Journal of Data Science 0 (0), 1–21
??? 2025
DOI: 10.6339/25-JDS1212
Statistical Data Science

Explainable Machine Learning for Functional Data

Katherine Goode[1,*], J. Derek Tucker[1], Daniel Ries[1], and Heike Hofmann[2]

[1]*Sandia National Laboratories, Albuquerque, NM, United States*
[2]*Department of Statistics, University of Nebraska-Lincoln, Lincoln, NE, United States*

Abstract

Black-box machine learning models are recognized as useful tools for prediction applications, but the algorithmic complexity of some models causes interpretation challenges. Explainability methods have been proposed to provide insight into these models, but there is little research focused on supervised modeling with functional data inputs. We argue that, especially in applications of high consequence, it is important to explicitly model the functional dependence in a black-box analysis to not obscure or misrepresent patterns in explanations. As such, we propose the ***V**ariable importance **E**xplainable **E**lastic **S**hape **A**nalysis (VEESA) pipeline* for training supervised machine learning models with functional inputs. The pipeline is an analysis process that includes the data preprocessing, modeling, and post-hoc explanations. The preprocessing is done using elastic functional principal components analysis, which accounts for vertical and horizontal variability in functional data and, ultimately, allows for explanations in the original data space that identify the important functional variability without bias due to correlated variables. Here, we demonstrate the pipeline on two high-consequence applications: explosives classification for national security and inkjet printer identification in forensic science. The applications exhibit the VEESA pipeline's ability to provide an understanding of the characteristics of the functional data useful for prediction. Code for implementing the pipeline is available in the *veesa* R package (and supplemental python code).

Keywords *elastic shape analysis; explainability; functional principal components; interpretability; variable importance*

1 Introduction

Many machine learning models are considered "black-boxes" since their algorithmic complexity results in the inability to assign a physical meaning to the model parameters. The interpretation of model parameters provides an understanding of the relationships between model inputs and outputs, which helps with model assessment, data insight, and building appropriate model trust. In low consequence applications (e.g., movie recommendations or personalized advertisements), a lack of model "interpretability" may be permissible when incorrect predictions result in less severe repercussions. However, in high-consequence areas (e.g., national security and forensics science), it is difficult to motivate the use of a non-interpretable model when model transparency is essential to avoid serious mistakes.

With that said, there are scenarios where black-box models are still selected for analyses in high-consequence scenarios. This could be due to meaningful improvement in prediction perfor-

*Corresponding author. Email: kjgoode@sandia.gov.

Received March 27, 2025; Accepted December 8, 2025

mance over inherently interpretable models or an interest in using a data-driven approach due to a lack of understanding of the scientific mechanism. We do not intend this methodology as a replacement for fully interpretable methods (e.g., LASSO), which are preferable when they meet the needs of an analysis. Instead, we propose an analysis approach intended to provide more opportunities for diagnosing and understanding a model in the scenario when a black-box model is used with functional data inputs for supervised learning.

The desire to contextually explain how predictor variables relate to black-box decisions has led to an explosion of research on explainable/interpretable machine learning (Adadi and Berrada, 2018; Rudin et al., 2022; Sankaran, 2024). Many approaches are post-hoc methods applied to a model after training that aim to quantify variable relationships captured by the model. Previous explainability work considers different types of data such as image (Chen et al., 2019) and spatio-temporal (Goode et al., 2024) data. However, there is minimal literature on explainability with functional data.

Functional data consist of observations each composed of a collection of points representing a continuous curve or surface over a compact domain (e.g., a fixed length of time or region of space). Examples include heights of children measured over time (Ramsay and Silverman, 2005) and silhouettes of animals extracted from images (Srivastava and Klassen, 2016). Modern technology has made the collection of functional data commonplace with devices such as glucose monitors (Danne et al., 2017) and environmental sensors (Butts-Wilmsmeyer et al., 2020). Functional data provide detailed information of a continuous process but require special consideration to appropriately account for the functional structure.

One approach to using functional data as predictive model inputs is to compute relevant summary statistics of the functions, but this may result in loss of information and incorrect inference (Srivastava and Klassen, 2016). Another approach is to create vectors of values across functions at each domain location (e.g., time or location) and use these *"cross-sectional"* vectors as model input features (Tian, 2010). A clear downside to the cross-sectional approach is that the dependence between points within a function is ignored. Alternative methods have been developed that treat the underlying curve as an infinite dimensional continuous function such as functional regression and functional principal component analysis (fPCA) (Ramsay and Silverman, 2005). In the machine learning literature, dimension reduction techniques, such as fPCA, have been used as a preprocessing step to capture the functional dependence (Li et al., 2014; Ries and Gabriel Huerta, 2023).

The few papers that consider explainability with functional data inputs include Martin-Barragan et al. (2014) and Thind et al. (2023). These two works focus on model-specific methods that adapt the methodology of support vector machines and neural networks, respectively, to account for functional data. Ha et al. (2021) explore another approach for neural networks that uses adaptive wavelet distillation to distill information from the model into an interpretable wavelet transform. Goode et al. (2020) suggest a model agnostic approach that uses fPCA for dimension reduction and feature importance for explainability. These previous approaches, however, only account for *vertical* variability in the functions (also known as y or amplitude variability). Tucker et al. (2013) highlight that functional data possess both vertical and *horizontal* variability (also known as x or phase variability), and if one variability is ignored, the resulting analysis may not accurately capture the structure in the data.

Here, we build on the approach of Goode et al. (2020) and propose the ***V****ariable importance* ***E****xplainable* ***E****lastic* ***S****hape* ***A****nalysis (VEESA)* pipeline as a novel explainable machine learning analysis approach for supervised learning with functional data inputs that accounts for both vertical and horizontal variability. We use the term *pipeline* to refer to the analysis process

that includes the data preprocessing, modeling, and post-hoc explanations. With the VEESA pipeline, we aim to provide more trustworthy insight into how a black-box model uses the data for prediction by first transforming the functional data via *elastic fPCA (efPCA)* (Tucker et al., 2019). efPCA directly captures vertical and horizontal variability and provides orthogonal inputs, which allows for the application of post-hoc explainability techniques without the concern of bias in the results due to correlation (Hooker et al., 2021). After explainability is used to identify important elastic functional principal components (efPCs), the pipeline uses visualizations of the variability captured by the efPCs in the original data space to understand the characteristics of the functions used by the model for prediction.

Code for implementing the VEESA pipeline is available via the *veesa* R package (Goode and Tucker, 2025). The package contains functions for preparing the input training/testing/validation data for the pipeline based on the *fdasrsvf* R package (Tucker, 2025b). The prepared data may then be used with any modeling package. Functions are also provided in *veesa* for visualizing the important efPCs, and demonstrations of the code are provided in the GitHub repository: https://github.com/sandialabs/veesa. Python code for implementation of the VEESA pipeline (based on the *fdasrsf* Python package (Tucker, 2025a)) is provided in the supplemental material.

The pipeline works for classification and regression, but in this paper, we focus on classification. Further, any explainability approach that is suitable for the model selected may be used with the pipeline. For simplicity, we only consider permutation feature importance (PFI) (Fisher et al., 2019) in this paper. PFI is a model-agnostic method, which allows us to demonstrate the method across applications with different model types. Additionally, we use PFI for global explanations (i.e., explanations over a set of data). However, in practice, a combination of other model-agnostic, model-specific, global, and local (i.e., explanations for a specific prediction) are encouraged for a more holistic understanding of a model. Examples include partial dependence plots (PDPs) (Friedman, 2001), individual conditional expectation (ICE) plots Goldstein et al. (2015), and Gini random forest feature importance (Breiman et al., 1984).

We demonstrate the VEESA pipeline on two high consequence classification tasks. The first task aims to identify explosive materials given hyperspectral computed tomography scans for national security purposes. The second is a forensic science application with the goal of identifying the source printer for illicitly printed documents (e.g., counterfeit currency) based on Raman spectroscopy. In both spaces, it is important to account for the functional dependence to not obscure or misrepresent patterns. We highlight how visualizing the important efPCs in the original data space aids model developers and subject matter experts (SMEs) in determining whether a model is drawing on reasonable phenomenological characteristics for prediction.

