3]. The motivation behind random forests is predicated on the theory that "a large number of weak learners forms one strong learner." The random forest algorithm utilizing a random selection of features to determine the split of each node reduces error rates and increases robustness to noise when compared to Adaptive Boosting[3]. A visual representation of a random forest is shown in Figure 20. Here, numerous decision trees are trained in parallel, and each tree is tasked with predicting the class of every sample in the training set. A majority vote of the predictions made by each decision tree is used to determine the ultimate prediction of the random forest model.

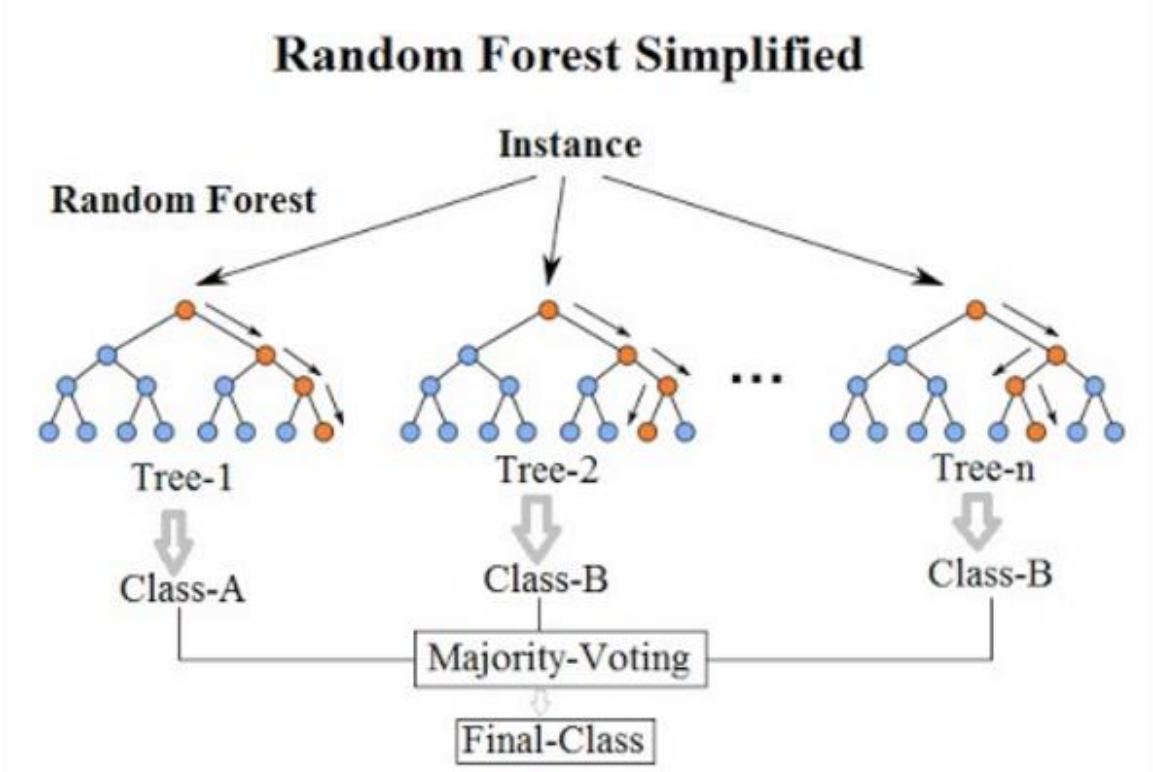


Figure 20: A visualization of a random forest classifier. Each parallel tree votes on the class of the sample and the majority vote is the ultimate ensemble prediction.

Random forests are designed to address the fundamental shortcoming of decision trees - decision trees are prone to overfitting. In principle, if no hyperparameters are imposed on the growth of the decision tree, trees could be grown to extreme depths, and every single data instance could be assigned its own leaf node which perfectly predicts the value of that instance. However, such a complicated tree is unlikely to effectively generalize to new data. Therefore, it is essential to limit the complexity of a single decision tree, and thus, limit its performance. There are many hyperparameters for limiting the complexity of a decision tree: the maximum allowed depth, the maximum allowed number of leaf nodes, the minimum number of samples which must terminate at a given leaf node, the maximum number of features to consider during the formation of a node, etc. The random forest addresses the shortcomings of simple decision trees by building many of

these restricted trees. In this way, each individual tree will have higher bias than a single complex decision tree, but significantly lower variance[32]. In total, the ensemble will have lower variance than a single decision tree, at the expense of slightly increased bias[26].

Ensemble methods: Adaptive Boosting

Adaptive boosting[33] is a serial ensemble method in which each predictive model in the sequence more strongly weights the data instances which had high errors in the previous model's predictions. This is accomplished by updating the objective function weights of individual instances of data seen by the next model in the series by an amount proportional to the prediction error associated with those individual data instances in the previous model. This process is graphically represented in Figure 21. In this figure, the first model in the sequence has a high classification error on one data point. The weight of that data point as seen by the next model is increased, causing the next model to correctly predict that data point at the expense of other predictions.

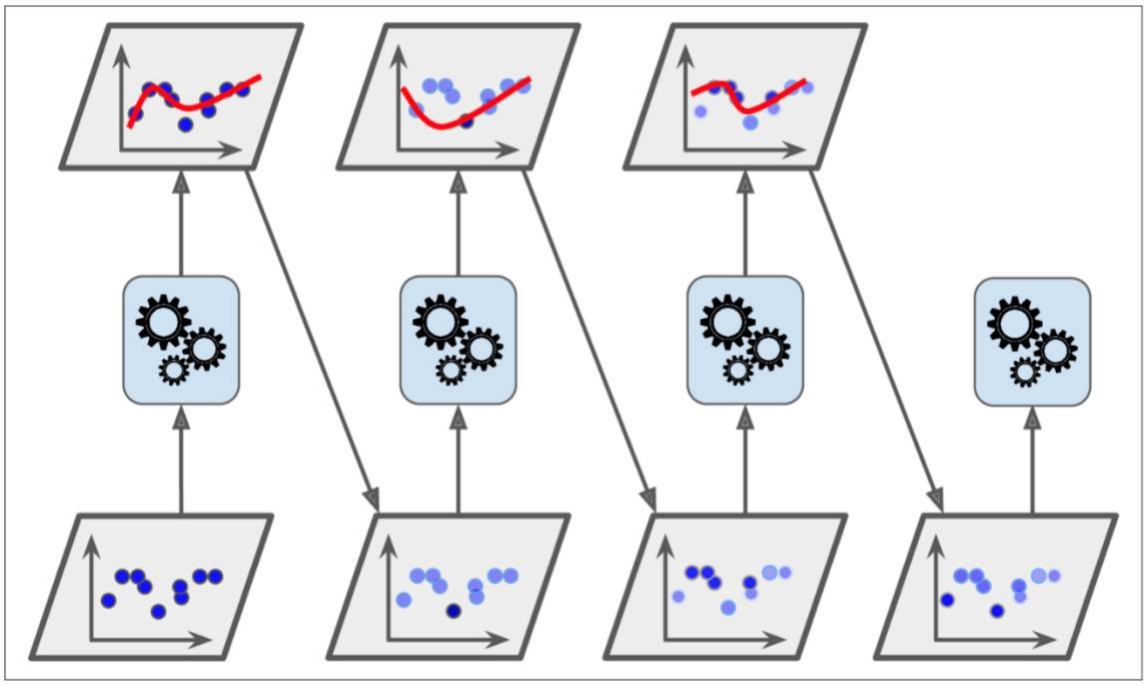


Figure 21: A visualization of the adaptive boosting process[26]. High error data instances have increased weight for the next model, improving the predictions on those data instances and sequentially correcting for errors.

The following section provides a mathematical description of the steps in the Adaptive Boosting ensemble method. The weight updates of individual data instances are given by:

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} \text{ if } \hat{y}_j^i = y^{(i)}, w^{(i)} \exp(\alpha_j) \text{ if } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases} \quad \text{(eq. 13).}$$

The ensemble of models then votes on the ultimate label of the data instance. But this time, the weights of the individual model votes are proportional to each model's error rate. The error rate of each model is given by:

$$r_j = \frac{\sum_{i=1, \hat{y}_j^{(i)} \neq y^{(i)}}^{m} w^{(i)}}{\sum_{i=1}^{m} w^{(i)}} \quad \text{(eq. 14).}$$

The error rates of each model are used to determine their voting weights in the ultimate ensemble decision, where the predictor voting weight is given by:

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j} \quad \text{(eq. 15).}$$

Therefore, higher-performing models contribute more strongly to the ultimate prediction of the ensemble.

Principal Component Analysis

Linear Principal Component Analysis

The goal of PCA is to find a lower-dimensional representation of the dataset D, where k is the dimensionality of the PCA projection and is less than the initial dataset dimensionality j of the dataset D. The simplest form of PCA is a linear projection of the data matrix X onto its covariance matrix eigenvectors. PCA via a linear projection requires calculation of the covariance matrix of the entire dataset and calculation of the corresponding eigenvectors and eigenvalues associated with this covariance matrix. The entries in the covariance matrix S at matrix coordinates (i, j) are defined by:

$$S^{i,j} = \frac{1}{N}\sum_{n=1}^{N}(x_n^i - \bar{x}^i)(x_n^j - \bar{x}^j)^T$$ (eq. 16),

where N is the number of samples in the dataset, and the elements (i, j) refer to the i_{th} and j_{th} features in the dataset. This quantity is calculated for every pair of features, and is equal to the variance of a given feature pair when i=j. The eigenvalues associated with the covariance matrix are calculated by the following:

$$det[S - \lambda I] = 0$$ (eq. 17),

and the corresponding eigenvectors are given by:

$$(S - \lambda I)\vec{x} = 0$$ (eq. 18).

However, the implementation of PCA in scikit-learn utilizes randomized truncated singular value decomposition for increased computational efficiency. This process is outlined next.

Randomized truncated singular value decomposition

The implementation of PCA in sklearn utilizes a randomized truncated singular value decomposition (SVD) developed by Halko et al. (2009)[34]. Here I outline the procedure for this method as presented in Halko et al. Given our initial matrix of observations and features **A** of dimensionality m x n (7,000 x 8,192), a target number of singular vectors k, and an exponent q, we compute an approximate rank-2k factorization:

$$U\Sigma V^*$$ (eq. 19),

where U and V are orthonormal, and **Σ** is nonnegative and diagonal. The steps are as follows:

1. Generate an n x 2k Gaussian test matrix Ω.
2. Form

$$\mathbf{Y} = (\mathbf{A}\mathbf{A}^*)^q\mathbf{A}\boldsymbol{\Omega}$$ (eq. 20)

by multiplying alternately with **A** and **A**∗.

3. Construct the matrix **Q** whose columns form an orthonormal basis for the range of **Y**.
4. Form

$$\mathbf{B} = \mathbf{Q}^{*}\mathbf{A} \quad \text{(eq. 21).}$$

5. Compute an SVD of the small matrix

$$\mathbf{B} = \tilde{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^{*} \quad \text{(eq. 22).}$$

6. Set

$$\mathbf{U} = \mathbf{Q}\tilde{\mathbf{U}} \quad \text{(eq. 23).}$$

The rows of **U** represent the principal components, or eigenvectors of the initial matrix **A** and the diagonal entries of **Σ** represent the corresponding eigenvalues.

Technical Description of Neutron Activation Analysis

Neutron activation analysis (NAA) is a nondestructive assay (NDA) technique for the determination of concentrations of elements based on the measurement of radioactive emissions which occur when the measured elements capture a neutron, become unstable, and emit radiation.

Irradiation of Uranium in a Reactor

In a nuclear reactor, the composition of materials changes through interactions with the neutron field and subsequent decay of the radioactive isotopes generated through these interactions. In this section we will present the sets of equations which describe these changing material compositions due to the interactions that occur in a reactor[35]. For actinides such as ^{238}U and ^{235}U, the primary interactions are transmutation through neutron capture and thermal neutron induced fission, respectively.

Let's first consider the depletion of ^{235}U. The primary relevant reaction that occurs in ^{235}U in the presence of a thermal neutron field is neutron induced fission. The Bateman equation which describes the concentration of ^{235}U as a function of time is then:

$$\frac{dN_{U-235}}{dt} = -\sigma_f \phi N_{U-235} \quad \text{(eq. 24),}$$

where $N_{U\text{-}235}$ is the concentration of ^{235}U at time t, σ_f is the one-group microscopic fission cross section for ^{235}U associated with the neutron flux energy spectrum phi.

The fissioning of ^{235}U results in fission products. These fission products radioactively decay, which results in source terms from parents and loss terms from decay for each fission product. The complete equation which describes the concentration of each fission product i as a function of time is then:

$$\frac{dN_i}{dt} = \phi \cdot \sigma_{hn} \sum_{f}^{n} \cdot y_n^j \cdot N_{hn} + \sum_{j,k} (\alpha_i^j \cdot \lambda_{Pj} \cdot N_{Pj} + \beta_i^k \cdot \sigma_c^{Pk} \cdot \phi \cdot N_{Pk}) - (\sigma_a^i \cdot \phi + \lambda_i) \cdot N_i$$

(eq. 25).

Here, the first term describes generation from ^{235}U fissioning, where:

- σ_{hn} is the microscopic fission cross section for the heavy nucleus,
- y_{ni} represents the fission yield of fission product i from fissionable heavy nucleus n, and
- N_{hn} is the concentration of the heavy nucleus.

The next term describes the production of the fission product from parent nuclei by either radioactive decay or neutron capture, where

- α_{ij} and β_{ik} represent the branching ratios of the parents P_j and P_k respectively,
- λ_{Pj} is a decay constant for parent P_j,
- N_{Pj} and N_{Pk} are the concentrations of the two types of parent nuclei, and
- σ_{cPk} is the microscopic neutron capture cross-section for parent P_k.

The third term represents the loss term for each fission product. The two types of loss are by radioactive decay or subsequent neutron capture, where:

- σ_{ai} is the microscopic capture cross section of fission product i.

There are many of these fission products produced. As such, there will be many gamma-ray energy signals that correspond to the initial ^{235}U content of the sample.

Next, let's consider the depletion of ^{238}U. Neglecting radioactive decay due to their extremely long half-lives relative to irradiation time, the primary ^{238}U reaction is neutron capture. The equation describing the concentration of ^{238}U as a function of time is then:

$$\frac{dN_{U-238}}{dt} = -\sigma_c \phi N_{U-238} \quad \text{(eq. 26)},$$

where σ_c is the one-group microscopic capture cross section for ^{238}U. This neutron capture results in unstable ^{239}U, which rapidly decays to ^{239}Np via the reaction $^{238}U(n, \gamma)^{239}U \rightarrow_{\beta} {}^{239}Np$. ^{239}Np then beta decays with a 2.356-day half-life to ^{239}Pu, emitting several gamma-rays, the most prominent of which have energies (and intensities) of 106 keV (25.34%), 228 keV (10.73%), and 278 keV (14.51%). I expect these three gamma-rays to be the prominent signals from ^{238}U in my measurements.

Measurement via gamma-ray spectrometry

Measurement via gamma-ray spectrometry starts after the samples have been removed from the reactor and decayed for 8 hours. The detector utilized in this work is a high-purity germanium detector (HPGe), which is a semiconductor material ideal for measurement of gamma-ray spectra. Here, the mechanisms by which these detectors operate is described.

Gamma-rays are detected by semiconductors via their interactions with the electrons contained in the semiconductor material. The three mechanisms of gamma-ray interaction with electrons include: the photoelectric effect, Compton scattering, and pair production. Each of these interactions transfers energy from the gamma-rays to the electrons. Semiconductors have the important property that the band gap between the valence electron band and the conduction band is on the order of E_g=1 eV[36]. Consequentially, electrons which are excited via interactions with gamma-rays are readily elevated from the valence to the conduction band. At room temperatures, thermal excitations alone would be sufficient to cause a continuous rate of electrons throughout the body of the detector elevating spontaneously to the conduction band. Therefore, HPGe detectors must be cooled, typically with a large dewar of liquid nitrogen, to a temperature of 77 K to prevent thermal excitations. A high voltage is applied across the body of the semiconductor, which causes excited electrons and their associated holes to accelerate in opposite directions across the potential difference. During this acceleration, the electron-

hole pairs interact with neighboring electrons, creating additional electron-hole pairs in a quantity proportional to the incident gamma-ray energy. This collection of electrons creates an output current from the detector with a magnitude linearly proportional to the incident gamma-ray energy. A signal amplifier shapes this signal into a normal pulse, and an analog-to-digital converter sorts this signal into one of 8,192 digital bins based on its magnitude. Computer software such as Ortec's Maestro then records each of these events into an energy dependent histogram of counted pulses. This is the gamma-ray spectrum.

Chapter 5: Methods

This section will explain in detail the steps that were taken to generate data, prepare the data for machine learning, and develop predictive models both with the simulated and experimental datasets. Figure 22 shows a representation of each of the major steps that will be discussed in this section for the simulated dataset.

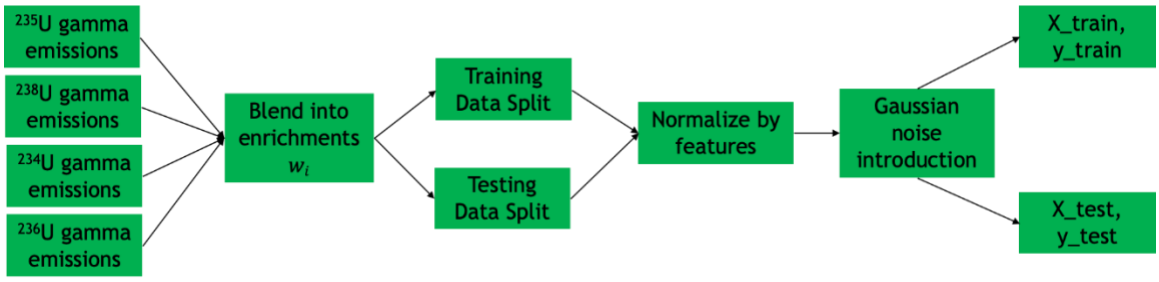


Figure 22: The major steps involved in preparation of simulated data for machine learning.

SYNTHETIC DATA GENERATION VIA ORIGEN

We begin with a discussion of the generation of simulated data using the Oak Ridge Isotope Generation Code (ORIGEN). ORIGEN is a software package which I have used to simulate the neutron irradiation of uranium and return the time-dependent gamma-ray emission intensities discretized into energy groups of the resultant material as the material radioactively decays[37]. I use ORIGEN to separately irradiate the four uranium isotopes ^{234}U, ^{235}U, ^{236}U and ^{238}U in the neutron field of the High Flux Isotope Reactor's pneumatic transfer facility at Oak Ridge National Laboratory for five different irradiation times: 5 minutes, 1 hour, 2 hours, 3 hours, and 4 hours, and return the gamma-ray emission data of the resultant material as a function of decay time, discretized into 180 bins ranging from 0 to 18 days post-irradiation. The gamma-ray emission data is discretized into 100 linearly spaced energy groups ranging from 0 to 2.5 MeV. These 100 energy groups are the "features" that I evaluate through machine learning. The samples are the 180 decay times at which the gamma-ray emission data is calculated.

Blending ORIGEN data into intermediate enrichment levels

Next, I blended the ORIGEN output data corresponding to each uranium isotope (^{234}U, ^{235}U, ^{236}U, and ^{238}U) in proportions commensurate with each of the seven ^{235}U

enrichment levels that I will experimentally measure. During the enrichment process, the concentrations of each of these isotopes vary as a function of the ^{235}U enrichment. I attempted to match these concentrations to the values reported in the literature. It is important to consider the variable concentrations of minor uranium isotopic impurities that exist in uranium samples of varying enrichments because it is possible that gamma-ray signatures from ^{234}U and ^{236}U may inform on the predictions made by machine learning models. A 1995 report from the Oak Ridge Y-12 plant on HEU to LEU conversion and blending facilities measured the isotope concentrations of ^{234}U, ^{235}U, and ^{236}U at a number of different weight percent enrichment values of ^{235}U (0.2%, 0.71%, 0.9%, 4%, 50%, and 97%)[38]. I have taken the data from that report and performed interpolations to generate second-order polynomial functions which estimate the minor uranium concentrations in uranium as a function of the ^{235}U enrichment levels. These interpolations are shown in Figure 23.

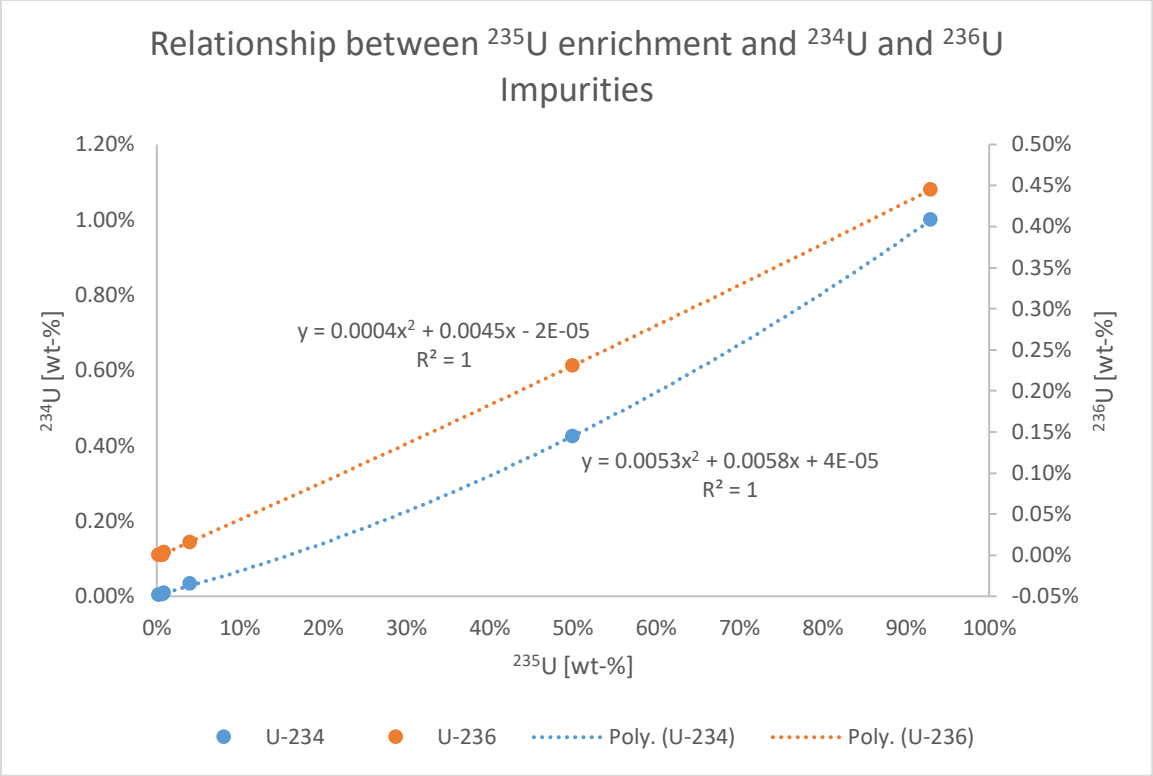


Figure 23: The concentrations of ^{234}U and ^{236}U as a function of ^{235}U weight percent enrichment.

