ORNL/TM-2018/1079

Updates to the Relevance Vector Machine: Multiclass Classification, Variable Selection, and Proof-of-Concept Application to Safeguards Fresh Fuel Verification using List-Mode Neutron Collar Data

Kenneth Dayman
Andrew Nicholson
Louise Worrall

June 2020

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2018/1079

Nuclear Nonproliferation Division

Updates to the Relevance Vector Machine: Multiclass Classification, Variable Selection, and Proof-of-Concept Application to Safeguards Fresh Fuel Verification using List-Mode Neutron Collar Data

Kenneth Dayman
Andrew Nicholson
Louise Worrall

Date Published: June 2020

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The authors would like to acknowledge and thank the United States Department of Energy (US DOE) National Nuclear Security Administration (NNSA) Office of Defense Nuclear Nonproliferation Research and Development (DNN R&D) for providing funding for this research under project OR16-List Mode for Collar-PD1La "List Mode Response Matrix for Advanced Neutron Correlation Analysis for Nuclear Safeguards".

ABSTRACT

To expand the capabilities of safeguards authorities to verify the integrity of fresh fuel assemblies, Oak Ridge National Laboratory has retrofit the existing electronics of the JCC-71 uranium neutron coincidence collar, which contains 18 ^{3}He neutron detectors and an external ^{241}AmLi(α, n) neutron interrogation source arranged to surround a fresh nuclear fuel assembly. The new electronics system allows analysts to record list-mode neutron multiplicity data in addition to the singles and doubles rates that are currently measured. Based on previous proof-of-concept research [1], analysis of these new data will identify off-normal fuel configurations in an assembly and characterize or localize the specific partial fuel defects. The purpose of this report it to document the analysis algorithm development and then to demonstrate its capability for the safeguards verification of fresh fuel assemblies using list mode neutron collar data. To analyze the complex list-mode data collected with the upgraded uranium neutron collar, multivariate classification algorithms are being developed using a novel classification method, the relevance vector machine. This approach may be applied to multiclass problems to estimate the probability that test data belongs to one of many possible classes of data. In addition, our method identifies the most useful variables/channels for making predictions, which illuminates the basis for the model's predictions, and this interpretability is largely unique among data analytics methods. Variable selection occurs during model training and parameter tuning and does not need any external hyperparameter tuning routines.

Finally, we apply the modified relevance vector machine to a simulated dataset of list-mode neutron collar data generated with the radiation transport code MCNP. The method can correctly identify off-normal fuel configurations, categorize the data according to four fuel defect scenarios, and rank the channels in the data according to prediction utility. For nuclear safeguards applications, it is concluded that this method has the potential to increase the sensitivity and reliability to detect missing fuel rods from a standard 17 x 17 Pressurized Water Reactor (PWR) fresh fuel assembly. Within this analysis, "off-normal" (i.e., missing fuel rods) were correctly classified in 17 simulated test scenarios with one quarter (25%) of the fresh fuel rods missing using a training data set of 58 simulated measurements.

1. INTRODUCTION

The International Atomic Energy Agency (IAEA) is required to verify the quantity of uranium in fresh fuel assemblies that are found in nuclear fuel fabrication plants and reactor facilities worldwide. While there are multiple approaches to performing such verification activities, we consider one technology: the uranium neutron coincidence collar (UNCL) [2]. Researchers at Oak Ridge National Laboratory (ORNL) are currently developing ways to expand the capabilities of the UNCL by upgrading the existing electronics to support list-mode data collection and analysis. This report focuses on corresponding algorithm development with the development of a statistically rigorous multivariate analysis method for analyzing the collected data with a novel, recently developed, statistical classification method.

1.1 NEUTRON COLLAR COINCIDENCE COUNTING FOR SAFEGUARDS

The specific UNCL system under development at ORNL is the Mirion Technologies Model JCC-71 Neutron Coincidence Collar [3]. The general UNCL design is shown as an MCNP rendering in Figure 1. Neutrons from the source incident on the fuel will either interact in the fuel or cladding and scatter out of the assembly toward the neutron detectors or cause a fission, which will lead to a temporary, localized growth in the neutron population by subcritical multiplication. Neutrons emitted from the fuel assembly will be collected in the neutron detectors.

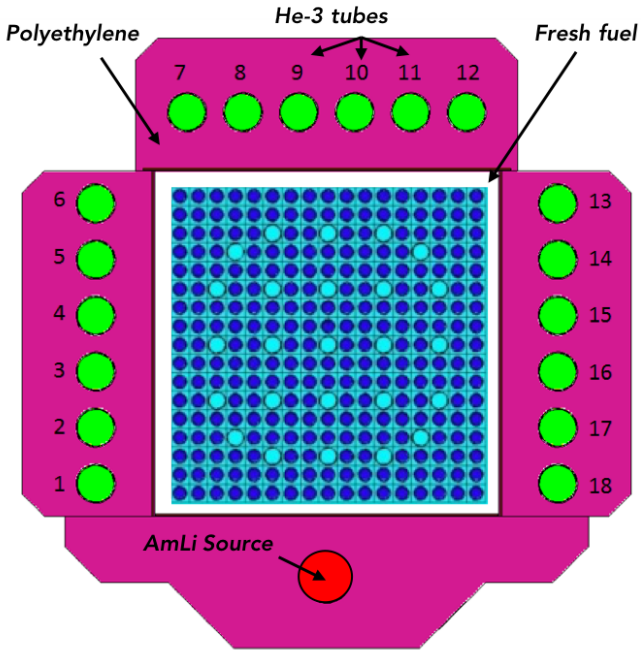


Figure 1. Uranium neutron coincidence collar. In the active configuration, 18 neutron detectors (^{3}He tubes) are arranged in three banks, which surround three sides of the fuel assembly under interrogation. The fourth side is exposed to an external ^{241}AmLi neutron source, which produces interrogation neutrons by the ${}^{7}_{3}\mathrm{Li}\ (\boldsymbol{\alpha}, \boldsymbol{n})$ reaction with alphas provided by the decay of ^{241}Am.

While neutrons born from fission and from scattering events are indistinguishable in the UNCL, analysis of the detection events' distribution in time (i.e., a pulse train[4], [5]) can estimate the average number of fission events. To first order, the distribution of neutrons not associated with a fission event (i.e., neutrons that do not cause fission nor are born from fission) will be distributed according to the Poisson distribution in time.* The distribution for fission neutrons is more complex. Initial fission events will be distributed according to the Poisson distribution; however, additional fission events will be distributed with higher frequency/intensity in a short time interval following a fission during subcritical multiplication. These bursts of neutrons are identified with shift register electronics. Counting the doubles rates and correcting for accidental doubles (spurious coincident events from the ^{241}AmLi source) yields the net doubles rate, which is proportional to the fissionable mass (e.g., ^{235}U) per unit length of the interrogated material.

In the existing neutron collar system, all six detectors in each bank are wired together before further electronic processing, meaning their signals are summed together. Furthermore, as discussed in [6], the logic signals produced from each bank are combined in a series of OR gates. Thus, all 18 detectors in the UNCL system are treated as a single neutron detector with $3\pi/2$ coverage of the interrogated material. This approach negates all spatial information that may be encoded in the distribution of neutron detections among the 18 tubes. Likewise, the singles count rate is largely dominated by non-fission neutrons coming from the ^{241}AmLi source and are of little additional use.

*The pulse train is a Poisson process [38] with parameter Λ, and the probability of observing k counts within a time window of width τ is given by the Poisson distribution with parameter $\Lambda = \lambda\tau$,

$$\mathbb{P}[x = k \,|\, \lambda, \tau] = \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau} = \frac{(\Lambda)^k}{k!} e^{-\Lambda}.$$

Given the need of the IAEA to expand and strengthen verification activities with little or no corresponding increase in resources—personnel or funding—we aim to expand the capabilities of the existing UNCL by moving toward list-mode data acquisition and multivariate data analysis. Changes to the electronic system of the JCC-71 are discussed elsewhere [6]. The remainder of this report will focus on the development and testing of multivariate pattern recognition methods for the list-mode neutron collar (LMCL).

1.2 A SELECTION OF OFF-THE-SHELF CLASSIFIERS

The goal of data analysis efforts for the LMCL is to develop and test multivariate methods to differentiate partial fuel defects (i.e., off-normal fuel rod configurations in an assembly) from normal assemblies. We treat this problem as a multivariate classification problem with multiple classes (see [7] for an overview of the canonical classification problem setup). In this section, we briefly summarize common classifiers that could be used to analyze LMCL data to identify and characterize partial fuel defects in nuclear fuel assemblies. It is not our intention to provide an exhaustive treatment of all classification methods and their properties (for this treatment see [7]); however, we summarize several methods to highlight their strengths and weaknesses in an effort to motivate the development of a novel method, discussed further in Section 2.

The k-Nearest Neighbors Classifier

The k-nearest neighbors (kNN) classifier, developed in the 1950s, is an intuitive nonparametric classifier that compares newly acquired (test) data to a library of training data, identifies the k training data points that are closest to the new data point as measured by some distance metric, takes a majority vote of the neighbors' classes, and returns the result of this majority vote as its classification decision [8].