The paper is structured as follows. Section 2 provides background efPCA and PFI. Section 3 describes the VEESA pipeline and methodological developments. Demonstrations of the VEESA pipeline with the explosive material and inkjet printer applications are included in Section 4. Finally, we discuss the advantages, limitations, and future research directions in Section 5. The code and implementation details for all analyses are available in the supplemental material.

2 Background

This section provides background on elastic shape analysis (ESA) (Section 2.1), efPCA (Section 2.2), and PFI (Section 2.3). We introduce the *shifted peaks* simulated dataset (based on an example from Tucker et al. (2013)) for method demonstration. The shifted peaks data contain 500 functions from two groups with 250 functions per group (Figure 1 (top left)). The functions

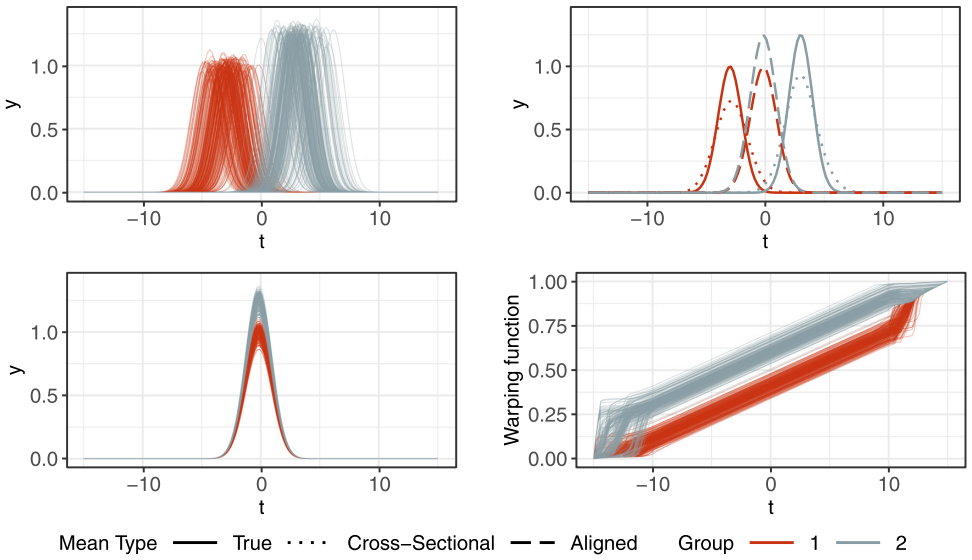

Figure 1: *Shifted peaks training data.* The simulated training data functions (top left), true, cross-sectional, and aligned functional means (top right), aligned functions (bottom left), and warping functions (bottom right).

are simulated as

$$y_{g,i}(t) = z_{g,i} e^{-(t-a_{g,i})^2/2},$$

where $t \in [-15, 15]$, $g = 1, 2$ indicates the group, $i = 1, 2, ..., 250$ identifies a function within a group, $z_{g,i} \overset{iid}{\sim} N(\mu_{z,g}, (0.05)^2)$, and $a_{g,i} \overset{iid}{\sim} N(\mu_{a,g}, (1.25)^2)$. We set $\mu_{z,1} = 1$, $\mu_{z,2} = 1.25$, $\mu_{a,1} = -3$, and $\mu_{a,2} = 3$, so the true functional mean of group 1 peaks earlier and with a smaller value of y than group 2 (solid lines in Figure 1 (top right)). The functions are generated with 150 equally spaced points per function. For prediction, the functions are randomly separated into training and testing sets with 400 and 100 functions, respectively.

2.1 Separation of Vertical and Horizontal Variability

As previously mentioned, there are two types of variability present in functional data: vertical and horizontal variability. Vertical variability is the variability in the height of the functions, and horizontal variability is the variability in the location of peaks and valleys of the functions. The shifted peaks data in Figure 1 (top left) exemplify vertical variability by the differences in peak intensity and horizontal variability by the difference in peak times. Figure 1 (top right) demonstrates an example of how ignoring horizontal variability can lead to inaccurate representations of the functional forms. The dotted lines represent the cross-sectional group means of the shifted peaks data (i.e., the mean of the values from all functions at one time point). Both cross-sectional means have different shapes than the true means, which are narrower and have higher peaks than the cross-sectional means.

Tucker et al. (2013) account for both variabilities in efPCA by using the elastic shape analysis (ESA) framework (Joshi et al., 2007; Srivastava et al., 2011; Tucker et al., 2013, 2019, 2020) to decompose observed functions into two new sets of functions: aligned and warping functions. *Aligned functions* match the peaks and valleys from the observed functions to capture

the vertical variability in the observed functions. *Warping functions* transform the observed functions to the aligned functions and, thus, capture the horizontal variability in the observed functions. Here, we provide an overview of the separation process. We refer the reader to Tucker et al. (2013) and Srivastava and Klassen (2016) for more details.

Start by letting f be a real-valued function with the domain $[0, 1]$; this domain can be easily generalized to any other compact subinterval of $\mathbb{R}^1$. We assume that all functions considered are observed on the same interval and, for concreteness, are absolutely continuous on $[0, 1]$. We let $\mathcal{F}$ denote the set of all such functions. In practice, the observed data are discrete, so this assumption is not a restriction. Also, let Γ be the set of orientation-preserving diffeomorphisms of the unit interval $[0, 1]$: $\Gamma = \{\gamma : [0, 1] \to [0, 1] | \ \gamma(0) = 0, \ \gamma(1) = 1, \gamma \text{ is a diffeomorphism}\}$. Elements of Γ play the role of warping functions. For any $f \in \mathcal{F}$ and $\gamma \in \Gamma$, the composition $f \circ \gamma$ denotes the time warping of f by γ (i.e., the aligned version of f).

Tucker et al. (2013) describe two metrics: one on the quotient space $\mathcal{F}/\Gamma$ for amplitude variability and the other on the group Γ for phase variability. Both metrics are proper distances. The *amplitude* or *y distance* for any two functions $f_1, \ f_2 \in \mathcal{F}$ is defined as

$$d_a(f_1, f_2) = \inf_{\gamma \in \Gamma} \|q_1 - (q_2 \circ \gamma)\sqrt{\dot{\gamma}}\|, \tag{2.1}$$

where $q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}$ is known as the *square-root velocity function (SRVF)* and $\dot{f} = df(t)/dt$. The SRVF transformation of f is used in the distance computation since if f is absolutely continuous, q is square-integrable, which allows $d_a(f_1, f_2)$ to be a proper distance. This is unlike other functional alignment processes (of the type $\inf_{\gamma \in \Gamma} \|f_1 - f_2 \circ \gamma\|$) that are not proper distances. For more details on the SRVF transformation, see Srivastava et al. (2011). The distance in Equation 2.1 is solved using the dynamic programming algorithm (Bertsekas, 1996).

The *phase* or *x distance* measures the distance between two warping functions $\gamma_1, \gamma_2 \in \Gamma$. Since Γ is an infinite-dimensional nonlinear manifold (i.e., not a standard Hilbert space), it is challenging to compute a distance on Γ. A transformation is applied to the warping functions to simplify the complicated geometry of Γ. Specifically, the SRVF of γ is computed: $\psi = \sqrt{\dot{\gamma}}$, where $\dot{\gamma} = d\gamma(t)/dt$. Note that $\dot{\gamma} > 0$. Let Ψ be the set of all such ψ's, which can be shown to be the Hilbert sphere (i.e., $\Psi = \mathbb{S}_\infty^+$, the positive orthant of the Hilbert sphere). For two warping functions $\gamma_1, \gamma_2 \in \Gamma$, the x-distance is computed as

$$d_p(\gamma_1, \gamma_2) = d_\psi(\psi_1, \psi_2) = \cos^{-1}\left(\int_0^1 \psi_1(t)\psi_2(t)dt\right),$$

which is the arc-length between the SRVFs of the warping functions on the Hilbert sphere. Note that ψ is invertible, which makes it possible to reconstruct γ from ψ as $\gamma(t) = \int_0^t \psi(s)^2 ds$ since $\gamma(0) = 0$.

