Todo list

Evaluating Statistical Methods for Nuclear Forensics Analysis

by

Arrielle C. Opotowsky

A preliminary report submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Nuclear Engineering & Engineering Physics)

at the

UNIVERSITY OF WISCONSIN–MADISON

November 2017

Preliminary Examination Committee:

Rebecca M. Willet, Professor, Electrical & Computer Engineering

Charles F. Weber, Distinguished Scientist, Oak Ridge National Laboratory

Jake P. Blanchard, Professor, Nuclear Engineering & Engineering Physics

Douglass L. Henderson, Professor, Nuclear Engineering & Engineering Physics

Paul P.H. Wilson, Professor, Nuclear Engineering & Engineering Physics

For Steve

ACKNOWLEDGMENTS

This proposal would not be possible with the wisdom and patience of my advisor, Paul Wilson. I honestly don't have words for how grateful I am and lucky I feel. I'm also appreciative of the CNERG community for technical and non-technical assistance; may quiche recipes be forever shared during important phone calls. Kelly Burton and Max Lagally have invested **much** effort into my success and convinced me that graduate school was the right path for me—more than once. My GERS friends have given me so much in and out of school, especially José Roberto, Richard, and Chandler. I have also received generous funding from the National Science Foundation and the Department of Homeland Security.

Mountains of personal support motivated me here and kept me here, which I do not take for granted. Steven "If you're gonna be dumb you gotta be tough" Harrell, my chosen family, inspired me to get all my KSAs, no matter where I wanted to find them. Robin, you have been such a light in my life for over a decade and always remind me why I came back to grad school. And to my friends for 15 years, thanks for keeping in touch despite the gaps. Denise and April, I'm so glad we are still in touch. Ruthie, you push me to be fierce, spittin' truths and slaying your way through life. And, always last and usually the least (jk), my family; although far too long to list, some have fought along with me this entire time. Maurice, thanks for the help when I was struggling. Lou, thanks for being one of my best friends and sources of laugh lines. Liz, I love finding my kitchen floor clean. For endlessly encouraging my academic and personal pursuits, I'm appreciative of my California family, Mel, Bonnie, Joelle, and Jamie.

Finally, my Madison family has blessed me in countless ways: Shan, Ninja, Peter, Drax, Samira, Heather, Burnie, Brarit, Sarah, the late Otto, James, Troy, BLou, Fetal, Martha, Matt.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The purpose of this work is to evaluate the utility of statistical methods as an approach to determine forensics-relevant quantities for commercial spent nuclear fuel as less information is available. Machine learning algorithms will be used to train models to provide these values (e.g., reactor type, time since irradiation, burnup) from the available information. The training data will be simulated using ORIGEN, which will provide an array of nuclide concentrations as the features (X) and the parameters of interest (y) are provided from the simulation inputs. Information reduction will be carried out using computationally generated gamma spectra; the radionuclide concentrations from the simulations can be converted into gamma energies, which then undergo a detector response calculation to represent real gamma spectra as closely as possible. Machine learning best practices will be used to evaluate the performance of the chosen algorithms, and inverse problem theory will be used to provide an interval of confidence in the model predictions.

copied from goals, fix up

1 INTRODUCTION

The realm of nuclear security involves many parallel efforts in nonproliferation (verification of treaty compliance, monitoring for smuggling, proper storage and transportation of nuclear materials), cyber security, minimizing stocks of weaponizable materials, disaster response training, and nuclear forensics. All of these efforts have been continually improving, but there was a gap regarding the ability of the United States (US) to coordinate and respond to a nuclear incident, especially with the technical portion of nuclear forensics: characterization and analysis. After all, the first textbook on the topic was published in 2005 [18]. In 2006, the US Department of Homeland Security (DHS) founded the National Technical Nuclear Forensics Center (NTNFC) within the Domestic Nuclear Detection Office (DNDO). The mission of the NTNFC is to establish a robust nuclear forensics capability to attribute radioactive materials with demonstrable proof.

There are many fields that contribute to the nuclear forensics capability, such as radiochemical separations, material collection techniques, improving detector technology, material library development, and identifying forensic signatures. These needs vary based on whether the material being collected is post-detonation (e.g., bomb debris) or pre-detonation (e.g., spent nuclear fuel (SNF)). In the pre-detonation realm, this project focuses on statistical methods to identify correlated material characteristics, which can lead to new forensic signatures.

1.1 Motivation

Nuclear forensics is an important aspect of deterring nuclear terrorism even though it is not, at first glance, thought to be preventative nuclear security. The most common defense of the field is that nuclear forensics capability deters state actors, not terrorist organizations. While it is true that a strong capability encourages governments to be more active in prevention of nuclear terrorism, it can also deter the terrorist organizations as well by increasing their chances of failure. Less destructive success tends to be more valued than high-risk mass destruction. In addition to influencing governments and making nuclear terrorism higher risk for organizations, nuclear forensics can assist in cutting off certain suppliers of nuclear materials or technologies (e.g., nuclear specialists that are only involved for financial reasons, access to state suppliers). Shutting off the sources builds a concrete barrier to nuclear terrorism. Therefore, nuclear forensics is considered impede this form of terrorism in both tangible and abstract ways [16].

Following the prevention value of nuclear forensics, it is important to understand the process of the technical portion of the investigation and how that can be improved. In the event of a nuclear incident, such as the retrieval of stolen special nuclear material (SNM) or the detonation of a dirty bomb, it is necessary to learn as much as possible about the source of the materials in a timely manner. In the case of non-detonated SNM, knowing the reactor parameters that produced it can point investigators in the right direction

in order to determine the chain of custody of the interdicted material. Section 1.1.1 covers the specific needs of the nuclear forensics community for SNF provenance, and Section 1.1.2 discusses how alternative computational approaches are useful, with a focus on why statistical methods in particular are being pursued.

1.1.1 Needs in Nuclear Forensics

The process of technical nuclear forensics includes the analysis and interpretation of nuclear material to determine its history, whether that be intercepted SNF, uranium ore concentrate (UOC), or the debris from an exploded nuclear device. After the technical portion is complete, intelligence data can be used to aid in material attribution; this is the overall goal of nuclear forensics.

After a nuclear incident, the material or debris is sampled and evaluated through many techniques that provide the following information: material structure, chemical and elemental compositions, and radioisotopic compositions and/or ratios. These measurements or ratios comprise the forensic signatures of the sample in question. These signatures can be analyzed with specific domain knowledge; for example, UOC will have trace elements depending on where it was mined from. They can also be analyzed against a forensics database in the case of SNF.

Measurement needs and techniques vary vastly depending on the material, as does the type of signature. This study focuses on non-detonated materials,

specifically, SNF. It is important to determine if some intercepted SNF is from an undisclosed reactor or a commercial fuel cycle to attribute it to an entity or state. This is typically done by obtaining select chemical and elemental signatures and isotopic ratios, and comparing these measurements to those in an existing forensics database of reference SNF. The signatures for SNF correlate to characteristics that can, in a best case scenario, point to the exact reactor from which the fuel was intercepted. The reactor parameters of interest are the reactor type, fuel type and enrichment at beginning of irradiation, cooling time, and burnup [1, 32, 33].

The current and future work of this study are designed based on two primary needs to bolster the US nuclear forensics capability: post-incident rapid characterization, and forensics database challenges and imperfection.

First, our best measurement techniques may not be available in an emergency scenario, and fast measurements typically yield inaccurate results. Currently, both radiological measurements and mass spectrometry are used in nuclear forensics exercises. Because these techniques have a multitude of variants within each category, there are differing levels of certainty of the results. Easily deduced is that the faster and cheaper methods also provide the most uncertain values. Thus, the main tradeoff is between time/cost and amount of information gained. A lofty goal would be to develop methods that provide instantaneous information, reliable enough to guide an investigation (e.g., within 24 hours). In the case of SNF, it takes weeks in a lab to measure isotopes via advanced (cooled detector) gamma

spectroscopy and mass spectrometry equipment. A handheld detector that measures gamma spectra could provide the fast measurements to calculate isotopic ratios for the above-mentioned fuel parameters of interest. However, while this nondestructive analysis is rapid, it is also difficult to evaluate because of the presence of overlapping peaks and the fact that uncertainties differ significantly because of the detector reponse, environment, storage, electronics, etc. Broadly speaking, gamma spectra give less information at a higher uncertainty than the near-perfect results of the destructive mass spectrometry techniques used for characterization, such as inductively coupled plasma mass spectrometry [11].

Second, forensics databases are imperfect; this is three-fold. Because of the values needed for material provenance and the number of measurement types, the forensics databases are 1. highly multidimensional and have 2. inconsistent uncertainties or missing data entries, respectively [12]. Thus, direct comparison between measurement results and a database therefore may not yield accurate parameter predictions. Furthermore, 3. forensics databases are kept by individual countries, and the reactor operation history information is somewhat well guarded, so it may be difficult to study SNM from a country that has a different fuel cycle. It is proposed that using a machine-learned model may be able to combat these issues; this is introduced next in Section 1.1.2.

1.1.2 Contribution of Statistical Methods

As previously mentioned, there are two main issues that are being addressed for forensics of SNF: database issues and speed of characterization. Many have begun considering computational techniques developed by nuclear engineers to calculate the parameters relevant to nuclear forensics analysis. One example is the INverse DEPletion THeory (INDEPTH) tool [1, 32, 33]. INDEPTH uses an iterative optimization method involving many forward simulations to obtain reactor parameters of interest given some initial values.

Another approach utilizes artificial intelligence to solve nuclear forensics problems, such as implementing searching algorithms for the database comparison step [5] and machine learning for determining reactor parameters from SNF characteristics [3, 9, 10, 19, 21, 22, 26, 27]. A variety of statistical and machine learning tools have been used to characterize SNF by predicting categories or labels (e.g., reactor type, fuel type) as well as predicting values (e.g., burnup, initial enrichment, cooling time). The former uses classification algorithms and the latter uses regression algorithms, many of which can be altered to perform both classification and regression. There is some promising work discussed in Section 2.4 that shows certain applications of machine learning can provide an additional tool for solving the forensics problem, both qualitatively (for visualization) and quantitatively (for prediction).

Statistical methods have the uniqueness of requiring minimal domain knowledge via machine learning algorithms that predict the characteristics or values of interest [3, 9, 10, 19, 21, 22, 27]. They first create a black-box

statistical model using the database entries, and can predict the reactor parameters of an unknown sample based on that model. Having a machine-learned model based on a large number of simulations may also overcome the challenges of missing data, irregular uncertainty, or lack of information on other fuel cycles. This logic also follows for other computational methods using a large number of simulations. Although not encompassed in a resuable model, they also could overcome missing data, irregular uncertainties, or ignorance of different or non-commercial fuel cycles. Also, it is generally known that statistical methods will be able to either use or reduce the dimensions in the forensics databases, which is another useful characteristic.

1.2 Methodology

As previously mentioned, the typical workflow of the technical portion of a forensics investigation is to take measurements of an unknown material and compare those measurements to databases filled with previously measured standard materials. As this work focuses on SNF, these measurements are elemental, chemical, and radiological in nature. Because creating databases from real measurements to represent SNF from reactor technologies from around the world is not within the scope of this project, the database in this study will be created from high-fidelity simulations via Oak Ridge Isotope GENeration (ORIGEN) [25] within the SCALE code system [23]

for modeling and simulation.

Figure 1.1 compares the INDEPTH and statistical methodologies, both of which use simulated SNF. While not all steps are required to be equivalent, the only difference here is the method one chooses to obtain reactor parameters. Both workflows address speed of characterization, as it is intended to have gamma spectra as the inputs. Because INDEPTH is better studied and validated than statistical methods, this work focuses on a statistical approach but with the intention to compare methodologies. Both workflows also address many of the database issues, described above. The bottom statistical methodology is described here.

cite

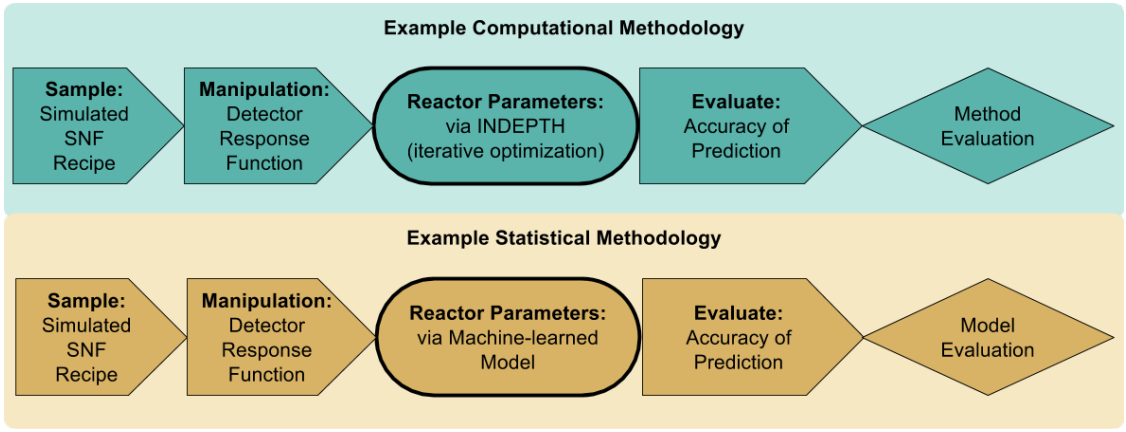


Figure 1.1: Computational Forensics Research Workflows

In the simulation and statistical learning paradigm, we need to determine how much information to what quality is needed to train a machine-learned model; the model must give appropriate predictions of reactor parameters

given a set of measurements from a test sample of interdicted SNF. Thus, the space that the training set encompasses must be chosen carefully so as to represent a likely scenario for stolen SNF.

The next step is to choose an algorithm that performs statistical learning. Statistical learners have varied strengths and weaknesses based on what is being predicted and how they implement optimization. Chosen for this study are simple regression algorithms for burnup prediction: nearest neighbor and ridge regression. For comparison, SVR is used because it is known to handle highly dimensional data sets well. These algorithms are introduced in Section 2.2.1.

After the training is complete, the results of each models' predictions must be evaluated. Typically, a test set is used to compare against the model created from the training set. The testing error can therefore be tabulated with respect to various specifications such as the training set size, number of features, or algorithm parameters. These results are broadly known as diagnostic plots and show if the algorithms' predictions are due to good performance or bad fitting.

After the models are evaluated using machine learning best practices, it will be important to compare them both against each other and against other computational forensics parameter methods. Thus, a Bayesian approach from the field of inverse problem theory will be used to give the probability density of the predictions so that the statistically generated predictions can be evaluated directly against other solutions, such as optimization-based

methods or direct computations.

Next, information reduction (within the training and/or testing data sets) must be used to investigate the extension of this workflow to the real world. The primary example is the reduction of information quality via gamma ray detectors, as they can provide fast results. If an algorithm could overcome the limitations of gamma detection and still provide useful results, this would warrant further studies and perhaps be field-applicable.

Thus, ultimately, the goal is to answer the question *How does the ability to determine forensic-relevant spent nuclear fuel attributes using machine learning techniques degrade as less information is available?*.