While the kNN classifier is easy to implement and flexible because it makes no assumptions on the data's structure or distribution, it has several shortcomings. There are multiple parameters and meta-parameters that must be tuned to achieve adequate performance. For example, the distance metric and the number of neighbors to consider must be chosen. These model parameters are specified by the user/analyst using additional resampling routines such a cross-validation or by using hold-out datasets. Furthermore, the notion of distance begins to break down in higher dimensions (i.e., when many variables are considered); however, there have been suggested alterations to the kNN classifier to mitigate these difficulties [9]. While there has been extensive research into bounding the error for kNN under various scenarios [10], [11], these results do not provide evidence for the quality of any particular classification decision (i.e., the obvious analog of measurement uncertainty) and they are only relevant to the asymptotic error behavior. Finally, like many other classifiers, the reasoning for the classifier's decisions are not immediately apparent to the analyst, which makes it difficult to make actionable conclusions from the output of kNN.

Linear and Quadratic Discriminant Analyses Classifiers

In contrast with kNN, linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are probabilistic, namely they compute the probability that test data were drawn from the distributions associated with each class in the training data. This statistical feature is predicated on the assumption that the data in each class within the training dataset are drawn from multivariate Gaussians. Thus, training an LDA amounts to estimating the mean vector μ_k and covariance matrix Σ_k for each class $k = 1,2,\ldots, K$ using the maximum likelihood estimator sample mean and sample covariance, respectively. LDA results from assuming the covariance matrix is the same for each class of data, whereas QDA does not make this assumption. To classify new data, the probability of observing the new data is calculated assuming it originated from each of the training data classes in turn, and the class that maximizes this value is assigned as the best-estimate classification decision.

Because of the probabilistic interpretation to the classification decision, interpretation of LDA and QDA can allow estimates of uncertainty. However, while this parametric approach sometimes works well in practice, both methods assume the data are drawn from multivariate Gaussians. If this assumption is violated (i.e., the data are drawn from distributions that are not well approximated by Gaussians), LDA and QDA are not appropriate. Like kNN above, interpreting the decisions made by LDA and QDA are not possible, because one may not immediately be able to discern which variables/features are most important for identifying each class, the relative importance of each variable, or how each data point relates to the rest of the data clouds (i.e., the remaining collection of training data). Finally, by assuming smooth, unimodal, compact Gaussian distributions for each of the classes in the training data, deriving highly irregular and nonlinear decision boundaries are not possible.

Support Vector Machines

Developed by Cortes and Vapnik [12], the support vector classifier (more commonly known as support vector machines [SVMs] when the generalization to regression problems is included) is a nonparametric classifier that is applicable to non-separable data (i.e., the training data clouds of the two classes overlap in their original representation). The SVM relies on finding a hyperplane that separates the two classes of training data and aims to maximize the distance between the data and the hyperplane. This maximum-margin objective is then relaxed to allow some of the training data to fall on the incorrect side of the hyperplane, leading the optimization problem shown in Equation 1.

$$\min_{\beta,\,\beta_0}\left\{\frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{N}\xi_i\right\} \qquad (1)$$

$$\text{subject to } \xi \in \mathbb{R}_+^N,\ y_i\left(x_i^T\beta + \beta_0\right) \geq (1-\xi_i)\ \ \forall\ i$$

Here the measured input data are vectors, $x_i \in \mathbb{R}^d$, with an associated classification y_i, and each example indexed by i. Furthermore, β and β_0 define the data-separating hyperplane, the norm $\|\beta\|$ is the inverse of the margin (distance between the hyperplane and nearest data point) to be maximized, and ξ are the "slack variables" that allow some training data to fall on the incorrect side of the hyperplane (defined in the constraint of the problem). The compromise between minimizing the size of ξ while maximizing the size of the margin between the training data and separating hyperplane is controlled by the trade-off parameter C.

Using the method of Lagrange multipliers for optimization (see [13]), the projection of new data, x_*, onto the optimal hyperplane defined by β and β_0 found above may be written (using the dual form of Equation 1) as shown in Equation 2.

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i x^T x_i + \beta_0 = \sum_{i=1}^{N}\alpha_i y_i \langle x \,|\, x_i\rangle + \beta_0 \qquad (2)$$

As discussed in [7], the vector α is sparse, and nonzero entries are associated with significant training data points Vapnik called the "support vectors."† Finally, the expression of the decision function entirely in terms of inner products on the data may be exploited to generalize the SVM to non-separable cases by transforming the data using a nonlinear transform. The efficacy of such a transform is shown in a simple

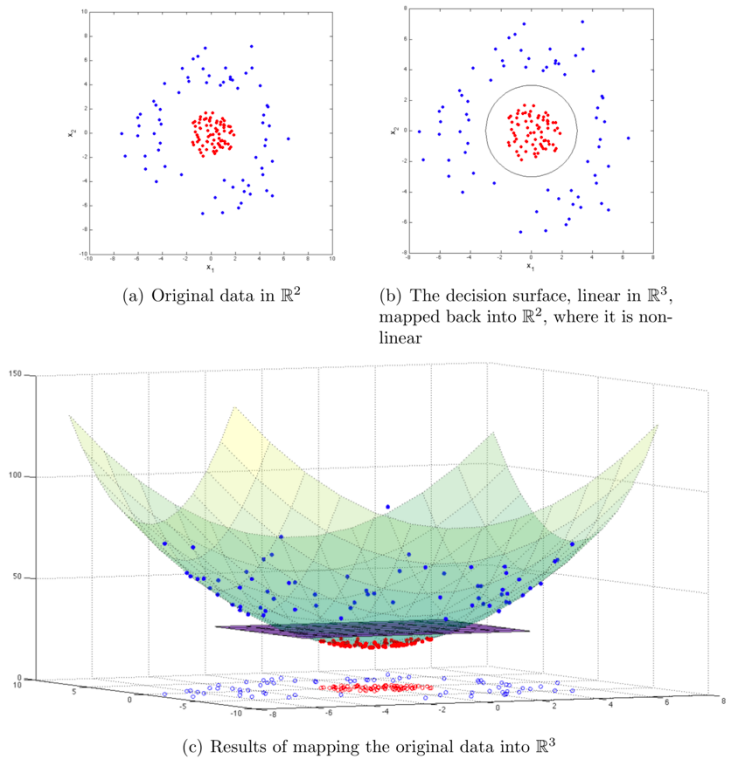
† The term "support vectors" is used because the hyperplane is defined entirely in terms of these training data points (i.e., the plane is supported by these points), and the training data are assumed to be multivariate, making them vectors.

example illustrated in Figure 2. To make final classification decisions (i.e., determine on which side of the separating hyperplane the data x falls) the sign of Equation 2 is taken.

While the SVM is applicable to highly nonlinear and non-separable problems and has been shown highly effective in many applications (see [14] for examples), the SVM has shortcomings. First, the SVM as summarized above is only applicable to problems with two classes (i.e., binary problems). While the developer of the SVM has done much to advance the theory of pattern recognition and classifier performance [15]–[17] and there has been substantial research on expected performance of the SVM (see [18]), the formulation of the SVM is heuristic,[‡] and the signatures implicitly identified by the SVM are not exposed to the analyst for interpretation. While work has been done to estimate probabilities associated with the SVM's classification decisions (e.g., Platt scaling [19]), these methods are ad hoc, and accordingly, can perform inconsistently. For example, we have seen in our work that the deterministic decisions made by an SVM classifier do not agree with the probabilities estimated with Platt scaling.[§] Other authors have noted that Platt scaling can potentially be computationally expensive because of additional cross validation routines, but this is less of an issue with modern computing resources; however, generalizing external ad-hoc probability estimation routines to problems with more than two classes is nontrivial [20]. Finally, like other classifiers, the SVM will always return a classification decision (i.e., assign the new data to one of the classes in the training dataset), even if the data is ambiguous (e.g., the data does not belong to a class in the training data or there is insufficient information to confidently assign a unique classification decision) and little to no indication of the decision quality is given (i.e., there is no estimate of uncertainty for any individual prediction made with a trained SVM model).

[‡] For example, it is not apparent why one should expect the separating hyperplane found by the SVM to necessarily coincide with fundamental structure in the data tied to the data-generating process (e.g., physical phenomena).

[§] We have applied a deterministic SVM classifier to test data generated with simulations of the LMCL to identify off-normal fuel configurations in pressurized water reactor assemblies. We then trained and tested a probabilistic SVM model using Platt scaling using this same dataset and observed inconsistencies in the results of these two classifiers. For example, the probability of belonging to class 1 (off-normal configuration) for one measurement was estimated to be approximately 45%; however, the deterministic classifier assigned this measurement to class 1.



(a) Original data in $\mathbb{R}^2$

(b) The decision surface, linear in $\mathbb{R}^3$, mapped back into $\mathbb{R}^2$, where it is non-linear

(c) Results of mapping the original data into $\mathbb{R}^3$

Figure 2. An example of a nonlinear mapping for a non-separable problem. The first pane (a) shows the original data in $\mathbb{R}^2$, which is mapped into $\mathbb{R}^3$ in the bottom pane (c). In three dimensions, the data is easily separated with a linear plane (purple). The mapped data has been offset vertically to aid visualization. The original 2D data (open circles) are also shown. Finally, the decision surface (black line) is projected back into $\mathbb{R}^2$ in (b). Image by the author from [21].