For separating the phase-amplitude components of a set of functions $f_1, f_2, ..., f_n$, we first compute a Karcher mean of the given functions under the metric d_a:

$$\begin{aligned}
(\text{In } \mathcal{F} \text{ space}) : \mu_f &= \arg\min_{f \in \mathcal{F}} \sum_{i=1}^n d_a(f, f_i)^2 \quad \text{and} \\
(\text{In SRVF space}) : \mu_q &= \arg\min_{q \in \mathbb{L}^2} \sum_{i=1}^n \left(\inf_{\gamma_i \in \Gamma} \|q - (q_i, \gamma_i)\|^2\right),
\end{aligned}$$

where $(q_i, \gamma_i) = (q_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}$. These formulations are equivalent with $\mu_q = \text{sign}(\dot{\mu}_f)\sqrt{|\dot{\mu}_f|}$. The algorithm for computing the Karcher mean also results in the aligned functions $\left\{\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_n\right\}$ representing the amplitude variability, where $\tilde{f}_i = f_i \circ \gamma_i$, and the warping functions $\{\gamma_1, \gamma_2, ..., \gamma_n\}$ used in aligning the original functions and representing the phase variability.

Figure 1 (bottom row) shows the aligned and warping functions computed from the shifted peaks data. The dashed lines in Figure 1 (top right) are the cross-sectional group means from the aligned functions. With the horizontal variability removed, the group means of the aligned functions correctly capture the shape of the true means (ignoring peak timing).

2.2 Elastic Functional Principal Component Analysis

There are three efPCA methods that model functional variability: *vertical, horizontal, and joint fPCA (vfPCA, hfPCA, and jfPCA)*. vfPCA and hfPCA are proposed in Tucker et al. (2013) and, as their names suggest, provide separate evaluations of the horizontal and vertical variabilities. Lee (2017) proposed jfPCA (or combined fPCA), which jointly accounts for vertical and horizontal variability by concatenating the warping and aligned functions into a combined function, g^C, before computing principal components. Tucker et al. (2020) modify the methodology proposed by Lee (2017) by constructing the combined function g^C using the SRVF $\tilde{q}$ of the aligned function $\tilde{f}$, since $\tilde{q}$ is guaranteed to be an element of $\mathbb{L}^2$. We use the modified version by Tucker et al. (2020). Here, we provide a generalized version of the process for computing efPCs; see Lee (2017), Lee and Jung (2017), and Tucker et al. (2013, 2020) for further details.

Table 1: Functional principal component domains.

	Vertical fPCA	Horizontal fPCA	Joint fPCA
Representation	$\tilde{q}$	γ	$g^C = [\tilde{q} \quad Cv(t)]$
Variability	Amplitude	Phase	Amplitude + Phase
Metric	Fisher-Rao	Fisher-Rao	Fisher-Rao

For a set of functions $\{f_1, f_2, ..., f_n\}$, separate the functions into aligned $\left\{\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_n\right\}$ and warping $\{\gamma_1, \gamma_2, ..., \gamma_n\}$ functions using the method described in Section 2.1. Next, compute a sample covariance function on the functional representation shown in Table 1 based on the variability of interest. For joint fPCA, C is a constant such that $C > 0$, and $v(t)$ is a tangent space representation of $\psi(t)$, which is used for computational ease. See Lee (2017) for a data-driven approach for estimating C and Tucker et al. (2020) for details on the computation of $v(t)$. For a generalization, let $z_1, z_2, ..., z_n$ represent the set of functions capturing the specified type of variability (i.e., $\tilde{q}$, γ, or $g^C = [\tilde{q} \quad Cv(t)]$). The sample covariance function is computed as

$$K = \frac{1}{n-1}\sum_{i=1}^{n}\left(z_i - \hat{\mu}_z\right)\left(z_i - \hat{\mu}_z\right)^T,$$

where $\hat{\mu}_z$ is the sample mean function. Next, apply singular value decomposition (SVD) to the covariance matrix to obtain $K = U_K \Sigma_K V_K^T$ where U_K contains the directions of principal variability. The principal coefficients are computed as $\langle z_i, U_{K,j}\rangle$ for $j = 1, ..., J$.

The principal directions can be visualized in the original function space, $\mathcal{F}$, to interpret functional variability captured by fPCs. A common visualization is to plot the Karcher mean

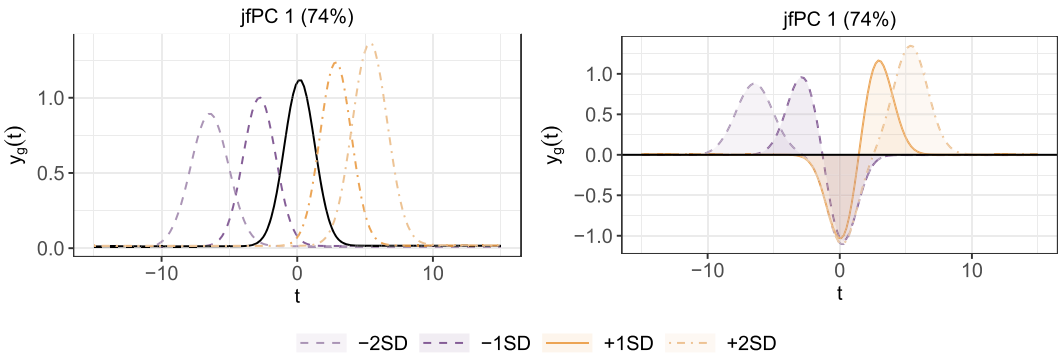

Figure 2: *Shifted peaks data first jfPC.* (Left) Principal direction plot for understanding the functional variability captured by jfPC 1 from the shifted peaks data. (Right) The corresponding principal differences plot for jfPC 1.

and the variation in the principal directions. In an abuse of notation, we represent a generalized form of this as

$$\mu_z \pm \sqrt{\tau \Sigma_{K,jj}} U_{K,j},$$

where $\tau \in \mathbb{N}^+$. By plotting a series of curves given various values of τ, the visualization provides a visual spectrum of the variability of shapes of functions captured by the principal component. We refer to these visualizations as *principal direction plots*. For an additional view to assist with interpretation of the principal components, we plot the difference between the principal directions and the Karcher mean. We refer to this second type of visualization as the *principal differences plot*.

Figure 2 shows the principal direction plot (left) and principal differences plot (right) for the first joint functional principal component (jfPC) computed on the shifted peaks data. In the principal direction plot, the solid black line represents the training data Karcher mean. The dashed/dashed-dotted lines represent the principal directions minus/plus 1 and 2 standard deviations. The visual indicates that the jfPC 1 captures a large amount of variability in peak time and intensity. In the principal differences plot, the solid black line represents a horizontal line with an intercept of 0. The dashed/dashed-dotted lines represent the principal directions minus/plus 1 and 2 standard deviations minus the Karcher mean. This plot helps highlight regions where the variability is important: between the times of -10 and 10 with a particular contrast between the regions of -10 to -2 and 2 to 10, approximately. We include shading with the minus/plus standard deviations in the principal differences plots in this paper to help distinguish these plots from the principal direction plots.

2.3 Permutation Feature Importance

Permutation feature importance (PFI) was initially developed as a tool for random forests by Breiman (2001) and generalized to any model type by Fisher et al. (2019). PFI quantifies the importance of a predictor variable by measuring how randomly permuting the variable changes the model performance. A variable is considered important if the model performance worsens considerably when the variable is permuted.