1.3 Goals

The main purpose of this work is to evaluate the utility of statistical methods as an approach to determine nuclear forensics-relevant quantities as less information is available. Machine learning algorithms are used to train models to provide these values (e.g., reactor type, time since irradiation, burnup) from the available information. The training data is simulated using ORIGEN, which provides an array of nuclide concentrations as the features (X) and the parameters of interest (y) are provided from the simulation inputs. Information reduction is carried out using computationally generated gamma spectra; the radionuclide concentrations from the simulations can be converted into gamma energies, which then undergo a detector response

calculation to represent real as-measured gamma spectra as closely as possible. Machine learning best practices are used to evaluate the performance of the chosen algorithms, and inverse problem theory is used to provide an interval of confidence in the model predictions.

The necessary background is covered in Chapter 2. First, an introduction to the broader field of nuclear forensics is in Section 2.1 to place this work in the context of the technical mission areas. After that, a short discussion of the field of machine learning, the algorithms used, and validation methods are in Section 2.2. Section 2.3 includes information about the codes used to generate the training data, via fuel cycle simulation, detector response function, and isotope identification of gamma spectra. Lastly, a review of statistical methods being used in studies of forensics analysis is covered next in Section 2.4.

After the existing work is discussed, the methodology and a demonstration of the experimental components is introduced next in Chapter 3. This will cover the simulated training data in Section 3.1, the parameters behind the learned models in Section 3.2, and the process of model evaluation in Section 3.3.

Finally, Chapter 4 summarizes the official thesis research proposal. After the preparatory tasks are covered in Section 4.1, there are three experiments outlined in Sections 4.2, 4.3, and 4.4. Qualitiative hypotheses as well as alternative directions for risk mitgation are discussed throughout this chapter as well.

2 BACKGROUND AND LITERATURE REVIEW

This chapter provides a background and literature review of the necessary components for this project. Section 2.1 outlines the broader field of technical nuclear forensics, with a focus on the area that motivates this project. Section 2.2 introduces the field of machine learning for an uninitiated audience, covers the relevant algorithms, and presents the methods field practitioners use for validation. Next, Section 2.3 covers the computational methods used to generate the training data for the machine learning input. Finally, the marriage of Sections 2.1, 2.2, and 2.3, is presented in Section 2.4, which is a review of previous work applying statistical methods to the nuclear forensics analysis of pre-detonated nuclear materials.

2.1 Nuclear Forensics

Nuclear forensics comprises a large part of an investigation into a nuclear incident, such as interdicted nuclear material or the detonation of a weapon containing radioactive components. The forensics portion of the investigation encompasses both the analysis of nuclear material and/or related paraphrenalia as well as the interpretation of these results to establish nuclear material provenance. The former has many technical aspects, relying on a range of nuclear science and chemistry. The latter involves intelligence and political considerations of the material analyses for attribution. This review

will only consider the technical portion of the nuclear forensics workflow.

First discussed are the types of forensic investigations in Section 2.1.1, followed by an introduction to inverse problem theory in Section 2.1.2 as a way to evaluate the results of forensic methods.

2.1.1 Types of Nuclear Forensics Investigations

The technical programs researching improvements to the US's nuclear forensics capabilities are split between the type of material being investigated. The analysis of irradiated debris from a weapon has different collection and measurement requirements than a mass of SNM. This separates the field into post-detonation and pre-detonation nuclear forensics. While both are discussed below in Sections 2.1.1.1 and 2.1.1.2, respectively, there is more focus on pre-detonation topics since this work is based on SNF.

2.1.1.1 Post-Detonation

Post-detonation nuclear forensics requires a diverse set of measurements to obtain the following information: identification of nuclear material, reconstruction of the weapon device design, and reactor parameters for nuclear material provenance. This could apply to an improvised nuclear device or a nuclear bomb. In conjunction with the measurements and characterization are a large array of logistical concerns, including recovery efforts, personnel safety, and material collection cataloging and transportation.

In the case of a full explosion using fissile material, the collection of materials and debris occurs as quickly as possible. It can be in the crater created by the explosion, further away from the center in the fallout, and in the atmosphere above or downwind from the detonation. These are collected by finding glass-like material near the epicenter, debris swipes in the fallout region, and advanced particle collection in the atmosphere via an airplane, respectively. While the epicenter cannot be reached for some time, the debris and atmosphere measurements of radioactive material can provide the yield of the weapon and whether it was made using uranium or plutonium. This along with other physical and chemical measurement allow device reconstuction to begin. Attribution begins to narrow to specific countries or organizations based on this information. [16]

The research needs for post-detonation focus on material collection and analysis as well as nuclear device modeling for rescontruction purposes. Ideally, most material sample collection would be done using automatic instrumentation. Additionally, bolstering the existing device modeling code for reverse engineering is needed. And, as with pre-detonation, a database of standard materials must be both strengthened and centralized. [16]

2.1.1.2 Pre-Detonation

Pre-detonation nuclear forensics investigations occur for every scenario in which non-detonated nuclear material has been found or intercepted. Although this could be an intact bomb, it is more likely that SNM intended

for a weapon would be the target of an investigation. Thus, the range of intact materials for measurement could be as small as one fuel rod. The goal is to determine the provenance of the SNM, which is generally done by reconstructing the irradiation process that created the material.

For SNF, where the material was obtained is the first step of the investigation. This would be gleaned from the reactor parameters and storage history (e.g., reactor type, cooling time, burnup), which requires first measuring and calculating certain values: isotopic ratios, concentration of chemical compounds, or existence of trace elements. Both radiological methods (e.g., gamma spectroscopy) and ionization methods (e.g., mass spectrometry) measure these quantities.

Although this is less of a humanitarian emergency than a post-detonation investigation, it is still important to have rapid characterization capabilities via on-site non-destructive analyses. As previously discussed in more detail in Section 1.1, however, the faster measurements result in poor measurement quality. Also, there is a need for research to combat the database issues, as an insufficient forensics database can reduce the accuracy and/or certainty of a reconstructed set of reactor parameters. Another area of research is deeper study of known forensics signatures or discovering new signatures with modeling, simulation, or statistical methods.

The 'Real World Methodology' section in Figure 2.1 shows a typical workflow for the technical portion of a pre-detonation forensics investigation. After a sample is obtained, characterization begins. Next, the results of these

techniques are then compared against existing standard materials databases to obtain the desired reactor parameters. These steps would be performed iteratively in a real investigation, first using non-destructive measurements, and destructive measurements last. The following steps in Figure 2.1 are seeking out reactor history information, if available, and reporting all results to the investigators.

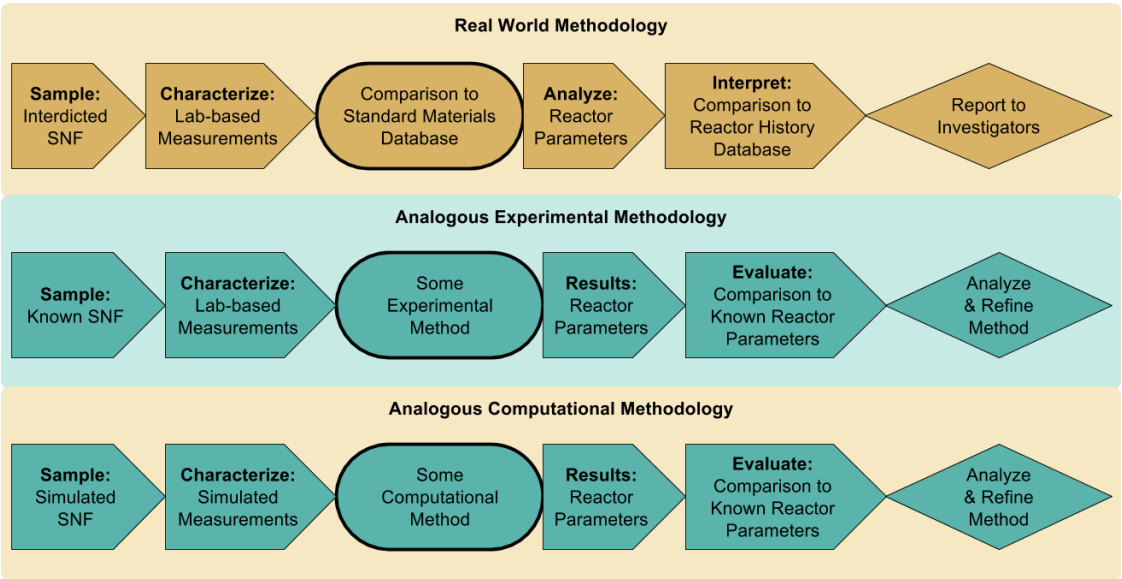


Figure 2.1: Example Forensics Workflows in Real-World Scenario

Below the example of a real-world workflow in Figure 2.1 are analagous experimental workflows, both physical and computational. For researchers studying alternative measurement techniques or a slight difference in the overall approach, it is necessary to iterate through multiple studies using known materials to probe sensitivities or other weaknesses in the procedure.

Fix up diagram to be more aligned with discussion

2.1.2 Nuclear Forensics as an Inverse Problem

Nuclear forensics is a traditional inverse problem, which has been well documented in mathematics and many scientific disciplines. Understanding inverse problem theory can help systematically define both the solution methods and their limitations. This section provides an introduction to the topic as well as its application to nuclear forensics.

As outlined in a textbook on the formal approach to inverse problem theory [30], the study of a typical physical system encompasses three areas:

1. *Model parameterization*
2. *Forward problem:* predict measurement values given model parameters
3. *Inverse problem:* predict model parameters given measurement values

First, this shows that it is important to consider the parameters that comprise a model; this is denoted as the *model space*. This is not every measurable quantity; domain knowledge is necessary to determine the model space. In the nuclear forensics context for SNF, this would consist of, e.g., several isotopic ratios because they are known to have a relationship with the reactor parameters that created the fuel of interest.

Second, understanding the physical system also requires an understanding of the forward problem: predicting how a certain set of model parameters will affect the resulting measurements. The breadth of these end measurements provides the *data space*, which are all the conceivable results of a given

forward problem. So for SNF this would be, perhaps, the range of isotopic ratios typical of a commercial reactor.

Lastly, the inverse problem is statistical in nature: given some solution, there is a probability that the data measured is caused by some value(s) of a model parameter. Including measurement uncertainties broadens the linear model to probability densities of the parameters. The opposite is also true in the forward case: including parameter uncertainties broadens the forward problem results to probability densities of the potential measurement values. [30]

In this way, we can define some probability that the answer is correct, given a set of measurements and their uncertainties. Inverse problem theory states that this follows the general form of Bayes' theorem, which is commonly expressed as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

where A and B are events, $P(A)$ and $P(B)$ are the probabilities that events A and B will occur, respectively, $P(B|A)$ is the likelihood that event B will occur given a known result for A, and $P(A|B)$ is the posterior probability that event A will occur given a known result for B.

This is can be mapped easily to the inverse physical system problem scenario. A would represent an occurence of a parameter in the model space, and B would represent the measurement of some value. Thus, $P(A)$ is the probability of a parameter existing without any knowledge of B. This is

known as the prior probability, usually given by some theory about the system. $P(B)$ is the probability of some measurement existing without any knowledge of A. This is known as the marginal likelihood, which is some homogeneous concept for the potential measurements that could be made (this only serves to scale to absolute probabilities and does not affect the relative probabilities). The likelihood, $P(B|A)$, is the chance that a measurement is observed from a given parameter, representing the forward problem. Lastly, the posterior probability is the chance of some parameter existing given some measurement, representing the inverse problem solution [29, 30]. It may be more intuitive to consider the conceptual version of Bayes' theorem below. A discussion of how these values are obtained takes place in Section 2.2.3.3.

$$Posterior = \frac{Likelihood * Prior}{Marginal\ Likelihood} \tag{2.2}$$

This framework is helpful for an experiment that intends to compare different methods for calculating the posterior probability of a system given some measurements [31]. In the nuclear forensics context of pre-detonated materials, this would be a a set of probabilities for different parameters of interest, e.g., reactor type, burnup, cooling time, and enrichment of some interdicted SNF.

2.2 Machine Learning

Machine learning is a sub-field of artificial intelligence (AI) within the broad category of computer science. The goal of AI is to create computer systems that respond to their environment according to some set of criteria or goal. For example, self-driving vehicles have computers on board that learn to avoid curbs and humans. While its use has been increasing in the commercial sector, there is also much anecdotal evidence to support the existence of a rapid increase of AI use in academic research across many disciplines beyond robotics. AI systems have been used in detection (e.g., fraud or spam), medical diagnostics, user analysis (e.g., Netflix ratings), and a host of scientific disciplines that have increasing amounts of multivariate data.

Much of the recent advances to the field of AI have occured in the statistical realm, which forgoes domain knowledge in favor of large data sets. Thus, machine learning and statistical learning have become somewhat separate fields [13]. Machine learning research focuses on the underlying algorithms using mathematical optimization, methods for pattern recognition, and computational statistics. Since this work is only an application of machine learning, there is no distinction made between the terminology. Additionally, this study is not concerned with computational time, but rather the ability to correctly predict values and categories relevant to the nuclear forensics mission. This restricts the relevancy of the algorithms to the underlying theory and its impact on the resulting model's accuracy.

Machine learning algorithms can be separated into two main categories: unsupervised and supervised learning. The former groups or interprets a set of input data, predicting patterns or structures. The latter includes both the input and output data, enabling the trained model to predict future outputs. Broadly speaking, the unsupervised learning algorithms are designed for clustering data sets or dimensionality reduction (i.e., determining some subset or linear combination of features most relevant to the input data) of data sets. Supervised learning algorithms predict both discrete and continuous values via classification and regression, respectively. Some algorithms can perform both classification and regression, and neural networks can even be modified to perform either supervised or unsupervised learning.

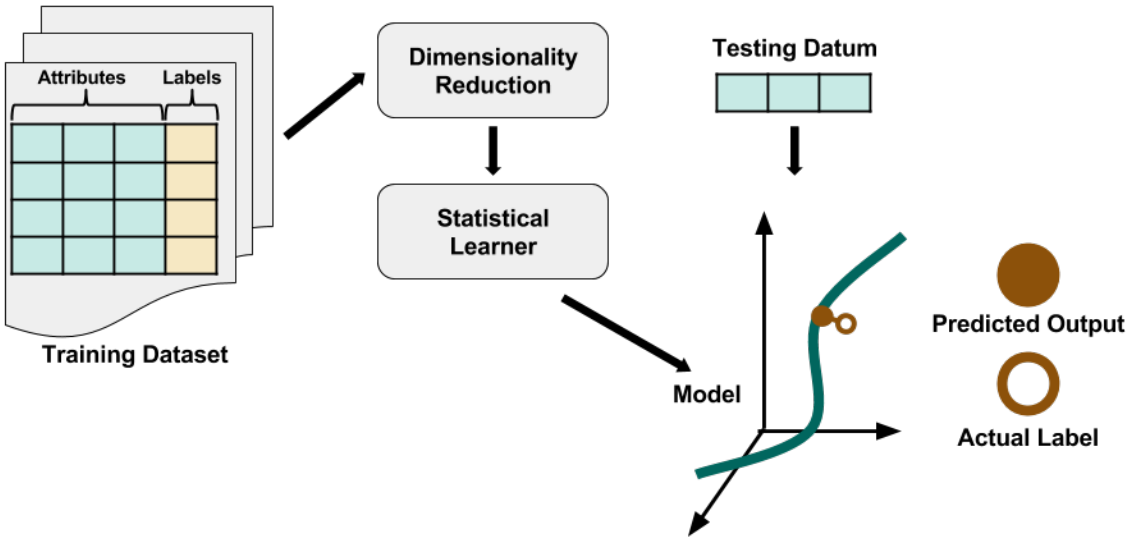


Figure 2.2: Schematic of Regression with Machine Learning

As shown in Figure 2.2, a typical (supervised) machine learning workflow begins with a training data set, which has a number of *instances*, or rows of observations. Each instance has some *attributes*, also referred to as *features*, and a label, which can be a categorical label or discrete/continuous values.