Neural Networks

Increased availability of large computational resources has led to a resurgence of artificial neural networks (ANN) for machine learning applications and the development of so-called "deep learning." Deep learning leverages massive datasets, large distributed computing platforms, and GPU acceleration to generate networks with hundreds of layers and tens of thousands of connections. ANNs are flexible and relatively simple to implement: The primary training mechanism relies on tuning the weights governing each connection in the network using gradient descent and exploiting a recursive algorithm for calculating the required derivatives (the backpropagation algorithm) [7].

While deep networks are flexible and have been shown extremely capable in areas such as image analysis, the ultimate accuracy of the method requires analyst intervention and subject matter expertise to design the topology of the network. Network designs are typically motivated with physical considerations (e.g., the analyst initializes nodes/connections to represent a physical feature of the problem/data), but there is no guarantee the network will respect the problem physics and desired feature within the trained network. Moreover, it is well documented that ANNs can incorrectly classify test data with high confidence and misclassify slight perturbations of input training, which suggests the trained networks are not extracting physically meaningful features, are overfitting to the training data, or a combination of both [22], [23]. For nuclear nonproliferation applications, the black-box nature of ANNs makes it difficult to draw actionable conclusions from the results of ANN models since no interpretation of the signatures underpinning the

decisions of ANN models is typically possible, and massive amounts of data are needed to train complex networks.

Goals for Classification Methods

In summary, we aim to develop a classifier that may be applied to the analysis of LMCL (and other) data that mitigates the weaknesses of the classifiers summarized above. Our classifier should:

1. be applicable to problems with more than two classes[**],
2. be grounded in a statistical formulation rather than resorting to heuristic or ad-hoc methods,
3. give an indication of *a posteriori* (sample specific) uncertainty in each classification decision as opposed to overall statistics computed on a set of data assumed to represent the quality of all future classification decisions,
4. indicate the most useful variables/features for making predictions, giving insight to the signatures the classifier identifies, and
5. be amendable to non-unique classification decisions (e.g., return a decision of "None-of-the-Above" or identify multiple feasible classification decisions if a single decision cannot be established).

[**] Many classifiers are applicable for two-class problems, but many applications (e.g., image classification) have more than two classes.

2. CLASSIFICATION WITH THE RELEVANCE VECTOR MACHINE

2.1 OVERVIEW

In this section, we summarize our work to develop a novel statistical learning method called the relevance vector machine (RVM). The primary objective of the RVM is to learn a relation between a set of *predictor* variables such as nuclide assay data or counts in each channel of spectrum and a desired *response* that cannot be directly measured such as fuel burnup or type of fuel. If the response is continuous (e.g., fuel burnup), the resulting problem is a regression problem; if the response is discrete (e.g., type of fuel), the problem is one of classification.

First, we present the basic formulation of the RVM, which estimates a real-valued function f that maps multivariate predictor variables to some target variable of interest,

$$f: X \subset \mathbb{R}^d \mapsto \mathbb{R}, \tag{3}$$

using a linear combination of simple functions (i.e., a basis expansion). This function is estimated by analyzing a set of training data for which the predictors and response are known, and this function may then be applied to new predictor data to estimate the unknown response.

The main power of the RVM derives from three key features:

1. The RVM can learn highly nonlinear functions without explicit need for *a priori* knowledge of the underlying data's structure,
2. The statistical framework of the RVM admits statistical interpretation to results (e.g., uncertainty calculations) and scientific defensibility, and
3. The analyst can interpret and understand the implicit signatures identified by the RVM.

After summarizing the RVM for regression problems, including the new developments of the method added in FY2016 and FY2017 to enable integrated feature selection and model tuning, we summarize modifications made in FY2018 to apply the RVM to classification problems with more than two classes, estimate class membership probabilities, and identify variable(s) that drive classification conclusions.

2.2 BASE RELEVANCE MACHINE FORMULATION

Here we summarize the basic form of the RVM as originally developed by Tipping [24], [25]. As summarized in Section 2.1, the RVM is a general function estimator that aims to approximate an unknown function f that maps measurable *predictor* variables (e.g., counts in each channel of a gamma-ray spectrum or the concentration of isotopes in a material) to an unmeasured *response* variable of interest (e.g., burnup of spent nuclear fuel). Herein measured predictor data will be written as d-dimensional vectors, $x \in \mathbb{R}^d$, and responses (sometimes also called target values[††]) that we aim to determine using the predictor data are denoted $t \in \mathbb{R}$. The unknown mapping between predictor data and target values we wish to estimate is shown in Equation 3.

[††]The term "target values" is used as a synonym for response values to highlight our acknowledgment that these values are only known with some average error or noise, σ).

To estimate the function f, the RVM uses a linear basis expansion where each of the basis functions, $\phi_i(x)$, is a kernel function evaluated against one of the training data points, and the basis functions are combined using a weighted sum with weights w_i, as shown in Equation 4.

$$t = f(x) = \sum_{i=1}^{N} w_i \phi_i(x) = \sum_{i=1}^{N} w_i K(x, x_i) \tag{4}$$

Using a kernel function, K, is equivalent to mapping the data to a higher (potentially infinite) dimension using a nonlinear map.‡‡ In addition to capturing nonlinear behavior, using kernel functions computed at each of the input training data points ensures the basis functions are scaled to the problem (i.e., the basis functions' interesting behavior is located within the range of the training data and uninteresting behavior occurs far from typical training data values, where the estimated function is no longer applicable).

While we have tested multiple kernel functions, we have found the radial basis function, shown in Equation 5, to work best for our applications (see [26] for discussion of testing different kernels). Note superscripts are used to identify elements of the vectors x and x_*.

$$\phi_i(x) = K(x, x_i) = \exp\left(-\eta \sum_{k=1}^{d} \left(x_i^k - x^k \right)^2 \right) \tag{5}$$

The parameter η controls the width of the basis functions (and thus controls the locality of the kernel functions). Typically, an appropriate value for η is not known *a priori* and a reasonable value is estimated using cross validation or another similar model selection routine external to training (see [7] for further description).

Given these basis functions, training data $\{x_i, t_i\}_{i=1}^{N}$, of multivariate predictors, and associated target values, t, we need to determine the values for the weighting terms in the basis expansion, $\{w_i\}_{i=1}^{N}$ in Equation 4. To do this, we define a Bayesian inference problem [27], where the values for the vector w are derived from the posterior distribution, as shown in Equation 6.

$$\pi(w \mid X) = p(t \mid X, w, \sigma^2)\, \pi(w) \tag{6}$$

‡‡ This is the so-called "kernel trick," which exploits the equivalence of a kernel function evaluation to an inner product between two data points that have been transformed using a nonlinear map [42]. In other words,

$$K(x_i, x_j) = \langle \varphi(x_i) \mid \varphi(x_j) \rangle_{\mathcal{F}},$$

where $\mathcal{F}$ is the space in which the input data is mapped, $\varphi: \mathbb{R}^d \mapsto \mathcal{F}$. Assuming a learning algorithm may be written entirely in terms of inner products, a nonlinear generalization using φ may be realized using a linear technique.

Here the collection of N vectors of training data are denoted X, and σ is the irreducible noise in the training data, estimated during model training. Next, we define the prior distribution for the weights, $\pi(w)$, and the likelihood function for measuring the training target values given the training predictor data, $p(t \mid X,w,\sigma^2)$.

First, we postulate that the desired function f is simple, and many of the basis expansion weights, w_i, are zero. Thus, we take the prior distribution for w to be a mean-zero Gaussian[§§] as shown in Equation 7.

$$\pi(w) = \mathcal{N}(w \mid 0, \alpha^{-1}) \tag{7}$$

The vector α is a vector of hyperparameters that controls how strongly the weights are driven toward zero. Using the mean-zero prior distribution on the weights (each of which are uniquely associated with a single basis function and associated training data point) encourages sparsity in the expansion shown in Equation 4. This enforces a penalty on the model complexity as opposed to other regression methods such as the Lasso [28], [29], or Elastic Net [30] that use separate regularization terms, which themselves require additional tuning parameter(s) for which appropriate values must be estimated using routines such as cross validation [7]. This sparse form for the function estimate is analogous to the support vector machine (see [7], [12]); and the training data points associated with nonzero weights are termed "relevance vectors" by Tipping [24]; however, the statistical formulation of the RVM admits additional insight into the data's structure and the meaning of the relevance vector(s).[***] The values for α are further discussed below.

Second, by assuming a perfect model with independent, additive Gaussian noise, the likelihood function is assumed to be Gaussian.[†††] This choice allows the first term on the right side of Equation 6 to be expressed as shown in Equation 8.

$$p(t \mid X,w,\sigma^2) = \mathcal{N}(t \mid X, w,\sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(t_i - \sum_{j=1}^{N} w_j \phi_j(x_i) \right)^2 \right) \tag{8}$$

With the choice of likelihood and prior functions, the Bayesian inference problem shown in Equation 4 is almost complete; however, to fully define these functions, the unknown values for the hyperparameters α

[§§] To justify the choice of a mean-zero Gaussian prior for the vector w, we hypothesize the function f is simple, and thus the basis expansion approximation should be sparse (i.e., many entries of the vector w are zero). A mean-zero prior distribution for the expansion weights enforces this simplicity (an informative prior distribution in Bayesian inference is a form of regularization), and the strength of this regularization is controlled by the width of the prior, α^{-1}. Using a Gaussian distribution allows an analytic solution of the posterior distribution for the weights w by conjugacy and is justified by the principle of maximum entropy, where the least restrictive distribution while maintaining a given mean and variance is the Gaussian distribution.