There is some variability in how PFI has been previously defined. In this paper, we define PFI as follows. Let $\mathcal{A}$ be a model, $\mathbf{X}$ be an $n \times p$ data matrix with n observations and p predictor

variables, and X_1, X_2,..., and X_p be the columns of $\mathbf{X}$. The data matrix, $\mathbf{X}$, may be training, testing, validation, or another dataset of interest. We let $\mathfrak{m}$ be a performance metric computed on $\mathbf{X}$ with model $\mathcal{A}$, where $\mathfrak{m}$ may be any metric of interest such that larger values indicate better performance (e.g., negative mean squared error for regression and accuracy for classification). Then for variable $j \in \{1, ..., p\}$ and repetition $r \in \{1, ..., R\}$: (1) Randomly permute X_j. Define the permuted variable as $\tilde{X}_{j,r}$. (2) Create a new dataset, $\tilde{\mathbf{X}}_{j,r}$, by replacing X_j in $\mathbf{X}$ with $\tilde{X}_{j,r}$. (3) Compute $\mathfrak{m}_{j,r}$ as the performance metric on $\tilde{\mathbf{X}}_{j,r}$ for model $\mathcal{A}$. (4) Compute the PFI for variable j as

$$\mathcal{I}_j = \frac{1}{R}\sum_{r=1}^{K} I_{j,r} = \mathfrak{m} - \frac{1}{R}\sum_{r=1}^{K} \mathfrak{m}_{j,r}, \quad \text{where} \quad \mathcal{I}_{j,r} = \mathfrak{m} - \mathfrak{m}_{j,r}.$$

The PFI value for j, $\mathcal{I}_j$, is interpreted as the average change in model performance when j is randomly permuted. For example, if $\mathfrak{m}$ is accuracy, PFI is interpreted as, "On average, the accuracy of model $\mathcal{A}$ decreases by $\mathcal{I}_j$ when feature j is randomly permuted". We note that it may also be valuable to consider the variability of $\mathcal{I}_{j,r}$ across repetitions. A large variation in $\mathcal{I}_{j,r} = \mathfrak{m} - \mathfrak{m}_{j,r}$ indicates that the change in model performance is dependent on the random permutation of variable j, and more repetitions may be needed to obtain a good estimate of the average PFI.

3 Methods

In this section, we present the VEESA pipeline methodology. Supervised machine learning models are typically fit to a set of training data and then used to obtain predictions on a set of held out data (test/validation data) to tune and/or obtain a more accurate estimate of model predictive performance on new data. Section 3.1 includes the VEESA pipeline steps as applied to training data, and Section 3.2 develops the pipeline for test and validation data.

3.1 VEESA Pipeline (Training Data)

Here, we describe the VEESA pipeline steps for training data. We demonstrate the steps on the shifted peaks data. See the supplemental material for a comparison to the cross-sectional modeling approach applied to the shifted peaks data that highlights advantages of the VEESA pipeline.

Step 1: Smoothing Since the SRVF transformation in the separation of vertical and horizontal variability requires the computation of a derivative, it is recommended that smoothing is applied to observed functions to compute an accurate estimate of the derivative. Thus, if the observed functions are not smooth, begin by applying a smoothing technique to the functions in the training data such as a box-filter (Tucker et al., 2013) or splines (Ullah and Finch, 2013). The choice of the smoothing method is left to the analyst, but we suggest treating the amount of smoothing applied to the functions (e.g., the number of times to run a box-filter) as a tuning parameter in the VEESA pipeline. For example, cross-validation (CV) may be conducted to identify the amount of smoothing that results in the best predictive performance. *Shifted Peaks Data:* We do not apply smoothing since the functions are already smooth.

Step 2: Separation of Functional Variability Apply the ESA alignment process (Section 2.1) to the training data. *Shifted Peaks Data:* Figure 1 shows the aligned and warping functions.

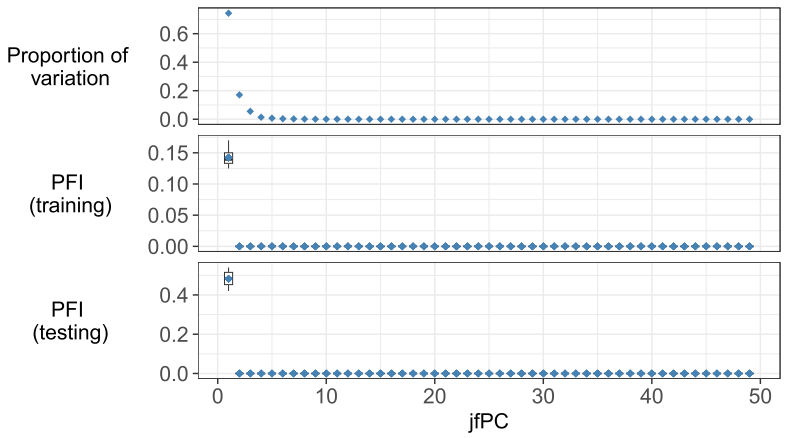

Figure 3: *Shifted peaks data jfPC metrics.* (Top) Proportion of variation explained by the jfPCs. (Middle and bottom) Blue diamonds represent PFI values from the training and testing data, respectively. Boxplots represent the variability across repetitions.

Step 3: Elastic Functional Principal Component Analysis Apply one of the efPCA methods described in Section 2.2 to the training data. The selection of a method will depend on the relationship between the function shapes and the response variable. If predictive information is only in the vertical/horizontal direction, then vfPCA/hfPCA is applicable. If predictive information is contained in both types of variability, then jfPCA should be used. If it is unknown which variability type contains predictive information, all methods could be applied to determine which leads to the best predictive performance. *Shifted Peaks Data:* We apply jfPCA since we know the data contain both vertical and horizontal variability predictive information. Figure 3 (top) shows the proportion of variance explained by jfPCs.

Step 4: Model Training Train a model using the efPCs obtained in Step 3 as predictor variables. Since the amount of variability explained by a principal component does not necessarily indicate the predictive ability of a principal component, we recommend initially training a model with all (or many) efPCs. The PFI results may inform feature selection. *Shifted Peaks Data:* We fit a random forest using the *randomForest* R package (Liaw and Wiener, 2002) with group as the response variable and the jfPCs obtained in Step 3 as the predictor variables. The model returns a training data accuracy of 1.

Step 5: Post-Hoc Explanations Any applicable explainability method may be applied at this time. For simplicity, we only consider PFI for explainability in this paper, which is applied as described in Section 2.3 to quantify importance of the efPCs. We consider PFI since it is model agnostic, which allows for comparison of variable importance across model types. Further, PFI is relatively simple to understand, which makes it possible to explain to SMEs who may have little familiarity with machine learning models. We note that PFI is known to be biased when there is correlation between predictor variables (Hooker et al., 2021), but due to the orthogonal nature of the efPCs, this concern is alleviated. *Shifted Peaks Data:* We compute PFI values on the training data using a metric of accuracy and 10 repetitions. The results are shown in Figure 3 (middle). The diamonds indicate the PFI values, and the boxplots depict variability across repetitions. jfPC 1 clearly has the highest importance, which we interpret as on average,

the random forest accuracy (on the training data) decreases by 0.14 when jfPC 1 is randomly permuted. All other jfPCs have PFI values of 0.

Step 6: efPC Visualizations Use visualizations such as principal direction and principal differences plots as discussed in Section 2.2 to understand the variability captured by important jfPCs. These plots depict the functional variability captured by a principal component on the original data space. This representation is advantageous to SMEs who are familiar with functions on this space and may have an intuition as to the functional variability that would be useful for prediction. *Shifted Peaks Data:* Recall that Figure 2 shows that jfPC 1 captures variability between earlier/shorter peaks and later/higher peaks. Based on our understanding of the data generating mechanism, this is a reasonable variability for distinguishing between groups.

3.2 VEESA Pipeline (Testing/Validation Data)

Here, we describe the steps to apply the VEESA pipeline to testing, validation, or other non-training data. The general process is the same as the training data steps but with some implementation adjustments. For conciseness, we only use the term of test data to describe the method going forward, but this approach is applicable to any dataset considered after a model is fit to training data.

Step 1: Smoothing Apply the same smoothing process used with the training data.