The training data are then inserted into a statistical learner; this calculates some objective, minimizes or maximizes that objective, and provides some model. This model is typically evaluated using a testing set that has the same set of attributes and labels (but different instances). The comparison of what the model predicts and the actual label gives the *generalization error*. Depending on the performance and application, the model may need improvement from more training and/or some changes in the algorithm parameters. Once the model is performing well enough and validated, it is finalized; then a user can provide a single instance and a value can be predicted from that.

This study performs regression tasks using supervised learning algorithms. Differences among the underlying mathematics of the algorithms impact the trained models. Therefore the algorithms used in this study will be discussed in Section 2.2.1. Next, model selection and assessment is covered in Section 2.2.2. Evaluating and optimizing algorithm performance is discussed in Section 2.2.3, as well as robustly comparing different algorithms for validation.

2.2.1 Algorithms for Statistical Learning

For relevant nuclear forensics predictions, both classification and regression algorithms must be used. For example, one may want to predict the reactor type based on some measurements (referred to as features) of SNF of an unknown source, and this would require a classification algorithm. Or perhaps the input fuel composition is relevant to an investigation on weapons intent, so a regression algorithm would be used to train a model based on some set of features. Since this work trains models to predict burnup of SNF, the algorithms are presented in a regression context.

2.2.1.1 Linear Models

One of the simplest and most utilized methods of prediction is a linear model from a least-squares fit. Thus, it is the most natural place to begin a demonstration of machine learning algorithms. Since linear models must have all parameters linearly related, there are many restrictions regarding the shape of the model. However, this makes the resulting models stable to peturbations.

An example of a linear model is in the equation below, where the vector of input features of size p, $\boldsymbol{X}$, provides a model, $F(\boldsymbol{X})$ ($\boldsymbol{Y}$ is the vector of known data), by determining the unknown coefficients, or weights, β_j's. The reword algorithm calculates these by minimizing the value of a loss function over all the training data. This is usually the least squares error from minimizing the sum of squared errors, $\sum_{i=1}^{n}(y_i - f(x_i))^2$. But it could instead be the least

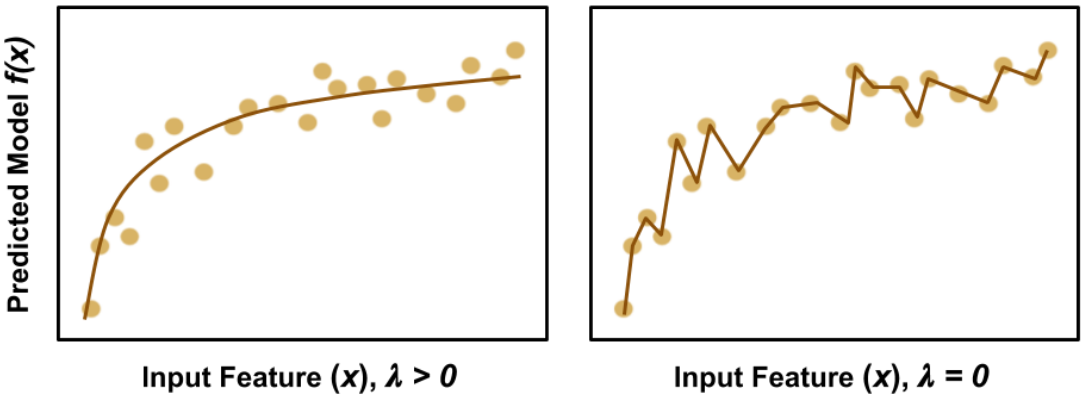


Figure 2.3: Effect of Regularization on Prediction

absolute deviations from minimizing the sum of absolute error differences, $\sum_{i=1}^{n} |y_i - f(x_i)|$. These are referred to as the L_2 and L_1 norms, respectively.

$$F(\boldsymbol{X}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j \tag{2.3}$$

The form of linear regression used here is called ridge regression. This algorithm performs optimization using the L_2 norm, and also uses the form of the L_2 norm for *regularization*. Regularization, sometimes called *shrinkage*, is a term introduced into the predicted model to prevent overfitting. (It is used in many machine learning algorithms.) This works by further reducing the weights, β_j, on the input features, x_j. This shrinkage term also includes a complexity parameter, λ [6]. Thus, the predicted linear model from ridge regression is updated to the following equation. A visualization of what this regularization can accomplish is in Figure 2.3.

$$F(\boldsymbol{X}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.4}$$

2.2.1.2 Nearest Neighbor Methods

Nearest neighbor regression is considered to be 'model-free' in that it does not actually generalize; it tracks the observations in the training set. During prediction, the algorithm will calculate a value based on the instance that is closest to the current test sample. Thus, there is not any learning, but instead a direct comparison between an unknown sample and the space that the training set populates. The predictions from nearest neighbors can be quite accurate, but are highly unstable to peturbations [6].

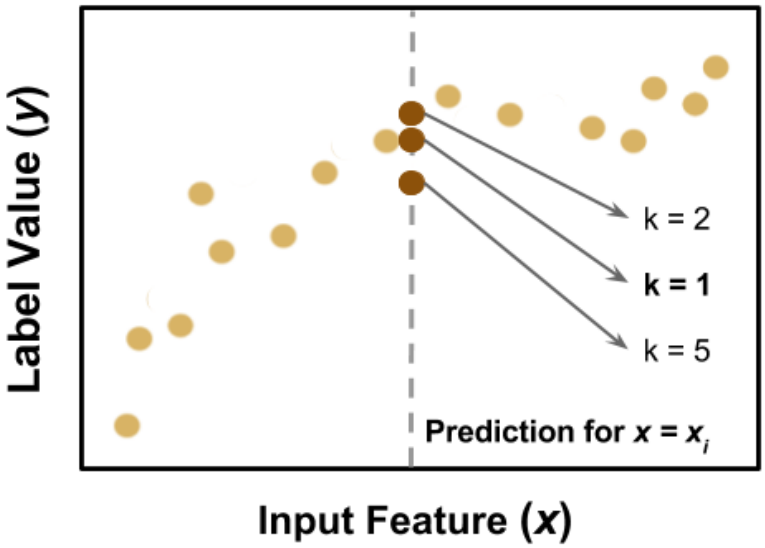


Figure 2.4: Schematic of k-Nearest Neighbors Regression

An extension of nearest neighbor is k-nearest neighbor regression. The closest k neighbors are averaged to provide an estimate of the unknown sample as shown in the equation below. Figure 2.4 provides a pictoral explanation of how this is done for a single prediction. The equation below

shows how this algorithm does this to determine a predict a value, Y, from the input features, $\boldsymbol{X}$, in the neighborhood, $N_k(\boldsymbol{X})$ [6].

$$Y(\boldsymbol{X}) = \frac{1}{k} \sum_{x_i \in N_k(\boldsymbol{X})} y_i \tag{2.5}$$

A measure of distance dictates what the neighborhood is. The metrics for distance differ, but in this study, the Euclidian distance was used. Another parameter is the population of the neighborhood, k. In this initial work, $k = 1$ is used. This can perform very well, but can also easily overfit the data and thus not generalize well.

2.2.1.3 Support Vector Machines

Support vector regression (SVR) is an extension of the popular classification algorithm, support vector machine (SVM). This algorithm was chosen because of its ability to handle highly dimensional data well, which in this study is a maximum of approximately 300 features.

As seen in Figure 2.5[1], the SVM algorithm separates two classes by determining an optimal hyperplane between them. The algorithm evaluates the quality of the line that separates the two classes by maximizing the margin width, $m_w = \frac{2}{\|w\|}$. The hyperplane is defined in the equation below, where $\boldsymbol{w}$ is the vector that is normal to the hyperplane.

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \tag{2.6}$$

[1]This schematic is based on the tutorial on SVMs by Dr. Saed Sayad at http://www.saedsayad.com/support_vector_machine.htm

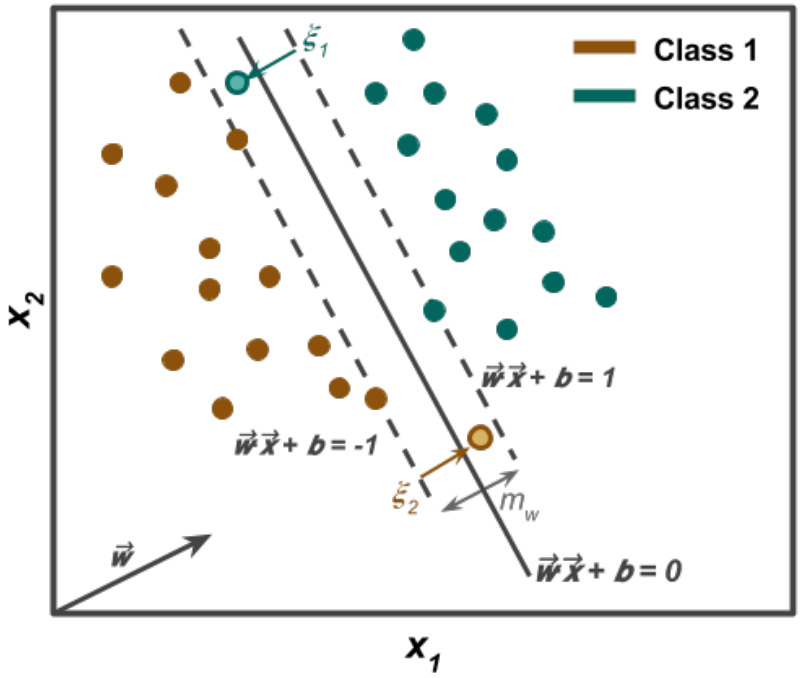


Figure 2.5: Schematic of SVM Classification

Figure 2.5 also shows a case of soft margins. Some problems are not linearly separable, and thus a penalty term, ξ_i, is introduced to allow for some misclassifications. The algorithm then simultaneously minimizes the misclassifications while maximizing the margin. The objective function of the algorithm below shows this as well (using quadratic programming). In the equation below, C is responsible for the margin width/misclassification tradeoff.

$$\begin{aligned} min\ & \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ subject\ to:\ & y_i(wx_i + b) > 1 - \xi_i \end{aligned} \tag{2.7}$$

Figure 2.6[2] demonstrates how SVM can be altered slightly from classification to nonlinear regression with SVR. SVR has a similar objective function but instead *minimizes* the margin, as shown in Figure 2.6a. This changes the constraints to the objective function as below:

$$\begin{aligned} &min\ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ &subject\ to:\ \ |y_i - (wx_i + b)| \leq \varepsilon + \xi_i \end{aligned} \tag{2.8}$$

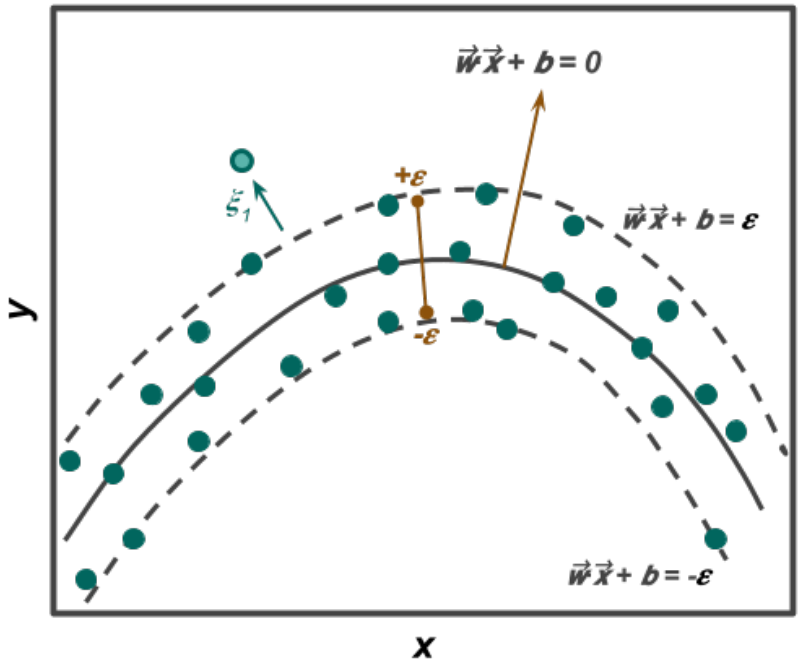
Further, this can be extended to nonlinear analysis via what is called the *kernel trick*. First, using a nonlinear kernel function maps the data into higher dimensionalfeature space. Then the algorithm can find a linear separation in this space, as shown in Figure 2.6b.

check wording

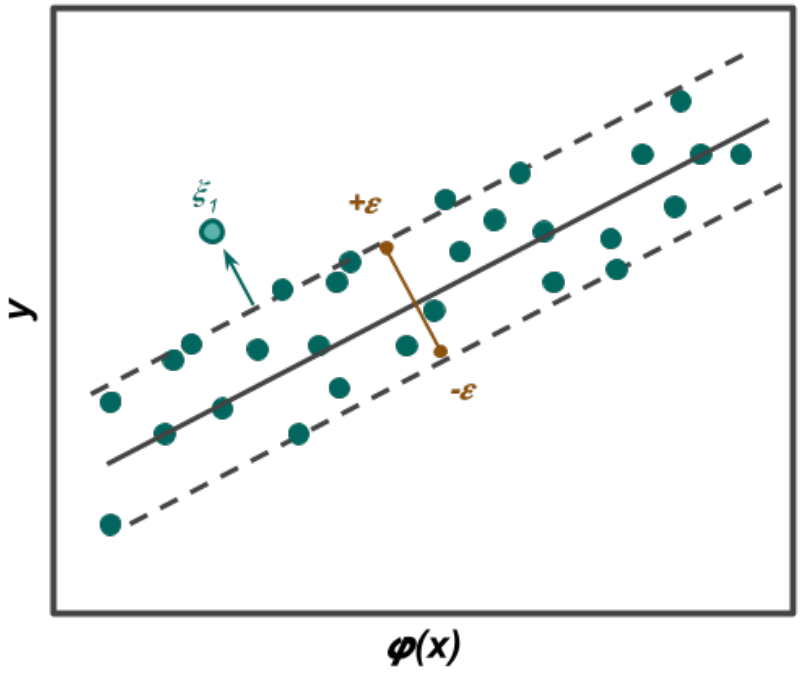
The kernel chosen for this study is the Gaussian radial basis function, shown below. This has two tuneable parameters, γ and C. The γ controls the width of influence of individual training instances, which strongly affects the fitting of the model. Low values correspond to underfitting because the instances have too large of a radius (low influence) and high values correspond to overfitting because the instances have a small radius (high influence).