[***] Nonzero weights identify the most important basis functions in the function estimate. Because each basis function is defined as a kernel evaluation against the training data, the nonzero weights identify important training data points. While this allows the analyst to draw additional insight on the structure of the data, the interpretation is more natural in the classification setting. See the section below on classification with the RVM for further discussion.

[†††] A Gaussian likelihood function is standard in regression models that rely on the assumption that the errors in the observed training data are additive and independently and identically distributed according to the Gaussian distribution. This is the standard assumption underpinning ordinary least squares as a maximum likelihood estimator.

and irreducible noise σ need to be determined. Combining Equations 7 and 8 gives the expression for the posterior distribution for the vector of weights w shown in Equation 9.

$$\pi(w \mid X) = \mathcal{N}(t \mid X, w,\sigma^2)\mathcal{N}(w \mid 0, \alpha^{-1}) \tag{9}$$

To find the values for α and σ, we define an optimization problem. We first compute the probability of observing the training data given values for the weight vector w (which itself depends on the hyperparameter α) and marginalizing (integrating) over the weights to isolate the dependence on the variables to be optimized (α and σ^2) as shown in Equation 10.

$$\mathcal{L}(\alpha) = p(t \mid \alpha, \sigma^2) = \log \int_{-\infty}^{\infty} \mathcal{N}(t \mid X, w,\sigma^2)\mathcal{N}(w \mid 0, \alpha^{-1})dw \tag{10}$$

Using Gaussian distributions admits the analytical solution to Equation 10, as shown in Equation 11 [24].

$$\mathcal{L}(\alpha) = -\frac{1}{2}\left(N\log 2\pi + \log\ \left|\sigma^2 I + \Phi A^{-1}\Phi^T\right| + t^T\left(\sigma^2 I + \Phi A^{-1}\Phi^T\right) t\right) \tag{11}$$

Note the new data structures used in Equation 11 to simplify the computation: the matrix A is a diagonal matrix containing the vector α along the diagonal, and the matrix Φ contains all the basis functions defined in Equation 4 evaluated at each of the data points. Further details of Equations 10 and 11 are not especially relevant here, but they are important to generalizing our implementation to classification problems with variable selection.

Equation 11 may be maximized using conventional methods of optimization (see [24], [31]) or with a constructive method developed by Tipping and Faul (see [25]). After optimizing the values for α using Equation 11, the prior distribution $\pi(w \mid 0,\alpha)$ in Equation 6 is defined, the likelihood function is easily computed from the training data, and the posterior distribution for the basis expansion weight vector w is analytically calculated. Thus, regardless of the chosen optimization method, training the RVM amounts to maximizing the marginal likelihood function shown in Equation 11, and the posterior distribution for w is calculated analytically. The best fit values for the weights are taken as the mean of the posterior distribution in Equation 6, μ.

To estimate the response values associated with new data x_*, we evaluate Equation 4, $f(x_*)$, and we calculate the uncertainty associated with this estimate, $u_{\hat{t}_*}$, using the covariance of the posterior distribution in Equation 6, Σ. These predicted target values are shown in Equation 12, and the standard uncertainty in the predicted response is given by Equation 13 [32].

$$\hat{t}_* = \Phi_* \mathrm{w} = \sum_{i=1}^{N} w_i \phi_i(x_*) = \sum_{i=1}^{N} \mu_i \phi_i(x_*) \tag{12}$$

$$u_{t_*}^2 = \sigma^2 + \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial f}{\partial w_i}\frac{\partial f}{\partial w_j}\Sigma_{ij} = \sigma^2 + J^T\Sigma J = \sigma^2 + \Phi_*^T\Sigma\Phi_* \quad (13)$$

Here, σ is the irreducible noise (the variance of the target values about the true response values in the training data, see [7]), and J is the vector of 1st partial derivatives of the function f with respect to the entries of the vector w (Jacobian). Examining Equation 12, these partial derivatives are the basis functions evaluated at the test data point x_*, $\Phi_* = [\phi_1(x_*)\ \phi_2(x_*)...\phi_N(x_*)]$.

2.3 VARIABLE SELECTION AND BASIS SHAPING

As suggested by Tipping in [24], we may replace the single shaping parameter η in Equation 5 with a vector of parameters with an entry for each of the input predictor variables and a unique value that is optimized during training using conventional optimization methods. Equation 14 shows the new basis functions.

$$\phi_i(x) = \exp\left(-\sum_{k=1}^{d}\eta_k\left(x_i^k - x^k\right)^2\right) \quad (14)$$

Note, superscripts are used to identify components of the predictor vectors x_i and x. This generalization can improve the fidelity of the estimated function (see [32] for examples illustrating the importance of η values).

As discussed in [32], we have implemented a sparse hill climbing routine that updates the vector η that is interleaved with the main iterations of α optimization. This routine requires the gradient of the marginal likelihood (Equation 11) with respect to η, shown in Equation 15.

$$\frac{\partial\mathcal{L}}{\partial\eta_k} = \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\partial\mathcal{L}}{\partial\Phi_{ij}}\frac{\partial\Phi_{ij}}{\partial\eta_k}, \quad \Phi_{ij} = \phi_i(x_j), \quad k = 1,2,..., d \quad (15)$$

Tipping provides a convenient expression for the first term in the summation [24],

$$\frac{\partial\mathcal{L}}{\partial\Phi_{ij}} = D_{ij} = \left[\left(C^{-1}tt^T - C^{-1}\right)\Phi A^{-1}\right]_{ij}, \quad (16)$$

where $C = \sigma^2 I + \Phi A^{-1}\Phi^T$, and $\Phi_{ij} = \phi_i(x_j)$. The second term is obtained by differentiating Equation 14 with respect to η_k,

$$\frac{\partial \Phi_{ij}}{\partial \eta_k} = -\Phi_{ij}\left(x_i^k - x_j^k\right)^2. \tag{17}$$

While simple to write, there are several complications we have found in the implementation (see [32] for discussion). By optimizing η to a sparse solution, variables that are most useful for computing the desired response/target are identified, and the relative importance of each variable is computed. The variables that are not useful (or confusing relative to making predictions) are discarded by the second optimization algorithm by setting the associated entry in the η vector to zero.

2.4 CLASSIFICATION WITH THE RVM

For binary classification problems, the likelihood function in the Bayesian setup of Equation 10 is changed to a Bernoulli distribution, as shown below in Equation 18. This function quantifies the probability of the collection of training data falling into the two classes, encoded as 0 or 1.

$$p(t \mid X,w) = \prod_{i=1}^{N} \xi(f(x_i \mid w))^{t_i}(1 - \xi(f(x_i \mid w)))^{1-t_i} \tag{18}$$

Here the function $\xi(\cdot)$ is the logistic function (shown in Equation 19) that maps the continuous function of Equation 6 to the range $[0, 1]$ and aims to approximately map the continuous function to the discrete values $\{0,1\}$.

$$\xi(x) = \frac{1}{1 + e^{-x}} \tag{19}$$

Following the procedure outlined in the previous sections, we need to compute the marginal (log) likelihood function by taking the log of Equation 18 to exchange the product for a summation and then integrate the result over w. However, we note that each target value can only take two values: 0 or 1 by our choice of class label encoding. In other words, Equation 18 has a support of only two values. This discrete structure of Equation 18 makes the log-likelihood impossible to compute exactly. Tipping suggests two approximations [24]. First, approximate the centroid‡‡‡ of the posterior distribution (μ in the regression case where the posterior distribution is Gaussian because all the utilized functions are conjugate) with the most probable value (mode):

$$\mu \rightarrow w_{MP} \tag{20}$$

This is found through iterative optimization (e.g., Newton's method). Second, assume the log-posterior is quadratic about the mean, which we have approximated with w_{MP} above. Defining

‡‡‡ We generalize the notion of the expectation (mean) and most probable (mode) of the distribution.

$$B = \text{diag}\big(\xi(f(x_i \,|\, w))^{t_i}(1 - \xi(f(x_i \,|\, w)))^{1 - t_i}\big), \tag{21}$$

the approximate log-posterior is given by Equation 22 (adapted from [24]).

$$\begin{gathered}\Sigma_{\mathrm{C}} \approx \left(\Phi^T B \Phi + A\right)^{-1} \\ w_{MP} = \Sigma \Phi^T B t.\end{gathered} \tag{22}$$

Note, Σ_C is the approximate covariance matrix in Equation 22 from the exact solution that is possible in the case of regression and the associated Gaussian likelihood function.

To make predictions, we use Equation 23, which gives a statistical estimate for the probability of belonging to class 1 (i.e., the class in the training data that was encoded with $t_i = 1$). Recall the function $\xi(\,\cdot\,)$ is used to map the continuous, unbounded function $f(x)$ to be approximately binary, $\{0,1\}$.

$$\hat{t}_* = \xi(f(x_* \,|\, w)) \tag{23}$$

This native statistical interpretation of the output of the relevance vector machine classifier (RVM-C) precludes the need for postprocessing classifier results to estimate class membership probabilities with routines such as Platt scaling [19], which is predicated on the classifier's output being a good estimate for the inverse of the desired probability, transformed by the logistic function as shown in Equation 24.

$$f(x_*) = \xi^{-1}(\mathbb{P}[t_* = 1]) \tag{24}$$

There is no reason to expect this condition to be met in general for an arbitrary classifier, unless the classifier is designed to satisfy this expression, as is the RVM-C by virtue of the Bernoulli likelihood function (see Equation 18).