Step 2: Separation of Functional Variability The separation of variability with the test data is implemented by aligning the test data SRVFs to the Karcher Mean of the training data SRVFs. That is, for a set of training data functions $\{f_1, f_2, ..., f_n\}$ and a set of test data functions $\{f_{n+1}, f_{n+2}, ...f_{n+m}\}$, compute SRVFs of the test data functions $\{q_{n+1}, q_{n+2}, ..., q_{n+m}\}$. Let $\hat{\mu}_q$ represent the sample Karcher mean of the training data in SRVF space. Compute warping functions $\{\gamma_{n+1}, \gamma_{n+2}, ..., \gamma_{m+l}\}$ that align the test data SRVFs to $\hat{\mu}_q$ as

$$\gamma_{n+l} = \underset{\gamma \in \Gamma}{\arg\min} \|\hat{\mu}_q - (q_{n+l} \circ \gamma)\sqrt{\dot{\gamma}}\|,$$

where $l = 1, ..., m$. The warping functions computed on the test data, γ_{n+l}, are then used to compute the aligned test data functions (in SRVF space):

$$(q_{n+l} \circ \gamma)\sqrt{\dot{\gamma}_{n+l}}.$$

Step 3: Elastic Functional Principal Component Analysis As described in Section 2.2, the specifics of computing efPCs on the test data will vary based on the efPCA method applied to the training data, but the concept is the same. Let $z_{n+1}, z_{n+2}, ..., z_{n+m}$ represent the set of functions computed in the alignment of the test data associated with the desired efPCA method. The principal coefficients are computed as $\langle z_{n+l}, U_{K,j} \rangle$ for $l = 1, ..., m$, where $U_{K,j}$ contains the directions of principal variability computed from applying efPCA to the training data. This multiplication is done on the SRVF space but can be converted back to the original space.

Step 4: Model Predictions Use the previously trained model to obtain predictions for the test data efPCs.

Step 5: Post-Hoc Explanations Compute explanations such as PFI on the test data. The efPCs that are important may vary between training and testing data. If there is a difference, it may suggest that test data contains observations that are out of distribution from the training data.

Step 6: efPC Visualizations Create visualizations of the principal directions for any important efPCs not previously considered.

Shifted Peaks Data: We apply the above steps to the shifted peaks test data using the random forest model from Section 3.1. The test data accuracy is 1, and the resulting PFI values are depicted in Figure 3 (bottom). Again, jfPC 1 appears as the only variable with importance. Note that the magnitude of the importance is larger for the test data than the training data (i.e., PFI values of 0.48 and 0.14 for jfPC 1 on the test and training data, respectively). It would be interesting to explore this further in future work to understand the reasons for the magnitude difference.

4 Examples

In this section, we apply the VEESA pipeline to two examples from high consequence application spaces. Section 4.1 applies a neural network to identify an explosive from a set of materials using hyperspectral computed tomography (H-CT) scans. Section 4.2 applies a random forest to predict the source inkjet printer of a document using Raman spectroscopy.

4.1 Hyperspectral Computed Tomography Data Material Classification

H-CT scans of materials produce a signature across a set of frequencies unique to an observation. There is interest in using H-CT scans to identify explosives from a set of materials for applications such as airport security (Jimenez et al., 2017; Gallegos et al., 2018). In this example, we consider a set of 1,980,409 H-CT scans simulated by experts at Sandia National Laboratories that contain five materials: water (H_2O), hydrogen peroxide (H_2O_2) solutions diluted by water with 100%, 50%, and 10% H_2O_2, and an explosive (Gallegos et al., 2019). The percentages of observations per material are 34%, 17%, 8%, 7%, and 34%, respectively. The signatures are separated into training and testing sets by randomly sampling 80% of scans from each material for the training data. The top row of Figure 4 shows subsets of 1,000 randomly selected signatures per material from the training data. The scans record observations at 128 frequencies, which are visualized as normalized frequencies between 0 and 1.

We apply the VEESA pipeline with a neural network to predict the material of an H-CT scan. Figure 4 shows the functions after alignment and warping (middle and bottom rows, respectively) when smoothing is implemented with 20 box-filter runs. Note that the alignment process is applied to the functions from all materials jointly (as opposed to separately by material). As an exploratory analysis step, we compute the training data cross-sectional means of the aligned signatures and the Karcher means of the warping functions for each material (Figure 5). There are minimal differences between the warping function means, which indicates there is little horizontal variability, on average. However, the aligned means show clear vertical variability between the materials with a pattern change just before the normalized frequency of 0.25. For example, water and the explosive have similar intensities before 0.25 but noticeably different intensities after 0.25. This agrees with SME knowledge of the differences between material signatures: In

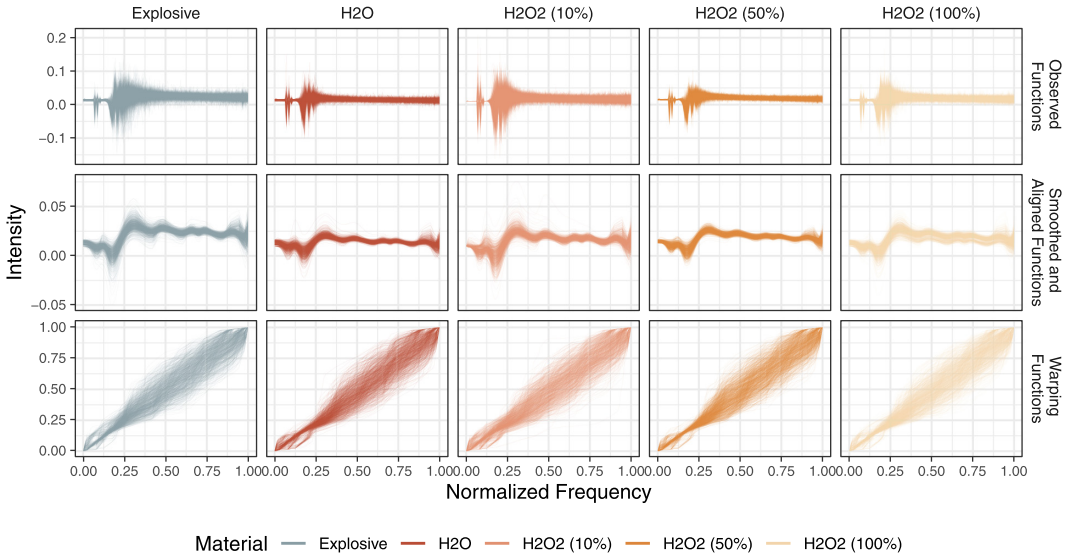

Figure 4: *H-CT data material signatures.* Observed (top row), smoothed and aligned (middle row), and warping functions (bottom row) of a subset of 1,000 H-CT signatures for each material.

general, the definition of the higher order frequencies (> 0.25) are important in the distinguishment between water and explosive material. Additionally, the response from 0.15 to 0.25 have small differences between water and hydrogen peroxide that can help in distinguishment and get more pronounced as the concentration increases.

We fit neural networks using jfPCs, vfPCs, and hfPCs as inputs with smoothing implemented with 20, 20, and 25 box-filter runs, respectively. For comparison, we also fit neural networks using the cross-sectional approach before smoothing, after smoothing (1 box-filter run), and after alignment (1 box-filter run). The number of box-filter runs included here are chosen based on the best test data predictive performance. See the supplement for details. All neural networks are trained with all 128 PCs/times as the inputs using the Python package *scikit-learn* (Pedregosa et al., 2011) and all default values (i.e., one layer with 100 neurons and a ReLU activation function).

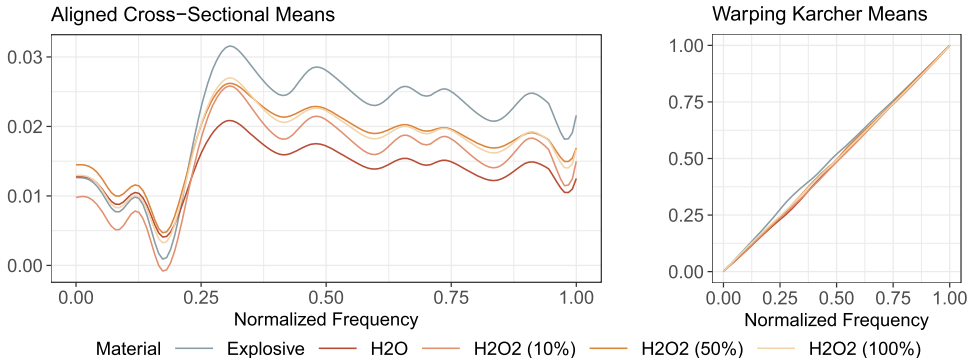

Figure 5: *H-CT training data material means.* For each material, cross-sectional functional means of the aligned functions (left) and Karcher means of the warping functions (right).