The C parameter also affects the fitting of the model by allowing more or less support vectors, corresponding to more or less misclassification, respectivey. A lower C smooths the surface of the model by allowing more misclassifications, whereas a higher C classifies more training examples by

[2]These schematics are based on a tutorial on SVR by Dr. Saed Sayad at http://www.saedsayad.com/support_vector_machine_reg.htm



(a) Demonstration of regression with SVR



(b) The kernel trick with SVR

Figure 2.6: Illustrations of SVR and Nonlinear Analysis

allowing fewer misclassifications. Thus, too low or too high of a C can cause under- or overfitting, respectively.

Since there is a tradeoff of fitting strength provided by both parameters, it is common to run the algorithm on a logarithmic grid from 10^{-3} to 10^{3} for each parameter. If plotted on a heatmap of accuracies given γ and C, there will be a diagonal of ideal combinations that emerges. The element with the lowest of each parameter is usually chosen.

2.2.1.4 Dimensionality Reduction Techniques

In addition to utilizing various algorithm parameters for regularization as discussed above, dimensionality reduction can improve generalizability by removing the noise of features that do not affect the regression task. This can be thought of in the following way: shrinkage techniques reduce the weights of noisy features, whereas dimensionality reduction removes them completely. Although one could use domain knowledge to manually reduce the number of features in a data set (e.g., only including certain nuclide subsets such as actinides), statistical feature reduction may also prove helpful in this work. The mathematical treatment of the methods described below are in Ref. [6].

Principal Components Analysis

Principal components analysis (PCA) is considered the most common dimensionality reduction technique. The PCA algorithm learns a linear

transformation of a data set $\boldsymbol{X}$ with which to construct a transformation matrix according to a user-chosen number of variables/components. This matrix is part of the singular value decomposition of the data matrix $\boldsymbol{X}$. The decomposition step is what provides the principal components, which are the result of maximizing the variance in the original data while minimizing the squared reconstruction error between the original data and the transformed data. The mathematical assumptions are that the variables are all Gaussian and uncorrelated.

Because the principal components are obtained purely statistically with no model assumptions, they are usually uninterpretable. However, they can still provide clues with which to obtain new information. In the case of this work, this could provide insight into new forensics signatures.

Factor Analysis

Factor analysis is similar to PCA in that it also calculates linear combinations of the data set features using the above-mentioned decomposition and the mathematical assumptions are the same. It is different in that the decomposition is rearranged so that it represents *latent variables* including random error disturbances rather than principal components. Latent variables are constructed by maximizing the correlation/shared variance among the variables rather than the total variance. However, different optimizations can be chosen so that the solutions are parameter-dependent.

Furthermore, the initial model assumptions, while enabling interpretable

results, also increase the dependence of the solutions on the algorithm inputs. If PCA does not perform well with the type of training data in this work, it is possible that factor analysis will, since there are many nuclides in the training data set that are in a decay chain together.

Independent Components Analysis

Independent components analysis (ICA) has characteristics of both factor analysis and PCA, but with the goal of finding independent measurements from multiple 'sources'. It uses the same form of decomposition as factor analysis but without the inclusion of random error. The mathematical assumptions are a bit different: the variables are statistically independent and non-Gaussian. This allows the algorithm to minimize higher-order statistics of the data set (PCA and factor analysis only minimize the first two orders).

Since the independent components are useful in signal processing of multiple signals, this technique could prove useful for reprocessed nuclear materials where multiple source streams converge.

2.2.2 Model Selection and Assessment

After a model is trained, the first step is model selection and assessment. Selection is estimating model performance among a set of trained models using a single validation set. After one model is chosen, assessment takes

place by determining the prediction capability on new data via a previously unseen testing set. Both selection and assessment can be done in a single step using k-fold cross-validation, which is described below.

2.2.2.1 Sources of Error

In statistical learning, there are two sources of error that need to be simultaneously minimized: bias and variance. Bias is caused by simplifications in the model, so the error is caused by missed relationships in the data; an underfit model is due to high bias. Variance is caused by including random noise in the model, so the error is caused by oversensitivity to that noise; an overfit model is due to high variance.

As shown in Figure 2.7, the shape of the total error curve shows that there is a tradeoff between the bias and variance that can be minimized. Some bias is desired in order to generalize to future unknown data. But some variance is also positive for the model because it captures the relationships in the data that the bias counteracts.

2.2.2.2 Types of Error

While the sources of the model prediction error are well known, the creation of a statistically learned model is a hidden process. Although the model emerges from a black box, there are ways to evaluate the generalization (i.e., prediction) capability of it. This is done by removing a small portion of

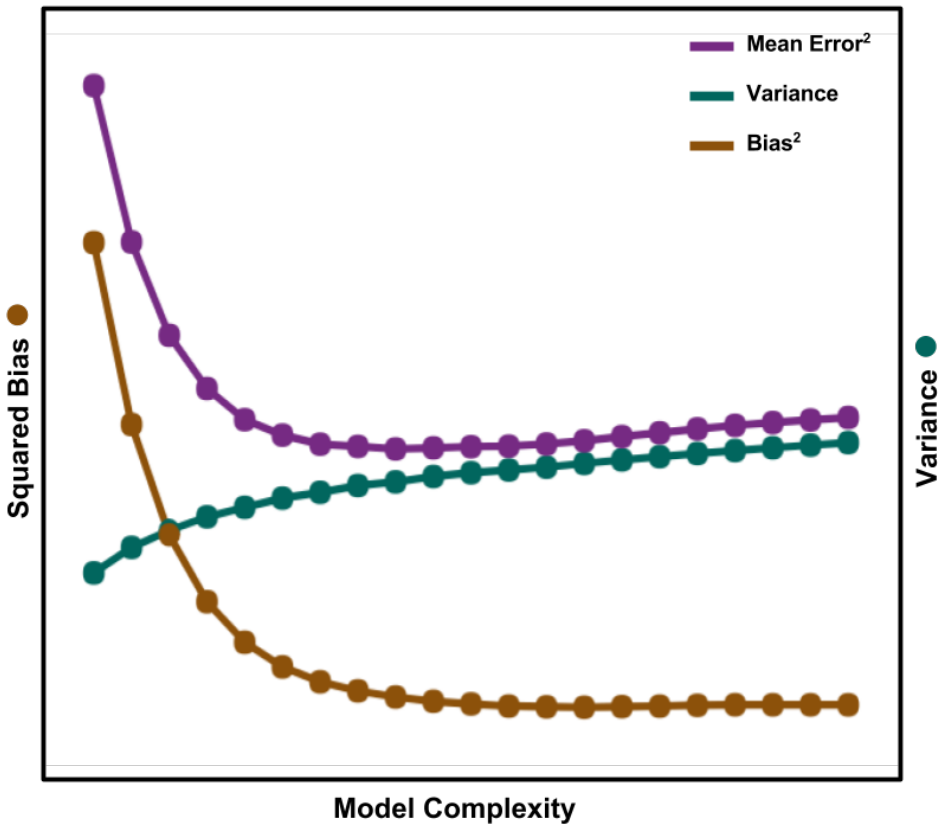


Figure 2.7: Bias and Variance Comprising Prediction Eror

the data for use as a testing set. The rest of the data set is known as the training set and is used to train a model. After training, the test set is used to test the model.

The generalization error is typically referred to as the *testing error*, as it is measuring the ability of the model to predict future cases that were not introduced in the training phase (i.e., the testing set entries). Next, the *training error* is provided by comparing the model predictions to the training set, as the model would likely be smoother than the potential noise

the training set would include. This is useful to determine the fitness of the model, the application of which is discussed below in Section 2.2.3.

Although one could just train and test their model, there is a way to test the model while still in the training phase. A testing set that would be used during training to give feedback, a *cross-validation* set, can provide a faster convergence to a satisfactory model. As shown in Figure 2.8, this can be done by splitting the data set into three groups: a large training set, a small cross-validation set, and a small testing set.

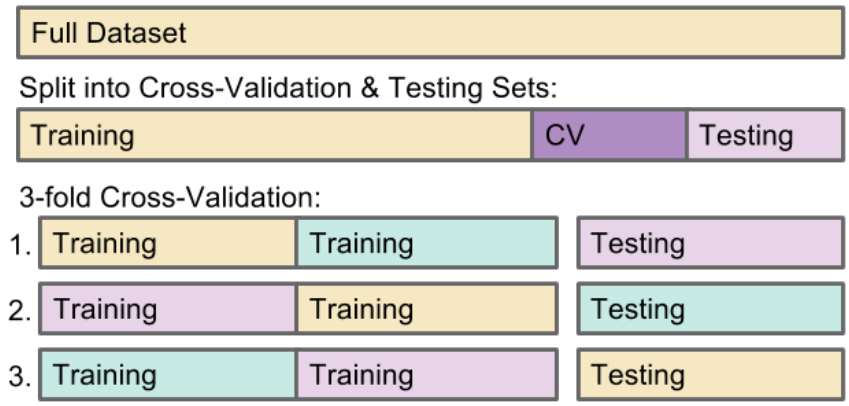


Figure 2.8: Illustration of Cross-Validation

However, in practice, multiple rounds of cross-validation steps are used, referred to as *k-fold cross-validation*. This allows a user to use all data entries as a testing entry once. As illustrated in Figure 2.8, this splits the dataset into k subsets. One set is designated as the testing set, and a model is trained with the rest. Following the first training phase, another begins, this time with a different subset as the testing set. This process is performed

k times to give k models. The models are 'averaged' by taking the mean of the predictions in the case of regression or voting in the case of classification. This provides an additional level of model validation than can be achieved with a single testing set.

2.2.3 Model Optimization and Validation

It is unlikely to have a model perform as one expects the first time. There are therefore a few techniques for optimizing the performance. It should be noted that much of the discussion here and in Section 3.3 focuses on the diagnostics aspect rather than the validation aspect of these techniques. In practice, these are used for both purposes, but in this work the formal comparison of model performance will be used, introduced and detailed in Sections 2.2.3.3 and 4.5, respectively.

However, the increase in performance from over-optimization could be linked to the training set performance and might not generalize outside of the specific type of input data used. A workaround for this scenario is to obtain more data for the set or to obtain a completely different data set altogether.

2.2.3.1 Training Set Size

The first diagnostic plot for optimizing the model performance is called a *learning curve*, which provides information about the bias-variance tradeoff with respect to the data set size. More specifically, learning curves compare

the training and cross-validation errors to the size of the training set (i.e., number of instances in the training set). This is done by randomly selecting a percentage of the the training set, inputting that into a statistical learner, and tabulating the error of the learned model.

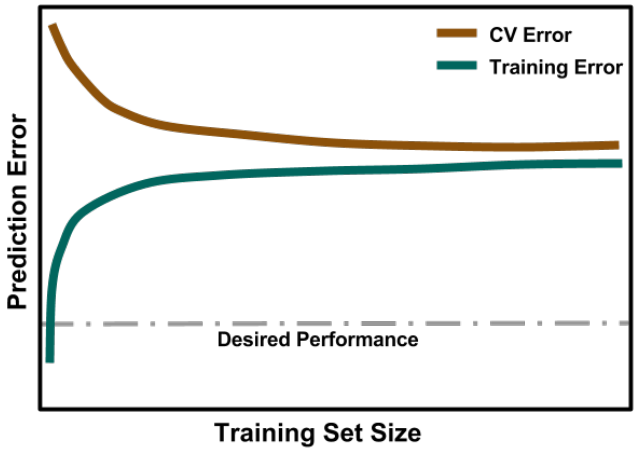
Typically, a learning curve will look somewhat like one of the three examples in Figure 2.9[3]. A learning curve tests the model for high bias or high variance, which can correspond to an under- or overfit model, respectively.

Figure 2.9a suggests underfitting because the model is missing important features in the data. It is characterized by a small gap between the curves but high overall errors. The cross-validation error remains consistently high and the training error increases drastically with increasing data, since it is not generalizing well.

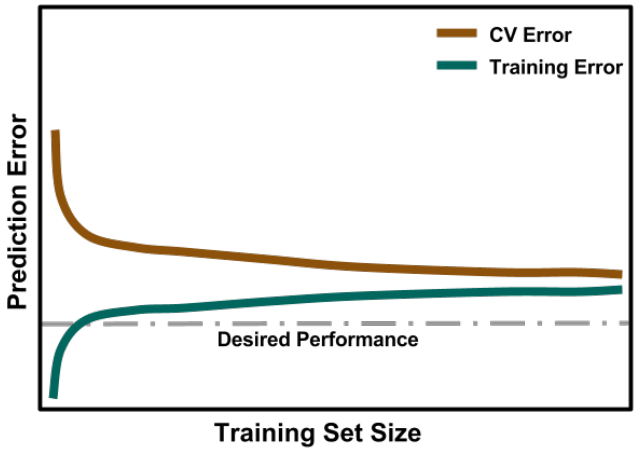
Figure 2.9c suggests overfitting because the model has too much sensitivity to variations in the data. It is characterized by a very large gap between the curves. It has an extremely low training error, as it has taken into account every detail of the training set, but a high cross-validation error because it cannot generalize beyond the testing set.

Figure 2.9b is an example of a more ideal model fit. It is characterized by a small gap between the two errors, and they are at a reasonable level with respect to the desired performance. The training error should increase with respect to the training set size due to a larger amount of bias (preventing

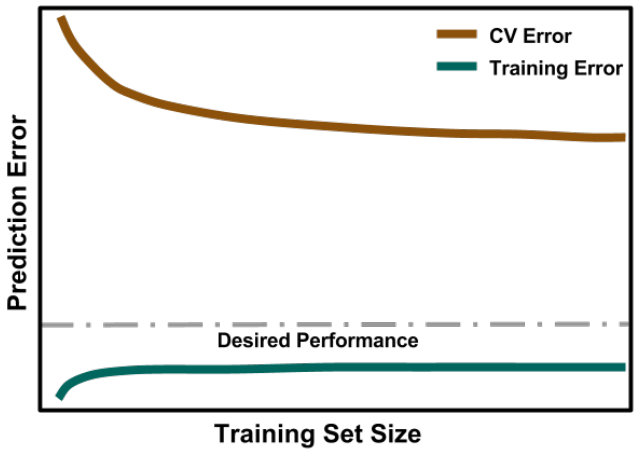
[3]These schematics are based on hand-drawn diagrams by Ritchie Ng on http://www.ritchieng.com/applying-machine-learning/



(a) High bias



(b) Ideal



(c) High variance

Figure 2.9: Learning Curves for Three Training Scenarios

overfitting). However, the cross-validation error should decrease quickly with respect to the training set size due to being close to the minimum of the bias-variance tradeoff.

add description of i.i.d. training set

2.2.3.2 Model Complexity

After ensuring the appropriate training set size is selected, the models must be further optimized using *validation curves*. These provide information on the bias-variance tradeoff with respect to model complexity. Two main factors affecting model complexity can cause the model to be under- or overfit to the data: number of features in the data set and algorithm parameters that vary the regularization.

Figure 2.10 adapted from Ref. [6] shows the optimum as the minimum of the cross-validation error curve. There is some gap between it and the training error, much larger than the left side of the plot and much smaller than the right. The plot above is a visualization of an approximately well-fit model. The left region is marked by both errors being quite high, and above is an illustration of how an underfit plot (high bias) could provide high errors. The right region shows the training error being quite low but the cross-validation error being high. The diagram above shows that it is obvious how the training error would be negligible, but generalizing beyond that probably will not yield accurate results.