2.5 VARIABLE SELECTION AND BASIS SHAPING WITH CLASSIFICATION

To adapt the variable selection and basis shaping routine, we need to modify Equation 15 to reflect the Bernoulli statistics of Equation 18. We note the second partial derivative term, Equation 17, is independent of the marginal likelihood function, so we only need to alter the matrix D in Equation 16. We use a preliminary form for the matrix (see Equation 16 and the associated discussion in [24]) and then substitute the results of the approximations described above (see Equations 20 and 22). The result is shown in Equation 25, and the final expression for the gradient of the marginal likelihood function (i.e., Equation 15) as implemented for classification problems is shown in Equation 26.

$$D = B\big((t - \xi(f(X|w_{MP})))w_{MP}^T - \Phi\Sigma_C\big) \tag{25}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_k} = \sum_{i=1}^{N}\sum_{j=1}^{M} - \left[B\left((t - \xi(f(X|w_{MP})))w_{MP}^T - \Phi\Sigma_C\right)\right]_{ij}\Phi_{ij}\left(x_i^k - x_j^k\right)^2 \tag{26}$$

2.6 CLASSIFICATION WITH MULTIPLE CLASSES

2.6.1 Solution Setup

To analyze problems with more than two classes, we have adopted a one-versus-all decomposition [33], [34], which leverages the statistical framework of the RVM to natively produce multiclass probability estimates. This is a relatively new capability, as estimating the probability of class memberships for problems with multiple classes using conventional classifiers has been an ongoing area of research [35], [36].

With this approach, for a problem with K nonoverlapping classes, we generate K binary RVM classification models, where each model is trained to predict whether data is drawn from a class or the union of all other classes. This setup is shown in Equations 27 and 28, where we denote a subset of the data using a superscript, so the data drawn from the k^{th} class (and the k^{th} class itself) is denoted $X^{(k)}$.

$$X = \bigcup_{k=1}^{K} X^{(k)}, \;\; X^{(i)} \cap X^{(j)} = \emptyset \;\; \forall \;\; i,j \tag{27}$$

$$f_k(x_*) = \mathbb{P}\left[x_* \in X^{(k)} \mid x_* \in X^{(k)} \vee x_* \in \bigcup_{j \neq k} X^{(j)}\right] \tag{28}$$

This is natural as the binary classification RVM (RVM-C) is defined to give the probability that the data is drawn from one of two classes. The multiclass classification decision is then given by applying each of the K binary models and assigning the label associated with the greatest estimated probability value as shown in Equation 29. The uncertainty (*a posteriori* probability of having made an incorrect classification decision) is then computed as one minus this maximal probability value as shown in Equation 30.

$$\hat{t}_* = \underset{k}{\text{argmax}}\left(f_k(x_*)\right) \tag{29}$$

$$u_{\hat{t}_*} = 1 - f_{\hat{t}_*}(x_*) \tag{30}$$

2.6.2 Data-driven Problem Insight

In addition to making (hopefully) accurate classification decisions, the trained RVM-C provides the analyst additional insights to the structure of the data and associated physical processes that govern the data generation and measurement. We illustrate these capabilities with two examples

Interpretation of Relevance Vectors: A Synthetic Data Example

As noted above, by using a mean-zero prior distribution for the weights defining the basis expansion (see Equation 4), the basis expansion is sparse and only a few of the weights are nonzero (i.e., the expansion is comprised of as few elements as possible). The nonzero weights that identify important basis functions are analogous to the support vectors in the SVM [12] and are were termed "relevance vectors" by Tipping [24]. Recall each basis function is generated from a kernel evaluation against one of the training data points. Therefore, each relevance vector is associated with a training data point that is salient to the problem. What do these relevance vectors represent in the classification problem setting?

In reference [24], SVM and RVM classifiers were trained using the synthetic two-dimensional data generated by Ripley [11]. The training data (dots and *x*'s), derived decision boundaries (dashed), and support vectors and relevance vectors (circled) are shown in Figure 3. In this example, a singular η value was used for all variables in the basis functions used in the RVM (see Equation 5), and the value for η was tuned using cross validation [7].

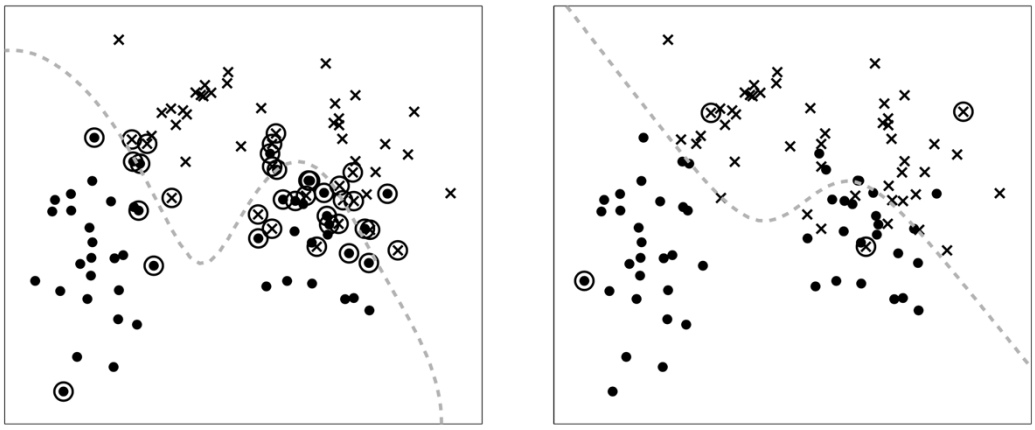


Figure 3. An SVM classifier (*left*) and RVM-C (*right*) for two-dimensional Gaussian mixture data from Ripley [11]**.** For each model, the decision boundary is shown as a dashed line and support vectors and relevance vectors are circled. In general, the support vectors are close to the decision boundary while the relevance vectors are in the center of the data cloud. Adapted from [24].

Ignoring the accuracy of classifications made by the models, two observations about the nature of classifiers may be observed. First, there are fewer relevance vectors than support vectors (whether this difference in complexity affects the ability of each model to generalize remains to be seen). Second, the relative position of the support vectors or relevance vectors within the clouds of training data suggests they are fundamentally different. The support vectors are concentrated near the decision boundary,[§§§] and the relevance vectors are concentrated within the higher density (in terms of number of data points) portions of the data clouds associated with each class. This suggests that the support vectors are associated with the training data edge cases (i.e., the limit of a signature associated with a class of data), whereas the training data points associated with the relevance vectors are prototypical examples of the data class. In some problems, this could help elucidate the underlying structure (the essence) of a class of data. Tipping also suggests basis functions constructed from (and thus centered on) edge cases are unlikely to be reliable estimators of class membership.

[§§§] In actuality, the decision boundary is shaped by the location of the support vectors since the separating hyperplane is written in terms of inner products relative to the support vectors (see [7]); however, interpreting the support vectors by reversing this logic is more intuitive.

Variable Selection: A Reactor Analysis Example

When an RVM model (with basis shaping) is trained, a unique weighting factor is assigned to each variable (dimension of the predictor data) and optimized, with a preference for sparsity. Nonzero variable weights select which variables are relevant to determining the desired target values (if a predictor variable is not correlated to the response or with another predictor that is correlated, its weight is driven toward zero). If all predictor data are in the same units or sufficiently normalized (in our work, we mean-center the input predictor data and rescale to unit variance), then the nonzero variable weights give a ranking of the relative importance of each feature of the training data. An example of this is shown in Figure 4 and further described below.

Nuclide concentration values were calculated using the TRITON routine within Standardized Computer Analyses for Licensing Evaluation (SCALE) 6.2 for 995 discrete positions within a graphite-moderated nuclear reactor. Using a quarter-core model (see reference [32] for more information), the concentration of 90 nuclides were calculated as a function of time and position within the core. An RVM model (in regression mode) was trained to determine the core-average burnup using a single sample (i.e., one vector of concentration values associated with a single position within the core). At the end of training, only 10 entries in η were nonzero (i.e., 10 nuclides were selected and used for making core-average burnup estimates), shown in Figure 4. See [32] for further discussion.

Figure 4. Nuclides used to make determinations of core-average burnup using a single fuel specimen. Bar heights indicate the relative importance of each nuclide. Image by the author from [32].

2.7 SUMMARY

To recap the development of the RVM algorithm for classification problems with multiple classes, prior to its proof-of-concept demonstration for safeguards, fresh fuel verification using simulated list-mode neutron collar data, we briefly enumerate the main points below.

1. Given multivariate data, $x \in \mathbb{R}^d$, and known target values, t, we aim to estimate some function $f(x)$ such that $t \approx f(x)$.
2. The target/response data can be continuous (i.e., a regression problem) or discrete (i.e., a classification problem).
3. The RVM uses a basis expansion to construct the function, $f(x) = \sum_i w_i \phi_i(x)$, where
    a. The library of possible basis functions are generated using the training data and some kernel function, where the i^{th} basis function is a kernel evaluation against the i^{th} training data vector, $\phi_i(x) = K(x,x_i)$, and
    b. Bayesian inference is used to determine the values for the weights, w_i, that combine the basis functions. The likelihood function is chosen based on the type of problem: a Gaussian likelihood for regression problems and a Bernoulli distribution for classification problems. A mean-zero Gaussian is used for the prior, which natively adds regularization to the problem and enforces an inherent simplicity to the basis expansion and drives many w_i towards zero.