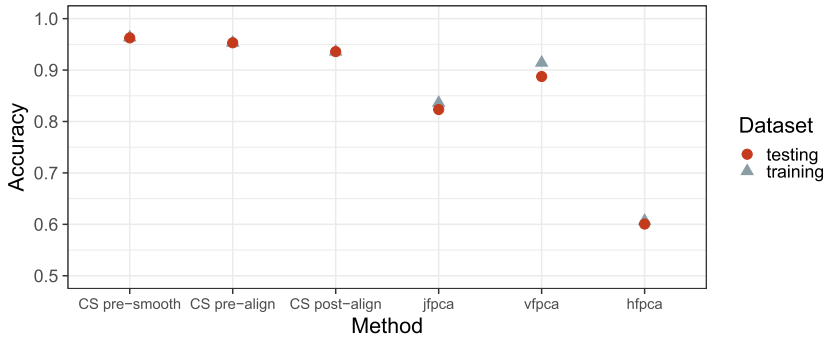

Figure 6: *H-CT data accuracies.* Best model accuracies from neural networks applied using the VEESA pipeline and the cross-sectional (CS) approach.

Figure 6 shows the best accuracies from each of the methods. The model fit with vfPCs has a test data accuracy of 0.89 and performs better than the models fit with jfPCs and hfPCs. This is unsurprising based on the results in Figure 5. The vfPCA model performs worse than the cross-sectional models in this initial implementation, but if we applied PFI to the cross-sectional methods, the feature importance results would be biased. As such, there may be a trade-off between higher accuracy and more trustworthy explanations. However, in this case, we have not tuned the neural network. Future work could investigate whether tuning affects these results and how the methods compare on validation data that has additional noise (as may be encountered in practice). The results from such analyses would help to determine if a decision between higher accuracy and more trustworthy explanations needs to be made.

We proceed with the VEESA pipeline using the vfPCA model to understand what characteristics of the data this model uses for prediction. PFI is computed on the test data using 5 repetitions and a performance metric of accuracy. Figure 7 shows the proportion of variation and PFI values associated with the vfPCs. The variability across PFI repetitions is not depicted since it is much smaller than the variability across PCs; see the supplement for details. The proportion of variation has a major drop after the first PC and drops close to zero after a small number of PCs. There is a much different trend with PFI, which provides a clear example where the predictive importance of a principal component is not directly related to the amount of variability explained. PC 128 has an extremely high PFI value and a handful of earlier PCs have elevated PFI values.

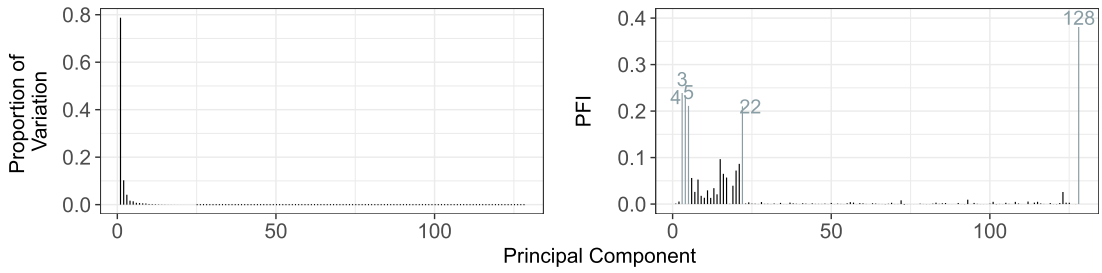

Figure 7: *H-CT data vfPC metrics.* Proportion of variation and PFI values associated with vfPCs. vfPCs with the five highest PFI values are labeled and colored.

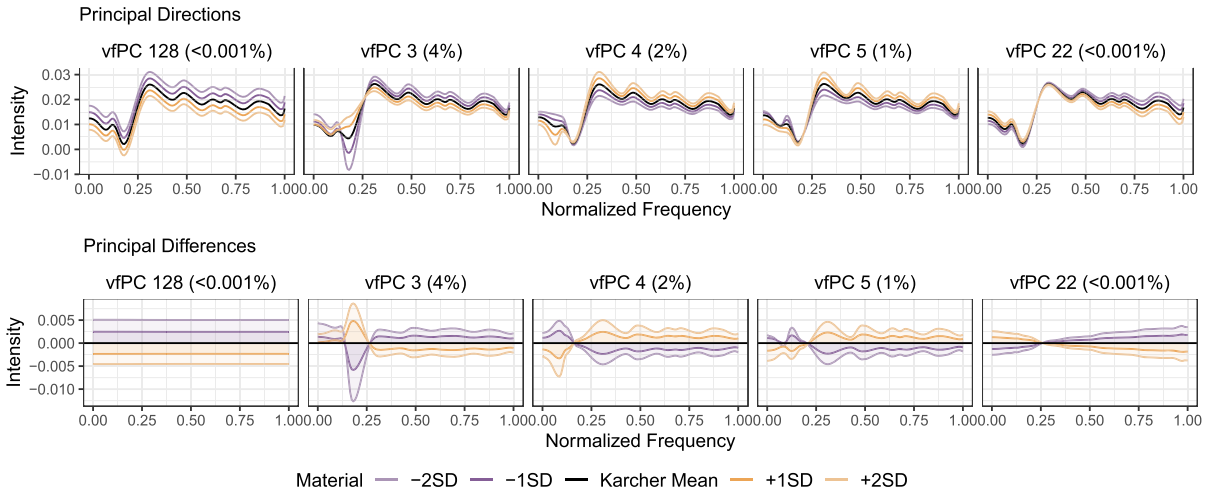

Figure 8: *H-CT data important vfPCs.* Principal directions and differences from the five vfPCs with the highest PFI values from the H-CT data example.

For space reasons, we only consider the principal direction and principal differences plots associated with the 5 vfPCs with the highest feature importance values in Figure 8. More vfPCs are included in the supplement. The PC with the highest PFI (vfPC 128) captures a consistent vertical variability across all frequencies. The other four vfPCs capture more nuanced aspects of the variability, but all capture (mostly) consistent vertical variability after a frequency of 0.25 with different types of vertical variability captured before 0.25. This aligns with the SME knowledge of distinguishing variability occurring after 0.25 and the patterns seen in Figure 5.

4.2 Inkjet Printer Identification with Raman Spectroscopy

Forensic investigators are interested in techniques that provide evidence connecting printed documents to the source inkjet printer to assist with investigations into illicit activities such as counterfeit currency. Buzzini et al. (2021) present one approach where Raman spectroscopy is used to extract signatures from documents generated by inkjet printers. The authors then use different variants of linear discriminant analysis (LDA) to predict the source printer given a sample of Raman spectra. Going forward, we will refer to a sample of Raman spectra as a signature for simplicity. Figure 9 shows signatures from the printers considered in Buzzini et al. (2021), which we refer to as the *inkjet dataset*.

The inkjet dataset contains signatures collected from eleven documents belonging to the Counterfeit Forensic Section of the Criminal Investigative Division of the US Secret Service. Each document is printed from a different device, but some devices have the same manufacturer; printers 7 and 8 are also the same model. For each document, 7 replicates were collected from the three main colored dot components (cyan, magenta, and yellow). Each replicate contains observations at 231 spectra between 1800 and 250 cm^{-1}. Buzzini et al. (2021) converted the replicates to have 1129 observations per signature with a path of approximately 1.4 cm^{-1}. See Buzzini et al. (2021) for more details.