In practice, plotting learning and validation curves can be iterative. But

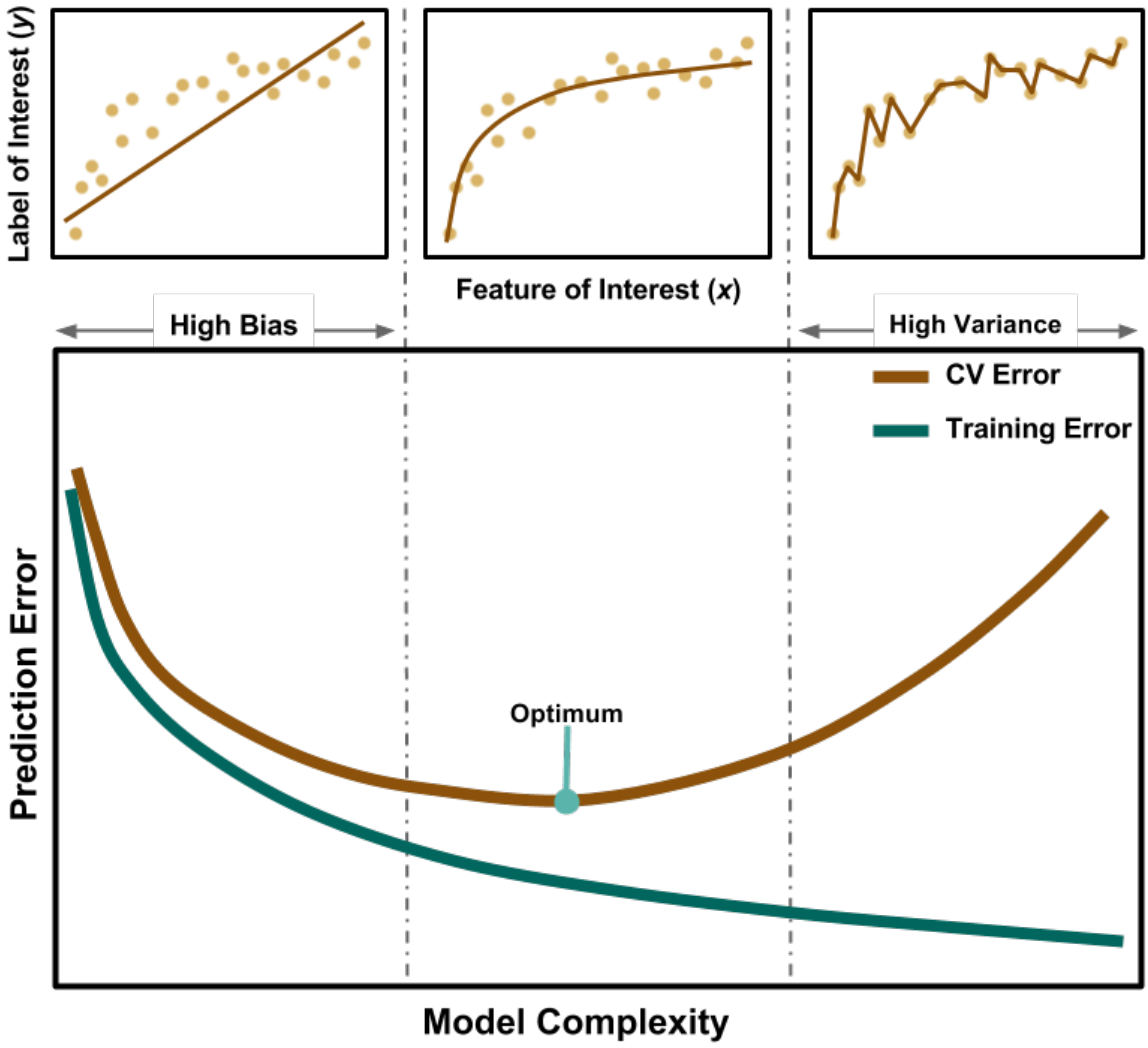


Figure 2.10: Validation Curve with Different Model Fits

as previously mentioned, too many optimizations will result in a poorly performing model when exposed to data outside of the training set.

2.2.3.3 Comparison of Methods

In addition to evaluating a single learned model, it will be beneficial to compare models. Moreover, there are potential degeneracies in the solution space. This is because most inverse problems are *ill-posed*, usually because the solution is not guaranteed to be unique [28].

This can be done using Bayesian inference as discussed in Section 2.1.2. Equations 2.1 and 2.2 show that there are three values to obtain to calculate a posterior probability, i.e., the probability of a parameter estimated from a machine-learned model being correct: the likelihood, the prior probability, and the marginal likelihood. Comparing posterior probabilities among different methods reveal the method with the most probable correct answer, i.e., the highest posterior probability. [29, 30] Each value is explained futher below.

The posterior probability represents the solution to an *inverse* problem, where model parameters are predicted from some given measurement values. It is not directly computable and thus the remaining probabilities discussed below indirectly allow its computation. In this context, it is the probability that a predicted reactor parameter from a chosen method is correct given input based on an SNF recipe. For example, it is the probability that a plutonium-239 concentration of $y\%$ is attributed to a uranium oxide fuel in a boiling water reactor (BWR) with a burnup of $x\ GWd/MTU$.

The likelihood represents the solution to a *forward* problem, where measurement values are predicted from some given model parameters. In

this context, this is calculated from the instances in the training data set. The likelihood is the probability that the output SNF composition of a simulation is correct given the input of reactor operation parameters. In practice, it will be calculated from a large number of forward simulations using ORIGEN, i.e., the training data set. For example, this would be the probability that uranium oxide fuel from a BWR having a burnup of $x\ GWd/MTU$ contains $y\%$ of plutonium-239.

The prior probability represents the spread of plausible *model parameters*, so it is a hypothesis based on the breadth of the model space with no evidence provided. In other words, it is given by model parametertization from a number of potential sources. One method is expert-elicited values. Another is a predicted model from some established theory or previously known relationship, e.g., empirical relations between isotopic ratios and certain reactor parameters or a direct calculation of the reactor parameters. In this context, it is obtained from the estimated model parameters from a given statistical model. For example, it is the probability that a model from SVR predicts a burnup of $x\ GWd/MTU$ for uranium oxide fuel from a BWR with no other direct measurements provided.

The marginal likelihood represents the spread of plausible *measurements*, so it is based on the breath of the data space with no model-based information provided. In practice, however, it is calculated by summing the joint probabilities of all possible model parameter hypotheses and measurements. This in essence provides a normalization constant. Thus, the marginal

likelihood is only needed for absolute posterior probability calculations; it does not affect the relative probabilities, which are all that is needed for model comparison. [30, 31]

For reference, Table 2.1 is a summary of the Bayes' theorem components described above as related to this work. In the table, **P** is a probability, **D** is a set of measurements (i.e., data), and **M** is a set of model parameters. Examples of all of these quantities are also provided.

2.3 Computational Methods

There can be a large number of computational tools within a given field of study. They are typically developed with one aspect of analysis in mind, so one must understand the simplifying assumptions the developers made for their purposes. Thus, choosing a computational tool is not always obvious. Thus, this section both introduces and defends the choices for the various experimental components.

2.3.1 Fuel Cycle Simulation

Nuclear fuel cycle studies involve tracking the material flow of nuclear fuel. This can be anywhere from mining to waste management, or focus on a process step anywhere in between. Fuel cycle studies are not necessarily nuclear-specific. They can be used to evaluate, e.g., economic predictions, environmental impact, transportation planning, etc. In order to draw

Probabilities	Examples
P(D|M) Likelihood	*Given:* M = BWR U-oxide with burnup = ***x*** GWd/MTU *This is true:* D = nuclide vector with Pu-239 = ***y***% *With:* ***z***% probability *Calc'd from:* ORIGEN simulations in training set
P(M) Prior	*Given:* No direct information on D *This is true:* M = BWR U-oxide with burnup = ***x*** GWd/MTU *With:* ***z***% probability *Calc'd from:* Machine-learned model prediction
P(D) Marginal L.	*Given:* No direct information on M *This is true:* D = nuclide vector with Pu-239 = ***y***% *With:* ***z***% probability *Calc'd from:* Summation of training set instances
P(M|D) Posterior	*Given:* D = nuclide vector with Pu-239 = ***y***% *This is true:* M = BWR U-Ox with burnup = ***x*** GWd/MTU *With:* ***z***% probability *Calc'd from:* All quantities above

Table 2.1: Summary of Bayes Theorem Components

conclusions from these studies, it is common to use a nuclear fuel cycle simulator that tracks the quantities of interest. These allow the comparison of different fuel types, reactor technologies, material processing steps, etc.

As mentioned, there are simplifications researchers need to make in order to experiment in a controlled way. Fuel cycle simulators are built for specific needs as well and also must remove complicating factors that are generally less relevant to the study. For example, one tool might be suited well to large-scale systems analysis with little nuclear physics included in the models, and another might focus on detailed isotopics within a system to track plutonium isotopes.

more needed?

Because a large portion of a nuclear forensics investigation relies on measuring isotopics, ORIGEN-Automatic Rapid Processing (ORIGEN-ARP) [25] within the SCALE code system was chosen for its physically detailed models. ORIGEN calculates time-dependent nuclide concentrations (or quantities derived from these) that result from activation and depletion calculations. The physics (i.e., neutron transport and decay) calculations are carried out in other SCALE modules that solve the depletion equations. This generates libraries for ORIGEN that include the probabilities of reaction (i.e., cross sections) for the system.

To obtain an SNF recipe from a reactor simulation, ORIGEN uses the desired input power generation with the cross section library to calculate a flux, the resulting depletion, and the end composition (i.e., isotopics, nuclide

vector). Another output is decay; the composition is computed using decay equations with nuclear data [2]. These compositions provide source terms for other calculations, such as decay emission spectra from neutrons, or alpha, beta, gamma rays, which use the same resource for nuclear data. Other quantities like activity, decay heat, or radiological hazard factors are also an option.

ORIGEN-ARP allows users to access a wider range of simulations by interpolating between the already calculated libraries for a wide array of reactors instead of creating new libraries. It is known to be validated for light water reactor (LWR) SNF [15]. Additionally, recycled SNF in the form of mixed oxide fuel has been benchmarked with the relevant reactors [4]. Thus, given an initial material composition, some reactor operation parameters, and a reactor type, one can quickly perform many different nuclear reactor simulations and obtain SNF recipes.

2.3.2 Statistics Toolkit

The statistics toolkit chosen for this work is scikit-learn [24], a machine learning package in python. Virtually all modern machine learning toolkits will have acceptably fast and reliable algorithms, but the use of python provides a platform for seamless integration of all the tools in the workflow.

2.3.3 Computational Gamma Spectra

Although the data modification shown later in Section 3.1.2 does not use any standalone tool, the code GAmma Detector Response and Analysis Software (GADRAS) [7] developed at Sandia National Laboratories will provide information reduction in a physically valid manner. Test studies will be carried out using the gamma spectra available from ORIGEN simulations, but the detector response will need to be varied more than this tool can provide. Thus, given the nuclide vectors from ORIGEN, GADRAS computationally generates gamma spectra from the gamma energies and a chosen detector response function (DRF). This will enable a more robust study of statistical performance with respect to information reduction.

2.4 Applications of Statistical Methods to Nuclear Forensics Analysis

Although the body of literature on the area of proposed research is not expansive, there have been a number of relevant studies on the prediction of forensically relevant categories or quantities of nuclear materials using statistical methods.

2.4.1 Special Nuclear Materials Studied

With regards to broader forensics capabilities, materials from different steps of the nuclear fuel cycle are must be studied. Even though each material has its own forensics signatures, the process of applying statistical methods to the analysis of material provenance is similar for each.

For example, on the front end of the fuel cycle, an entity may have obtained UOC if they have enrichment capabilities. One study performed statistical analyses on UOC from 21 sources (throughout seven countries) using 30 concentration measurements of various elements, isotopes, and compounds, e.g., sodium, magnesium, thorium, uranium-234, or halide compounds [27]. The goal of classifying the source and the country was reached 60% and 85% of the time, respectively.

On the back end, an organization might have interest in SNF if they have reprocessing capabilities. Or, perhaps already separated plutonium from SNF has been intercepted and needs to be traced. Another study addresses this by performing factor analysis on theoretical separated plutonium from various sources of ORIGEN-simulated SNF based on their composition at the end of irradiation [20]. Since in this study all materials are the same age, five plutonium isotopes (238–242) correctly predicted a test sample. However, taking different times since irradiation and reprocessing into account requires more isotopic measurements.

Note: there was a ton of skew in their data and the correction for that was weak sauce

2.4.2 Statistical Methods Employed

There are statistical methods studies that focus on classification, usually focusing on the prediction of the reactor type for unknown samples [10, 20, 21, 27]. However, this work is focused on burnup prediction as well. Although the results for both regression and classification are based on a number of features that are usually isotopic in nature, it is not clear if the regression counterparts of these algorithms will perform similarly for this task.

Promising regression work using factor analysis has been published [19, 22]. Although factor analysis explicitly requires the input of domain knowledge, it is a valuable first step towards understanding how statistical methods can provide insightful models that predict fuel enrichment and burnup. In the following cases, the features included for the analysis are only the uranium and plutonium isotopes remaining in the SNF. Ref. [19] shows the initial work done on predicting enrichment and burnup from a range of simulated SNF recipes and comparing an 'unknown' sample to the results of the factor analysis. Ref. [22] extends the study to real measured samples from the spent fuel isotopic composition (SFCOMPO) database [17]. This work also highlights and addresses a known problem: reliable discrimination between SNF from pressurized water reactor (PWR)s and BWRs.

The most closely related work to this study involves not only statistical

could also discuss PCA+PLSDA work here?

methods but an investigation of those methods when faced with information reduction via random nuclide measurement errors in the training data set [3]. Additionally, feature reduction was investigated by using various nuclide compositions: the top 300 by concentration, fission products only, and a PCA-derived shortened nuclide list. Three methods were compared: nearest neighbor classification and regression using both L_1 (sum of absolute differences of Cartesian coordinates) and L_2 norms for reactor type classification and burnup prediction, and ridge regression, which uses an L_2 norm for regularization, for burnup prediction. In both prediction cases, using the fission products nuclide list with both nearest neighbor methods performed the best. All other nuclide lists quickly devolved to random guesses with an increase in nuclide error in the case of reactor prediction, and more than 100% error in the case of burnup prediction.

For reliable prediction, it seems to be promising to use actinides [19, 22] and/or fission products [3] for feature reduction using domain knowledge. However, this work intends to still investigate statistical methods for dimensionality reduction, e.g., PCA. This could be beneficial in prediction with different parameters, or could be useful in other ways, such as visualization or discovering new correlations among SNF properties as new reactor technologies are deployed.

3 METHODOLOGY AND DEMONSTRATION

This chapter first covers the methodology of the proposed work by introducing each experimental component and a demonstration of each component. This has been split into three sections, summarized below.

Section 3.1 discusses how the training data is obtained. After the initial training data is simulated in Section 3.1.1, with a possible information reduction step in Section 3.1.2, it will be input to a statistical learner.