4. The basis functions (and associated training data used to construct the function) associated with nonzero w_i are called *relevance vectors*. The relevance vectors are canonical examples of each class of data and/or important training data used to discern the structure of the function $f(x)$ and the underlying data generating process that is being estimated.
5. A second optimization problem is added to select the most important variables (i.e., features) for predicting the desired targets, t.
6. The Bayesian setup of the RVM allows for rich statistical interpretation and results. Estimates of target values for new data are expected values from the Bayesian posterior predictive distribution, and the associated variance gives sample-specific estimates of uncertainty.
7. The discrete support of the Bernoulli likelihood distribution used for classification requires modification of many terms in expressions used during training of the RVM.
8. A one-versus-all problem decomposition was implementation to give the RVM the ability to analyze classification problems with more than two classes. The result gives estimated probabilities for class membership and estimates of misclassification probability. Future work will further analyze these probabilities to determine "None-of-the-Above" classification decisions.

3. ANALYSIS OF LIST-MODE NEUTRON COLLAR DATA

3.1 INTRODUCTION

We now turn our attention to the motivating problem for the development of Section 2: classification of normal and off-normal fuel rod configurations in a fresh fuel assembly using list-mode data collected from an array of neutron detectors and active neutron interrogation. A basic schematic for the LMCL is shown in Figure 5. As discussed in Section 1, an array of 18 neutron detectors, arranged in three banks or arrays of six detectors each, are embedded in moderating material (not shown), and assembled into a collar configuration that is placed around a fresh commercial nuclear fuel assembly. The fourth side of the collar contains no neutron detectors but instead contains a sealed neutron source (e.g., AmLi, AmBe, PuBe). Neutrons from the source are incident on the fuel where fission and associated neutron multiplication occurs. The fresh fuel assembly used for this work is a pressurized water reactor (PWR) fuel design with a 17 x 17 fuel rod configuration. Figure 5 shows the fuel assembly divided into 4 quadrants for this analysis. To date, the singles and doubles neutron count rates have been recorded treating the output of all tubes in series (i.e., combining their signals with a logical OR) to verify the integrity of the fuel rod configuration. See [3] and [4] for more information.

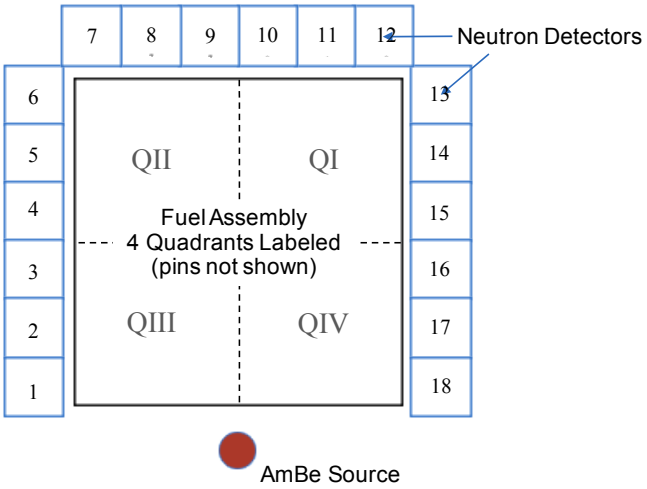


Figure 5. Basic geometry for the LMCL.

Ongoing work aims to explore the efficacy of using list-mode data taken from the neutron collar (i.e., analyzing each event in each detector separately as well as coincidences between events) to verify fuel configuration integrity. To this end, list-mode data has been simulated with radiation transport methods and analyzed with RVM-C models to identify and exploit multivariate, time-dependent signatures of off-normal (i.e., missing fuel rods) fresh fuel rod configurations.

3.2 SIMULATED DATA

MCNP was used to simulate 75 LMCL measurements, and neutron multiplicity values were derived using the PTRAC feature [37]. A random collection of 71 "detector networks" were used to evaluate coincident detection events. Signals from detectors within a network were combined with a logical OR, and doubles rates were determined by applying a logical AND between detection events, with the additional constraint

that events must occur within a specified time of each other to be considered coincident. Thus, 142 variables were computed and analyzed (a singles and doubles rate for each of the detector networks).[****] Combing counts from each detector in the detector network complicates the physical interpretation of the various count rates (i.e., associating count rates in a detector network with count rates in each neutron detector in the collar). For simplicity, we have accordingly decided to neglect the doubles in this initial analysis. Thus, 71 predictors are analyzed.

The 75 simulated measurements fell into one of 5 classes. The first class, where all fuel rods are present, was defined as the normal configuration. In each of the other four classes, 25% of the fuel rods were removed, and all missing fuel rods were taken from one of the four quadrants of the assembly. Two such off-normal configurations are shown in Figure 6. Due to neutron transport and the fission process, changes in the count rates in each of the neutron detectors are expected to be indicative of the type of partial fuel defect (i.e., how many and which rods are removed). The sensitivity and specificity of these potential signatures is a subject of ongoing research.

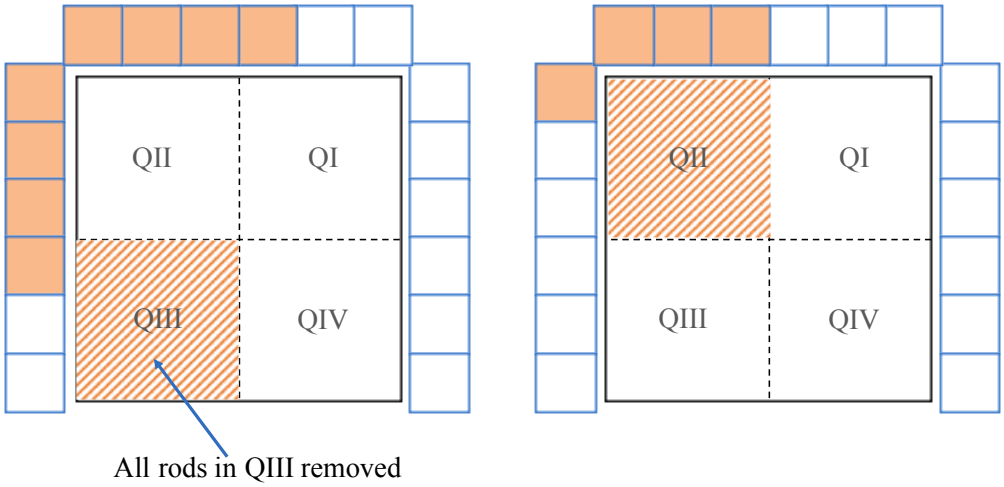


Figure 6. Difference in singles count rates from two off-normal fuel rod configurations. Neutron detectors for which the average singles count rate increases relative to the average normal fuel configuration values are highlighted in orange. The changes in count rates when all rods in QIII are removed (left) are different than when rods in QII are removed (right).

Of the 75 MCNP simulations, a training set of 58 measurements was randomly chosen, and the remaining 17 were reserved for testing the models. Both sets contained measurements from all five classes of data.

3.3 RESULTS

For this initial problem of the detection of fuel rod removal scenarios (i.e., off-normal fuel rod configurations) all 17 testing samples were correctly classified. The probability of class membership (i.e., $\{f_k(x_*)\}_{k=1}^{5}$) for each of the 17 testing samples are reported below in Table 1. The relatively constant

[****] In the future, we will aim to compute the pairwise coincidence rates between each individual detector. This would yield a matrix of $\Sigma_{n=1}^{18} n = 171$ coincidences in addition to the singles rates of each of the 18 detectors. Computing the doubles rates for single-detector networks as well as networks with multiple detectors contains redundant information. Ideally, each variable/dimension in the predictor data should be unique to enable the clearest interpretation of the optimized, sparse η vector and remove potentially confusing conflated information that may be implicitly reported in more than one detector network.

values for $f_k(x_*)$ may be associated with the apparent ease of the classification problem. In addition, we note the probabilities of class membership associated with the normal fuel configuration (class 1) are essentially constant (to the reported precision) and small compared to the values associated with each of the off-normal fuel configurations (classes 2-5). This suggests the data associated with the four off-normal fuel configurations (i.e., missing fuel rods) are more distinct than data associated with the normal configuration. These types of behavior have not been observed in other verification analyses where partially overlapping synthetic datasets (i.e., more difficult classification problems) were analyzed. We believe the nonintuitive results associated with Class 1 data may be remedied by utilizing larger datasets to better capture trends in the data indicative of in-tact (normal) fuel assemblies with no missing fuel rods.

Table 1. Results from a one-versus-all RVM-C analysis of the initial LMCL dataset. All classification decisions are correct. Probabilities of class membership associated with ultimate classification decision (shown in bold) are very high, suggesting this initial dataset is "easy" to classify.