While forensic scientists have some notions of what characteristics of the signatures may be useful for distinguishing between printers (e.g., pigment-based inks tend to have more peaks in cyan (printer 4) compared to dye-based inks (printer 3)), the exact ties between signature char-

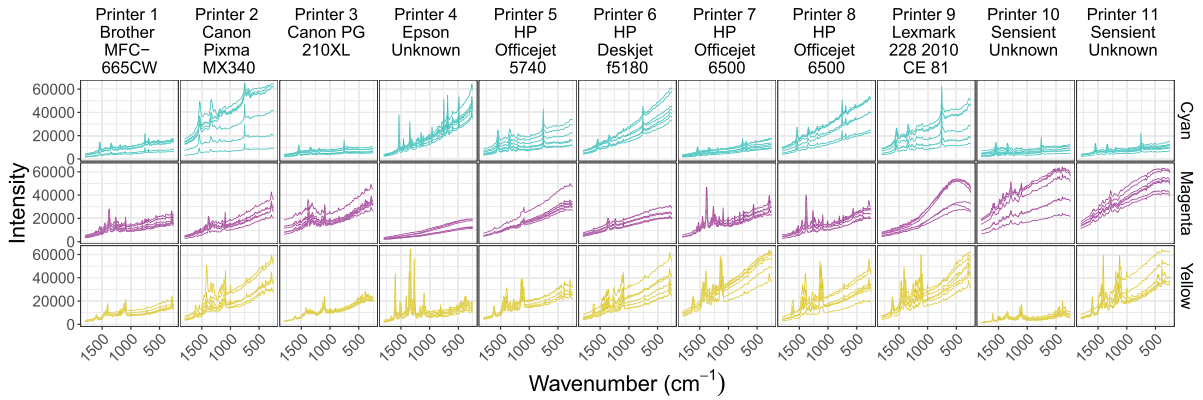

Figure 9: *Inkjet data signatures.* Raman spectra signatures from 11 inkjet printers for the colors of cyan, magenta, and yellow with labels for printer manufacturer and model.

acteristics and their generating mechanism is not well understood. As such, this is an exemplar where data-driven black-box models may be able to pick up on distinguishing characteristics that may be overlooked by the current scientific understanding. However, there are also likely characteristics in the data that are not useful for distinguishing between printers such as noise due to human measurement variability. As such, the VEESA pipeline could provide insight into the types of variability used for prediction to determine if the model is drawing on meaningful characteristics or noise.

We implement the VEESA pipeline using jfPCA since Figure 9 shows variability in both the vertical and horizontal directions in the signatures, and we do not have prior knowledge of which type of variability contains information useful for prediction. We implement smoothing using a box-filter, and we select a random forest as the predictive model as an example of commonly used statistical learning model. The models are fit using the *randomForest* R package (Liaw and Wiener, 2002) with all tuning parameters set to the default value besides for the number of trees. We follow the predictive assessment in Buzzini et al. (2021), so we can compare models. Specifically, we build models separately for each color, and we implement a 3-fold CV procedure with 10 replications to compare performance across different values of box-filter runs, number of PCs input to the model, and number of random forest trees. The details of the CV procedure are included in the supplemental material.

Table 2 contains the VEESA pipeline best and worst CV accuracies and corresponding tuning parameters. Note that the worst VEESA pipeline accuracies have no smoothing. The last

Table 2: Inkjet data cross-validation average accuracies.

Scenario	Color	Box-Filter	PCs	Trees	VEESA	Buzzini
Best	Cyan	20	40	1000	0.9260	0.91
Best	Magenta	5	40	250	0.8896	0.92
Best	Yellow	20	20	1000	0.9286	0.92
Worst	Cyan	0	100	50	0.3532	-
Worst	Magenta	0	90	50	0.5416	-
Worst	Yellow	0	100	50	0.5182	-

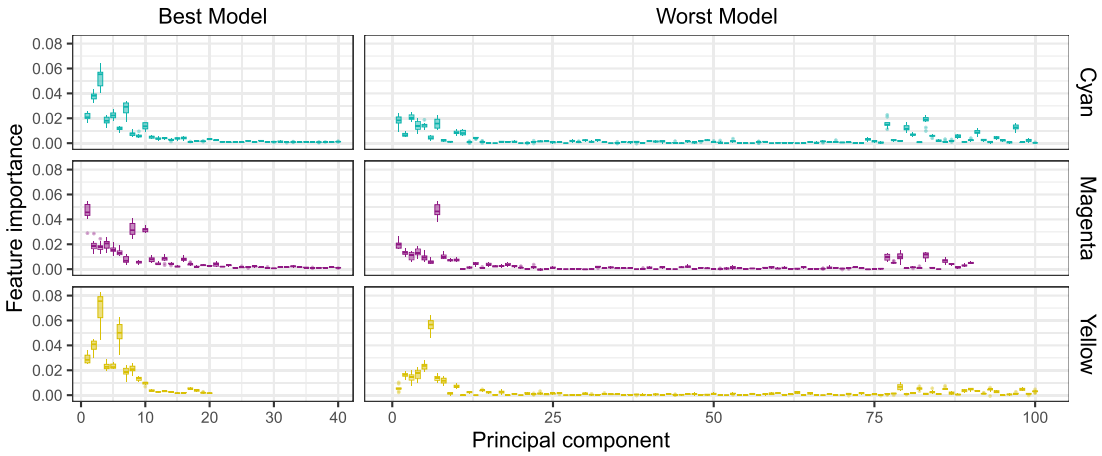

Figure 10: *Inkjet data feature importances.* Boxplots of PFI values across 10 repetitions from the best and worst performing models for cyan, magenta, and yellow inkjet signatures.

column contains the highest CV accuracies from Buzzini et al. (2021). For cyan and yellow, the VEESA pipeline achieves approximately the same or better CV accuracies. For magenta, the VEESA pipeline's best CV accuracy is close but lower. Additional investigations (included in the supplement) show that the low VEESA pipeline accuracy for magenta results from misclassifications of printers 7 and 8, which share the same manufacturer and model.

For each of the best and worst tuning parameter scenarios, we apply the VEESA pipeline using a random forest trained on the full inkjet dataset (within a color). We use the full data to mimic the joining of all available data when building a model for predictions on new data. We consider the best and worst cases to study how the feature importance varies between good and poor performing models. We use jfPCA and compute PFI for the jfPCs using 10 repetitions.

The PFI results are included in Figure 10. The best performing models place high importance on the earlier PCs, and the worst performing models distribute importance between early PCs (less than 15) and later PCs (greater than 75). In the supplement, we explore how the removal of PCs 15 to 75 in the worst performing scenarios affects the model accuracies. Figure 11 shows principal directions and differences from the best performing models for cyan signatures. Again, for space reasons, we only consider the five PCs with the highest feature importance values and place the magenta and yellow results in the supplement.

While there are nuances in the variability captured, jfPCs 1, 2, and 3 generally capture increasing vertical variability as the wavenumber approaches 500 while also capturing small amounts of horizontal variability. jfPC 7 captures approximately the same amount of vertical variability across wavenumbers, and jfPC 5 captures small amounts of vertical variability around the three main peaks. It seems reasonable that the model would focus on these modes of variability based on Figure 9 where a key distinguishing factor between printers in cyan signatures is the difference in slopes. Another distinguishing factor is the height of the main three peaks and small amounts of variability in the wavenumbers where the main peaks occur, which aligns with the SME knowledge about how peak types differ between dye and pigment based inks.