Section 3.2 is about algorithms that use the features and labels of the training data to statistically formulate a model. Algorithm choice and parameters are discussed in Section 3.2.1. Next, the main goal for these machine-learned models is to supply reactor parameters associated with some unknown SNF. Section 3.2.2 shows the results of testing this goal: the prediction of a new instance that has only features and no label.

Finally, the algorithms are evaluated for accuracy and validated, as shown in Section 3.3. In practice, validation is more than just making sure the models are properly fit to the data. Perhaps the training set was not representative of the actual data space, whereas other methods do not rely on the data space for results. To both understand the performance and validate the models, the results are then evaluated for over- or under-fitting.

This work incorporates some methods and suggestions from previous work on the subject [3] regarding machine learning model performance with respect to information reduction. This is to establish some baseline

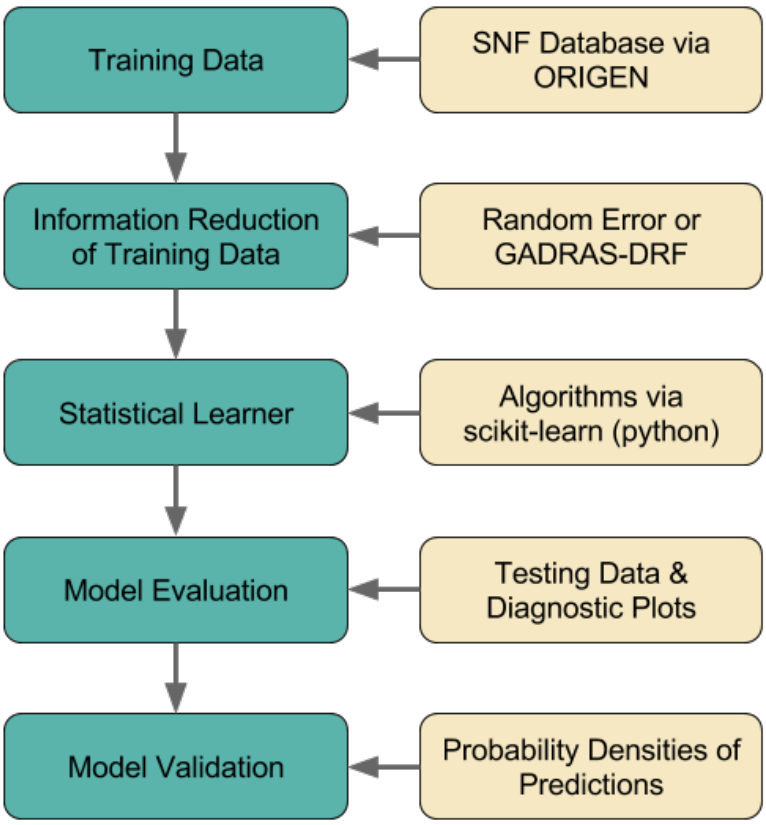


Figure 3.1: Methodology of Proposed Experiment

expectations of reactor parameter prediction and how the different algorithms perform.

Next, this work will expand upon the previous work in two ways. The first is adding a different information reduction technique via applying a gamma spectroscopy DRF to the SNF nuclide recipes, which can calculate various spectra based on the types of gamma detectors available to the forensics community. Secondly, a more advanced machine learning algorithm, support

vector regression, is included so as to compare more complex models against simpler models. A schematic of the workflow involving the experimental components is shown in Figure 3.1.

3.1 Training Data

3.1.1 Spent Nuclear Fuel Simulations

Because creating databases from real measurements to represent reactor technologies from around the world is impossible, the database in this study will be created from high-fidelity simulations via ORIGEN [25], an activation and depletion code within the SCALE 6.2 modeling and simulation suite [23]. Specifically, the ARP module of the activation and depletion code ORIGEN was used: ORIGEN-ARP.

A set of simulations of SNF at different burnups and cooling times will comprise the database. Of interest to an entity trying to create a weapon is partially irradiated fuel if they have plutonium separations capabilities or any radioactive substance in the case of a dirty bomb. Addressing the former, a smaller burnup than is typical for SNF from a commercial reactor is used in the previous work.

It should be noted that many algorithms are developed on an assumption that the training set will be independent and identically distributed (i.i.d.). This is important so that the model does not overvalue or overfit a certain area in the training space. A truly i.i.d. training set would go beyond the

Fuel	Reactor	Enrichment
CE14x14	PWR	2.8
CE16x16	PWR	2.8
W14x14	PWR	2.8
W15x15	PWR	2.8
W17x17	PWR	2.8
S14x14	PWR	2.8
VVER440	PWR	3.60
VVER440_3.82	PWR	3.82
VVER440_4.25	PWR	4.25
VVER440_4.38	PWR	4.38
VVER1000	PWR	2.8
GE7x7-0	BWR	2.9
GE8x8-1	BWR	2.9
GE9x9-2	BWR	2.9
GE10x10-8	BWR	2.9
Abb8x8-1	BWR	2.9
Atrium9x9-9	BWR	2.9
SVEA64-1	BWR	2.9
SVEA100	BWR	2.9
CANDU28	PHWR	0.711
CANDU37	PHWR	0.711

(a) Reactor types and uranium-235 enrichment [weight%]

	PWR	BWR	PHWR
Power Density [MW/MTU]	32	23	22
Burnup [MWd/MTU]	600–17700	600–12300	600–12300
Cooling Time		{1m, 7d, 30d, 1y}	

(b) Simulation space defining reactor parameters and cooling time

Table 3.1: Design of the training set space

lower burnups, but this is purely for demonstration with a single use case in mind. The training database is thus constructed by simulating the same training set space as described in Ref. [3], shown in Table 3.1. For each entry shown here the simulations included

oops, finish!

Fuel	Reactor	Enrichment	Cooling Time	Burnup
CANDU28	PHWR	0.711	{1m, 7d, 30d, 1y}	{1400, 5000, 11000}
CANDU28	PHWR	0.711	{3m, 9d, 2y}	{5000, 6120}
CE16x16	PWR	2.8	{1m, 7d, 30d, 1y}	{1700, 8700, 17000}
CE16x16	PWR	2.8	{3m, 9d, 2y}	{8700, 9150}
CE16x16	PWR	3.1	{7d, 9d}	{8700, 9150}
GE7x7-0	BWR	2.9	{1m, 7d, 30d, 1y}	{2000, 7200, 10800}
GE7x7-0	BWR	2.9	{3m, 9d, 2y}	{7200, 8800}
GE7x7-0	BWR	3.2	{7d, 9d}	{7200, 8800}

Table 3.2: Design of the testing set space

While in most machine learning studies the testing set is chosen randomly from the training set, the previous work used an external one, shown in Table 3.2. Although the test set was designed to have values in between the trained values of burnup, it was chosen systematically. Therefore it was implemented in this study for comparison, but cross-validation will be used moving forward. More specifically, using k-fold cross-validation is expected to better indicate the model performance.

3.1.2 Information Reduction

Since the overall goal of this project is to determine how much information to what quality is needed to train a machine-learned model, there will be an information reduction manipulation applied to the training data set. This study evaluates the impact of randomly introduced error of varying amounts on the ability of the algorithms to correctly predict the burnup.

The three algorithms will be evaluated with error applied to each nuclide vectors in the training set. A maximum error is ranging from $0 - 10\%$ is chosen for each round of training, and a random error within the range of $[1 - E_{max}, 1 + E_{max}]$ is applied to each component of the nuclide vector.

However, error in a nuclide vector is not random, in fact it is systematic and dependent on a number of known sources of uncertainty. The next study will introduce error by limiting the nuclides to only those that can be measured with a gamma spectrometer. Although this is initially done using the availability of gamma spectra in ORIGEN, GADRAS can provide more DRFs to further reduce information given to the algorithm.

3.2 Statistical Learning for Models

3.2.1 Algorithms Chosen

Choosing which algorithms to test is usually based on what is being predicted and intuition regarding strengths and weaknesses of different optimization

methods.

For a benchmarking excercise, some machine learning approaches here were chosen based on previous work [3]: nearest neighbor and ridge regression. These are useful because they are simple, providing a dissimilarity-based model and a linear regression-based model, respectively. If more complex algorithms are not required to obtain useful results, then there is no need to use more computationally expensive options. However, hedging on the fact that more complex models will be needed, this work also employs an algorithm that is known to handle highly dimensional data sets well: support vector regression. These algorithms were introduced in Section 2.2.1.

Algorithm	Parameter	Value
Nearest Neighbor Regression	*n*-neighbors	1
	Weights	uniform
	Distance Metric	L2: Euclidian Distance
Ridge Regression	Regularization, α	1.0
	Normalization	False
	Stopping Tolerance	0.001
Support Vector Regression	Kernel	Radial Basis Function
	Gamma, γ	0.1
	C	1.0
	Epsilon, ε	0.1
	Stopping Tolerance	0.001

Table 3.3: Defaults.....

A python-based machine learning toolkit, scikit-learn [24], is used to train the models. The default parameters used for the algorithms are in Table 3.3.

3.2.2 Reactor Parameter Prediction

The prediction of reactor parameters here is done with the burnup of the SNF to provide algorithm and corresponding model generalizability. Following the training phase of the models, next it is important to estimate the reactor parameter predication capabilities of those models. This is done with a set of measurements from a test data set with samples that mimic interdicted SNF. The testing set has the same features as the training set, with labels that are compared to the predicted labels. The results of each model's prediction errors are shown in Table 3.4.

Algorithm	Error Origin	MAPE	RMSE
Nearest Neighbor Regression	Testing Set	78.24	3479.0
	5-fold Cross-Validation	127.84	4401.4
Ridge Regression	Testing Set	0.44	44.96
	5-fold Cross-Validation	0.05	3.24
Support Vector Regression	Testing Set	4.32	428.52
	5-fold Cross-Validation	0.49	163.61

Table 3.4: Model burnup prediction errors for three algorithms

First shown in Table 3.4 are testing set errors and cross-validation errors,

because although previous work uses the former, it is expected that the latter will provide better estimates. The models evaluated by the testing set do not have a validation set for pre-evaluation. The models that are evaluated via cross-validation do not use the testing set.

Table 3.4 includes two error types. For the sake of comparison to previous work and convenient interpretation, mean absolute percentage error (MAPE) is tracked. However, MAPE requires that no true values are 0. The preferred method in the community is to use root-mean-squared error (RMSE) for model error estimation, so both are tabulated. The MAPE shows that there are some extremely high and extremely low errors depending on the algorithm, both of which are quite concerning, as this indicates poor performance.

Figure 3.2 shows the three algorithm's MAPEs with respect to the reduction of information by the introduction of random error to the nuclide vectors as described in Section 3.1.2. SVR is shown to perform the best, but it quickly reaches 100% error. Although ridge regression rapidly increases with any amount of error, nearest neighbor regression shows more promise, although it reaches 257% at 10% error. Overall, this performance indicates that these algorithms are unlikely to predict burnup when faced with uncertainties in real-world measurements or the further reduction in information from gamma detection.

Given the results in Table 3.4 and Figure 3.2, next introduced are some diagnostic and optimization procedures that can shed light on the

Figure 3.2: Results from information reduction using random error

performance.

3.3 Validation

To obtain reliable models, one must both choose or create a training set carefully and study the impact of various algorithm parameters on the error. Although the title of this section suggests final steps of confirming a model's usefulness for predictions, what follows is more of a troubleshooting exercise. In practice, these analyses are used for both purposes.

Machine learning algorithms are heavily dependent on the inputs and parameters given to them, such as training set sizes, regularization, learning

rates, etc. From the results shown in Section 3.2, it is clear there is room for improvement. Diagnostic plots show the errors of the predicted burnup values to the actual burnup values with respect to some variable on the x-axis. As previously introduced in Section 2.2.3, the errors are compared to the training error to understand the generalization strength with respect to training set size (learning curves) and the algorithm parameters governing model complexity (validation curves).

In addition to machine learning best practices, another layer of comparison is added here. Because it is difficult to ensure consistently representative testing data, the accuracy of a learned model should not depend on only one testing set. The learned model's accuracy is better estimated by using a validation set, or even better, k-fold cross-validation, introduced in Section 2.2.2. This work includes both the testing error (using the testing set described in Section 3.1) and cross-validation error. The predetermined testing set will allow for comparison against the previous work it was obtained from [3], but it is assumed that cross-validation will provide a better indication of model performance.

The learning curves are obtained as follows, shown in Figure 3.3. For a given (randomly chosen) training set size between 15 and 100% of the total data set, training and prediction rounds were performed for each. The testing error scenario performs this k times and averages those results. This is equivalent to the k in k-fold cross-validation to provide some semblance of equivalent statistics. The cross-validation error scenario has no need for

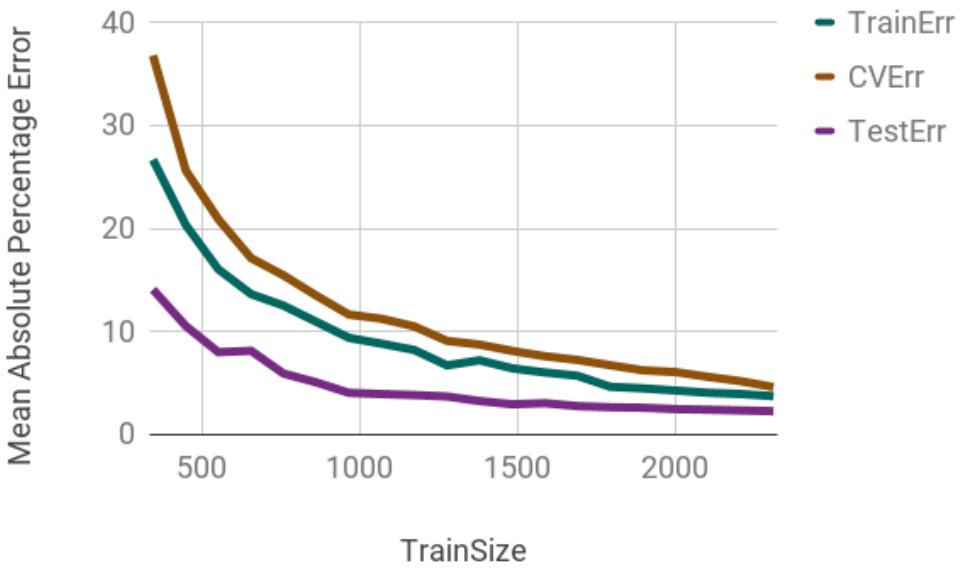


Figure 3.3: Learning curve for burnup prediction, $\gamma = 0.001$

averaging because it is performed automatically. In both cases, the learning curves do not provide a clear picture of over- or undertraining upon first glance.

The validation curves are obtained as follows, shown in Figure 3.4. The γ parameter in SVR, which influences model complexity, was varied from 10^{-4} to 10^{-1}. Training and prediction rounds were performed for different γ values in this range. Again, the testing and cross-validation errors are both used as described above. As with Figure 3.3, determining the robustness to over- or undertraining is difficult here although there is possibly a minimum at $\gamma = 0.001$.