With this information, the gamma-ray emissions data $\mathbf{X_w}$ corresponding to uranium samples of enrichment **w** are calculated via a linear combination of the ORIGEN output data for each individual uranium isotope ($\mathbf{X_{U\text{-}235}}$, $\mathbf{X_{U\text{-}236}}$, $\mathbf{X_{U\text{-}234}}$, and $\mathbf{X_{U\text{-}238}}$) as:

$$X_w = w * X_{U-235} + f_{U-236}(w) * X_{U-236} + f_{U-234}(w) * X_{U-234} + (1 - [w + f_{U-236}(w) + f_{U-234}(w)]) * X_{U-238} \quad \text{(eq. 27)},$$

where the functions $f_{U\text{-}236}(w)$ and $f_{U\text{-}234}(w)$ are the second-order polynomial fits of the minor uranium isotopic abundances to the enrichment value **w**, given by:

$$f_{U-236}(w) = 0.0004 * w^2 + 0.0045 * w - 2 \times 10^{-5} \quad \text{(eq. 28) and}$$

$$f_{U-234}(w) = 0.0053 * w^2 + 0.0058 * w + 4 \times 10^{-5} \quad \text{(eq. 29).}$$

It is important to acknowledge the fact that these ^{234}U and ^{236}U concentration values are specifically applicable to the measurements performed in a 1996 report on down-

blending of HEU. In general, the concentrations of the minor uranium isotopes vary significantly between materials of different provenance, and no single functional form can sufficiently capture the expected concentrations of minor uranium isotopes in general. Therefore, the data used here is but one example of potential minor uranium isotope concentration functions. As will be shown later in experimental measurements, the standard reference materials that I irradiated had minor uranium concentrations which differ significantly from the values shown here. I view these discrepancies as more of an opportunity than a problem however – it allows us to test the performance of models under an increased variety of conditions regarding minor uranium isotopic concentrations.

There are several shortcomings to the method I just described for generating uranium samples of varying enrichments. By combining the results of independent ORIGEN simulations for each pure isotope, I neglect the influence that each uranium isotope has on the neutron flux energy spectrum seen by the other isotopes. In particular, the large neutron capture resonances exhibited by ^{238}U suppress the neutron flux magnitude at those energies as seen by the other uranium isotopes. Figure 24 shows these neutron capture resonances. In hindsight, generating ORIGEN outputs directly corresponding to each uranium enrichment level with all four uranium isotopes included in the input file would allow my data to consider the influences of each uranium isotope on the neutrons seen by the others. The advantage of the procedure I chose to take is that it allows for a modular dataset, i.e., with ORIGEN outputs corresponding to each individual uranium isotope, I can rapidly post-process that data in python to generate whatever intermediate enrichment levels I want.

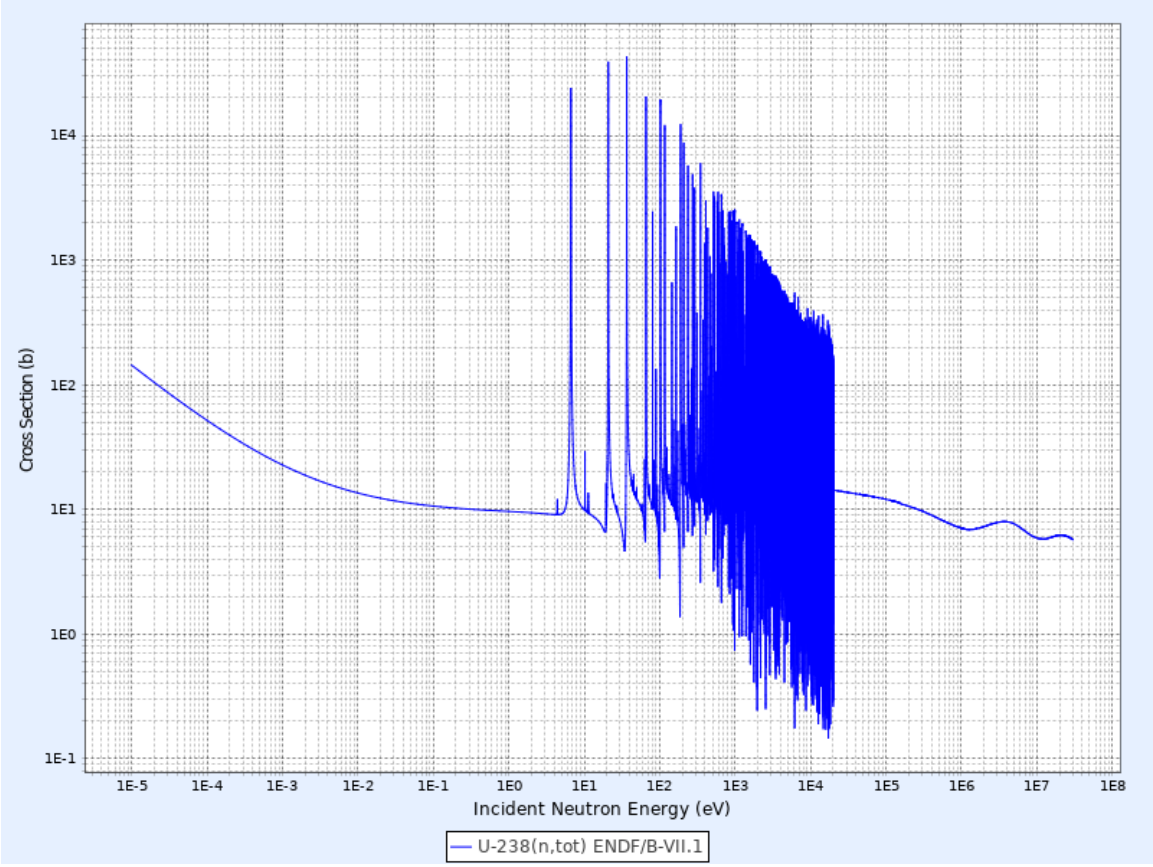


Figure 24: The total neutron interaction cross-section plot of ^{238}U. Taken from the Korean Atomic Energy Research Institute.

The next major shortcoming in this procedure for the generation of synthetic gamma-ray data is that there are no considerations of the logistical concerns of real-world gamma-ray spectrometry including: self-shielding and detector response functions (efficiency and energy resolution). These effects will strongly influence the very low energy gamma-ray emissions below approximately 150 keV. Gamma-rays at those energies will be much more strongly suppressed in the real-world data than these simulations suggest. I acknowledge that the simulated data generated here is not a high-fidelity one-to-one substitute for experimental measurements. That is not its purpose. The purpose of this dataset is merely to demonstrate the feasibility of machine learning and show that enrichment predictions can be made with low errors using datasets of gamma-ray emissions. The performance

results from the simulated data will inform on the ultimate design of the much more laborious and time-consuming experimental measurements.

Splitting the data into training and testing groups

The next step is to split the data into training and testing groups. I use the training data to tune the hyperparameters of a model through cross-validation, and then fit the model on the entire training set. Then, I test the fit model on the test dataset, and measure its performance using the scoring metric of mean-absolute error. In this work, I compared two different scenarios of data splitting into training and testing groups when the t_{irr} parameter is varied. The first scenario introduces a mismatch between the training and testing groups by training models exclusively on one t_{irr} value and testing their performance on all t_{irr} values to obtain a notion of the generalization performance of models. The second scenario holds the training fraction fixed at 0.8. For models that consider more t_{irr} values, the total dataset size is larger. As such, the number of training instances that the models fit to increases. Table 2 shows the total number of data instances for each number of considered irradiation times, and the training set sizes of each strategy.

Table 2: The sizes of the total dataset and training set for the five different numbers of irradiation times considered when: 1) holding the training samples fixed, and 2) holding the training fraction fixed.

Num. of irradiation times	Total dataset size	Fixed X_{train}	Fixed X_{train}/X_{test}
1	1440	1152	1152
2	2880	1152	2304
3	4320	1152	3456
4	5760	1152	4608
5	7200	1152	5760

By comparing these two methods for training/testing splits across a range of different discrete numbers of irradiation times, I compare the relative effects of training models with more data versus training models on datasets that are inherently more varied (by including more irradiation times) versus introducing mismatches into the training and testing groups to gauge the generalization performance of these models.

Normalizing individual data samples

The next data preparation step normalizes the data feature values of each data sample such that the total sum of all feature values is one. This step is necessary because the activity level, and thus total number of gamma-rays counted for an individual sample varies as a function of: irradiation time, decay time, sample mass, and uranium enrichment level. Because of this multi-variate dependency of total counts, a model that is generated via unnormalized data will be sensitive to all of these factors. The ultimate goal is to generate models that are invariant to as many parameters as possible. Performing the data normalization therefore helps to ensure that, for instance, sample mass will not significantly influence the reliability of a developed model.

I perform data normalization according to the following equation:

$$X_j^i(norm) = \frac{X_j^i}{\sum_j X_j^i} \quad \text{(eq. 30)},$$

where j is the index of the features corresponding to each sample, and i is the index corresponding to each observation. This normalization process allows samples of different activity levels to have their gamma-ray spectrum shapes compared without raw activity informing on the predictions. It is only the relative intensities of the gamma-ray energy bins in the spectra which inform on the models.

Gaussian Noise Introduction

The next step in the data preparation process introduces noise as data value perturbations sampled from a Gaussian distribution. I perturb the simulated data gamma-ray intensity values according to random, normally distributed Gaussian noise of 5% variance and mean of 0 via to the following equation:

$$X_{noise} = X(1 + \mathcal{N}(\mu, \sigma)) \quad \text{(eq. 31)},$$

where the gaussian normal equation is given by:

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{(eq. 32)}.$$

I perform this step using the numpy function `numpy.random.randn()`, which returns randomly selected samples from the "standard normal" distribution. The purpose of introducing gaussian noise into the simulated data is to more closely represent the errors and uncertainties associated with real-world counting statistics. Introducing this noise allows us to answer the following question with simulated data: will a statistical learning method be capable of ignoring the noisy, unimportant variations in the data associated with counting statistics and still make meaningful predictions?

At this point the synthetic data is prepared. The training and testing datasets are now stored in separate `numpy` arrays X_{train} and X_{test}, and their labeled enrichments are stored in the `numpy` arrays y_{train} and y_{test}.

EXPERIMENTAL DATA COLLECTION

This next section describes the steps taken to generate an experimental dataset of gamma-ray spectrometry measurements of irradiated uranium of varying enrichment levels.

Sample preparation

For this work I first acquired a set of certified reference materials (CRM) of uranium in the form U_3O_8 of varying enrichment levels from New Brunswick Laboratory.

These CRMs were then weighed out into 5 mg vials and prepared in rabbits (vials) for irradiation. Table 3 shows the CRM samples that were prepared in this experiment including their certified material compositions, and the sample masses that I weighed out for measurement. The mass uncertainties for all samples are +/- 0.01mg. Figure 25 shows the procedure for vial preparation. The 5 mg uranium standard is weighed out into the smallest vial on the left. This small vial is then placed into a medium sized vial. Two vial spacers are placed above and below the medium sized vial, and it is inserted into the large rabbit for irradiation.

Table 3: The sample information and certified isotopic composition data for this experiment.

Internal ID	**NBL CRM**	**Mass [mg]**	**Atom % ^{235}U**	**Atom % ^{238}U**	**Atom % ^{234}U**	**Atom % ^{236}U**
V0322	U-129A	4.99	0.72087 ±0.00039	99.27382 ±0.00039	0.0052962± .0000038	0.0000097± .0000012
V0324	U030A	5.06	3.0404 ±0.0016	96.9312 ±0.0016	0.02778 ±0.00006	0.000599 ±0.000005
V0328	U0002	5.03	0.01755 ±0.00005	99.9823 ±0.0001	0.00016 ±0.00001	<.00001
V0332	U0200	5.04	20.013 ±0.020	79.651 ±0.021	0.1246 ±0.0003	0.2116 ±0.0006
V0336	U500	4.94	49.696 ±0.050	49.711 ±0.050	0.5181 ±0.0008	0.0755 ±0.0003
V0340	U630	5.07	63.353 ±0.020	35.066 ±0.020	0.61894 ±0.00043	0.96230 ±0.00067
V0344	U970	4.91	97.663 ±0.003	0.5229±.00 06	1.6653 ±0.0017	0.1491 ±0.0005

Figure 25: From left to right, the procedure for preparing sample vials.

Irradiation

Next, we irradiate these samples in UT Austin's TRIGA Mark II Reactor in the 3L epithermal neutron irradiation facility at a power level of 10 kW for 5 minutes, plus additional neutron fluence during reactor startup and shutdown. Care was taken to perform the startup and shutdown procedures each day in the exact same manner to ensure that the total neutron fluence each sample was exposed to remains as consistent between samples as possible. We irradiate each sample separately so that we can perform consecutive gamma-ray spectrometry measurements during the same 8 to 58-hour decay time window for each sample. I attempted to measure the neutron fluence that each sample was subjected to so as to verify that they are very similar in magnitude with Co flux monitors. However, the irradiations had quite low total neutron fluences due to the power and time used, and the Co flux monitors had an undetectable amount of activation activity.

Gamma-ray spectrometry measurements

I use Ortec's Maestro gamma-ray spectrometry software with a high-purity germanium detector. I position the samples on a 25 cm spacer above the detector. I chose this distance to ensure that even for the 97% enriched sample, dead-times would never exceed 20%. The placement of the samples above the HPGe detector is shown in Figure 26. This system uses an analog nuclear instrumentation bin for all detector signal processing up to and including digitization into the gamma-ray spectrometry software ORTEC Maestro. This NIM bin is shown in Figure 27. Note that the full NIM bin is a Compton suppression system. However, in this experiment I am using basic single detector gamma-ray spectrometry without Compton suppression. Therefore, all components displayed in Figure 27 are not used. The relevant components are: the high voltage power supply, the signal amplifier, and the analog to digital converter. A flowchart showing all of the important components of the system for this experiment is shown in Figure 28.

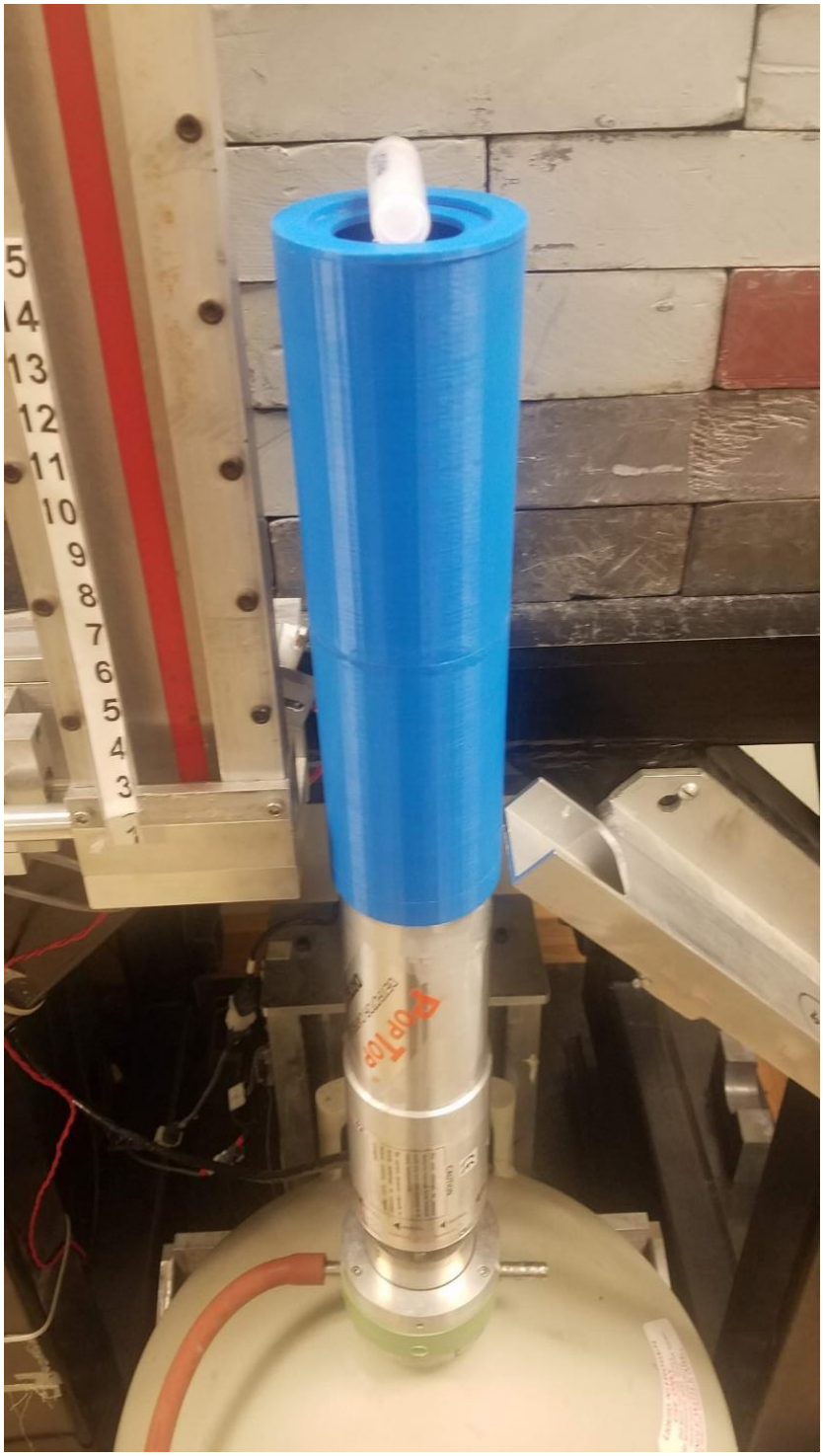


Figure 26: The counting geometry used in this experiment. The sample-HPGe detector blue spacer is 25 cm.

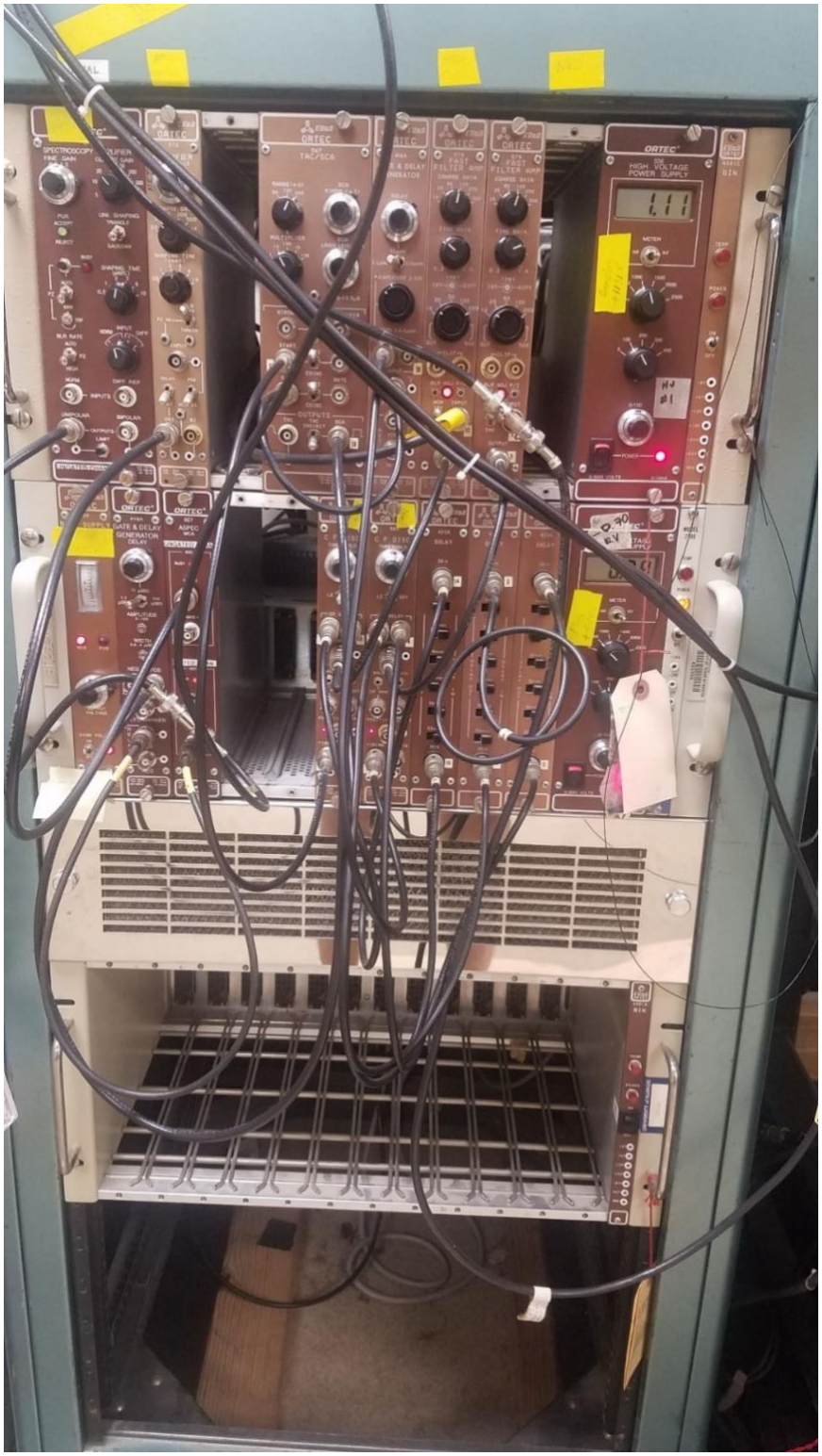


Figure 27: The NIM bin used for detector signal processing.