True Class	Probability of Class 1	Probability of Class 2	Probability of Class 3	Probability of Class 4	Probability of Class 5
1	**0.2241**	0.0016	0.0016	0.0016	0.0016
1	**0.2241**	0.0016	0.0016	0.0016	0.0016
2	0.2241	**0.9931**	0.0016	0.0016	0.0016
2	0.2241	**0.9933**	0.0016	0.0016	0.0016
2	0.2241	**0.9930**	0.0016	0.0016	0.0016
2	0.2241	**0.9932**	0.0016	0.0016	0.0016
3	0.2241	0.0016	**0.9926**	0.0016	0.0016
3	0.2241	0.0016	**0.9930**	0.0016	0.0016
3	0.2241	0.0016	**0.9932**	0.0016	0.0016
4	0.2241	0.0016	0.0016	**0.9936**	0.0016
4	0.2241	0.0016	0.0016	**0.9930**	0.0016
4	0.2241	0.0016	0.0016	**0.9932**	0.0016
4	0.2241	0.0016	0.0016	**0.9934**	0.0016
5	0.2241	0.0016	0.0016	0.0016	**0.9935**
5	0.2241	0.0016	0.0016	0.0016	**0.9934**
5	0.2241	0.0016	0.0016	0.0016	**0.9936**
5	0.2241	0.0016	0.0016	0.0016	**0.9933**

For each of the models associated with the four fresh fuel assembly partial defects (i.e., missing a fraction of the fuel rods), the entries of the final η vector were ranked, and the eight most important detector networks associated with each class are reported in Table 2. The identified detector networks for the QIII-missing data class are shown in Figure 7. When fuel rods in the third quadrant are removed, the fission rate is anticipated to decrease in QIII and increase in QII because of a decrease in fissile material content (QIII fuel rod removal) and additional streaming or penetration of interrogating source neutrons (QII no longer shielded by QIII) respectively. Note the identified detector networks surround either QII, QIII, or both—presumably to implicitly compare the two changes in fission rate. In the first network (top left), detectors near QI are included, which we believe are used to compare the changes in fission rate in QII compared to the unchanged baseline fission rate in QI.

Table 2. Eight most important detector networks for identifying each of the four partial fuel defects using the multiclass RVM-C. Important detector networks were identified using the entries of the optimized variable weighting vector (η) from each of the binary RVM-C models constructed in the one-versus-all decomposition. Recall, signals from each detector in the network are combined using a logical OR.

	QI Removed	**QII Removed**	**QIII Removed**	**QIV Removed**
Ranked Detector Networks	8 17	2 11	4 5 6 10 11 12	7 8 9 13 14 15
	7 8 9 16 17 18	1 2 3 10 11 12	3 4 9 10	9 10 15 16
	9 16	3 10	3 10	10 15
	7 8 9 13 14 15	4 5 6 10 11 12	4 9	9 16
	7 8 17 18	1 2 11 12	4	1 2
	2 11	7 8 9 16 17 18	5	14
	1 2 3 10 11 12	8 17	4 5 6	13 14 15
	16	3	1	15

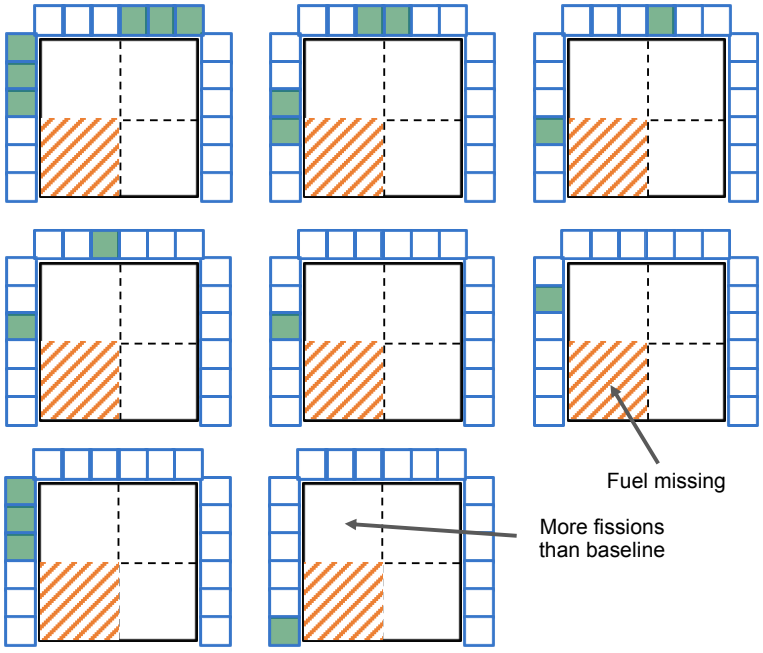


Figure 7. Eight most important detector networks for identifying QIII-missing partial fuel defect from LMCL data using RVM-C. When fuel rods in QIII are missing, two major physical effects are anticipated: (1) the fission rate in QIII goes down, and (2) the fission rate in QII increases as more neutrons are incident on QII fuel rods. All identified detector networks are highlighted in green.

4. CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

Work to date has indicated that improvements can be made to the reliability of partial defect detection by exploring classification methods. A novel classification method has been demonstrated for the correct detection of fuel rod removal within 17 simulated fresh fuel removal scenarios or test cases. Here we summarize several avenues for further research that could be explored outside of the current project scope.

Measurement Noise by Directly Simulating Measurement Time

To date we have computed each set of measurement data directly with MCNP6 and processed each neutron history online with an external script. This approach was chosen to limit the storage required to save the results from each simulation, but a new time-intensive simulation is needed for each synthetic measurement (potentially negating the storage savings) and it is difficult to correctly assess and control the measurement noise. In addition, changes to the coincidence logic are not possible. We aim to overcome these shortcomings with an entirely new approach to generating datasets. For each fuel configuration, we will generate a large pool of particle history data, and we will use nonparametric bootstrapping to generate realistic list-mode data by generating a Poisson process on which to overlay the output of the bootstrap resampling. The proposed algorithm is summarized below.

1. Generate pool of histories

   For each combination of fuel assembly type and partial fuel defect scenario (e.g., 1/8 of rods removed in a block at the back of the assembly oriented to the AmLi source), an MCNP6 model will be executed, and N histories will be simulated and PTRAC results will be recorded. Each history corresponds to a neutron emitted from the AmLi source. For each history, every neutron detection is recorded by a pair of numbers (tuple) denoting the time of the interaction in the detector relative to the beginning of the history and the detector in which the interaction occurred. Noting that multiple detections may occur for a given history, especially if fission occurs in the fuel assembly, the pool of histories is shown in Equation 28.

   $$\{(\Delta t_j, d_j)\}_{j \in \mathcal{J}_i} \quad i = 1, 2, \ldots, N \tag{28}$$

   MCNP6 restarts the time variable at the beginning of each history. Thus, the time of each detection hit is given relative to the beginning of the history. To make this explicit, the time variable is written in Equation 28 as a time difference, Δt, and the detector in which in the j^{th} interaction occurred is d_j , $d_j \in \{0, 1, 2, \ldots, 18\}$, where $(\Delta t_j, d_j) = (0,0)$ indicates no detection occurred in the detectors.

2. Calculate the number of histories for a measurement

   Radiation counting is described by a Poisson process. The parameter for the process is governed by the activity of the AmLi source, which itself depends on the half-life of ^{241}Am. Because the half-life of ^{241}Am is effectively infinite relative to the anticipated acquisition time of the LMCL, we will model the neutron emission of the AmLi source as a homogeneous Poisson point process [38]. For a source strength of λ s^{-1} and an acquisition time of τ s, the number of decays that will occur, X, is distributed as shown in Equation 29

   $$\mathbb{P}(X = n) = \frac{(\lambda\tau)^n}{n!} e^{-\lambda\tau} \tag{29}$$

Equation 29 will be sampled to decide the number of neutrons the AmLi source emits.

3. Generate Poisson process

For x neutron emissions, the time for each emission will be sampled according to a Poisson point process with parameter $\Lambda = \lambda\tau$. The time between each event (survival function) in the process is distributed exponentially. We will sample the survival function and construct the list of times for each neutron emission. Starting from the beginning of the measurement, $t_0 = 0$, the times for the x neutron emissions, $\{t_i\}_{i=1}^{n}$, are given recursively as shown in Equation 30.

$$\begin{aligned} t_{i+1} &= t_i + v_i \\ v_i &\sim (1 - e^{-\lambda v}) \end{aligned} \tag{30}$$

4. Sample histories

From the pool of N histories, n histories will be uniformly sampled with replacement, and this finite sample will asymptotically have the same distribution as the (unknown) population [39]:

$$\left(i, \{(\Delta t_j, d_j)\}_{j \in \mathcal{J}_i}\right) \quad i = 1, 2, \ldots, n \tag{31}$$

Note the index i in the first position within the tuple in Equation 31 is simply for identifying purposes.

5. Overlay histories onto Poisson process

For each of the n histories, we associate a time from the Poisson process generated above. This is shown in Equation 32.

$$\left(i, t_i, \{(\Delta t_j, d_j)\}_{j \in \mathcal{J}_i}\right) \quad i = 1, 2, \ldots, n \tag{32}$$

Next, we calculate the absolute time for each detector hit using Equation 33.

$$(t_i + \Delta t_j, d_j),\ j \in \mathcal{J}_i,\ i = 1, 2, \ldots, n \tag{33}$$

Finally, we sort the tuples in ascending order of time to form the list-mode data and reindex:

$$\left\{\{(t_i + \Delta t_j, d_j)\}_{j \in \mathcal{J}_i}\right\}_{i=1}^{n} \mapsto \{(t_k, d_k)\}_{k=1}^{K}. \tag{34}$$

6. Apply coincidence logic

From the list-mode data, we bin the data to calculate the number of singles counts for each of the 18 detectors, and we apply a logical test on the difference in time for detection events to calculate the pairwise coincidence rates. The singles counts are collected into a vector with an entry for each

detector, $s \in \mathbb{Z}^{18}$, and the coincidence counts are collected into a matrix, $C \in \mathbb{Z}^{18 \times 18}$. To calculate the coincidence counts, each entry of C is initialized to 0, and incremented according the rule below.