Although there is no example behavior of Figure 3.3's peculiar learning

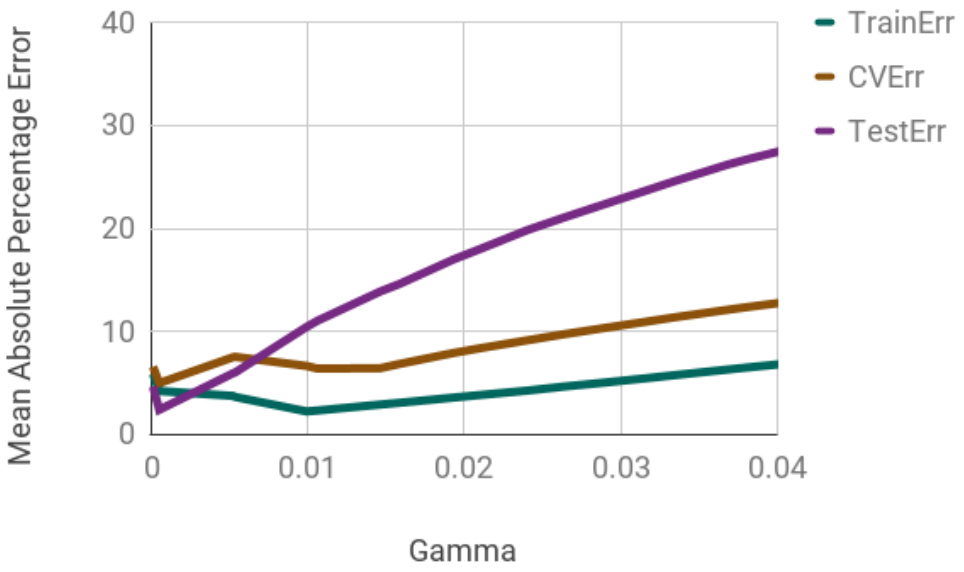


Figure 3.4: Validation curve for burnup prediction, $TrainSize = 2313$

curve in Figure 2.9, the curve mimics the squared bias curve from Figure 2.7. This indicates that the bias in the model is much higher than the variance. Next, the testing error is lower than the training error; this should never be the case, and indicates an issue with the systematically chosen testing set. While the cross-validation error is correctly higher than the training error, it follows along in parallel, producing no information on model fitness other than confirming a very high bias. It is presumed this is not the fault of the algorithms, but the training set itself. It is likely covering too small of a range of the simulation space.

Additionally, Figure 3.4's validation curve shows the testing error dropping below the training error for extremely small γ, around where a minimum

might be. Since the model suffers from high bias, no amount of model complexity can be optimized. The resolution to the underfitting is discussed in Section 4.1.

4 RESEARCH PROPOSAL

This document previously demonstrated the performance of machine learning on a set of nuclear material isotopics to calculate a reactor parameter of interest: burnup. Additionally, there have been various methods discussed for understanding the learned model's behavior and thus the quality of the results. Moving forward to a set of experiments is now possible.

Before describing the experiments, some topics and issues are addressed in Section 4.1. Next, the proposed research experimental design is presented. Sections 4.2, 4.3, and 4.4 describe how this work will probe the viability of statistical learning using direct isotopic information, information obtained from gamma spectra, and other fuel cycle flows, respectively. Discussed separately is the planned implementation of comparing methods using the Bayesian framework in Section 4.5. Finally, the timeline of this project is covered in Section 4.6.

4.1 Experiment Preparations

Expanding Training Set

As identified in Section 3.3, the testing set used for the demonstration was not suitable for further study without being expanded. Many algorithms are developed on an assumption that the training set will be i.i.d.. This is important so that the model does not overvalue or overfit a certain area

in the training space. The next step is to provide a larger, more diverse training set to the algorithms so they can better predict when faced with new instances. This diversity will be suggested from the SFCOMPO database [17], as it includes many common domestic and international reactors.

The SFCOMPO-2.0 relational database [17] has approximately 750 SNF measurements from 44 reactors. While this is not sufficient as a training set, it provides a better framework for simulating a larger training set using ORIGEN. After cross-validation, diagnostics, and optimization, the trained models can be tested against the entries in this database to provide a clear estimate of of the model performance.

Finalizing Set of Algorithms

The three algorithms in the demonstration (nearest neighbor, ridge, and support vector regression) are not necessarily the set that will be evaluated for the experiments. After these are used to train new models on a larger training set for comparison, other algorithms can be speedily assessed as well. Since support vector-, distance-, and linear-based models are already represented, other obvious choices include Bayesian methods, decision trees, neural networks, or ensemble methods [6].

Computational Framework and Resources

Thus far, all simulations and training have not required more processing than available on a personal computer. However, some algorithms do require larger amounts of computational time (e.g., neural networks). If necessary, the training stage can be done using the Center for High Throughput Computing (CHTC), which is available to University of Wisconsin researchers.

4.2 Experiment 1

Viability of Statistical Learning on Direct Isotopics

The first experiment will be a purposefully constructed version of the demonstration: evaluating the model performance with known isotopics. This sheds light on how this methodology will perform on the simplest scenario, providing an upper estimate of performance. *The main purpose of this experiment is to probe the usefulness of statistical methods for determining reactor parameters, ultimately choosing the best performing methods.*

Figure 4.1 shows what type of input can be used for training and prediction. The two horizontal boxes are the physical and computational forms of what these experiments are simulating, respectively. The lab-measured mass spectra correspond to the perfect information being referred to here. In the computational context, these measurements instead come from simulations. Mass spectra results are thus approximated as the direct isotopics given from the simulations, since they provide highly reliable and accurate information.

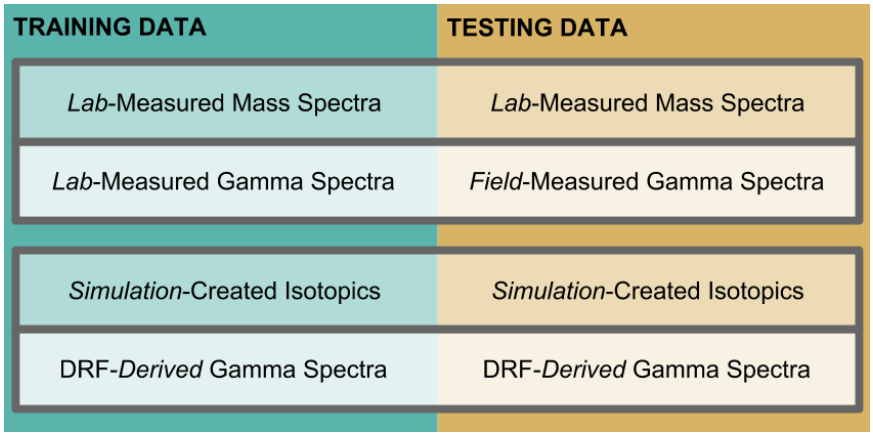


Figure 4.1: Motivation for Data Sets for Experiments 1 and 2

The variables for this experiment will include the following:

1. the complexity of the machine learning algorithm used,
2. feature reduction, and
3. different subsets of the desicion space.

As previously mentioned, there are complexity differences among the algorithms, which will be taken into account. The distance-based nearest neighbor algorithm is by far the simplest, as it does not do any learning. By contrast, SVR with a radial basis function kernel to work in many dimensions is much more elaborate. The feature reduction will be different subsets of isotopes, e.g., top n nuclides, fission products only, etc. Reducing the decision space can be done by, e.g., simplifying the regression task by fixing the reactor type. It is expected that a more complex algorithm (e.g., SVR) will be needed, and that preprocessing and/or manual feature reduction will assist in creating higher quality models. Simplifying the decision space

should always improve prediction, but it is not obvious how much it will be needed for burnup prediction specifically.

Risks

It is possible that statistical models trained on direct isotopic information do not perform well enough. Other than attempting different types of algorithms, it is possible to preprocess the data, statistically performing feature reduction via PCA. If this is not sufficient, it is possible SNF is has too many or too few correlated features to provide reliable models across the space of current reactor technologies. Since separated plutonium and UOC have been also studied using these techniques, it is possible these materials can provide useable learned models. Additionally, this methodology would also work if applied to post-detonation materials. There is work on creating standard materials to represent the "urban canyon", so this is another subject that could benefit from statistical correlations [8].

4.3 Experiment 2

Viability of Statistical Learning on Gamma Spectra

The second experiment will be the previously discussed extension of the demonstration by applying detector response functions to the SNF isotopics: evaluating the model performance with reduced isotopic information. This demonstrates the usefulness of this methodology in a real-world scenario

where exact isotopics are not always known. *The main objective of this experiment is to measure the reduction in statistical model parameter prediction reliability as the quality of the training information is reduced.*

The two bottom portions of the boxes in Figure 4.1 represent a more realistic measurement scheme, involving a model trained from gamma spectrometers rather than the lengthy process of performing mass spectrometry on the samples. In the physical context, the measurements for training would be done using a semiconductor gamma detector, but the testing or prediction step may be done outside of the lab on a different detector. This will be captured by applying different detector response functions to the radionuclide inventories from the simulations.

The variables for this experiment will include the following:

1. the complexity of the machine learning algorithm used,
2. feature reduction (implicit), and
3. quality of training and/or testing data set.

As in Section 4.2, the algorithm complexity will be used to explain performance. The feature reduction is implicit here, since gamma detection only includes radionuclides within the SNF isotopics. The indirect isotopic training data are likely going to reduce the prediction capability of the models, but it is not yet clear if a response function simulating a hand-held NaI gamma detector can provide any useful predictions. And while it is still expected that the complex algorithms will perform better, it is not yet

obvious if different algorithms than the ones used in Experiment 1 will be needed.

Risks

It is possible that statistical models trained on indirect isotopic information do not perform well enough. Again, here, different algorithms may perform better than others due to the underlying optimization processes. Further feature reduction could also prove useful, focusing on particular energy regions or particular peaks throughout the spectrum. The quality of the training information can be improved slightly by using an isotope identification algorithm; this may improve the performance, as they are developed to automatically report isotopics from gamma spectra. If this still is not sufficient, it may be that direct isotopic information (i.e., that obtained from mass spectrometry) is required for reliable statistical models of SNF. Although preprocessing could also be investigated here, the materials discussed above may also be more disposed to defined statistical correlations.

4.4 Experiment 3

Viability of Statistical Learning on Reprocessed Fuel

There is already nuclear fuel reprocessing in France for LWRs, and possibly in the forseeable future in China with fast breeder reactors. Given the existence

of reprocessing in fuel cycles, it is important to develop nuclear forensics capabilities on processed SNF. This is presents an additional prediction challenge because many of the isotopic, chemical, and elemental signatures are stripped away in the processing. However, there is a possibility that plutonium isotopes can provide enough information for characterization, even with reprocessing [26]. Although Ref. [26] predicts reactor type and enrichment, it is enough motivation to pursue burnup as well.

This experiment is intended to repeat those in Sections 4.2 and 4.3 with an added layer of complexity from the reprocessing by including mixed-oxide fuel. It will be carried out using direct isotopics as in Section 4.2, with a goal to use indirect information as in Section 4.3 pending performance. The training data set will also be simulated using ORIGEN. *The main purpose of this experiment is to probe the difficulties of reactor parameter prediction when some of the inputs are from a cyclical nuclear fuel cycle.*

The variables for this experiment will include the following:

1. the complexity of the machine learning algorithm used,
2. quality of training data set, and
3. type of preprocessing for feature reduction.

Because of the increased complexity of the training data, it is expected that a more involved algorithm will be necessary. The quality of the training data set will be varied as well as the type of preprocessing tool. It is not yet obvious if the feature reduction is absolutely necessary, but it is expected to drastically improve the accuracy of burnup prediction. Feature

reduction can be carried out via a range of discriminant, component, or factor analyses. Multiple studies have demonstrated success with some of these dimensionality reduction techniques [9, 10, 19, 21, 22, 26, 27]. Because reprocessing mixes streams of material, PCA is not expected to perform as well as ICA.

Risks

It is possible that this framework is not the best approach to understand fuel cycles with reprocessing. The first action upon poor performance results is to try more/different preprocessing tools; this may have to include using domain knowledge to manually reduce the dimensions. As one of the goals within the nuclear forensics community is to identify new signatures and/or correlated measurements, this is not desireable but it could still offer interesting results. Also, some creativity could be applied to find a method that works for this type of dataset, such as the iterative partial least squares discriminant analysis used for UOC country and source prediction in Refs. [26, 27].

4.5 Model Comparison

It is essential to be able to compare the methods proposed here against each other, but also against other analagous methods. For example, Experiment 2 (Section 4.3) aims to predict reactor history values using simulated gamma

spectra, and INDEPTH uses gamma spectra as inputs. These can be directly compared using the inverse problem structure introduced in Section 2.1.2. An application of Equation 2.2 is discussed further in Section 2.2.3.3.

However, the Section 2.2.3.3 discussion excluded some detail for clarity. If the training data set includes uncertainties, the values discussed will not just be numbers, but probability distributions. While initially uncertainties will be ignored for an approximate analysis, they cannot be ignored for long in the nuclear forensics context. These are quite important for obtaining some measure of confidence in the solution. Including the uncertainty broadens each predicted parameter to a range of probabilities. These are then analyzed to produce ranges of confidence in the predictions [31]. The methods for calculating the necessary probability distributions are outlined below.

Here, we change the meaning of the variables to represent probability distributions, as in Equation 4.1. C is a constant given by the marginal likelihood, which can be ignored when calculating relative probabilities, and $\boldsymbol{d}$ and $\boldsymbol{m}$ represent the training data set and model parameters, respectively. Thus, $P(\boldsymbol{d}|\boldsymbol{m})$ is the likelihood distribution function, $P(\boldsymbol{m})$ is the prior probability distribution, and $P(\boldsymbol{m}|\boldsymbol{d})$ is the posterior probability distribution.

$$P(\boldsymbol{m}|\boldsymbol{d}) = C \ * \ P(\boldsymbol{d}|\boldsymbol{m}) \ * \ P(\boldsymbol{m}) \tag{4.1}$$

Mathematically speaking, the distributions are obtained by integrating

over the relevant probability densities. For example, the prior probability distribution can be calculated from Equation 4.2, where $\boldsymbol{m}$ is the range of predicted model parameters, i.e. burnup values, and $\boldsymbol{d}$ is a set of nuclide vectors. Also, here, $\rho(\boldsymbol{x}) = \prod_i \rho(x_i)$.

$$P(\boldsymbol{m}) = \int_{\boldsymbol{m}} \rho(\boldsymbol{d}) d\boldsymbol{d} \tag{4.2}$$

Similarly, the likelihood distribution function is obtained as in Equation 4.3.

$$P(\boldsymbol{d}|\boldsymbol{m}) = \int_{\boldsymbol{d},\boldsymbol{m}} \rho(\boldsymbol{d}|\boldsymbol{m}) d\boldsymbol{m} \tag{4.3}$$

In practice, however, these functions are not calculated directly. Below, the methods chosen to estimate the functions in this work are addressed.

4.5.1 Estimating Density Functions

Estimating a probability density function $\rho(x)$ is not a straightforward task. One must have a 'feel' for the shape of this function to predict a smooth curve that will represent the probability density over a range of a parameter. One estimation method is binning the parameters in the training set over a given width and tallying up each bin to form a histogram; using some kernel, Gaussian for example, provides a function that should approximately follow the shape of the histogram. The shape of the histogram and function may strongly depend on the bin size, but this can be handled as a separate optimization step within the sci-kit learn package [24].