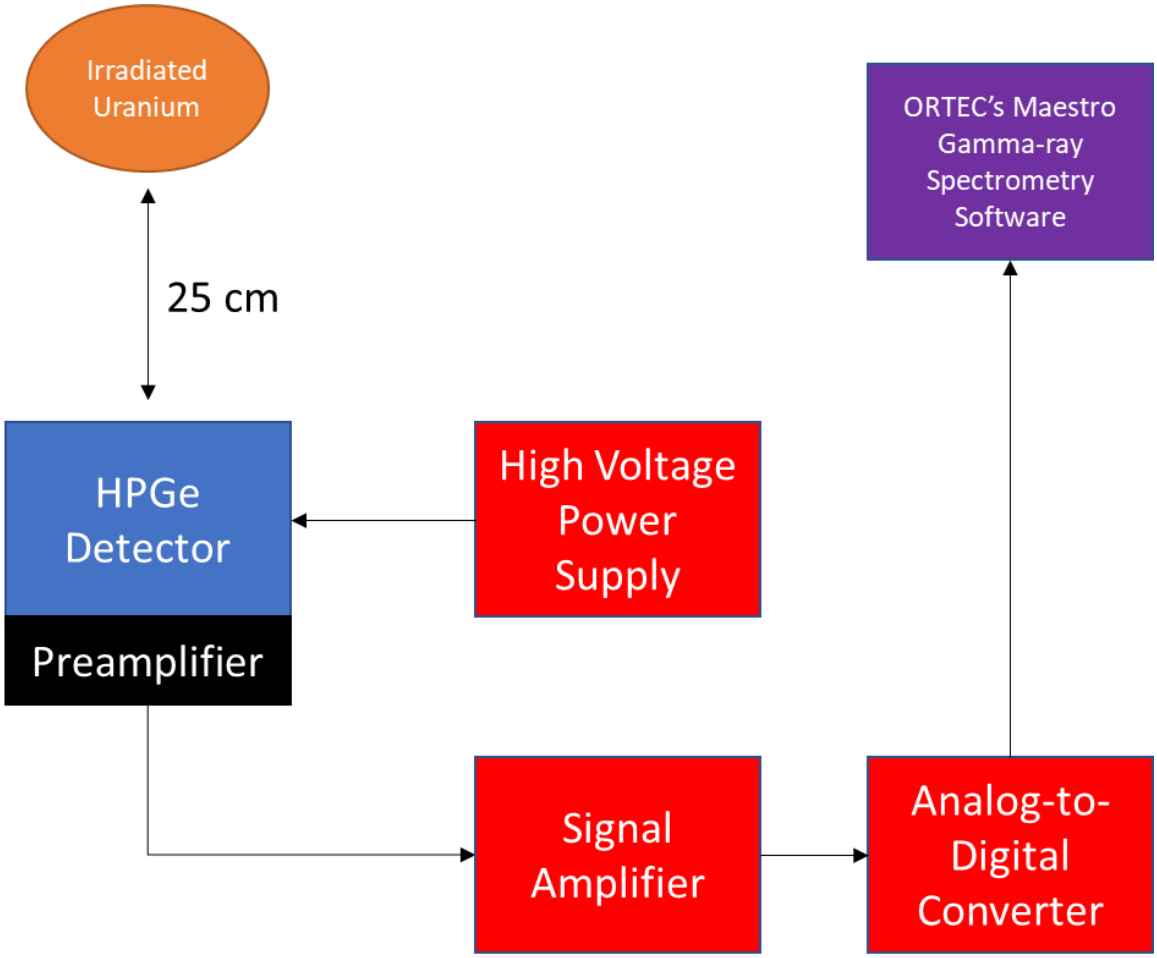


Figure 28: The full circuitry of the gamma-ray spectrometry system used in this experiment.

I created a Maestro job file which collects 1,000 sequential gamma-ray spectra of 180 seconds of real time each. The total dataset for a given sample thus represents 50 hours of continuous gamma-ray spectrometry measurement, discretized into 3-minute time bins. These spectra were saved in the `.spe` text file format. I chose this data acquisition protocol due to its post-processing flexibility: adjacent spectra can be easily combined in order to improve the counting statistics of individual gamma-ray spectra at the expense of total dataset size. Later, I will explore this tradeoff for the determination of the optimum time discretization that results in highest model performance. These results will also inform on future measurement campaigns or attempts to implement this method into another real-

world system by providing insight into what counting times are adequate, and how large of a dataset is necessary for high model performance.

Data processing for machine learning

I wrote a python script to handle the processing of all the experimental data. The text files have a basic structure consisting of: a fixed number of rows of header information, 8,192 rows representing the number of counts contained in each of the gamma-ray energy channels, and a fixed number of rows of footer. This consistent file structure made it easy to generate a pair of `numpy` arrays, `X` and `y`, that match the structure of the simulated data, and include all the data contained in the 7,000 individual gamma-ray spectrum text files. Each observation, or gamma-ray spectrum, is stored in one row of `X`, and the label corresponding to the enrichment of the sample is stored in the corresponding row of the `y` array.

However, the dimensionality of the full experimental dataset is 7,000 observations and 8,192 features per observation. Due to constraints on the processing power available to me, and the complexity of model training as a function of both the number of training observations and the number of features, I coupled into this data processing pipeline a function which generates coarser discretization of both the number of features and the number of observations. These adjustments correspond to decreasing the total number of gamma-ray energy bins and increasing the real time corresponding to each gamma-ray spectrum, respectively. In this way, I retain all of the data contained in the original dataset but can store it in an array of more compact dimensionality, thus decreasing the processing time for model fitting substantially. While this retains all of the data, it does sacrifice some information. Increased computing capabity would potentially allow for use of the full dataset. For the main analysis of experimental data, I utilized a spectrum collation factor of 5 and a gamma-ray energy bin collation factor of 64. This means that each observation in the final dataset corresponds to a data acquisition period with a (180*5) 900 second live time, and (8,192/64) 128 gamma-ray energy bins. Of course, there exists a tradeoff between having a more detailed dataset with more observations, thus potentially building superior

models, versus having a dataset that is small enough to train models in a reasonable timeframe. Next, I consider another method of data truncation: principal component analysis for dimensionality reduction.

Principal component analysis for experimental dataset truncation

As an alternative to degrading the energy resolution of the individual gamma-ray energy bins, I performed principal component analysis on the full 8,192 bin datasets. PCA was performed with the scikit-learn function `sklearn.decomposition.PCA`[2]. This function uses singular value decomposition to project the full dataset onto a lower dimensional space. SVD is implemented with the randomized truncated singular value decomposition as implemented by Halko et al. (2009)[34]. I chose to create a 10-dimensional PCA representation of the initial dataset, which will be shown to account for over 99% of the inherent variance in the original dataset. The performance of models generated on resolution-increased and PCA-truncated datasets is then compared.

At this point, the experimental dataset is stored in the pair of `numpy` arrays `X` and `y` that is identical to the format of the simulated ORIGEN data after the enrichment blending step. All further steps in the preparation of the experimental data for machine learning are identical to the process described above for the simulated data. The one notable exception is the lack of data replication across different irradiation times. All experimental data in this work has a fixed t_{irr}.

We have now discussed the preparation of both the synthetic and experimental datasets. Next, we will discuss the procedure for model construction and evaluation.

Model training and tuning via cross-validation and grid searches

At this point the data are stored in `numpy` arrays X_{train}, X_{test}, y_{train}, and y_{test}, where `X` are the gamma-ray emission data across all energy bins (columns), and across all training/testing instances (rows), and `y` are the enrichment value labels associated with each row in the corresponding `X` array. I use the training data to construct models using each considered algorithm and tune the hyperparameters of the models using grid searches and cross-validation. An example of this process is shown in Figure 29. Here, I tune a ridge

regression model to find the optimum value for the regularization parameter. I use five-fold cross-validation, which means that the training data are broken into five subsets. For each regularization parameter value considered, a model is trained on four out of the five subsets, and evaluated on the final "holdout" set. This process is repeated five times for each regularization parameter value. Figure 29 shows that a regularization parameter value of 15 results in models that have the highest cross-validation score across all splits. This figure plots the negative mean absolute error in predictions for each of the 5 cross-validation sweeps at each tested regularization parameter alpha. I retain the convention of plotting the *negative* of the MAE to preserve the notion that higher=better, thus aiding visual intuition.

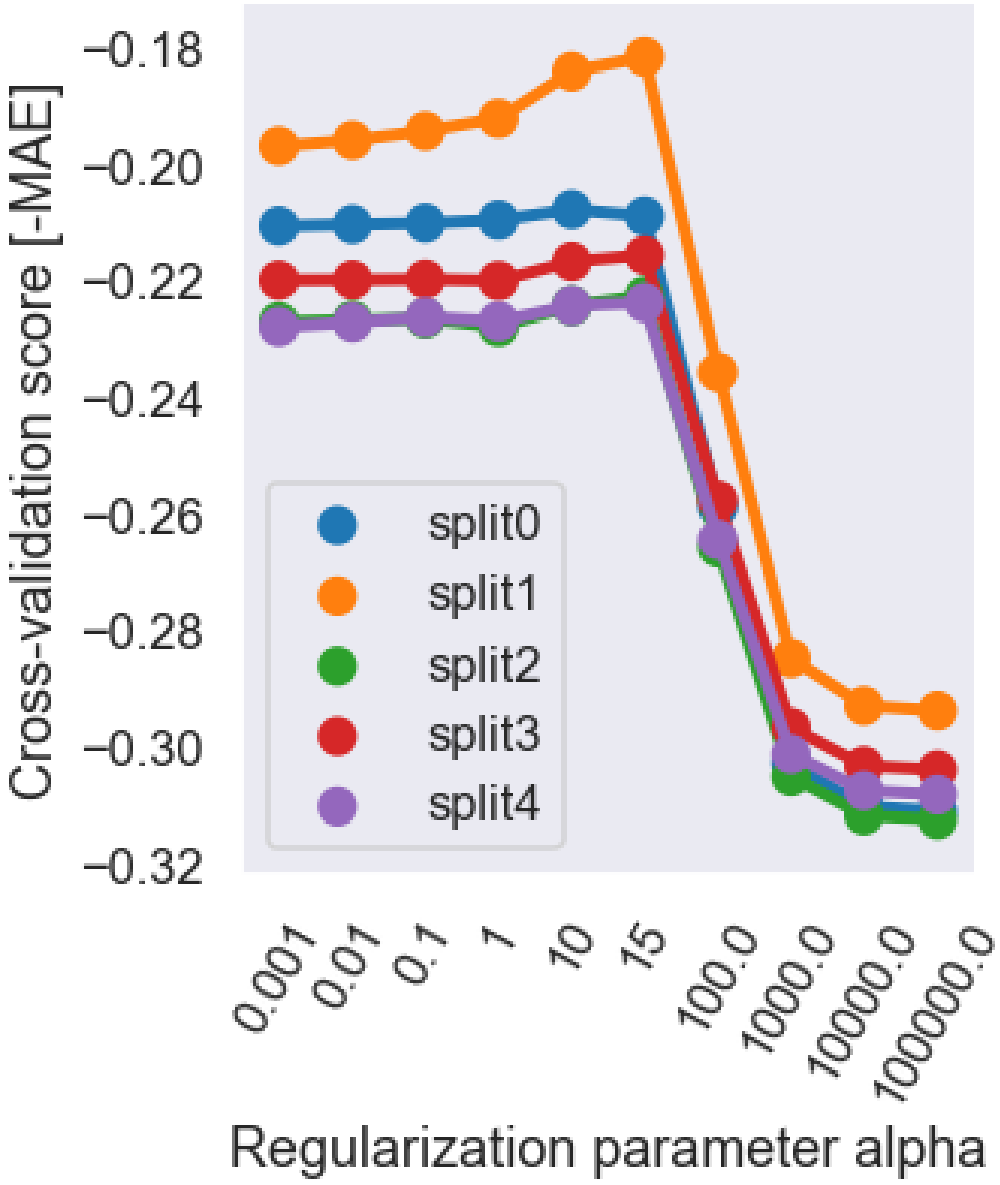


Figure 29: An example of the cross-validated grid search method for determining the optimum regularization parameter value for a ridge regression model with five-fold cross-validation.

For other algorithms, such as decision trees, there are many more hyperparameters to tune than just one regularization parameter. This results in a higher-dimensional grid search with multiple hyperparameters to simultaneously tune. The grid search considers

every possible combination of considered hyperparameter values. Figure 30 shows a slice of the data contained in a single high-dimensional grid search for a decision tree based on grid searching over two hyperparameters simultaneously. Creating visualizations while performing these grid searches is essential to understand the effects that these combinations of hyperparameters have on model performance. For instance, Figure 30 shows that there is an optimum decision tree configuration located at the hyperparameter coordinates of (10, 5) for these two hyperparameters.

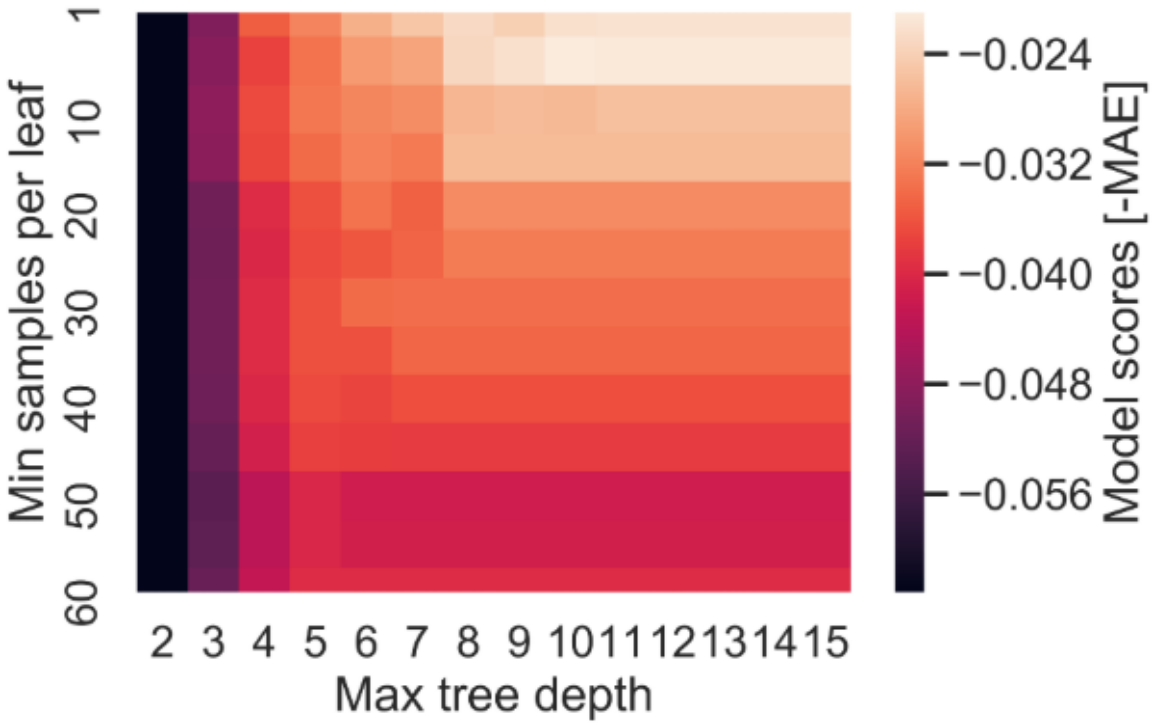


Figure 30: The [-MAE] scores for a decision tree regressor grid search over two hyperparameters. The best model uses a max tree depth of 10 and a min samples per leaf of 5.

MODEL EVALUATION AND COMPARISON

Next

, I use the optimum hyperparameters as determined by the cross-validated grid searches to build the final models with each algorithm. This is done with a single training fit of each model to the entire training dataset. Then, each model performs predictions on the test set

X_{test} which consists of new data that has not been seen by models during either hyperparameter tuning or training. I generate summary statistics of the performance of each model by comparing the predictions on the test set to the true labels y_{test} including: the distribution of absolute errors in prediction, the mean absolute error in prediction, and the standard deviation of the absolute errors in prediction. I repeat this process for each algorithm, across the synthetic data for all five sets of irradiation times, with fixed and variable training set sizes, and on the experimental data with fixed t_{irr}.

Chapter 6: Synthetic Data Results

The two datasets that I generated in this work each answer a different set of questions. Because it is comparatively trivial to generate large amounts of simulated data versus experimental data, the simulated dataset includes an extra dimension of variation in t_{irr} than the experimental data. The questions which I answer with the simulated data include:

- Is it feasible to predict uranium enrichment based off gamma-ray emissions if the decay time of the specimens are unknown?
- Which machine learning algorithms yield the best performing models?
- What is the effect on prediction performance if the dimensionality of the unknown parameter space of the dataset is increased? In particular, what if the irradiation time of the specimens is no longer fixed, but allowed to vary between two, three, four, and five values?
- How much data is really needed to train effective models? What is the effect on prediction performance if the fraction of the total dataset used for training is decreased?

SIMULATED DATA WITH IRRADIATION TIME FIXED

The first set of experiments uses a simulated ORIGEN dataset with the following properties: 7 enrichment levels, 1-hour irradiation time, 180 decay times linearly spaced up to 18 days post-irradiation, and 100 gamma-ray energy bins. Because the structure of the dataset is fixed in this section, the purpose of this section is to compare the prediction performance of each different machine learning algorithm. Results consist of performance metrics and selection of important gamma-ray energy features. Figure 31 below shows a density distribution of the absolute errors in prediction on the test set for each of six models after tuning and fitting on the training set.

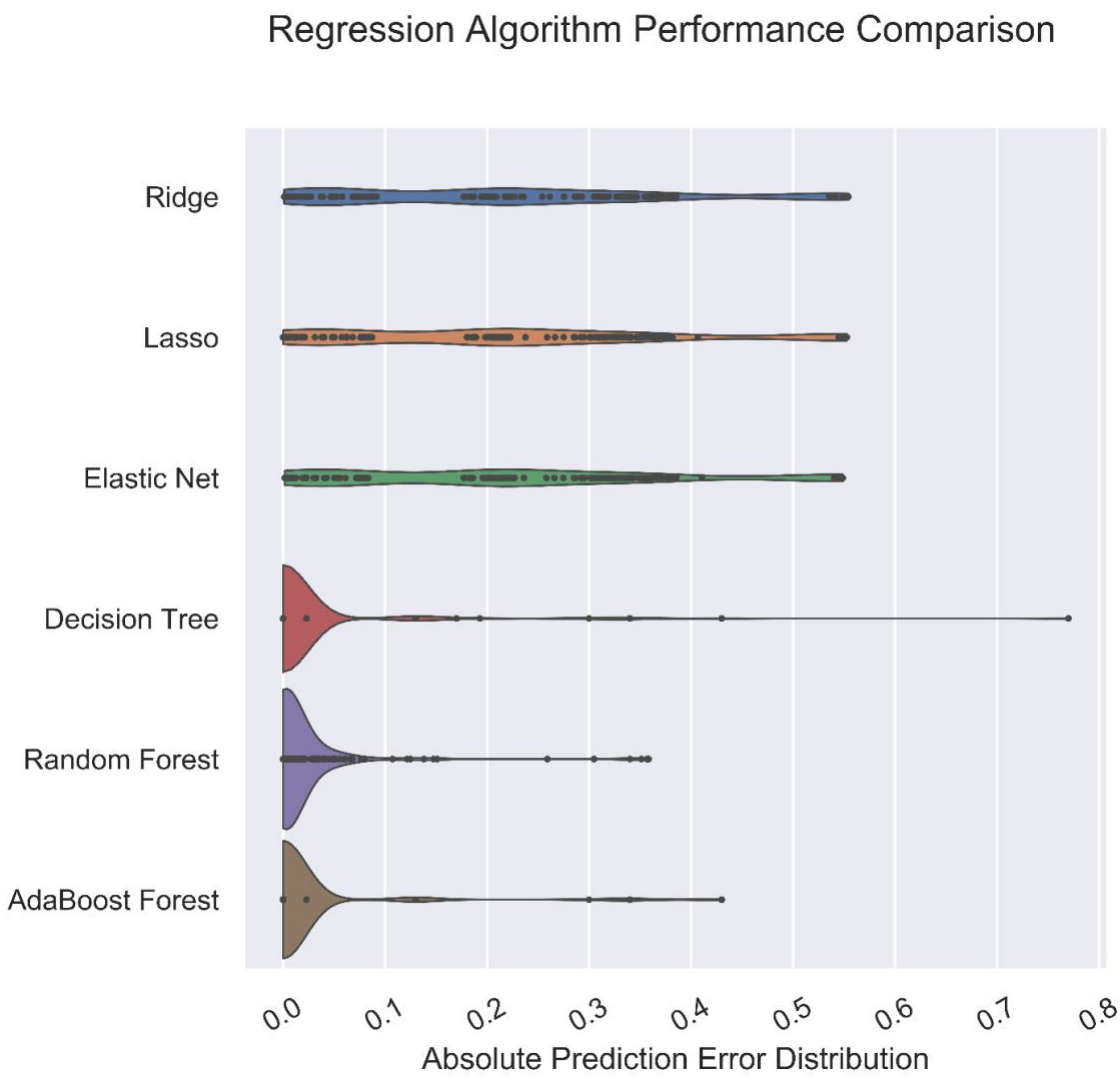


Figure 31: The distributions in absolute prediction error of uranium enrichment for each machine learning algorithm evaluated on the synthetic data test set with fixed tirr.

These results are replicated in Table 4. In general, the regularized linear models do not have performance competitive with the decision tree based models. However, all three decision tree based models had mean absolute errors less than 2%. In particular, the adaptively boosted decision tree ensemble had the best performance in this scenario, with a MAE score of only 1.7±7.0%.

Table 4: The performance summary statistics for each machine learning algorithm on synthetic data.

Model	Mean absolute error	Standard Deviation
Ridge	0.216	0.168
Lasso	0.216	0.171
Elastic Net	0.216	0.170
Decision Tree	0.019	0.076
Random Forest	0.018	0.051
Adaptive Boosting	0.017	0.070

Figure 32 shows the best single decision tree model. Left branches correspond to samples whose evaluation at the node was true, and right branches correspond to false samples. There are a few interesting features of this decision tree to note. Several leaf nodes nearly perfectly predicted many specimens after evaluating only a few features. For instance, many of the uranium samples enriched to 97% ^{235}U were among the easiest samples to predict. Tracing the leftmost branch of the decision tree, only a single feature was necessary to make these predictions: $\mathbf{X_{96}}$ (corresponding to gamma-rays in the energy bin 99 keV to 124 keV). Of the 1,152 samples used to train this model, the 112 samples with an $\mathbf{X_{96}}$ value of 0.014 or less were predicted to have an enrichment of 97% with a MAE score of 0.006. On the rightmost branch of the decision tree, three features ($\mathbf{X_{96}}$, $\mathbf{X_{88}}$, and $\mathbf{X_{74}}$) correctly predicted most of the lowest enriched samples corresponding to: depleted, natural, and 3% enriched uranium. For reference, Table 5 shows the upper bounds of each gamma-ray energy bin $\mathbf{X_i}$. Intermediate enrichment values were determined via the middle branches of the decision tree and require greater branch depth.