FOR each event $k = 1, 2, ..., K$,
IF $|t_k - t_j| \leq \epsilon$ for some j,
Increment the coincident rate $C_{d_k,d_j} = C_{d_k,d_j} + 1$,

New Defect Scenarios

To date, we have performed proof-of-concept data analysis on a single simulated dataset comprised of one fuel assembly type and 1 partial fuel defect scenario: (1) a 17 × 17 pressurized water reactor (PWR) fuel in a normal configuration and (2) one off-normal 25% pin removal in a block (i.e., one quadrant of the pins removed). This work has demonstrated list-mode data may be analyzed to differentiate normal fuel configurations from partial defects, and the position of the missing rods may also be identified.

In the future we hope to simulate additional fuel configurations to test the ability of the RVM-C to identify and localize more subtle fuel defects. Variables we wish to explore include

1. fewer numbers of removed fuel rods (e.g., 1/8 assembly, 8 rods, single rod),
2. removed rod position (e.g., back of assembly, front of assembly, center of assembly, edge of assembly), and
3. distribution of removed rods (i.e., a contiguous block versus randomly dispersed through the rod array).

Analyze Pairwise Coincidence Rates to Identify Signatures

As discussed in Section 3.2, 71 "detector networks" were used to define coincident neutron detection events. Owing to limitations in computing resources and memory requirements, singles and doubles rates were computed online using a user-defined subroutine that was interfaced with MCNP during PTRAC calculations. While we have demonstrated the capability of the variable-selection/basis-shaping routine integrated in the RVM-C training algorithm to rank the most useful of these detector networks for classifying the fuel defects (see Table 2 and Figure 7), because of the complexity of the chosen detector networks, physical interpretation of the most important networks is difficult. In the future, we intend to calculate singles count rates and all pairwise coincidence rates. This will yield 171 coincidence rates and 18 singles rates; however, this larger number of variables will be systematic, allowing interpretation of the most useful detector combinations for characterizing partial fuel defects. The results of these studies can inform future development and optimization of neutron collar instruments.

Noise-Mitigating Models

Our working hypothesis is the multivariate analysis of the list-mode data collected with the LMCL will be more robust to measurement noise than univariate analysis methods, and we expect our classifier's performance to degrade as shorter measurement times are simulated and the associated measurement noise increases. In previous work, we have demonstrated a simple method to develop RVM models that are more robust to measurement noise. To develop these noise-mitigating models, we use parametric bootstrapping [39], [40] to generate duplicates of our training data with new noise realizations. This effectively replaces the individual data points (vectors) associated with each training measurement with a finite sample representing the underlying distribution associated with the measurement. By training on these distributions of data (as opposed to points of data), preliminary studies suggest the RVM can distinguish multivariate

changes associated with the response (i.e., the presence and type of fuel defect) from measurement noise, and the accuracy of RVM predictions dependence on the noise in the testing data is reduced. In fact, in some cases, this "smearing" of the training data improves the RVM's performance even when no noise is present in the testing data.

Asymmetric Misclassification Cost (Decision Analysis)

The probabilistic nature of the RVM-C allows for novel analysis capabilities. First, additional decision analysis criteria may be applied to the probabilities of class membership to account for differences in the penalty for misclassifying data drawn from each class. This may be used to minimize a risk metric or enforce difference confidence requirements for each class of data. For example, it may be desired to make conservative decisions and only declare a fuel assembly as defective (and the operator in noncompliance) if the measurement is very precise and the associated classification decision made by the RVM-C carries a small uncertainty. In the case of a low-confidence, off-normal result, additional measurements may be performed and analyzed before declaring the operator in noncompliance.

None-of-the-Above Decisions

Using another novel probabilistic classifier under development at ORNL, we have previously shown that analysis of class membership probabilities may be analyzed to identify test data that are inconsistent with all classes in the known training data [41]. In such situations, it is desirable to report a "None-of-the-Above" result, rather than return a specious classification result. This capability is relevant to many classification problems; however, this capability is not widely available and its application to the RVM is under development.

5. REFERENCES

[1] L. Worrall, A. Nicholson, and S. Stewart, "List Mode Response Matrix for Advanced Correlated Neutron Analysis for Nuclear Safeguards," in *Proceedings of the Institute of Nuclear Materials Management Annual Meeting*, 2016.

[2] H. O. Menlove, "Description and performance characteristics for the neutron coincidence collar for the verification of reactor fuel assemblies," Los Alamos, 1981.

[3] M. Technologies, "Model JCC-71, 72, & 73 Neutron Coincidence Collars, Data Sheet C38898-07/2011," 2017.

[4] D. Reily, N. Ensslin, and H. Smith, *Passive Nondestructive Assay of Nuclear Materials*. Nuclear Regulatory Commission, 1991.

[5] N. Ensslin, W. H. Gesit, M. S. Krick, and M. M. Pickrell, "Active Neutron Multiplicity Counting," in *Passive Nondestructive Assay of Nuclear Materials (Addendum)*, 2007.

[6] L. Worrall *et al.*, "Verification Data Pattern Recognition and Change Detection at the Neutron Instrument Level," in *Advancements in Instrumentation Data Processing and Analysis*, 2018.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[8] E. Fix and J. Hodges, "Discriminatory analysis--nonparametric discrimination: Consistency properties," Randolph Field, TX, 1951.

[9] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Recognit. Mach. Intell.*, vol. 18, pp. 607–616, 1996.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-11, pp. 21–27, 1967.

[11] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[13] S. Boyd, *Convex optimization*. Cambridge University Press, 2004.

[14] O. Ivanciuc, "Applications of support vector machines in chemistry," *Rev. Comput. Chem.*, vol. 23, pp. 291–400, 2007.

[15] V. Vapnik and A. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to their Probabilities," *Theory Probab. its Appl.*, vol. 16, no. 2, pp. 264–280, 1971.

[16] V. Vapnik and A. Chervonenkis, "Ordered Risk Minimization I.," *Autom. Remote Control*, vol. 35, pp. 1226–1235, 1974.

[17] V. Vapnik and A. Chervonenkis, "Ordered Risk Minimization II.," *Autom. Remote Control*, vol. 35, pp. 1403–1412, 1974.

[18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer, 1996.

[19] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. large margin Classif.*, vol. 10, no. 3, pp. 61–74, 1999.

[20] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," *Proc. 22nd Int. Conf. Mach. Learn. - ICML '05*, no. 1999, pp. 625–632, 2005.

[21] K. Dayman, "Multivariate Analysis Applied to the Characterization of Spent Nuclear Fuel," 2012.

[22] C. Szegedy *et al.*, "Intriguing properties of neural networks," pp. 1–10, 2013.

[23] J. Yosinski, J. Clune, A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Comput. Vis. Pattern Recognit.*, 2014.

[24] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[25] M. E. Tipping and A. C. Faul, "Fast Marginal Likelihood Maximization for Sparse Bayesian Models," *Proc. Ninth Int. Work. Artif. Intell. Stat.*, pp. 1–5, 2003.

[26] K. Dayman, B. Ade, and C. F. Weber, "Sparse Bayesian Regression with Integrated Feature Selection for Nuclear Reactor Analysis," in *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, 2017.
[27] J. K. Ghosh, M. Delampady, and A. Samanta, *An Introduction to Bayesian Analysis. Theory and Methods*. 2006.
[28] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
[29] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 73, no. 3, pp. 273–282, 2011.
[30] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 67, no. 5, pp. 768–768, 2005.
[31] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 383–389, 2002.
[32] K. Dayman, B. Ade, and C. Weber, "Sparse Bayesian Regression with Integrated Feature Selection for Nuclear Reactor Analysis," in *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, 2017.
[33] L. Bottou *et al.*, "Comparison of classifier methods: A case study in handwriting digit recognition," *Proc. Int. Conf. Pattern Recognit.*, pp. 77–87, 1994.
[34] C. Hsu and C. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
[35] B. Zadrozny and C. Elkan, "Transforming Classifier Scores into Accurate Multiclass Probability Estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 694–699.
[36] T. Wu, C. Lin, and R. C. Weng, "Probability Estimates for Multi-class Classification by Pairwise Coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.
[37] J. Goorley, M. James, T. Booth, and F. Brown, "Initial MCNP6 Release Overview--MCNP6 version 1.0," 2013.
[38] J. F. C. Kingman, *Poisson Processes*. Clarendon Press, 1992.
[39] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *Ann. Stat.*, vol. 7, no. 1, pp. 1–26, 1979.
[40] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.
[41] K. J. Dayman and C. F. Weber, "Flexible classification with spatial quantile comparison and novel statistical features applied to spent nuclear fuel analysis," *J. Radioanal. Nucl. Chem.*, vol. 318, no. 1, pp. 605–618, 2018.
[42] J. Mercer, "Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations," *Philos. Trans. R. Soc. A*, vol. 2019, pp. 415–446, 1909.