As mentioned, the prior probability distributions are given by the model

space, e.g., reactor parameters as predicted from the machine-learned models. The above-mentioned method should be utilized to obtain a prior density function from the range of burnups. This will not be necessary because the current design of the training set will provide a flat density function (i.e., uniform distribution). However, this is not the case in a real-world scenario with mostly commercial reactor fuel comprising SNM inventories around the world. This topic will only be investigated if the training data set is changed. [31] Note: This implies the posterior is now only dependent on the likelihood.

The likelihood function can be obtained by summing the probability densities of each nuclide vector (d) given some predicted model parameter (m, e.g., a burnup prediction) using some algorithm (e.g., nearest neighbor regression). In essence, this is a large set of forward problems; it is information including the database of SNF recipes, and the known parameter inputs compared to the model's parameter predictions. The goal of this step is to find the *maximum* likelihood, usually carried out in a step called maximum likelihood estimation (MLE). In this framework, the training phase provides the maximum likelihood distribution through the use of cross-validation, since the results are reported as a mean error with a standard deviation (which can be converted to accuracy for likelihood) [24].

Unfortunately, MLE is not this simple for other methods that do not employ cross-validation [29, 31]. This will indeed have to be broached in order to compare the machine-learned models against other forensics

methods but is beyond the scope of this work.

4.5.2 Posterior Odds

Finally, a non-normalized posterior probability distribution, $P(m_i|d)$, can be calculated directly from the likelihood and prior distributions. The same can be done for a model obtained from a different algorithm, $P(m_j|d)$. The relative posterior probability distribution, aka *posterior odds* [31], can then be obtained as in Equation 4.4. Here, $B_{ij} = \frac{\rho(d|m_i)}{\rho(d|m_j)}$ and is referred to as the *Bayes factor*.

$$\frac{P(m_i|d)}{P(m_j|d)} = B_{ij}\frac{P(m_i)}{P(m_j)} \tag{4.4}$$

Given relatively uniform priors, the Bayes factor is the key component that determines relative model performance. Table 4.1[1] shows an approximation of posterior probabilities calculated from $|lnB_{ij}|$. Taking the logarithm of the Bayes factor is done for convenience, since taking the product of many likelihoods tends towards zero.

In summary, given a mean-squared error and its standard deviation from using cross-validation with any machine learning algorithm, the MLE can be formed. To compare two models, a ratio of MLE_i to MLE_j, called the posterior odds, provides the probability of model i being correct.

[1]This table is reproduced from Ref. [31].

$\lvert \ln B_{ij} \rvert$	Probability	Likelihood Strength
< 1.0	< 0.750	Inconclusive
1.0	0.750	Weak
2.5	0.923	Moderate
5.0	0.993	Strong

Table 4.1: Model Comparison using Likelihood Strength

4.6 Timeline

Figure 4.2 is an approximate timeline over which this work will occur. Multiple tasks will be complete before my internship is finished, and much of the computational framework is also complete. The experiments are expected to take a few weeks each, accounting for time spent on diagnostics and readying the results for the next step. Formal validation and model comparison will then be carried out. Because of reporting requirements, some preporatory writing will also be done during the internship. Thus, final analyses and writing are expected to take three months.

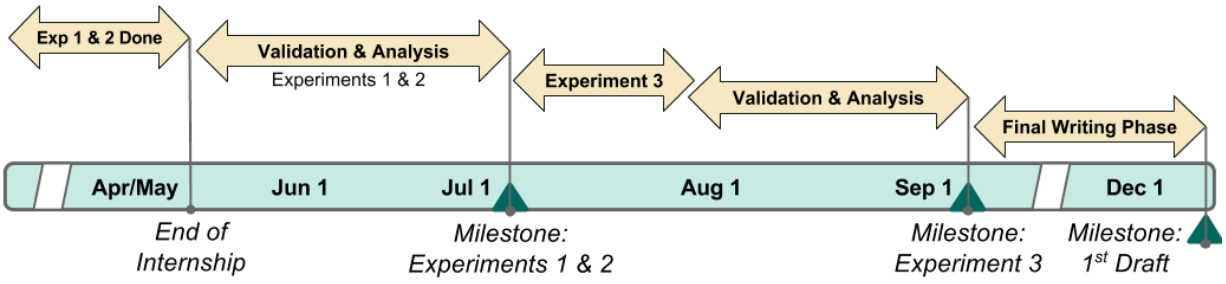


Figure 4.2: Timeline for Project

REFERENCES

[1] Broadhead, Bryan L, and Charles F Weber. 2010. Validation of inverse methods applied to forensic analysis of spent fuel. In *Proceedings of the Institute of Nuclear Materials Management 51st Annual Meeting*. Baltimore, MD, USA. https://www.osti.gov/scitech/biblio/1001291.

[2] Chadwick, M.B., M. Herman, P. ObloÅ¾inskÃ½, M.E. Dunn, Y. Danon, A.C. Kahler, D.L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D.A. Brown, R. Capote, A.D. Carlson, Y.S. Cho, H. Derrien, K. Guber, G.M. Hale, S. Hoblit, S. Holloway, T.D. Johnson, T. Kawano, B.C. Kiedrowski, H. Kim, S. Kunieda, N.M. Larson, L. Leal, J.P. Lestone, R.C. Little, E.A. McCutchan, R.E. MacFarlane, M. MacInnes, C.M. Mattoon, R.D. McKnight, S.F. Mughabghab, G.P.A. Nobre, G. Palmiotti, A. Palumbo, M.T. Pigni, V.G. Pronyaev, R.O. Sayer, A.A. Sonzogni, N.C. Summers, P. Talou, I.J. Thompson, A. Trkov, R.L. Vogt, S.C. van der Marck, A. Wallner, M.C. White, D. Wiarda, and P.G. Young. 2011. Endf/b-vii.1 nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data. *Nuclear Data Sheets* 112(12):2887–2996. http://www.sciencedirect.com/science/article/pii/S009037521100113X.

[3] Dayman, Kenneth, and Steven Biegalski. 2013. Feasibility of fuel cycle characterization using multiple nuclide signatures. *Journal of*

Radioanalytical and Nuclear Chemistry 296:195–201. http://link.springer.com/article/10.1007%2Fs10967-012-1987-4.

[4] Gauld, I. C. 2003. Mox cross-section libraries for origen-arp. Tech. Rep., UT-Battelle, LLC and Oak Ridge National Laboratory. ORNL/TM-2003/2, https://www.nrc.gov/docs/ML0410/ML041040058.pdf.

[5] Gey, Frederic, Chloe Reynolds, Ray Larson, and Electra Sutton. 2012. Nuclear forensics: A scientific search problem. In *Proceedings of the Lernen, Wissen, Adaption (Learning, Knowledge, Adaptation) Conference*. Dortmund, Germany. http://metadata.berkeley.edu/nuclear-forensics/Paper_9-12-12_lwa-2012-nuclear-forensics-scientific-search-problem_v7.pdf.

[6] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc.

[7] Horne, Steven M., Gregory G Thoreson, Lisa A. Theisen, Dean J. Mitchell, Lee Harding, and Wendy A. Amai. 2014. Gamma Detector Response and Analysis Software - Detector Response Function (GADRAS-DRF). User's Manual, Sandia National Laboratories, Albuquerque, New Mexico, USA. Version 18.5; SAND2014-19465, http://www.osti.gov/scitech/servlets/purl/1166695.

[8] Inn, Kenneth G.W., Jacqueline Mann, Jeffrey Leggitt, JoAnne Buscaglia, Simon Jerome, John Molloy, and William Pramenko. 2015. Nuclear forensic reference materials for attribution of urban nuclear terrorism. *Presentation for NIST*, https://www.nist.gov/sites/default/files/documents/oles/7-Inn_Kenneth-Nuclear-Forensics-UVC-Project.pdf.

[9] Jones, Andrew, Phillip Turner, Colin Zimmerman, and J.Y. Goulermas. 2014. Machine learning for classification and visualisation of radioactive substances for nuclear forensics. In *Techniques and Methods for Safeguards, Nonproliferation and Arms Control Verification Workshop*. Portland, Oregon. https://www.researchgate.net/publication/264352908_Machine_Learning_for_Classification_and_Visualisation_of_Radioactive_Substances_for_Nuclear_Forensics.

[10] Jones, Andrew E., Phillip Turner, Colin Zimmerman, and John Y. Goulermas. 2014. Classification of spent reactor fuel for nuclear forensics. *Analytical Chemistry* 86:5399–5405. http://pubs.acs.org/doi/ipdf/10.1021/ac5004757.

[11] Kristo, M.J., D.K. Smith, S. Niemeyer, G.D. Dudder, and R. Abedin-Zadeh. 2006. Nuclear Forensics Support: Technical Guidance Reference Manual. Tech. Rep., IAEA, Vienna, Austria. http://www-pub.iaea.org/MTCD/Publications/PDF/Pub1241_web.pdf.

[12] Langan, Roisin T., Richard K. Archibald, and Vincent E. Lamberti. 2016. Nuclear forensics analysis with missing data. *Journal of Radioanalytical and Nuclear Chemistry* 308:687. http://link.springer.com/article/10.1007/s10967-015-4458-x.

[13] Langley, Pat. 2011. The changing science of machine learning. *Machine Learning* 82:275–279. https://link.springer.com/article/10.1007%2Fs10994-011-5242-y.

[14] Lantzos, I., Ch Kouvalaki, and G. Nicolaou. 2015. Plutonium fingerprinting in nuclear forensics of spent nuclear fuel. *Progress in Nuclear Energy* 85(Supplement C):333–336. http://www.sciencedirect.com/science/article/pii/S0149197015300329.

[15] Leal, L. C., O. W. Hermann, S. M. Bowman, and C. V. Parks. 1998. Arp: Automatic rapid process for the generation of problem-dependent sas2h/origen-s cross-section libraries. Tech. Rep., Lockheed Martin Energy Research Corporation and Oak Ridge National Laboratory. ORNL/TM-13584, https://digital.library.unt.edu/ark:/67531/metadc684948/.

[16] May, Michael, Reza Abedin-Zadeh, Donald Barr, Albert Carnesale, Philip E. Coyle, Jay Davis, William Dorland, William Dunlop, Steve Fetter, Alexander Glaser, Ian D. Hutcheon, Francis Slakey, and Benn Tannenbaum. 2007. Nuclear Forensics: Role, State of the Art, and Program Needs. Tech. Rep., Joint Working Group

of the American Physical Society and the American Association for the Advancement of Science. https://www.aaas.org/report/nuclear-forensics-role-state-art-program-needs.

[17] Michel-Sendis, Franco, Jesus Martinez-GonzÃ¡lez, and Ian Gauld. 2017. Sfcompo 2.0 – a relational database of spent fuel isotopic measurements, reactor operational histories, and design data. *EPJ Web of Conferences* 146:06015. www.oecd-nea.org/sfcompo/.

[18] Moody, K.J., P.M. Grant, and I.D. Hutcheon. 2005. *Nuclear Forensic Analysis*. 1st ed. Boca Raton, Florida, USA: CRC Press. https://books.google.com/books?id=Q9mgDnWoPLYC.

[19] Nicolaou, G. 2006. Determination of the origin of unknown irradiated nuclear fuel. *Journal of Environmental Radioactivity* 86:313–318. http://nuclear.ee.duth.gr/upload/A13%20%20%20identification.pdf.

[20] ———. 2008. Provenance of unknown plutonium material. *Journal of Environmental Radioactivity* 99(10):1708–1710. http://www.sciencedirect.com/science/article/pii/S0265931X08000969.

[21] ———. 2009. Identification of unknown irradiated nuclear fuel through its fission product content. *Journal of Radioanalytical and Nuclear Chemistry* 279(2):503–508. http://link.springer.com./article/10.1007%2Fs10967-007-7300-x.

[22] ———. 2014. Discrimination of spent nuclear fuels in nuclear forensics through isotopic fingerprinting. *Annals of Nuclear Energy* 72:130–133. Technical Note, http://www.sciencedirect.com.ezproxy.library.wisc.edu/science/article/pii/S0306454914002308.

[23] Oak Ridge National Laboratory. 2016. SCALE: A Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis and Design. Code Suite, Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA. Version 6.2.1, ORNL/TM-2005/39, Available from Radiation Safety Information Computational Center as CCC-834, http://scale.ornl.gov.

[24] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830. http://scikit-learn.org/stable/index.html.

[25] Rearden, B.T., and M.A. Jessee. 2016. Ch. 5 Depletion, Activation, and Spent Fuel Source Terms. In *SCALE Code System: User Documentation*, 5–1–5–263. Oak Ridge, Tennessee, USA: Oak Ridge National Laboratory. Version 6.2.1; ORNL/TM-2005/39, https://www.ornl.gov/sites/default/files/SCALE%20Code%20System.pdf.

[26] Robel, Martin, and Michael J. Kristo. 2008. Discrimination of source reactor type by multivariate statistical analysis of uranium and plutonium isotopic concentrations in unknown irradiated nuclear fuel material. *Journal of Environmental Radioactivity* 99(11):1789–1797. http://www.sciencedirect.com/science/article/pii/S0265931X08001203.

[27] Robel, Martin, Michael J. Kristo, and Martin A. Heller. 2009. Nuclear forensic inferences using iterative multidimensional statistics. In *Proceedings of the Institute of Nuclear Materials Management 50th Annual Meeting*. Tuscon, AZ, USA: Institute of Nuclear Materials Management. LLNL-CONF-414001, https://e-reports-ext.llnl.gov/pdf/374432.pdf.

[28] Skutnik, Steven E., and David R. Davis. 2016. Characterization of the non-uniqueness of used nuclear fuel burnup signatures through a Mesh-Adaptive Direct Search. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 817:7–18. http://www.sciencedirect.com/science/article/pii/S0168900216001571?via%3Dihub.

[29] Tan Bui-Thanh. 2012. A Gentle Tutorial on Statistical Inversion using the Bayesian Paradigm. Note ICES REPORT 12-18, The University of Texas at Austin, The Institute for Computational Engineering and Sciences. http://users.ices.utexas.edu/~tanbui/teaching/Bayesian/Bayesian.pdf.

[30] Tarantola, Albert. 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation*, chap. 1. The General Discrete Inverse Problem, 1–40. Philadelphia, Pennsylvania, USA: Society for Industrial and Applied Mathematics. http://epubs.siam.org/doi/pdf/10.1137/1.9780898717921.ch1.

[31] Trotta, Roberto. 2008. Bayes in the Sky: Bayesian Inference and Model Selection in Cosmology. *Contemporary Physics* 49(2): 71–104. Invited review, https://ned.ipac.caltech.edu/level5/Sept13/Trotta/Trotta_contents.html.

[32] Weber, Charles F, Vladimir A Protopopescu, Michael H Ehinger, Alexander A Solodov, and Catherine E Romano. 2011. Inverse solutions in spectroscopic analysis with applications to problems in global safeguards. In *Proceedings of the Institute of Nuclear Materials Management 52nd Annual Meeting*. Palm Desert, CA, USA. https://www.osti.gov/scitech/biblio/1031530.

[33] Weber, Chuck F, and Bryan L Broadhead. 2006. Inverse depletion/decay analysis using the scale code system. In *Transactions of the American Nuclear Society Winter Meeting*, vol. 95, 248–249. Albuquerque, NM, USA. Track 4: Nuclear and Criticality Safety Technologies.