Table 5: The upper boundaries of the 100 gamma-ray energy bins.

X_i	E_{max} [MeV]	X_i	E_{max} [MeV]	X_i	E_{max} [MeV]	X_i	E_{max} [MeV]
0	2.50	25	1.88	50	1.26	75	0.644
1	2.48	26	1.86	51	1.24	76	0.619
2	2.45	27	1.83	52	1.21	77	0.594
3	2.43	28	1.81	53	1.19	78	0.569
4	2.40	29	1.78	54	1.16	79	0.545
5	2.38	30	1.76	55	1.14	80	0.520
6	2.35	31	1.73	56	1.11	81	0.495
7	2.33	32	1.71	57	1.09	82	0.470
8	2.30	33	1.68	58	1.06	83	0.446
9	2.28	34	1.66	59	1.04	84	0.421
10	2.25	35	1.63	60	1.02	85	0.396
11	2.23	36	1.61	61	0.990	86	0.371
12	2.20	37	1.58	62	0.965	87	0.347
13	2.18	38	1.56	63	0.941	88	0.322
14	2.15	39	1.54	64	0.916	89	0.297
15	2.13	40	1.51	65	0.891	90	0.272
16	2.10	41	1.49	66	0.866	91	0.248
17	2.08	42	1.46	67	0.842	92	0.223
18	2.05	43	1.44	68	0.817	93	0.198
19	2.03	44	1.41	69	0.792	94	0.173
20	2.01	45	1.39	70	0.767	95	0.149
21	1.98	46	1.36	71	0.743	96	0.124
22	1.96	47	1.34	72	0.718	97	0.099
23	1.93	48	1.31	73	0.693	98	0.074
24	1.91	49	1.29	74	0.668	99	0.050

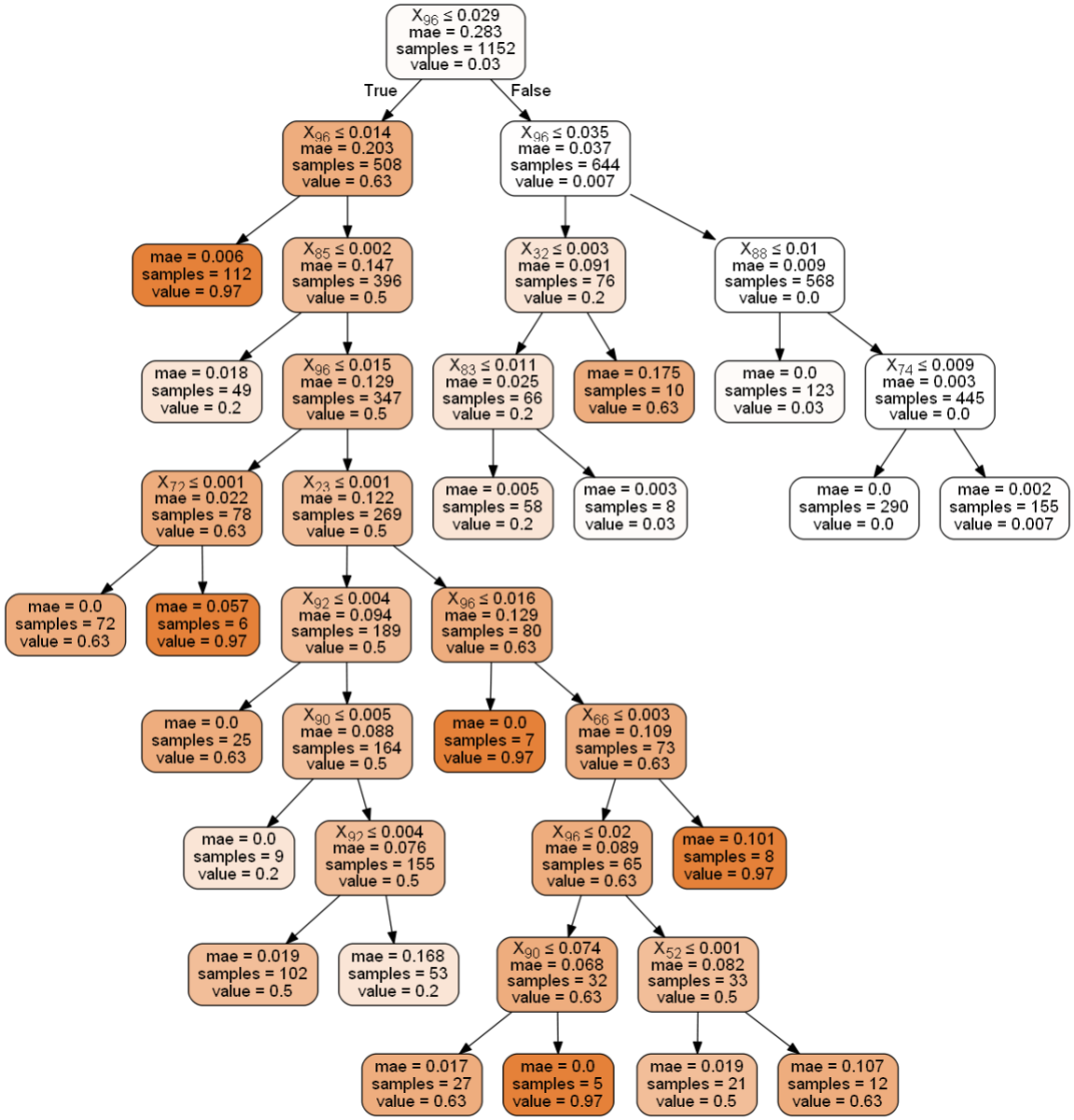


Figure 32: A representation of the decision tree model which had the best predictions on the synthetic data fixed t_{irr} test set, with a MAE score of 1.9±7.6%.

Next, let's compare the performance of each of the six machine learning algorithms. First, we will look at the three regularized linear models. Plots of the feature importances and learning curves as determined by each of the regularized linear models are shown in Figure 33. All models unanimously selected the energy bin 99-124 keV as the most important

feature in this dataset. In fact, the Lasso and Elastic Net models *only* selected this feature (due to the tendency of L_1 norms to induce sparse models with a small number of important features).

Next, we compare the decision tree model to the ensemble methods based on decision trees: random forests and adaptive boosting. The plots of feature importances and learning curves for the three decision tree based models are shown in Figure 34. All three models agree that the most important feature is again the gamma-ray energy bin 99-124 keV. The next most important feature as determined by all three models is 371-396 keV. The resolution of the gamma-ray energy bins in ORIGEN is not sufficient for us to analyze which fission products are responsible for the main emission in each of these energy groups. However, during experimental measurements, I will pay special attention to these energy regions, and I will use the gamma-ray spectra to determine which fission products are responsible for emissions in these "important" regions.

I have also generated the learning curves corresponding to each of these six algorithms to demonstrate the fitting and generalization performance of each model. All models have learning curves in which the training and cross-validation scores gradually converge and approach nearly the same value. This indicates strong generalization performance.

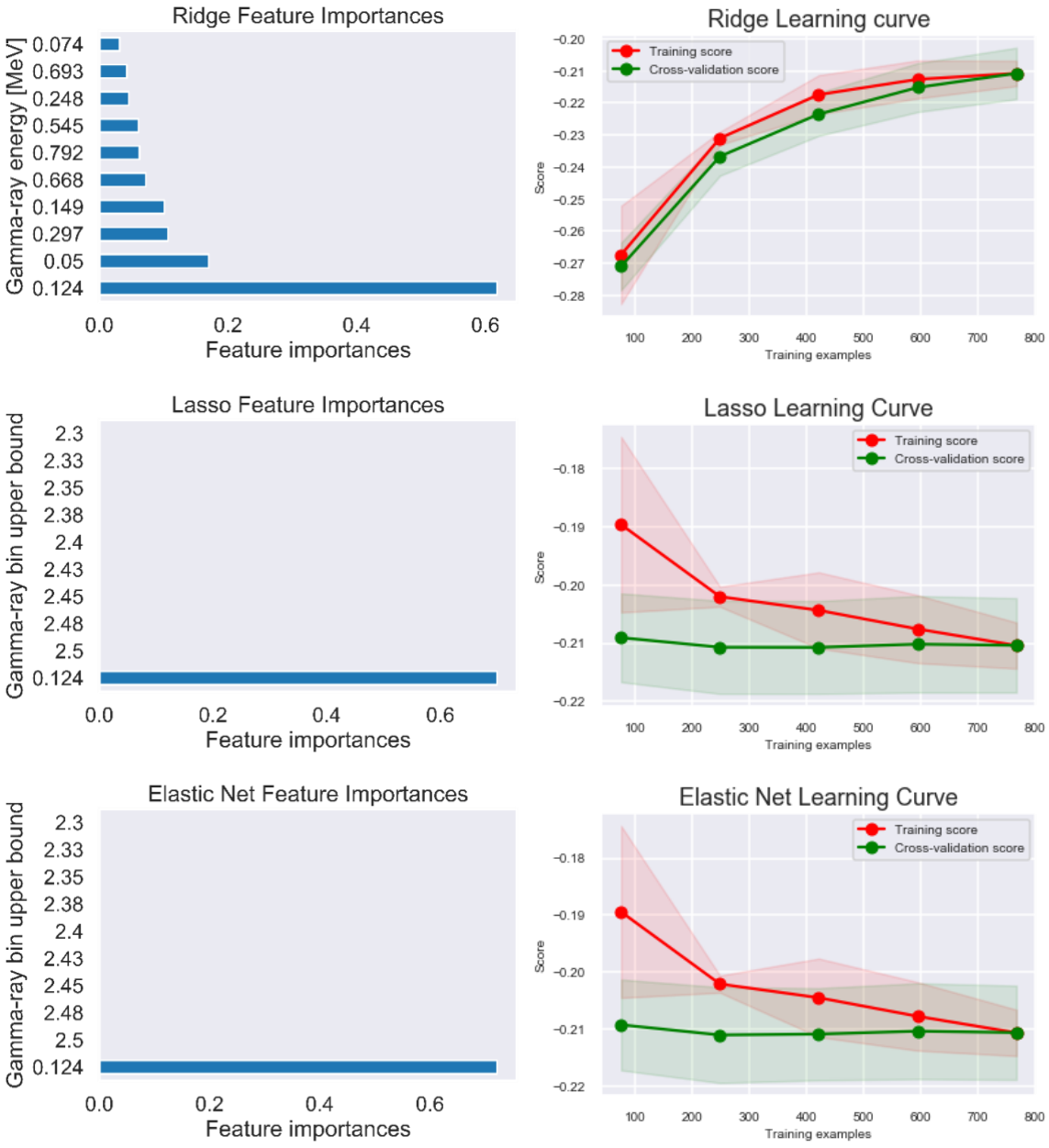


Figure 33: The feature importances and learning curves for the three regularized linear models of synthetic data with a fixed t_{irr}.

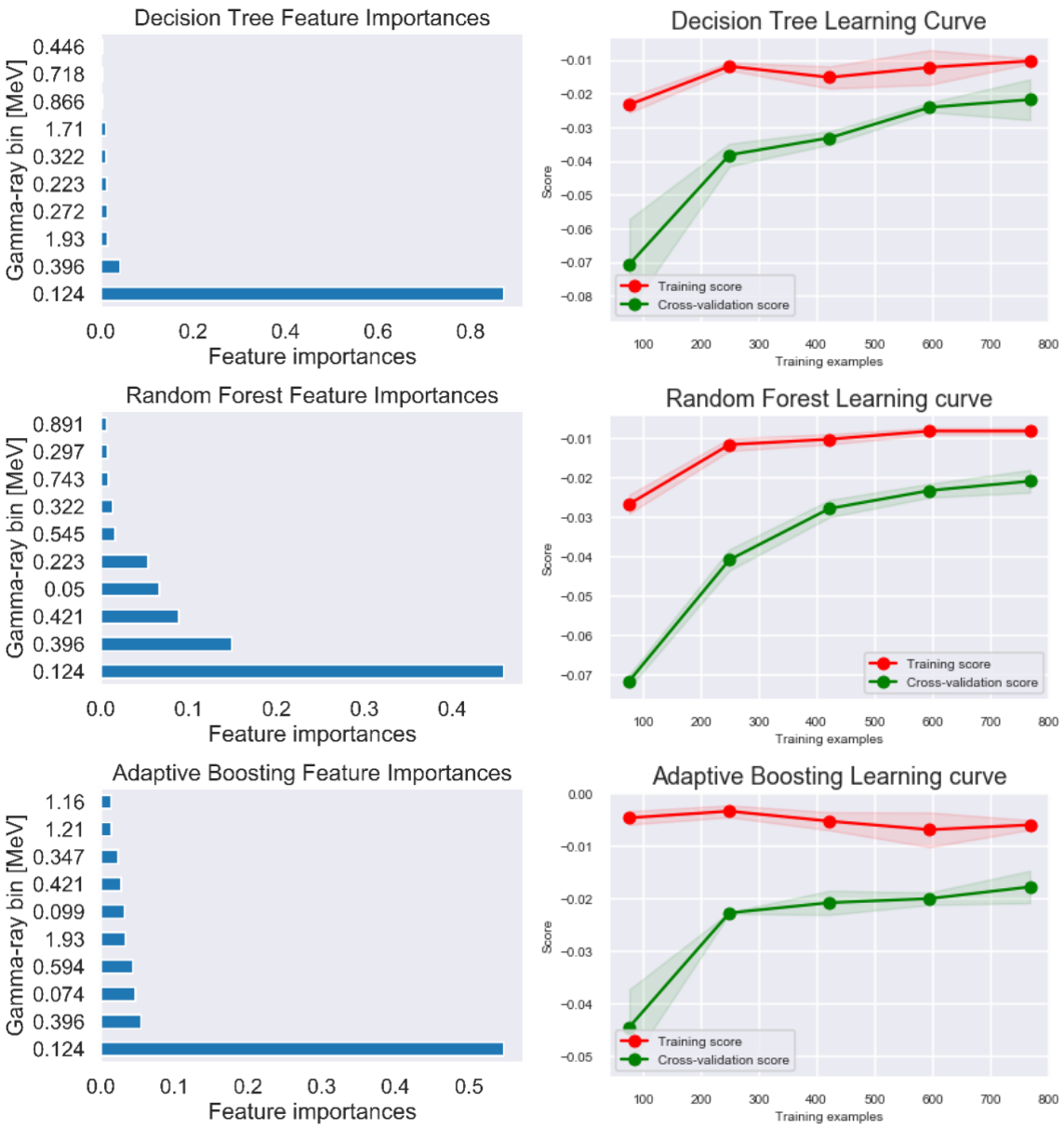


Figure 34: The feature importances and learning curves for: the decision tree model, random forest, and adaptive boosting of synthetic data with a fixed t_{irr}.

In summary, the results from these initial simulations demonstrate that decision tree based models are capable of performing reliable enrichment predictions without knowledge of the precise decay time of samples. Mild further improvements in performance can be obtained by utilizing ensemble methods with decision trees as the base estimator. Regularized linear models do not perform particularly well, however, they are

capable of determining the important features of the dataset and are shown to be in agreement with the decision tree based models.

SIMULATED DATA WITH VARIABLE IRRADIATION TIME

This section demonstrates the robust generalization performance of the predictive models. Here we look at two sample algorithms: ridge regression and decision trees and study the performance of these models as the size of the training set is varied, and as mismatches between the training and testing sets are introduced. These experiments help determine how much data is necessary for effective model construction and will test the abilities of these models to generalize to datasets which are inherently outside of the domain on which they were trained. Figure 35 shows the nine distinct combinations of training and testing datasets that were compared.

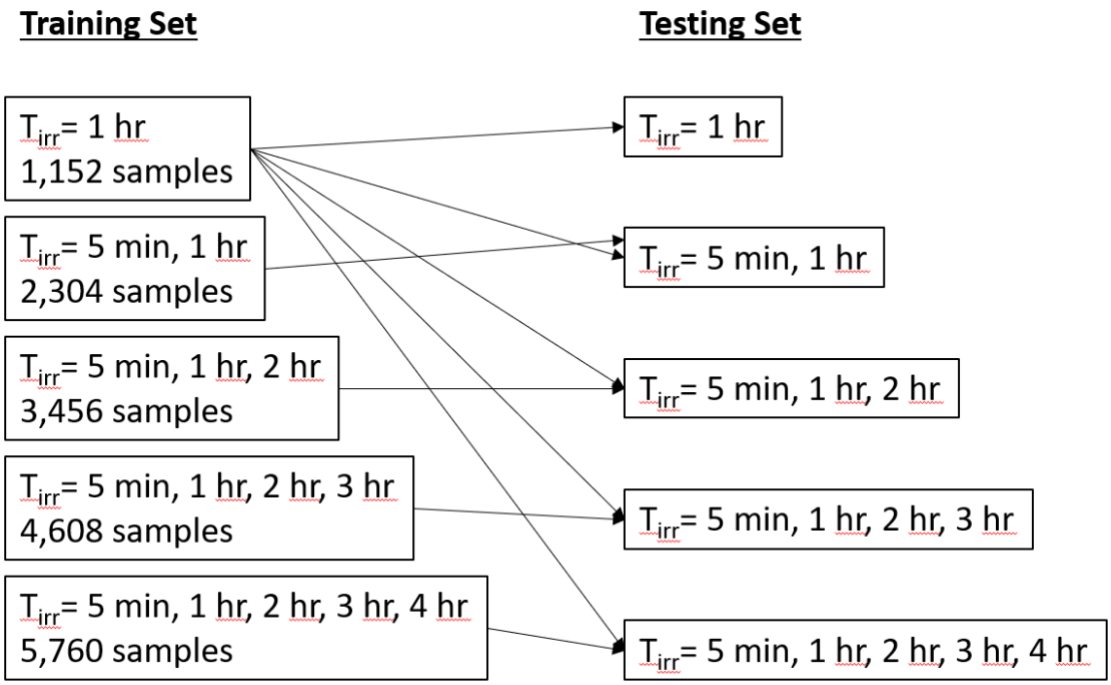


Figure 35: The nine combinations of training and testing datasets that were compared.

The prediction performance of each of these algorithms (ridge regression and decision trees) across all nine training/testing scenarios is shown in Figure 36 and Figure 37. These results are significant because there is not a clear trend in prediction performance as either: mismatches between training and testing sets are introduced, or: as the number

of considered t_{irr} values increases. The performance of these models does not degrade as the dimensionality of the dataset is increased or when the training set represents only a slice of the test dataset. The nearly constant prediction performance of each of these models across all of these variations demonstrates that these models have strong generalization performance and are not overfitted to the exact conditions on which they were trained.

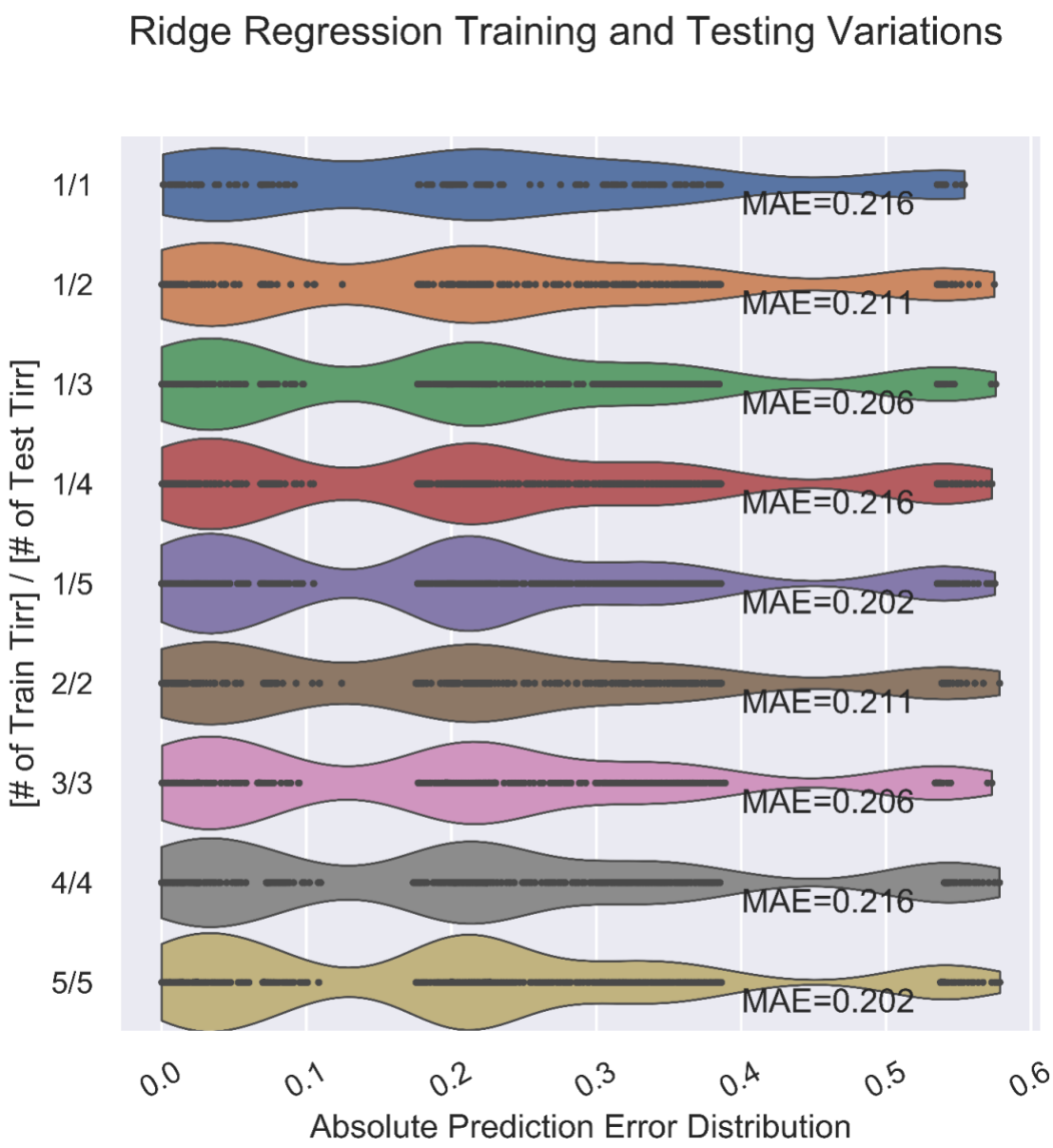


Figure 36: The distribution of absolute errors in prediction when varying the number of irradiation times considered in both the training and testing datasets of a ridge regression model.

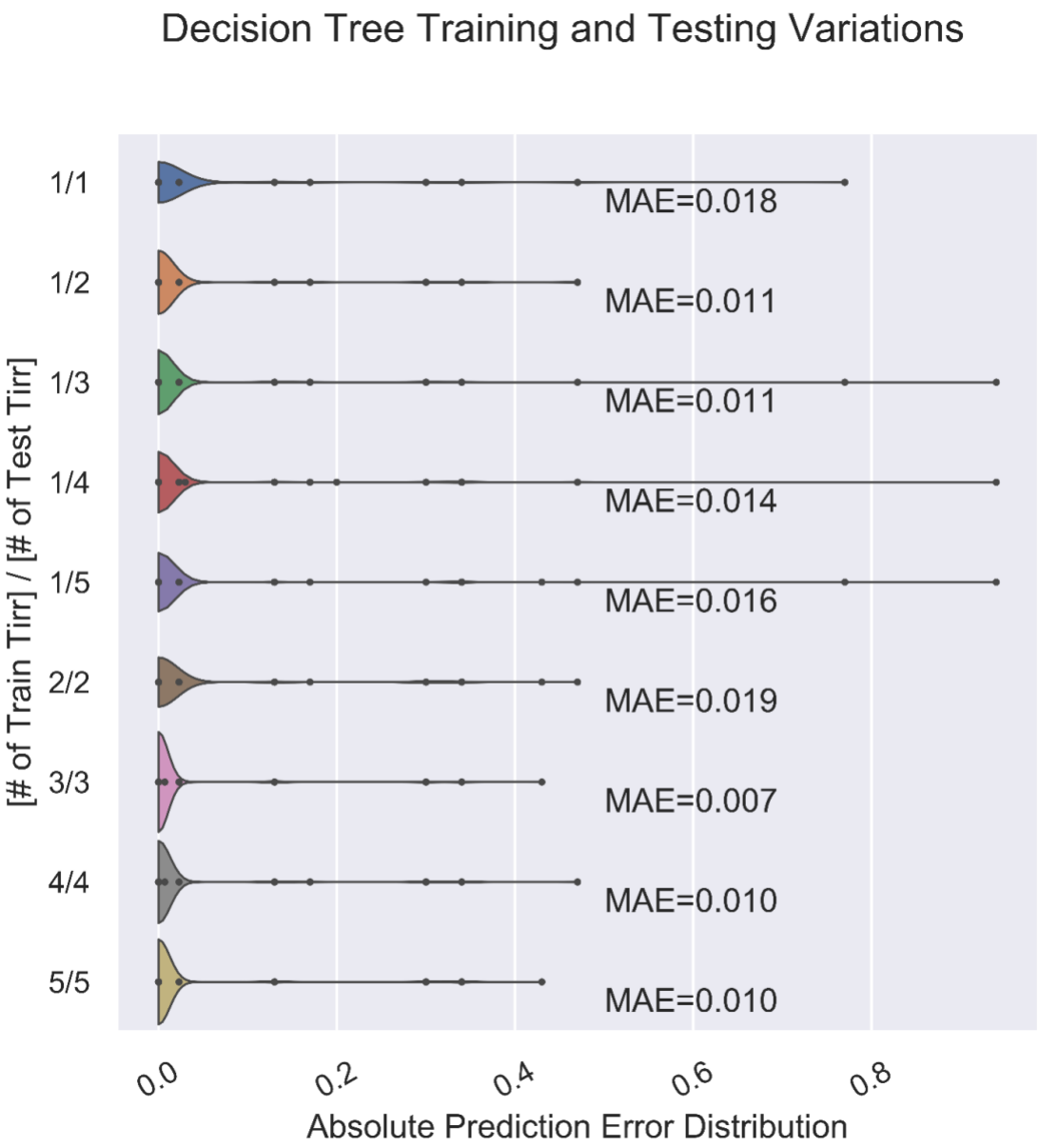


Figure 37: The distribution of absolute errors in prediction when varying the number of irradiation times considered in both the training and testing datasets of a decision tree model.

At this point, the simulated data have successfully demonstrated that robust models can be trained in these conditions. As such, we move on to experimental measurements and consider a dataset in which the irradiation time is held fixed.

Chapter 7: Experimental Measurement Results

By comparison, collecting experimental data is more laborious and time consuming than running sequential simulations in ORIGEN. Therefore, the experimental dataset lacks variation in irradiation time and will instead focus on determining uranium enrichment with decay time invariance. The questions that the experimental data answer include:

- How large are the prediction errors for enrichment determination with real-world data as compared to the simulated data for each machine learning algorithm?
- What are the most important gamma-ray energies as selected by each model in a gamma-ray measurement spectrum?
- What are the tradeoffs in prediction performance associated with decreasing the total dataset resolution in terms of energy bin discretization and gamma-ray spectrum time discretization to improve model training time (i.e. having 1,000 3-minute gamma-ray spectra (observations) versus 100 30-minute gamma-ray spectra with which to construct models, or having 8,192 energy bins (features) versus 128 energy bins)?
- Does principal component analysis effectively reduce computational time while preserving prediction performance?

The full set of experimental measurements is shown in Figure 38. This figure shows the seven full 50-hour gamma-ray spectra associated with each of the tested enrichment values. As expected, there is a direct correlation between the total number of counts in each spectrum and the enrichment associated with that spectrum. This trend demonstrates the importance of data normalization – we do *not* want raw count rates to inform on the ultimate decision returned by models because we are mostly interested in the shape of the spectra, and we want to eliminate other potential sources of error such as variable sample masses.

Recall that the dimensionality of this dataset is 8,192 gamma-ray energy bins and 7,000 3-minute observations. I have taken two approaches to reducing this dimensionality so as to speed up computation time and avoid having an excessive number of features relative to the size of the observation set. The first approach reduces each of these dimensions by summing adjacent gamma-ray energy bins and adjacent spectra with hyperparameters that control the number of each that are combined (the so called "down-binning "factors). The second approach utilizes principal component analysis to extract the top principal components, discard the rest, and construct models on a significantly reduced dimensionality dataset. These two approaches are discussed and compared next.

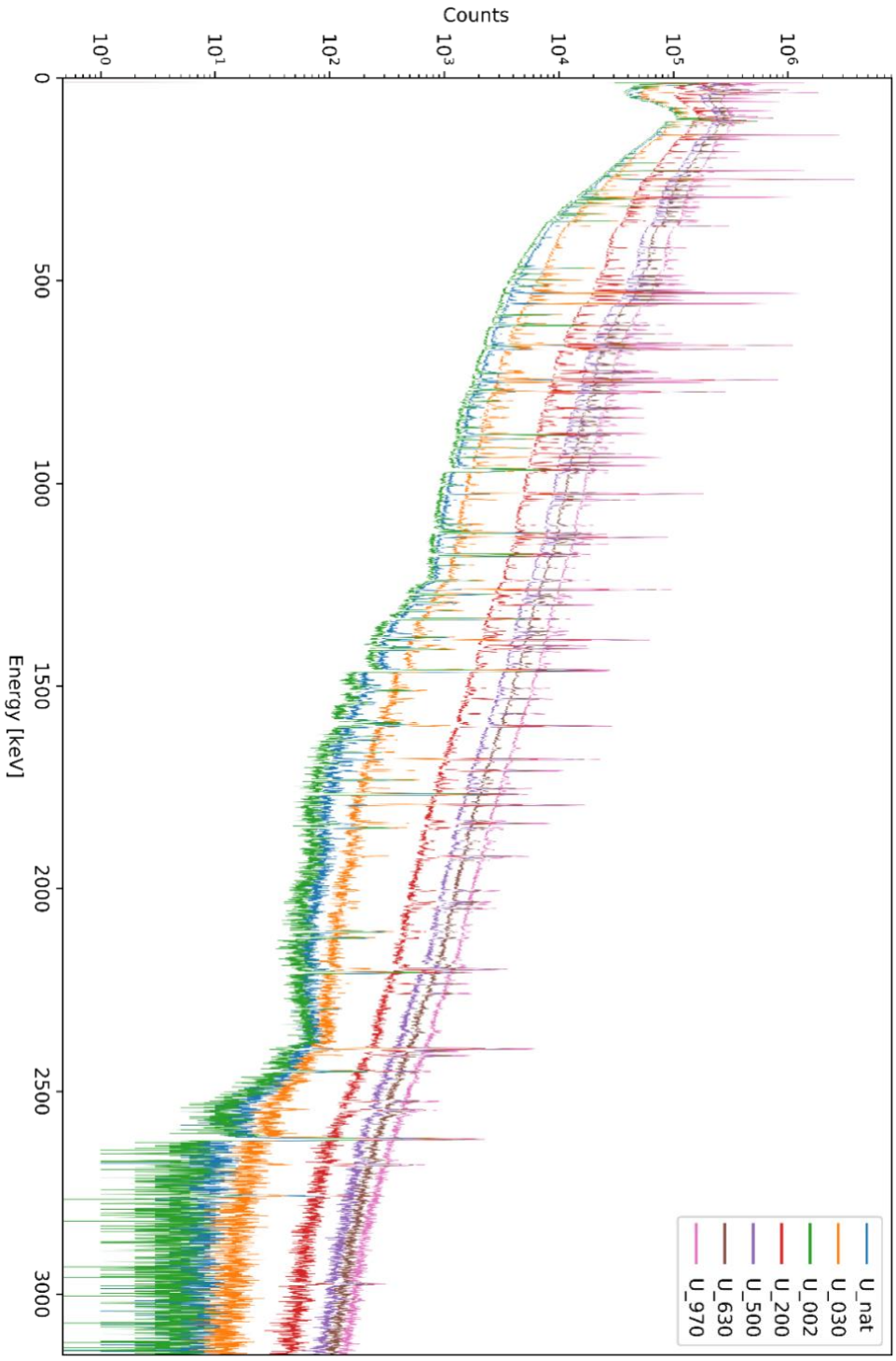


Figure 38: The full 50 hour gamma-ray spectrometry measurements corresponding to each of the seven irradiated uranium samples.

DOWN-BINNING METHOD FOR DATA TRUNCATION

The first method for data truncation relied on reducing the temporal and energy resolution of the dataset – these adjustments corresponding to increasing the real time of each gamma-ray spectrum and decreasing the number of energy bins contained in each spectrum, respectively. I utilized a spectrum down-binning factor of 5 and an energy bin down-binning factor of 64. This results in a new dataset dimensionality of: (7,000/5)x(8,192/64)=1400x128. This dimensionality is now of a comparable order to the synthetic dataset, facilitating relatively quick computation times and maintaining a sense of symmetry between the synthetic and experimental datasets. Figure 39 shows the absolute errors in prediction on the test set of models representing each of the six algorithms. These results are numerically summarized in Table 6.

There was a significant improvement in the performance of the regularized linear models when compared to the synthetic data: a factor of 4 decrease in MAE score. With MAE scores on the order of 5%, regularized linear models appear to be a useful tool for quick, computationally trivial estimations of enrichment.

However, the decision tree-based models still significantly outperform the regularized linear models. Again, the adaptively boosted forest had the highest performance. We can also see a manifestation of the bias-variance tradeoff associated with decision trees versus random forests. Decision trees have a tendency to overfit data. In this performance plot, that tendency to overfit results in a few very high error predictions – the highest errors out of any model in the group due to their high variance. On the other hand, when compared to decision trees, random forests have reduced variance but higher bias. This means that, on average, the predictions made by the random forest are not as strong as those of the decision tree. However, due to reduced variance, the random forest can avoid some of the very high error predictions made by the decision tree.

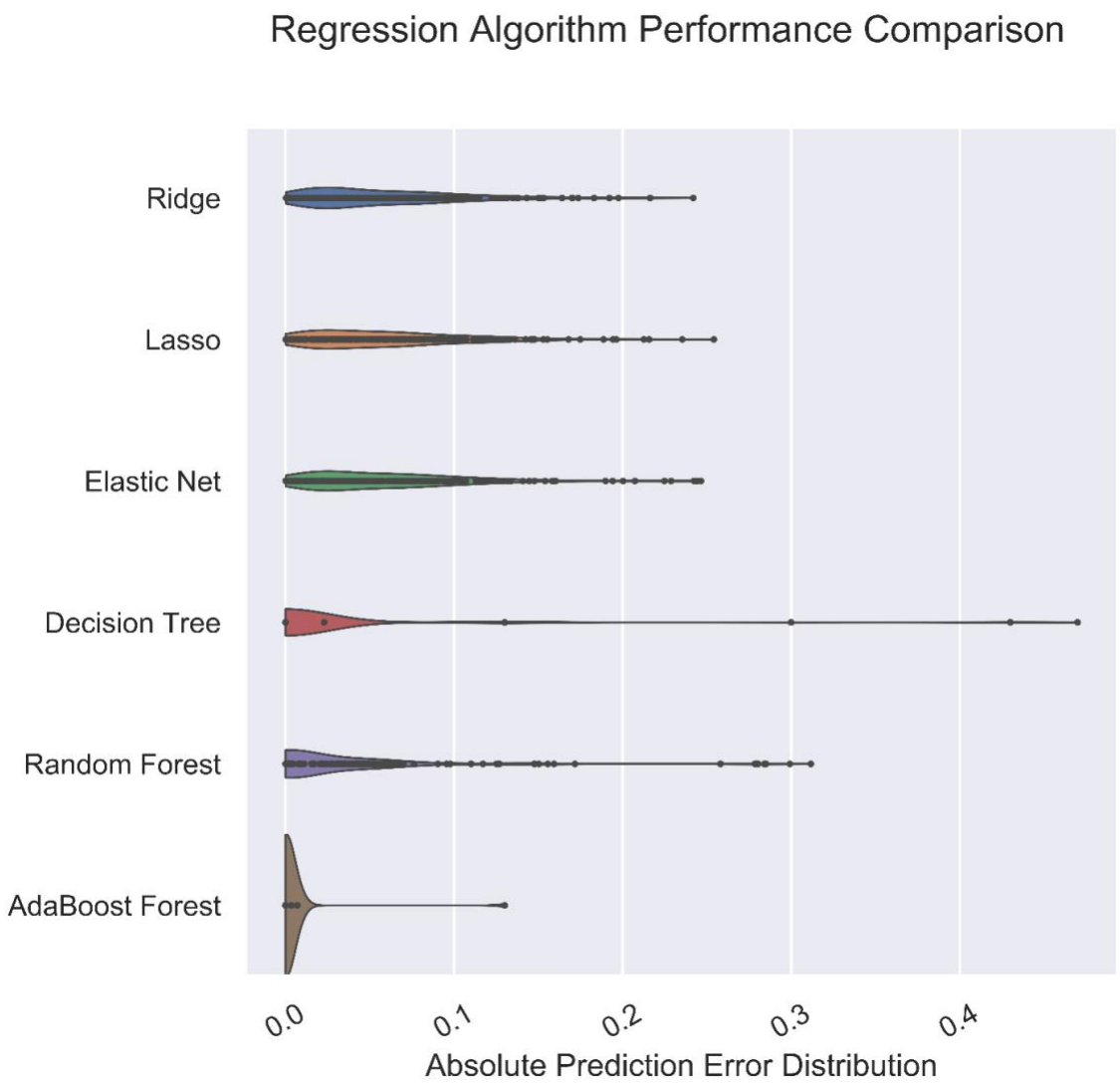


Figure 39: The distributions in absolute prediction error for each machine learning algorithm evaluated on the experimental data test set with energy discretization down-binning.

Table 6: The performance summary statistics for each machine learning algorithm on experimental data.

Model	Mean absolute error	Standard Deviation
Ridge	0.0566	0.0454
Lasso	0.0590	0.0467
Elastic Net	0.0587	0.0480
Decision Tree	0.0227	0.0818
Random Forest	0.0302	0.0577
Adaptive Boosting	0.0028	0.0155

The figure below shows the feature importance and learning curves for each of these six models on experimental data. Again, we see strong convergence of the training scores and cross-validation scores as the number of training examples for each model is increased – this indicates good generalization performance.

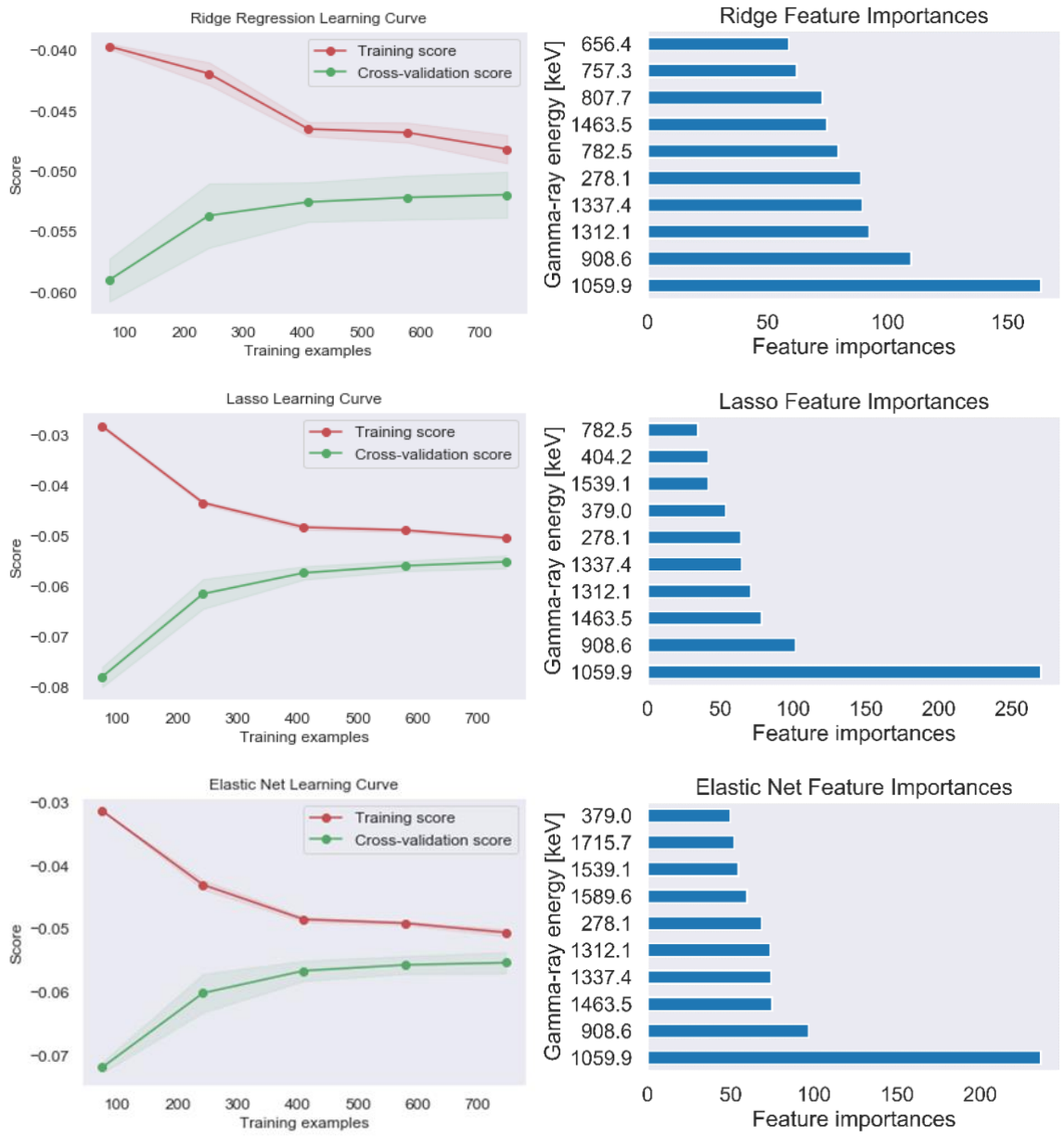


Figure 40: The learning curves and feature importances of the three regularized linear models for experimental data with energy discretization downbinning.

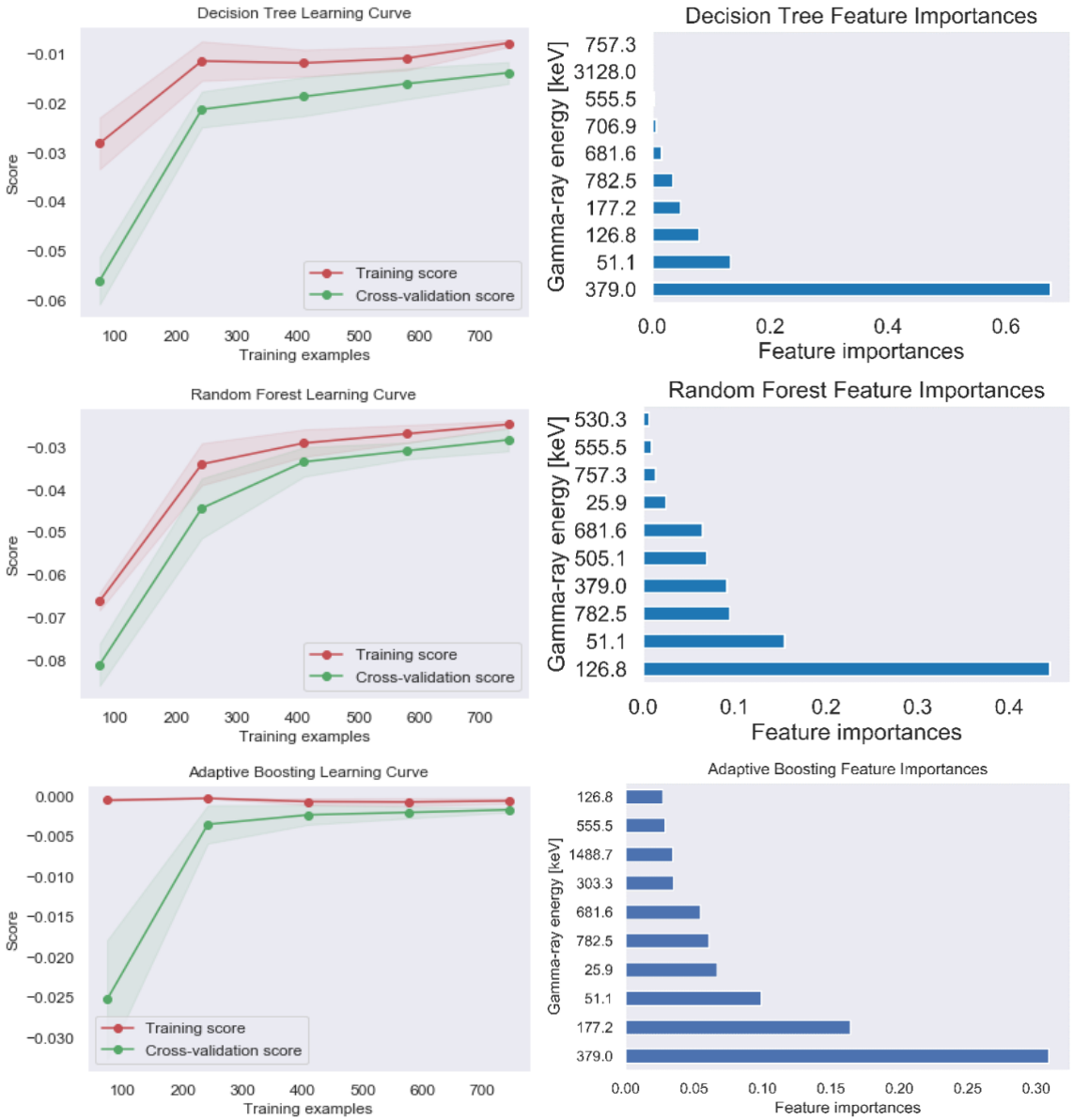


Figure 41: The learning curves and feature importances of the three decision tree based models for experimental data with energy discretization down-binning.

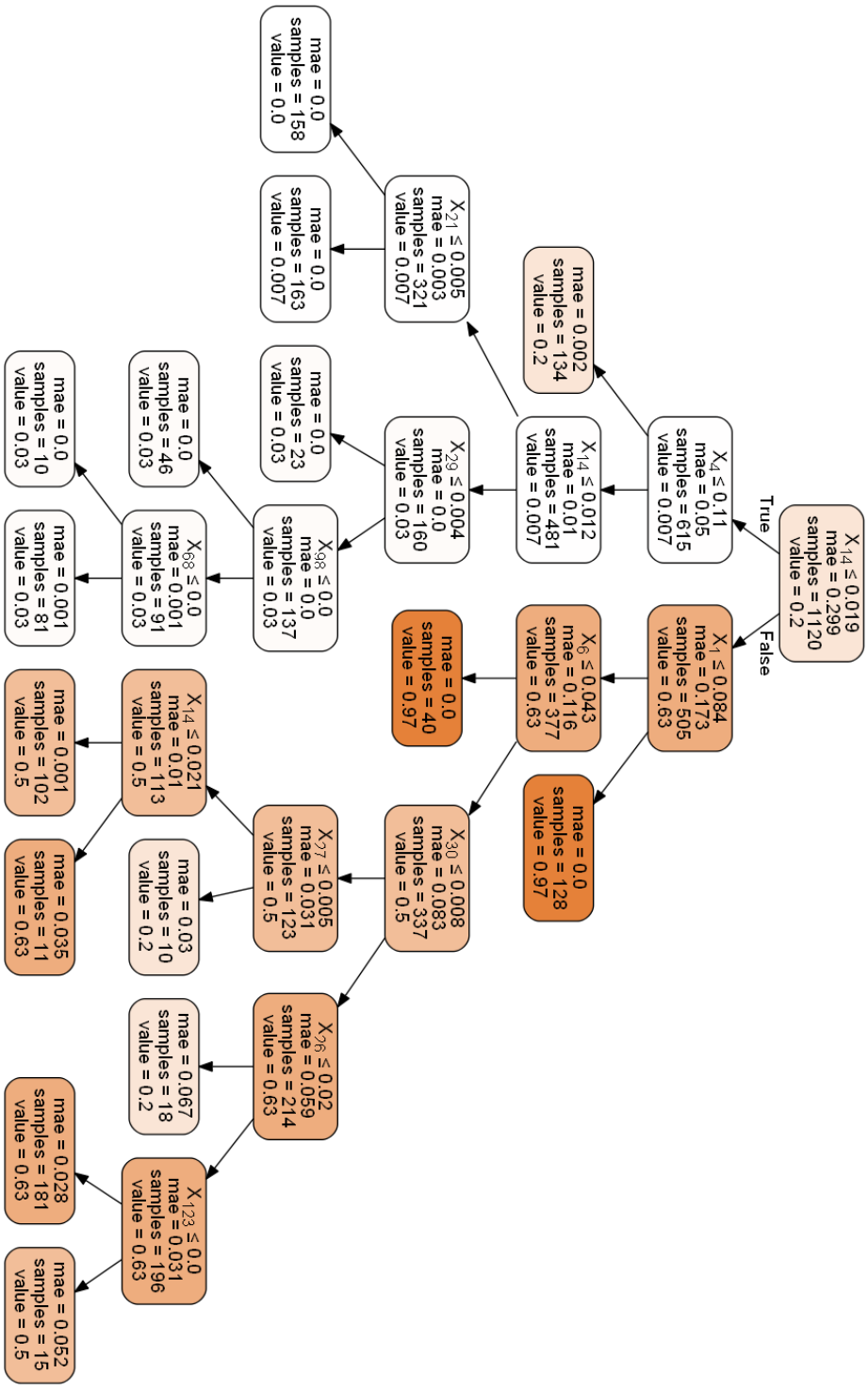


Figure 42: The decision tree structure based on the experimental dataset with energy discretization down-binning.

PRINCIPAL COMPONENT ANALYSIS FOR DATA TRUNCATION

The second method for data truncation utilizes a principal component representation of the initial dataset. As shown in Figure 43, the advantage of PCA representations of data is the significant dimensionality reduction that can be achieved while preserving nearly all of the underlying variance of the initial dataset. In Figure 43 we see that the first 10 principal components alone account for more than 99% of the variance in the initial dataset. As such, I selected the top 10 principal components of the data for this section, thus reducing the dimensionality of the feature space from 8,192 to 10.

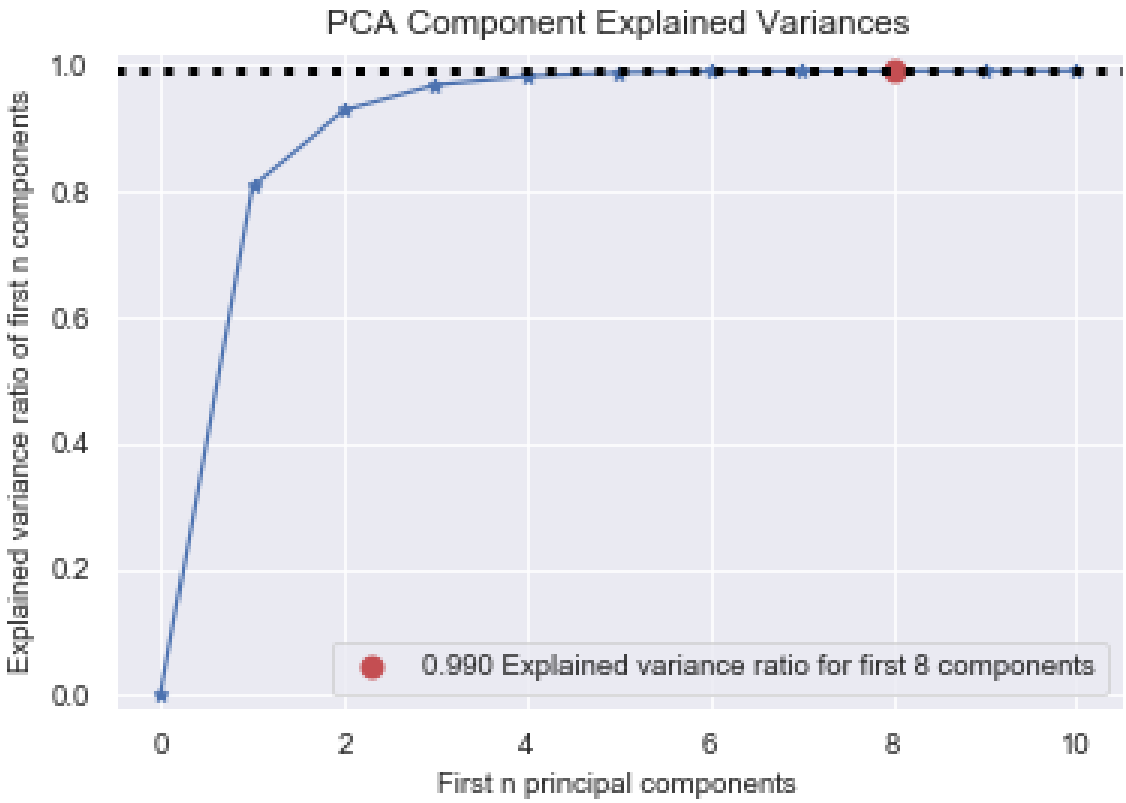


Figure 43: The cumulative amount of data variance explained by the first n principal components.

The top 10 prominent features in each of the top three principal components are shown in Figure 44. There are 17 individual features in this plot, indicating that there is substantial overlap in the most prominent features of the top three principal components. The gamma-ray energies and parent nuclides that this set of features correspond to include:

105 keV (^{239}Np), 140 keV (^{99}Mo), 228 keV (^{239}Np), 81 keV (^{133}Xe), 100 (^{239}Np) keV, and 250 keV (^{135}Xe). Recalling the method of Mandel et al. from Chapter 2, 105 keV and 140 keV were two of the gamma-rays that were chosen as representative of the ^{238}U and ^{235}U contents. The fact that my models have selected these gamma-rays as well is a promising indication that they have correctly identified the important signatures that are representative of each uranium isotope. Furthermore, the additional unique gamma-rays that my models have selected are quite reasonable: ^{99}Mo, ^{133}Xe, and ^{135}Xe are well-established as several of the most prominent fission products in short-lived fission product gamma-ray spectra.

Figure 45 shows the full representation of each of the top three principal components across the entire gamma-ray spectrum. The trends that we see here include: a strong emphasis on lower energy gamma-rays (~300 keV or less), with increasingly sparse important features as energy is increased, shared important features among multiple principal components but usually with opposite signs.

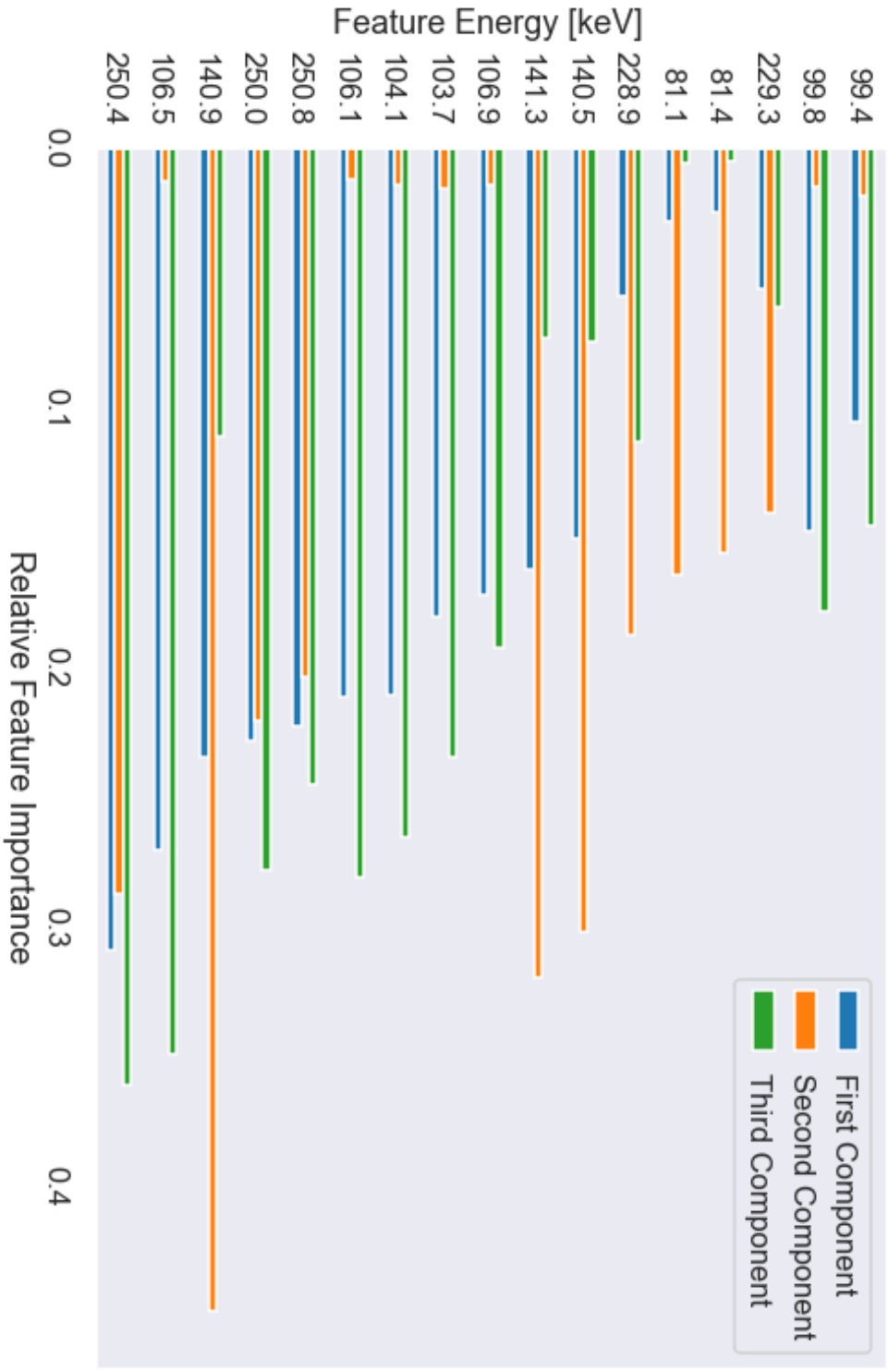


Figure 44: The top 10 most important gamma-ray energy features for each of the top three principal components of the experimental dataset.

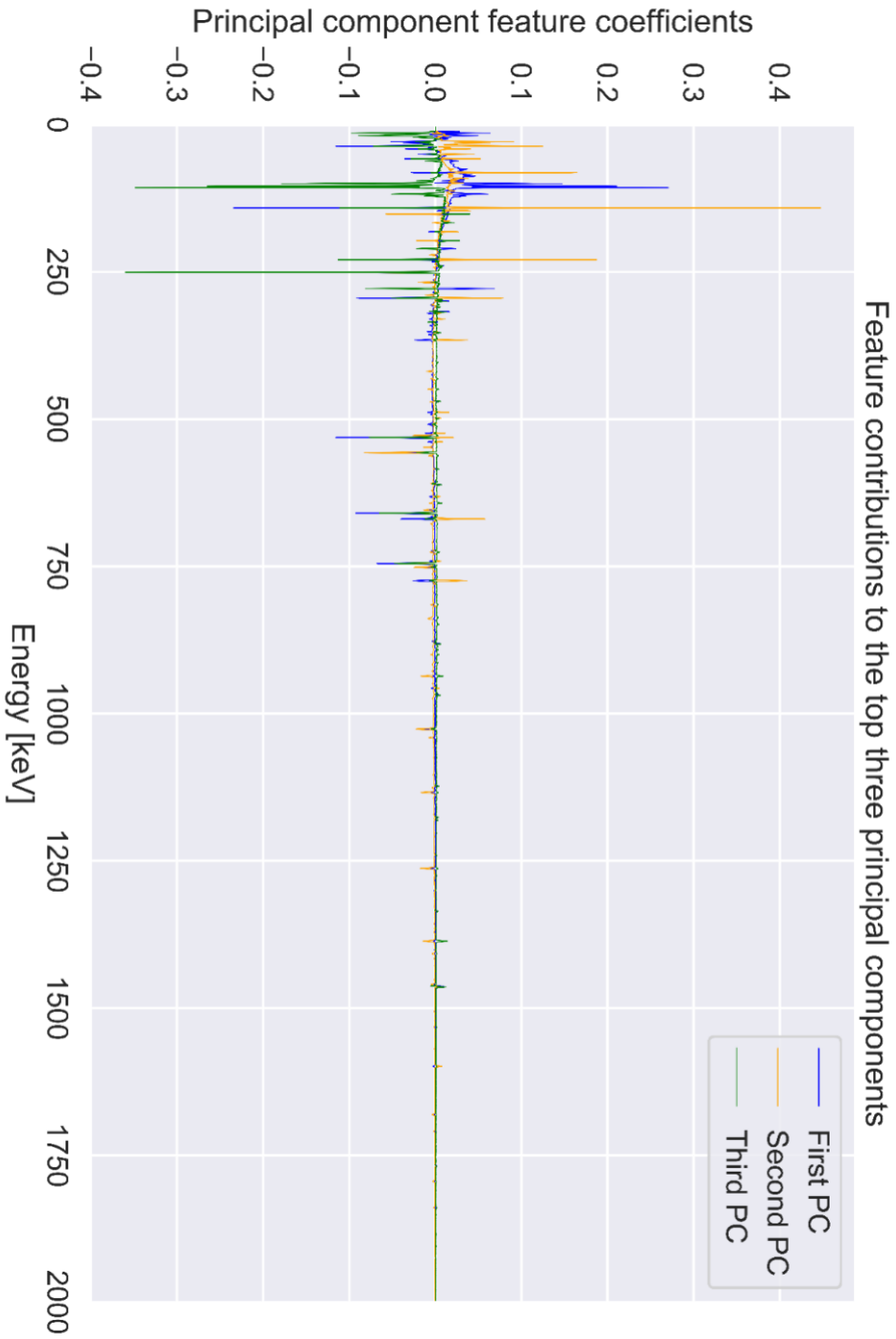


Figure 45: The weightings of each gamma-ray energy bin feature in the top three principal components.

Figure 46 shows the performance of each of the models built with PCA data. These results are numerically summarized in Table 7. Interestingly, the decision tree was the highest performing model – outperforming even the adaptively boosted forest of decision trees. Across the board however, the performance of all models with this PCA data is excellent and outperforms the models constructed with the down-binned dataset. I will also note that model tuning, fitting, and evaluating with the PCA data was usually on the order of 10 to 20 times faster than with the down-binned dataset. PCA is therefore a highly effective method for speeding up computation time, preserving, and potentially even improving model performance.

A few interesting anomalies exist in the prediction error plots shown in Figure 46. The decision tree model had essentially perfect prediction performance for all but a single outlier data point. The adaptively boosted forest had two outlier data points, one of which appears to have the same error as the decision tree. All three of these outliers correspond to measurements taken of 63% enriched uranium. The decision tree incorrectly predicted a single 63% enriched uranium sample as being 50% enrichment. The adaptively boosted forest predicted the same specimen as being 56.5% enriched, but also incorrectly predicted a second 63% enriched uranium sample as being 50% enrichment. These results suggest that these models occasionally run into challenges discriminating between intermediate levels of highly-enriched uranium. However, from a practical standpoint, this is of minimal concern. These models maintain perfect prediction capabilities for the low enriched uranium and the most highly enriched uranium. In practice, materials of either 63% or 50% enrichment would be treated similarly in a nuclear safeguards or nuclear forensics investigation.

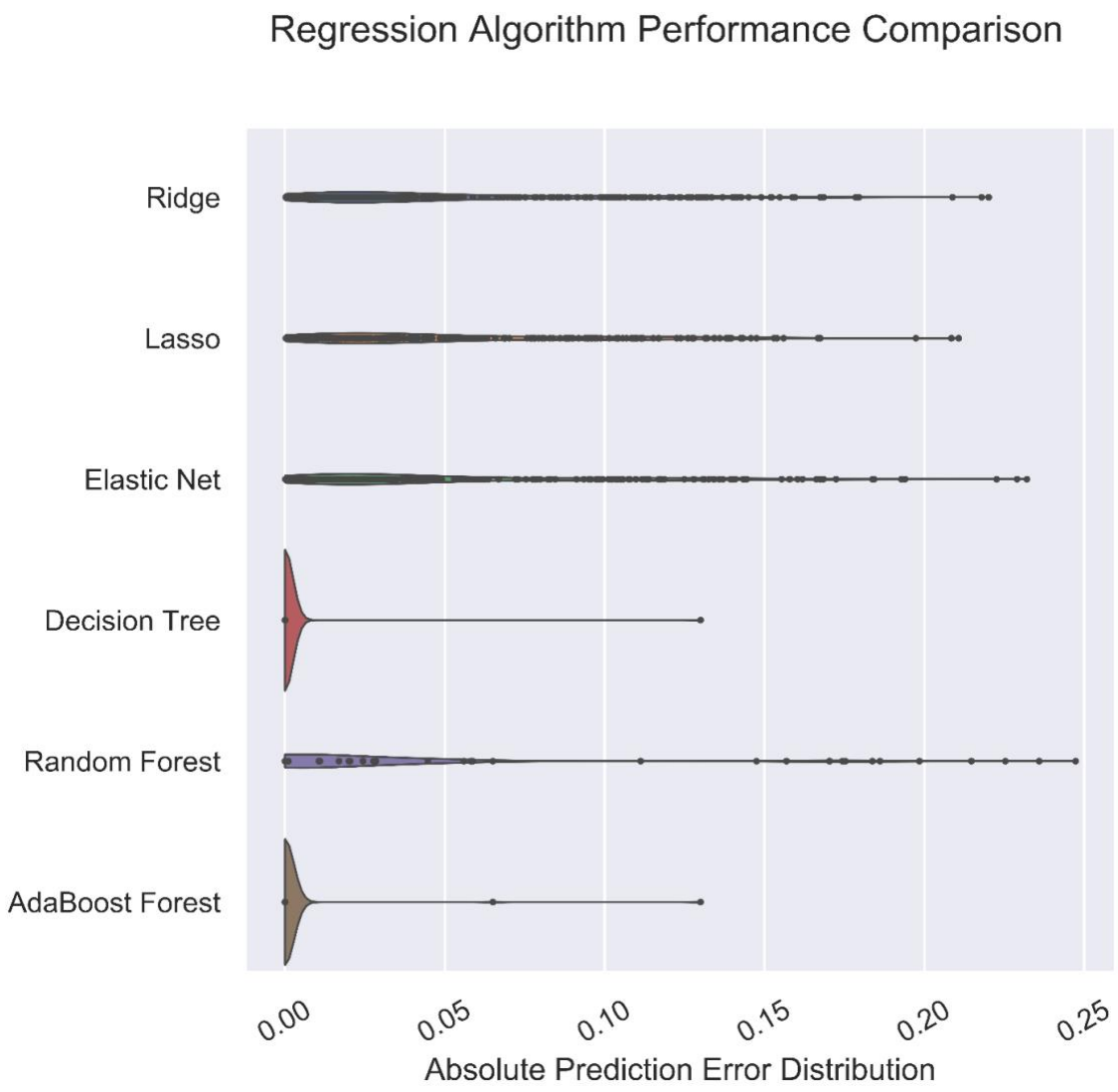


Figure 46: The distributions in absolute prediction error for each machine learning algorithm evaluated on the experimental data test set with principal component analysis preprocessing.

Table 7: The performance summary statistics for each machine learning algorithm on experimental data with principal component analysis preprocessing.

Model	**Mean absolute error**	**Standard Deviation**
Ridge	0.0530	0.0474
Lasso	0.0534	0.0452
Elastic Net	0.0529	0.0505
Decision Tree	0.0005	0.0078
Random Forest	0.0299	0.0546
Adaptive Boosting	0.0007	0.0087

Figure 47 shows the entire set of predictions made by each of these models on the test set broken down by target enrichment value. Here we can clearly see the single outlier prediction made by the decision tree and the two outlier predictions made by the adaptively boosted forest: all of these outliers occur at 63% enrichment.

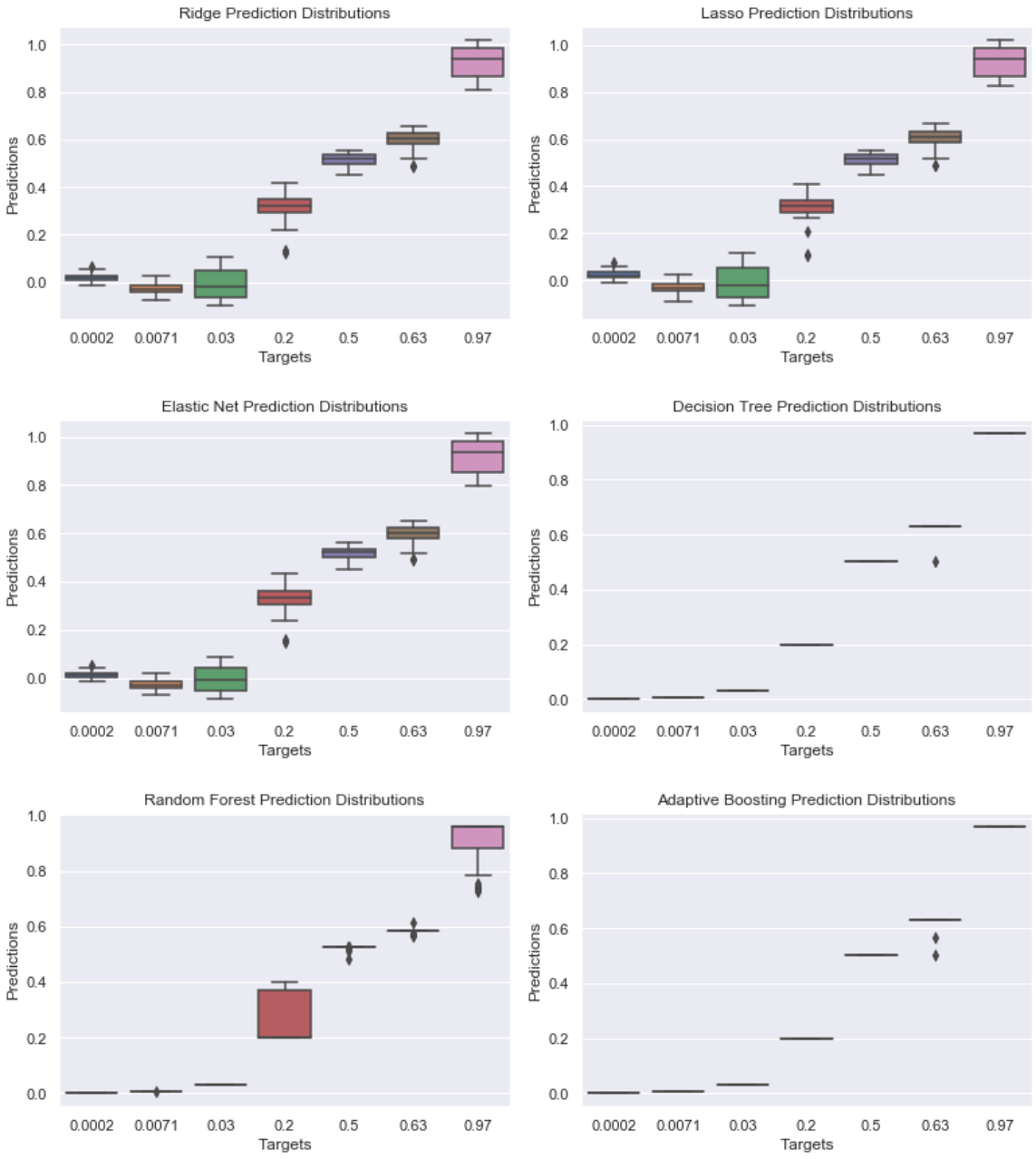


Figure 47: The predictions made by each model with PCA-truncated data on the test set broken down by target enrichment value.

Figure 48 shows the feature importances and learning curves for each of these six models on the PCA-truncated dataset. These models have excellent convergence between

the training and cross-validating curves across the board. This suggests that all of these models are well-fit. This is important because it demonstrates that the nearly perfect performance of these models is not the result of merely overfitting the training data. Rather, these models have successfully captured the inherent trends in the gamma-ray data which corresponding with enrichment.

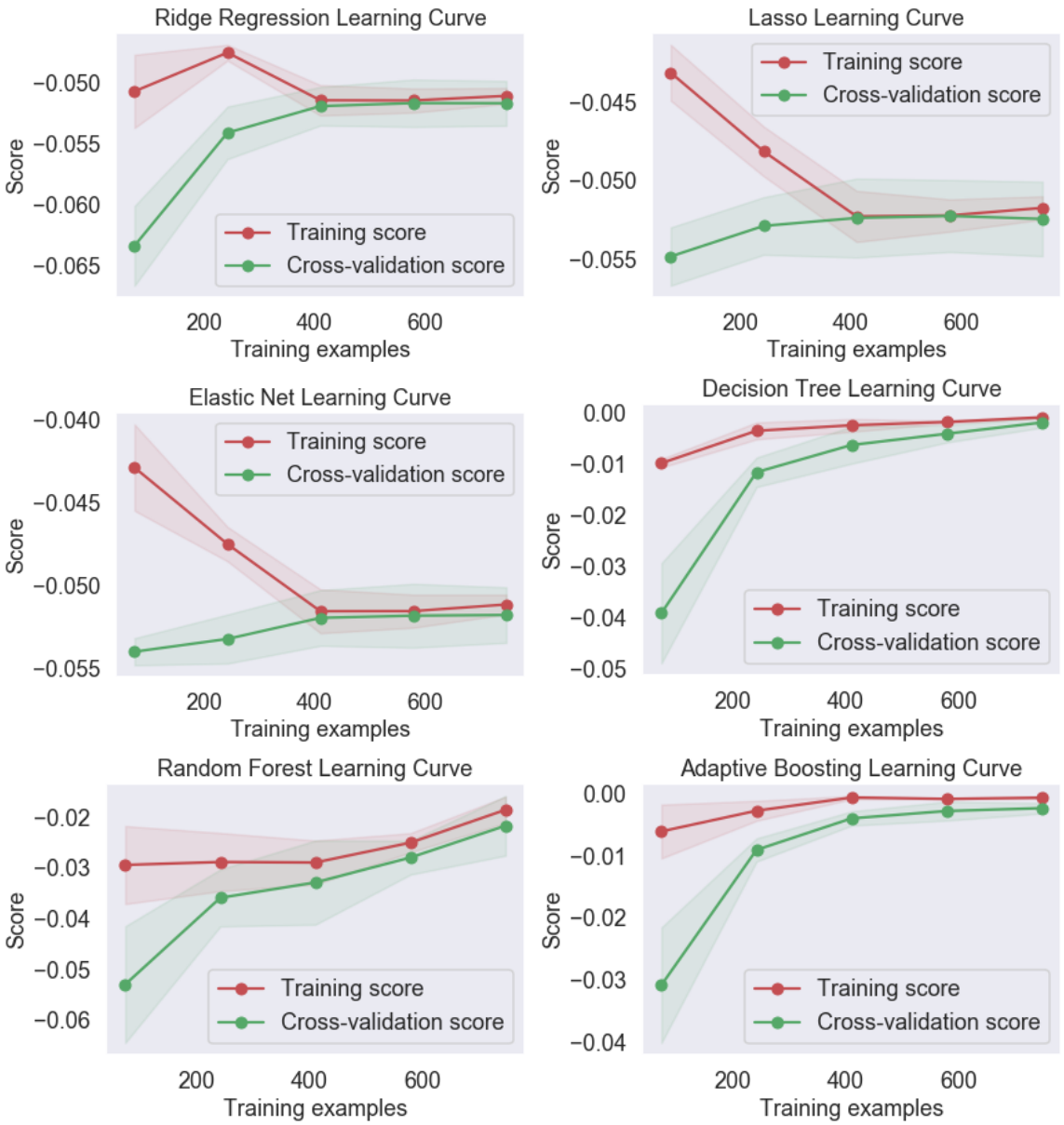


Figure 48: Learning curves for each of the six algorithms used on the PCA preprocessed experimental data.

Figure 49 shows the structure of the PCA-based decision tree. This decision tree has a strikingly simple structure while attaining the highest score out of any model developed thus far in this work: 0.05%. In general, we see that the ranking of the principal components matches the order in which they most strongly inform on the decision tree model.

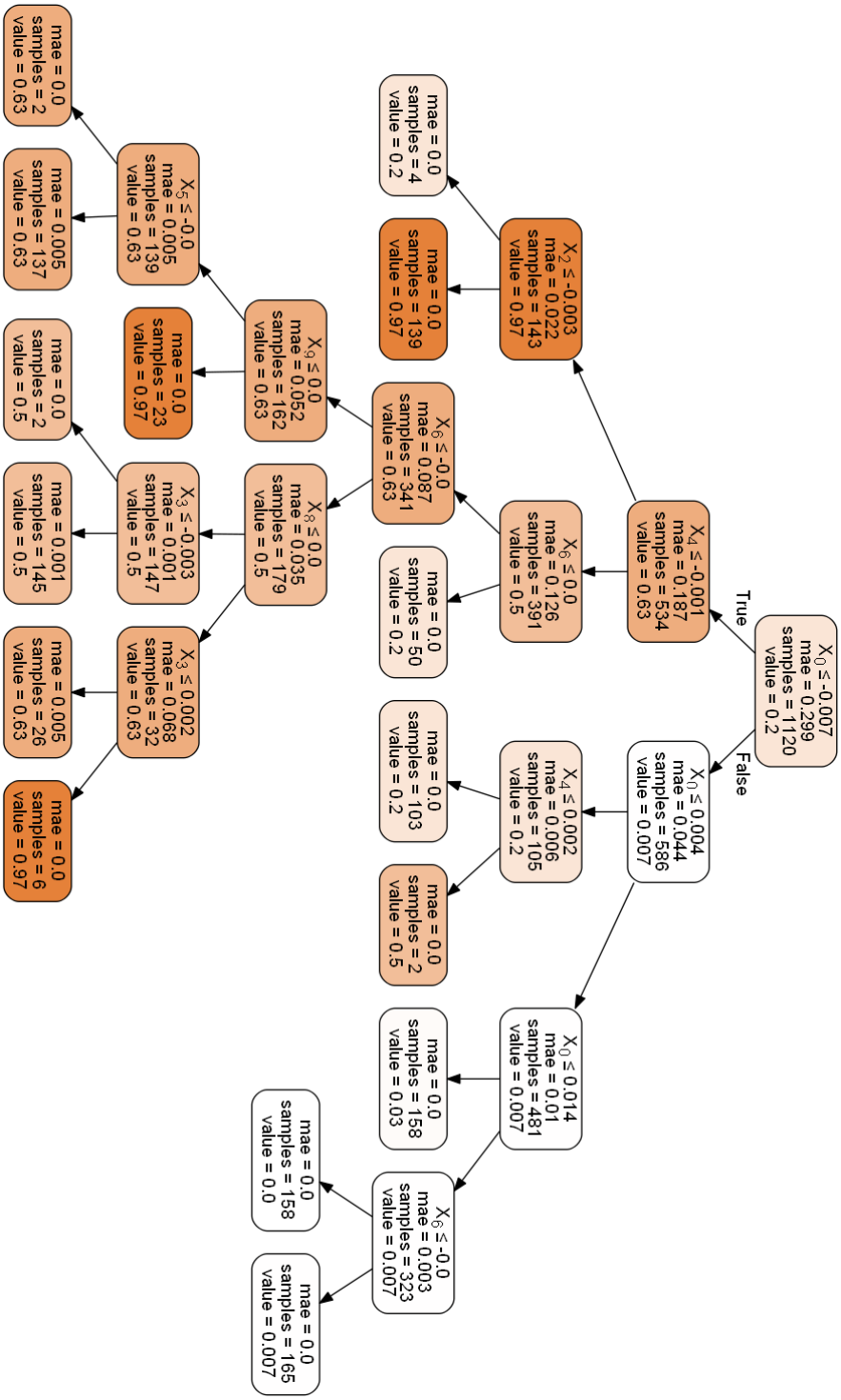


Figure 49: The decision tree structure based on the experimental PCA preprocessed data which achieved an error score of only 0.05%.

RESULTS: SUMMARY

Here we summarize and compare the findings from each of the simulated and experimental models, the important features that they selected during model construction, and how those results compare to previous methods in the literature. Table XXX shows the important gamma-rays that were frequently selected by the simulated data, the experimental data, and compares those results to the gamma-rays used by Mandel et al. Many of the same gamma-rays selected by Mandel et al. were also used by my simulated and experimental models. In fact, the only unique gamma-rays that were selected among my models are gamma-rays from ^{133}Xe (81 keV), ^{135}Xe (250 keV), and a 100 keV gamma-ray the origin of which I could not ascertain. The fact that there is general agreement between the different methods in terms of which gamma-rays are important is strong indication that all of these methods are valid. However, the differences suggest that there are many ways to build a robust model which can predict uranium enrichment based off of gamma-ray measurements.

Of course, I have restricted the analysis of feature importances of my experimental data by only considering the top 10 features of each of the top 3 principal components. It is possible that contained in lower feature/principal component pairs are many of the other gamma-rays utilized by Mandel et al.

In summary, these results show that, using the same amount of data as Mandel et al. (a few gamma-ray measurements) we are able to build predictive models of uranium enrichment that significantly outperform the conventional Mandel et al. technique whilst maintaining invariance across decay time (and potentially irradiation time as well).

Table 8: The important gamma-rays as selected by Mandel et al., my simulations, and my experimental data.

Photopeak	Parent	Mandel et al.	Simulations	Experimental
105 keV	^{239}Np	Y	Y	Y
210 keV	^{239}Np	Y		
278 keV	^{239}Np	Y		
228 keV	^{239}Np			Y
140 keV	^{99}Mo	Y	Y	Y
530 keV	^{133}I	Y		
556 keV	^{99}Y	Y		
658 keV	^{97}Nb	Y	Y	
743 keV	^{97}Nb	Y		
773 keV	^{132}I	Y	Y	
81 keV	^{133}Xe			Y
250 keV	^{135}Xe			Y
100 keV	^{239}Np			Y

Chapter 8: Conclusions and Future Work

CONCLUSIONS

Here, I summarize my accomplishments in this work, the main findings, and the impact that I expect this work to have on the fields of nuclear security and nuclear forensics. Utilizing simulated data, I have demonstrated that decision tree-based models can perform predictions that are accurate within 2% (of the range of values: 0-100%) on the uranium enrichment of irradiated samples with unknown decay times based exclusively off their gamma-ray emissions while maintaining relatively simple structures. These results compare favorably with existing methods in this field. Furthermore, I demonstrated that the models I developed have robust generalization performance, with virtually no degradation in performance as disparities in the irradiation history of the training and testing datasets are introduced.

I have reinforced the findings of my simulated data models with real-world experimental measurements. Even taking into account the difficulties of real-world measurement such as detector response functions, mass uncertainties, etc. the performance of these models does not deteriorate. An adaptively boosted decision tree based-model trained on real world data achieved errors as low as 0.3%. This result suggests that these methods are ready for a host of applications-focused real-world scenarios. Actually, the real-world data is so rich in dimensionality (compared to simulations) that the performance of the models improved (when compared to models built on simulated data), despite employing data truncation methods for reasonable computation times. I have also demonstrated that principal component analysis is highly effective at compressing the experimental data, while causing no prediction performance degradation. On the other hand, the PCA representation of the dataset yielded even still better results in prediction performance. The most impressive result of this work is captured in Figure 48, where an extremely simple decision tree evaluated on PCA-truncated data achieved a prediction error score of 0.05%.

In conclusion, this work demonstrates the exceptional power of performing statistical analysis via inverse methods on large sets of data to predict measurements which are otherwise intractable in the forward sense. I expect that these results will encourage the nuclear safeguards and nuclear forensics communities to embrace statistical modeling techniques to effectively leverage the large reserves of data available to them to dramatically improve their measurement capabilities in a wide variety of applications. The prediction capabilities and insights provided by these models can be naturally extended to application-focused measurements in the fields of nuclear safeguards, nuclear forensics, and nuclear nonproliferation such as: re-verification of spent nuclear fuel dry storage casks, forensics analysis on environmental samples of reactor or weapons debris, and more.

Many problems in nuclear engineering have complex nonlinear dependencies between the ground truths of the system and the observable signals which are used to determine these ground truths through the use of theoretical models. The machine learning techniques I have presented here have the power to characterize these relationships through approximations that lead to exceptional prediction capabilities even in the absence of a rigorous theoretical model of the forward problem. In particular, the feature and threshold value pairs identified by nodes in a decision tree can reliably capture boundary values which influence the ultimate labels assigned to observations in a nonparametric sense. The ability to circumnavigate the theoretical models when performing important real-world decision-based measurements is a key advantage of statistical prediction models using machine learning.

LIMITATIONS & FUTURE WORK

In this section I will discuss the limitations in my work, emphasizing the steps that would be necessary to continue this project towards application-focused systems. I will provide examples of a few specific applications to which the methods I have developed in this work could be applied.

Here I will list some of the main limitations of the work performed here, in no particular order. All of these limitations represent opportunities for future work in this problem domain.

I have arbitrarily limited the selection of machine learning algorithms to the three regularized linear models and three decision tree-based models used in this work. Of course, there are many more machine learning algorithms that could be useful for this work such as support vector machines, k neighbors, neural networks, etc.

I could have spent more time performing feature engineering on the input data. In particular for the regularized linear models, it could have been useful to engineer more features based off of the initial features. For instance, as was shown in Mandel et al., photopeak count ratios tend to be strong indicators of uranium enrichment. I could have engineered photopeak ratios into the data provided to the regularized linear models, which would have given the models much more flexibility to identify meaningful trends in the data beyond simple independent coefficients for each gamma-ray energy bin feature.

I have limited the sets of data which are presented to the machine learning models to only seven discrete enrichment levels. Furthermore, I have used the same enrichment levels between training and testing sets. More work is needed to determine the generalization performance of such models when presented with a continuous range of all enrichment values, and to determine whether such models can interpolate between enrichment values on which they are trained so as to perform predictions on previously unseen intermediate values.

It is important to discuss the effects of random and bias errors in the observation data on the prediction capabilities of this method. The counting statistics of gamma-ray spectrometry in particular place limitations on reasonable time windows for individual observations. This work focused on experimental data in which each observation represented 15 minutes of gamma-ray spectrometry data. Even in this relatively well-homogenized dataset, the range of decay times and enrichment values significantly influence the activities, and thus random errors present in each individual observation. Measurements of lower enrichment samples and measurements of higher decay time will

necessarily have poor counting statistics and thus high random errors when compared to measurements of higher enrichment or lower decay time. The dependencies of model performance on differences in errors present in the data must be studied in more detail. The dataset that I have collected is appropriate for this analysis, as the time discretization can be reduced to 180 seconds per observation. However, although that analysis was considered outside of the scope of this present work, this would be an excellent focus for future research on predictive modeling methods.

POTENTIAL APPLICATIONS

The foundations that I have developed in this work can be adapted to a wide variety of applications in nuclear security and nuclear forensics. Here, I will discuss an example and outline the additional steps that would be necessary towards building a deployable application-focused system.

Spent Nuclear Fuel Autonomous Re-Verification Assay

As I alluded to in the literature review, the current methods of spent nuclear fuel re-verification could be improved. Spent nuclear fuel re-verification offers an excellent opportunity to develop a robust machine learning platform due to: the abundance of storage facilities, the wide variety of inherent variation in the compositions of the spent fuel elements, and the rigorous documentation which provides large quantities of labeled data. The complexity and size of this dataset affords the opportunity to develop models that could: perform anomaly detection, verify reactor type of origin, identify material swapping, estimate enrichment, cooling time, or burnup, and more.

Currently, several competing experimental measurement facilities exist for the re-verification of spent nuclear fuel dry storage casks. These measurement facilities include the INL Compton Dry-Cask Imaging System and the LANL BN-350 Dual Slab Verification Detector. Each of these systems operate on the principle of establishing a unique measurement fingerprint for each spent fuel canister. The measurements which compose this fingerprint can be used to uniquely identify each spent fuel canister. Furthermore, these systems have the capability to measure minute deviations in the

composition of each fingerprint with sensitivity sufficient to detect the removal of a single fuel element.

The Compton Dry Cask Imaging System places a high-purity germanium detector system directly above each fuel element contained in the dry storage cask and measures the ratio of scattered to unscattered photons at each location. The complete set of measurements at each location constitutes a unique fingerprint for this spent fuel cask. If a single fuel element were removed, that would change the amount of photon scattering that occurs in that column, and the measured scattered to unscattered photon ratio at that location would be different. This system has the benefits of nondestructive passive measurement capability with sensitivity sufficient to detect the removal of individual fuel elements, thus supporting nuclear safeguards re-verification objectives. Figure 50 shows a photograph of the Compton Dry Cask Imaging System. The HPGe detector rests on a set of movable platforms which place it directly above each fuel element in the measured cask.

Figure 50: A photograph of the Compton Dry Cask Imaging System.

The Dual-Slab Verification Detector[39] is another spent nuclear fuel re-verification system which relies on establishing uniquely identifying fingerprints of each cask. Here, the measurement is of neutrons rather than gamma-rays. The system consists of an array

of ^{3}He neutron detectors placed around the perimeter of the storage cask. These neutron detectors take measurements at 50-100 unique locations around the periphery of the casks, and the full set of measurements represents one unique fingerprint for that cask. Figure 51 shows a top-down schematic view of the Dual-Slab Verification Detector and its positioning relative to the measured cask.

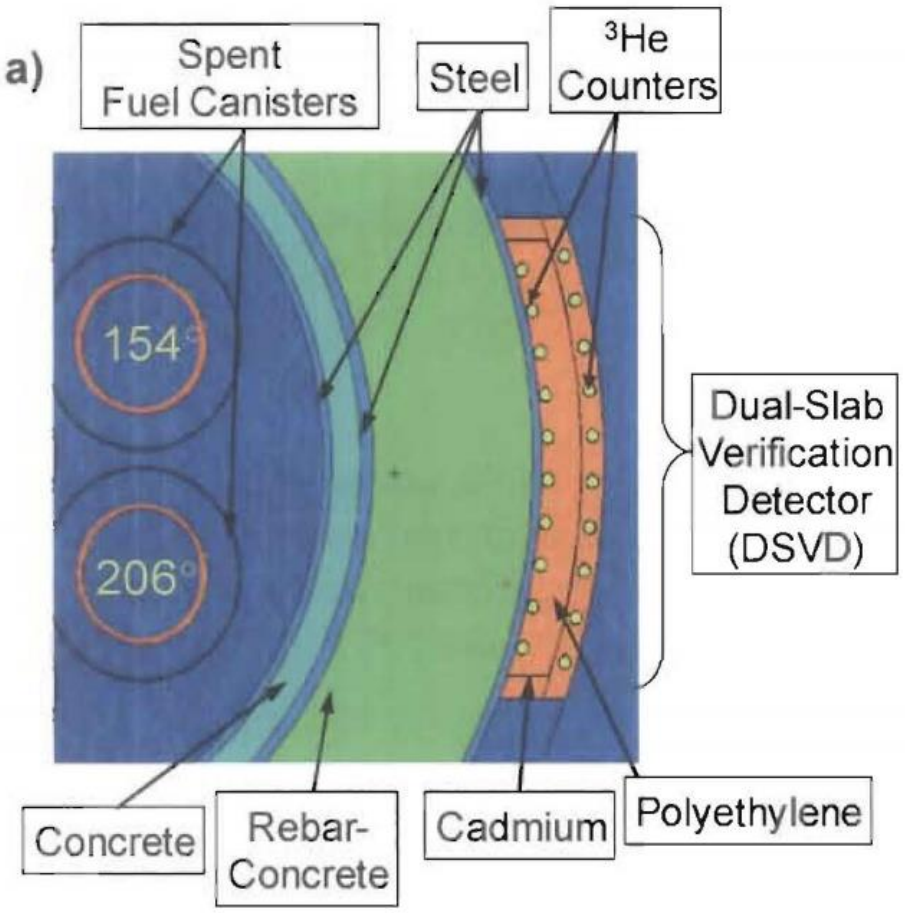


Figure 51: A top down schematic view of the Dual-Slab Verification Detector.

As a potential future work application of the methods I developed here, I propose utilizing these machine learning techniques to analyze the composition of gamma-ray and neutron fingerprints of each of these re-verification systems. The idea here is to move from merely establishing uniquely identifying fingerprints for each spent fuel cask to actually

learning from the composition of each fingerprint to predict important characteristics of each spent fuel cask based on these fingerprint compositions. This could readily be accomplished through supervised learning provided that a sufficient dataset of existing measurements by each of these systems exists. For each measurement made by CDCIS or DSVD, the dataset should be labeled according to important material properties corresponding to that storage cask such as (but not limited to): burnup, cooling time, uranium enrichment, reactor type of origin, number of fuel elements present, etc. The features of each measurement are the unique positions at which each measurement takes place, and the feature values are the gamma-ray or neutron measurements at each of those locations. Models could then be trained to predict any of these important material properties as a function of the features and feature values obtained with the CDCIS & DSVD.

Figure 52 shows a flowchart representing this idea. The CDCIS & DSVD measurements are combined into a data-fused fingerprint consisting of both gamma-ray and neutron measurements. That fingerprint is labeled according to all important material properties. This process is repeated for many spent nuclear fuel canisters, and the resultant dataset is used to train and tune predictive models for important quantities/material properties. This is the natural next step for these measurement systems – moving from merely memorizing fingerprints representative of each individual fuel cask to identifying meaningful trends in the compositions of these fingerprints.

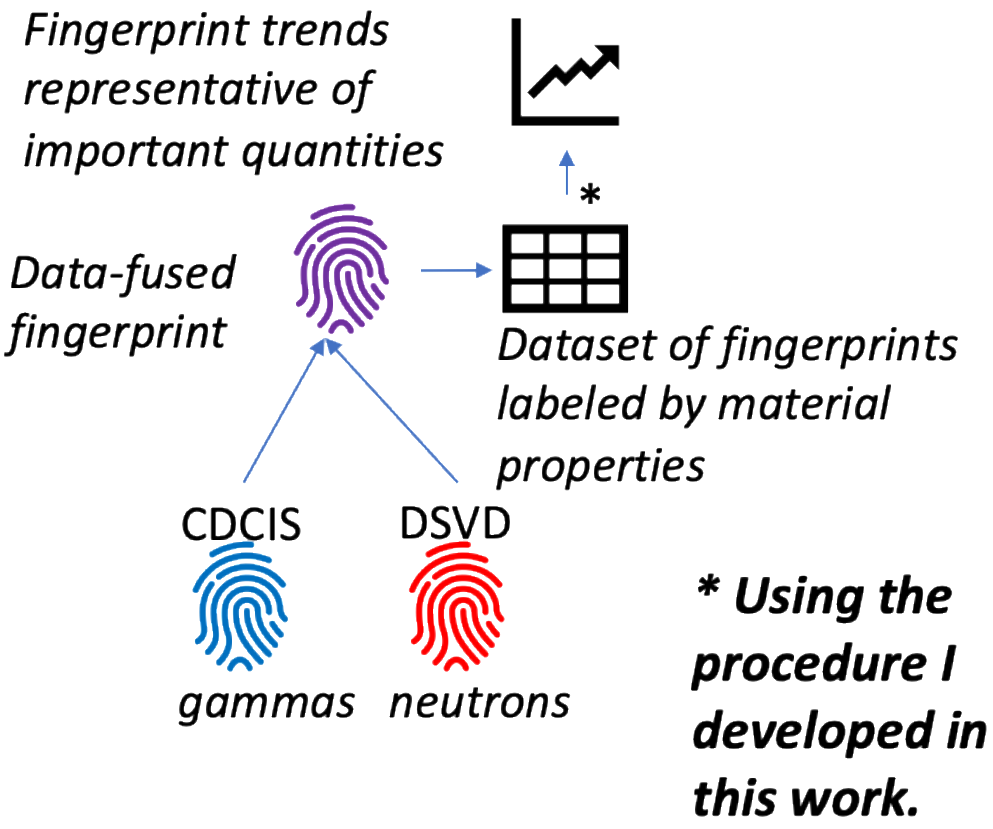


Figure 52: A flowchart of the salient idea presented in this proposal of future work.

The methods for building predictive models that I have demonstrated in this work can be applied to virtually any measurement-based multi-variate application conceivable. This proposal for improving the use of data in spent nuclear fuel re-verification is just one such example. It is my hope that the scientific community embraces the powerful predictive capabilities of machine learning to transform the way that many of these types of problems are solved.

Appendix

These appendices exclude copies of the main Jupyter Notebooks that I developed while performing this work. There are four main Jupyter Notebooks: two for the synthetic data experiments, and two for the real-world measurement experiments. In lieu of replicating them here, live copies of these notebooks will be made available on my website at awdrescher.github.io.

ORIGEN INPUT

Here is a sample ORIGEN input that was used to generate the synthetic data for this experiment. This particular deck corresponds to the irradiation of 1 g of ^{235}U for 1 hour in the High Flux Isotope Reactor at Oak Ridge National Lab. Thank you to Justin Knowles for helping me develop this code.

```
'=========================================================================
' |\      |       /\            /\
' | \     |      /  \          /  \
' |  \    |     /    \        /    \
' |   \   |    /______\      /______\
' |    \  |   /        \    /        \
' |     \\| /          \ /          \
'
'This model was created for irradiations conducted at the Neutron
'Activation Analysis lab at ORNL's High Flux Isotope Reactor. It is
'meant to be used as an estimator of isotope activities and gamma
'spectra created from irradiation and decay of small samples.
'It is not in any way intended to be used for safety purposes.
'
'=========================================================================
'
=couple

0$$ a3 80 a6 33 e
1$$ a15 -1 a16 8 0 238 e t
'-------------------------------------------------------------------------
'HFIR PT-1 MOC flux spectrum (238group)
'Each number in this block represents the relative neutron flux
'for each of the ENDF specified 238 group neutron energies
'-------------------------------------------------------------------------
9**
1.265E+08  9.175E+09  6.418E+09  4.817E+09  3.177E+09  5.106E+09  2.495E+10
1.900E+10  2.477E+10  3.179E+10  8.000E+09  3.361E+10  1.282E+10  3.090E+09
1.492E+10 1.238E+10 4.020E+09 1.860E+09 1.690E+09 3.040E+09 2.380E+09
5.060E+09 4.970E+09 5.430E+09 1.270E+09 1.640E+09 9.300E+08 2.850E+09
5.190E+09 5.780E+09 7.800E+08 6.410E+09 2.680E+09 2.380E+09 5.600E+09
```

```
3.540E+09 3.840E+09 2.700E+09 2.840E+09 1.119E+10 1.168E+10 1.746E+10
1.676E+10 9.120E+09 1.453E+10 9.500E+09 2.090E+09 5.190E+09 1.580E+09
1.146E+10 8.350E+09 2.280E+09 6.160E+09 2.393E+10 1.072E+10 2.305E+10
1.608E+10 1.949E+10 1.013E+10 1.816E+10 2.936E+10 2.430E+09 1.290E+10
1.036E+10 6.960E+09 2.340E+09 1.371E+10 8.730E+09 1.910E+09 1.551E+10
1.151E+10 1.962E+10 1.120E+09 3.469E+10 4.697E+10 3.970E+09 1.011E+10
8.370E+09 7.000E+08 4.370E+09 2.010E+09 2.537E+10 1.450E+09 1.990E+09
3.660E+09 4.510E+09 6.140E+09 5.430E+09 1.440E+09 3.020E+09 3.150E+09
3.860E+09 2.190E+09 3.720E+09 2.020E+09 5.880E+09 1.550E+09 1.590E+09
1.630E+09 1.080E+09 1.590E+09 2.280E+09 1.570E+09 2.160E+09 1.960E+09
2.060E+09 7.400E+08 1.670E+09 1.550E+09 2.420E+09 1.500E+09 1.450E+09
8.700E+08 2.700E+09 9.200E+08 2.470E+09 5.100E+09 5.600E+09 6.230E+09
4.050E+09 2.860E+09 3.010E+09 1.560E+09 4.990E+09 3.570E+09 3.400E+09
2.790E+09 2.710E+09 3.770E+09 4.780E+09 2.000E+09 8.430E+09 5.640E+09
7.030E+09 7.610E+09 1.240E+09 2.150E+09 2.250E+09 2.340E+09 2.460E+09
7.900E+09 1.548E+10 1.080E+10 1.710E+10 4.400E+09 3.900E+09 6.600E+09
1.900E+09 1.000E+09 6.000E+08 2.000E+09 2.100E+09 2.200E+09 2.300E+09
2.400E+09 2.200E+09 2.000E+09 2.400E+09 2.500E+09 3.500E+09 1.800E+09
2.500E+09 3.000E+09 3.100E+09 3.400E+09 3.500E+09 2.000E+09 2.100E+09
2.200E+09 2.200E+09 2.400E+09 1.200E+09 1.200E+09 1.200E+09 1.300E+09
5.000E+08 5.000E+08 5.000E+08 5.000E+08 6.000E+08 5.000E+08 6.000E+08
5.000E+08 5.000E+08 6.000E+08 6.000E+08 5.000E+08 6.000E+08 6.000E+08
6.000E+08 1.400E+09 1.600E+09 1.600E+09 1.600E+09 3.600E+09 4.000E+09
4.200E+09 5.300E+09 7.100E+09 4.000E+09 4.500E+09 9.800E+09 1.070E+10
1.260E+10 1.650E+10 7.700E+09 8.300E+09 1.240E+10 1.650E+10 2.000E+10
2.600E+10 4.320E+10 1.004E+11 2.258E+11 5.372E+11 1.321E+12 2.011E+12
1.013E+12 1.453E+12 2.072E+12 2.960E+12 4.201E+12 5.789E+12 5.799E+12
4.481E+12 1.093E+13 1.181E+12 9.420E+11 2.230E+11 1.960E+11 6.000E+10
4.700E+10 4.600E+10 2.000E+10 1.100E+10 1.300E+10 1.100E+10 2.600E+10
e t
done
end
=origen
-1$$ 50000000
1$$ 1 e t
PT-1 irradiation in HFIR
3$$ 33 a3 1 0 a16 2 a33 100 e t
35$$ 0 e t
'-------------------------------------------------------------------
'Card 56$$ requires the number of ISOTOPES to be entered in a13
'-------------------------------------------------------------------
'------------------>| |<------------------------------HERE!
56$$ 1 1 1 a10 0 a13 1 1 3 a17 3 e
57** 0 a3 0.0 e
95$$ 1 e t
IRRADIATION CASE
PT-1
'-------------------------------------------------------------------
'Card 59** contains the value of TOTAL neutron flux for all
'energy groups. Recommend benchmarking this value to flux monitors.
'-------------------------------------------------------------------
59** 4.7017E+13 e
```

```
'-----------------------------------------------------------------------
'Card 60** contains the irradiation time in SECONDS.
'-----------------------------------------------------------------------
60** 3600 e
'-----------------------------------------------------------------------
'Card 73$$ requires each isotope to be entered in as format:
'Z*10000+A*10
'Examples: U235=922350, Na23=110230
'Note: Last digit =1 for first metastable state, >1 for additional lvls
'-----------------------------------------------------------------------
73$$ 922350
'-----------------------------------------------------------------------
'Card 74** requires a mass (g) for each isotope. Must have a mass for
'every isotope, and number of entries must equal 73$$ entry
'-----------------------------------------------------------------------
74** 1
'-----------------------------------------------------------------------
'Card 75$$ an entry for each isotope
'1=activation product, 2=actinide, 3=fission product
'-----------------------------------------------------------------------
75$$ 2 e t
54$$ a8 1 a11 0 e
56$$ a2 1 a10 1 a14 4 a15 3 a17 3 e
57** 0 a3 0.0 e
95$$ 1 e t
DECAY CASE
POST PT-1 IRRADIATION
'-----------------------------------------------------------------------
'Card 60** contains the decay time in DAYS.
'-----------------------------------------------------------------------
60** 0.01 e t
54$$ a3 1 a8 1 a11 0 e
56$$ a1 0 a2 179 a10 1 a14 4 a15 3 e
57** 0 a3 0.0 e
95$$ 1 t
DECAY DURING MEASUREMENT
USED FOR SOURCE INTEGRATION
'-----------------------------------------------------------------------
'Card 60** contains the measurement period and number of linearly
'interpolated intervals. (18i 0 2) means 0 to 2 hrs with 18 divisions
'-----------------------------------------------------------------------
60** 250i 0.01 25.01 e
81$$ 2 a3 26 1 e
82$$ 20r-2 e
'-----------------------------------------------------------------------
'Card 83** represents gamma ray groups printed. 40i1000000 means 40
'linearly interpolated groups between 0MeV and 1MeV.
'-----------------------------------------------------------------------
83** 100i2500000 e t
56$$ f0 t
end
```

References

1. Aubert, G. & Lansdorp, P. M. Telomeres and Aging. *Physiol. Rev.* 557–579 (2009). doi:10.1152/physrev.00026.2007
2. Pedregosa, F., Weiss, R. & Brucher, M. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
3. Breiman, L. Random Forests. *Mach. Learn.* 5–32 (2001). doi:10.1023/A:1010933404324
4. M Mantel, J Gilat, S Amiel, (Nuclear chemistry department, Soreq Nuclear Research Center, Yavne, I. Isotopic analysis of uranium by neutron activation and high resolution gamma-ray spectrometry. **2**, (1969).
5. Hata, H. *et al.* Application of support vector machine to rapid classification of uranium waste drums using low-resolution γ-ray spectra. *Appl. Radiat. Isot.* **104**, 143–146 (2015).
6. Drescher, A. *et al.* Gamma-gamma coincidence performance of LaBr3_Ce scintillation detectors vs HPGe detectors in high count-rate scenarios. *Appl. Radiat. Isot.* **122**, 116–120 (2017).
7. Shuryak, I. Advantages of synthetic noise and machine learning for analyzing radioecological data sets. *PLoS One* **12**, 1–20 (2017).
8. Park, K. J. *et al.* Determination of burnup and Pu/U ratio of PWR spent fuels by gamma-ray spectrometry. *Nucl. Eng. Technol.* **41**, 1307–1314 (2009).
9. Tian, C., Guo, H., Lv, N., Xu, P. & Lv, W. Analysis on enrichment and mass of uranium materials with time-correlated measurement. *Health Phys.* **113**, 387–391 (2017).
10. Jones, A. E., Turner, P., Zimmerman, C. & Goulermas, J. Y. Classification of spent reactor fuel for nuclear forensics. *Anal. Chem.* **86**, 5399–5405 (2014).
11. Robel, M. *et al.* A composite position independent monitor of reactor fuel irradiation using Pu, Cs, and Ba isotope ratios. *J. Environ. Radioact.* **195**, 9–19 (2018).
12. Meier, D. *et al.* Contract No : Office of Environmental Management (EM). Disclaimer : Multi-Isotope Process (MIP) Monitor Deployment at H-Canyon.
13. Dayman, K. J., Coble, J. B., Orton, C. R. & Schwantes, J. M. Characterization of used nuclear fuel with multivariate analysis for process monitoring. *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* **735**, 624–632 (2014).
14. Åberg Lindell, M. *et al.* Discrimination of irradiated MOX fuel from UOX fuel by multivariate statistical analysis of simulated activities of gamma-emitting isotopes. *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* **885**, 67–78 (2018).
15. Durst, P. C. Guidance for Independent Spent Fuel Dry Storage Installations. 1–22 (2012).
16. Meyer, R. M. Nondestructive Examination Guidance for Dry Storage Casks. (2016).
17. Kamuda, M., Zhao, J. & Huff, K. A comparison of machine learning methods for automated gamma-ray spectroscopy. *Nucl. Instruments Methods Phys. Res. Sect. A*

Accel. Spectrometers, Detect. Assoc. Equip. (2018). doi:10.1016/j.nima.2018.10.063
18. Zhang, Y., Wang, S., Phillips, P. & Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowledge-Based Syst.* **64**, 22–31 (2014).
19. Bordarie, J. Predicting intentions to comply with speed limits using a 'decision tree' applied to an extended version of the theory of planned behaviour. *Transp. Res. Part F Traffic Psychol. Behav.* **63**, 174–185 (2019).
20. Li, Y. *et al.* Random forest regression for online capacity estimation of lithium-ion batteries. *Appl. Energy* **232**, 197–210 (2018).
21. Chen, W., Sun, Z. & Han, J. Landslide susceptibility modeling using integrated ensemble weights of evidence with logistic regression and random forest models. *Appl. Sci.* **9**, (2019).
22. Chen, Y. *et al.* Estimating deciduous broadleaf forest gross primary productivity by remote sensing data using a random forest regression model. *J. Appl. Remote Sens.* **13**, 1 (2019).
23. Chihab, Y., Bousbaa, Z., Chihab, M., Bencharef, O. & Ziti, S. Algo-Trading Strategy for Intraweek Foreign Exchange Speculation Based on Random Forest and Probit Regression. *Appl. Comput. Intell. Soft Comput.* **2019**, 1–13 (2019).
24. Chattopadhyay, M. A Comprehensive Review and Meta-Analysis on Applications of Machine Learning Techniques in Intrusion Detection. *Australas. J. Inf. Syst.* **22**, 1–27 (2018).
25. Ng, A. Y. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. *Twenty-first Int. Conf. Mach. Learn. - ICML '04* 78 (2004). doi:10.1145/1015330.1015435
26. Géron, A. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems* (2017).
27. Burkov, A. *The 100-Page Machine Learning Book*. *Expert Systems* **5**, (2019).
28. Singh, S. Understanding the Bias-Variance Tradeo. *Towards Data Science* 1–5 (2019). Available at: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229.
29. Joseph Rocca. Ensemble methods: bagging, boosting and stacking. *Towards Data Science* 1–21 (2019).
30. Hastie, T., Tibshirani, R. & Friedman, J. *Elements of Statistical Learning*. *Revista Espanola de las Enfermedades del Aparato Digestivo* **26**, (2009).
31. Brieman, L., Freidman, Olshen & Stone. *Classification and Regression Trees*. (1984).
32. T.K.~Ho. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 832–844 (1998).
33. Freund, Y. & Schapire, R. E. Experiments with a New Boosting Algorithm. *Proc. 13th Int. Conf. Mach. Learn.* 148–156 (1996). doi:10.1.1.133.1040

34. Halko, N., Martinsson, P. G. & Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**, 217–288 (2011).
35. Sehgal, B. R. Light water reactor safety. A historical review. *Nucl. Saf. Light Water React.* 1–88 (2012). doi:10.1016/B978-0-12-388446-6.00001-0
36. Gilmore, G. *Practical Gamma-ray Spectrometry 2nd Edition*. *Training* (2008). doi:10.1002/9780470861981
37. Wieselquist, W. A. The scale 6.2 origen API for high performance depletion. *Math. Comput. Supercomput. Nucl. Appl. Monte Carlo Int. Conf. M C+SNA+MC 2015* **4**, 2789–2799 (2015).
38. Nuclear Materials Disposition Program Office Y-12 Plant Defense Program. HEU to LEU Conversion and Blending Facility UNH Blending Alternative to Produce LEU UNH for Commercial Use. **13675**, (1995).
39. Santi, P. A., Browne, M. C., Parker, R. F. & Williams, R. B. Testing of the Dual Slab Verification Detector for Attended Measurements of the BN-350 Dry Storage Casks. **836**, (2009).

Vita

Adam Drescher grew up in New England. He moved to Austin, Texas and started attending The University of Texas at Austin in January 2013. He graduated in December 2015 with a Bachelor of Science in Physics degree. He remained at UT Austin for graduate school. He obtained his Master of Science in Engineering degree in May of 2017. He then obtained his Doctor of Philosophy in Engineering degree in December 2019. He has accepted a position as a Postdoctoral Research Associate in the Nuclear Security Modeling Group within the Nuclear Sciences Engineering Directorate at Oak Ridge National Laboratory, where he will start in January 2020. In his free time, Adam enjoys motorcycling, weightlifting, and music.

Permanent address: awdrescher@gmail.com

This dissertation was typed by the author.