Overall, we achieve similar predictive performance as Buzzini et al. (2021) while also gaining insight into how random forests make use of patterns in the data. We achieve this predictive per-

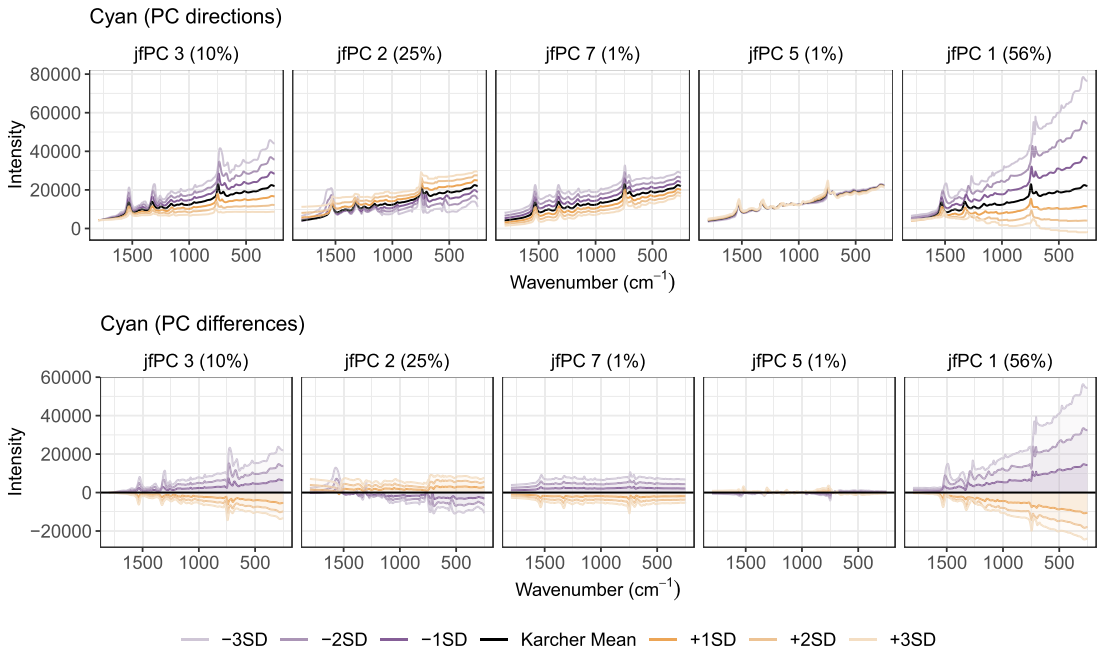

Figure 11: *Inkjet data important jfPCs.* jfPCs with the largest PFI values from the best (top) and worst (bottom) models for predicting cyan inkjet signatures. jfPCs ordered from left to right based on highest to lowest feature importance.

formance without as much preprocessing of the data as implemented in the analysis of Buzzini et al. (2021). We point this out, because this also highlights that the VEESA pipeline could be beneficial for analyses that do not have access to SMEs or scenarios where it is not understood how to best preprocess the data based on prior experiences. However, we also note that the principal components identified as important in this analysis capture many different characteristics of the functional variability in one principal component. This makes it difficult to separate the exact characteristic of variability being used by the model for distinguishment. Future work could try to remove certain characteristics believed to be noise through baseline adjustments or normalization as done in Buzzini et al. (2021). This may lead to principal components with less confounding of the sources of variability captured in applications with complex data such as this.

5 Discussion

We propose the VEESA pipeline to help fill the gap of methodology for explainable supervised machine learning with functional data inputs. With the VEESA pipeline, we mesh statistical methods (efPCA) with machine learning techniques to take advantage of strengths of both communities of practice. efPCA provides the ability to capture the data dependence structure that can be combined with the data-driven predictive ability of machine learning. By explicitly modeling the functional dependence in the inputs, we ensure the explanations produced by the pipeline do not obscure or misrepresent functional relationships and can be visualized in the original data space that is familiar to SMEs. Further, since efPCs are orthogonal, feature importance is computed without concern of bias due to correlation.

The VEESA pipeline offers one approach to explainable machine learning with functional data, but there are numerous directions for improvement and future research. We first note that the VEESA pipeline is structured such that other methods may be substituted for the warping, dimension reduction, and explanation methods. We focus on ESA dimension reduction since we have found accounting for horizontal variability to be important in practice. However, other analysis scenarios may find that different approaches to capturing the functional dependence may be warranted. With the explanations, as previously mentioned, the application of additional forms and approaches to explanations (e.g., global versus local and model-specific versus model-agnostic) would provide more perspectives. Future research could also explore how different types of explanations compare when used with the VEESA pipeline.

As presented in this paper, there are several assumptions underlying the VEESA pipeline. (1) We assume the efPCs have meaningful contextual interpretations. If not, this could potentially be resolved using a different transformation of the data (e.g., functional partial least squares (Febrero-Bande et al., 2017)). (2) The only inputs are efPCs from one set of functions. If there are additional variables, the guarantee of uncorrelated input variables may no longer hold. For example, a clear next step in the inkjet analysis is to train a model including efPCs of all three colors, but the PFI may be biased due to correlation. There is some work on computing feature importance that accounts for correlation between inputs (Strobl et al., 2008; Hooker et al., 2021), but this is an active research area. (3) The approach for implementing efPCA described in this paper assumes the functions have the same number of peaks. If this is not the case, the alignment process must choose between peaks. Future work could use the Bayesian multimodal alignment proposed for this scenario under the ESA framework (Tucker et al., 2021).

Another direction for future research would be to consider how best to conduct feature selection under this framework. Feature selection could lead to improved predictive ability and more interpretability (i.e., interpretation is easier with less PCs to consider). It is possible that a combination of knowledge of the proportion of variation captured by the principal components and the feature importance may suggest routes for feature selection. However, the recommended approach warrants further investigation.

Additionally, the VEESA pipeline can be computationally intensive with larger datasets. For example, the alignment process with the H-CT training data using took over 30 hours using Python code run on a computer with 44 cores and 252 gigabytes of memory. (Exact clock times for each step in the H-CT analysis are provided in the supplemental material.) Future research could consider methods for improved computational efficiency.

We end by discussing the important element of the intended audience for explanations. The VEESA pipeline is aimed at model developers and SMEs. However, if a model developed using the VEESA pipeline is deployed, additional steps should be taken to determine the form of explanations aimed at users or decision makers who do not possess the same technical knowledge. The way a model would be used in practice could take different forms. For example, in the inkjet analysis, a model developer may apply the model to a new prediction that could be used in a court case, but the developer may need to be creative to find a way to distill their understanding to provide explanations to lawyers, judges, and juries. In the H-CT material classification, if a model developed with the VEESA pipeline is used by airport security to scan luggage for explosives, simply providing PFI values will not be sufficient to help the security employees. Instead, an interactive tool that allows the employees to explore understandable explanations for individual predictions would likely be more valuable. In general, there is much room for the advancement of explanations intended for non-technical decision makers.

Supplementary Material

The supplementary materials include additional analyses on the shifted peaks, H-CT, and inkjet printer data, implementation details, R and Python code, and the shifted peaks and inkjet datasets. Due to proprietary reasons, the H-CT dataset is not able to be shared.

Acknowledgement

The authors thank Danica Ommen for pointing us to the inkjet application and Patrick Buzzini for providing access to the inkjet data and insights into the scientific understanding of the trends in the data. Additionally, we thank Philip Kegelmeyer for a careful reading of the manuscript and insightful feedback.

Funding

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Adadi A, Berrada M (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6: 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Bertsekas DP (1996). Dynamic programming and optimal control. *Journal of the Operational Research Society*, 47(6): 833–834. https://doi.org/10.2307/3010291

Breiman L (2001). Random forests. *Machine Learning*, 45(1): 5–32. https://doi.org/10.1023/A:1010933404324

Breiman L, Friedman J, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Butts-Wilmsmeyer CJ, Rapp S, Guthrie B (2020). The technological advancements that enabled the age of big data in the environmental sciences: A history and future directions. *Current Opinion in Environmental Science and Health*, 18. 63–69. Environmental Chemistry: Innovative Approaches and Instrumentation in Environmental Chemistry.

Buzzini P, Curran J, Polston C (2021). Comparison between visual assessments and different variants of linear discriminant analysis to the classification of Raman patterns of inkjet printer inks. *Forensic Chemistry*, 24:100336. https://doi.org/10.1016/j.forc.2021.100336

Chen C, Li O, Tao C, Barnett AJ, Su J, Rudin C (2019). This looks like that: Deep learning for interpretable image recognition. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA.

Danne T, Nimri R, Battelino T, Bergenstal RM, Close KL, DeVries JH, et al. (2017). International consensus on use of continuous glucose monitoring. *Diabetes Care*, 40(12): 1631–1640. https://doi.org/10.2337/dc17-1600

Febrero-Bande M, Galeano P, González-Manteiga W (2017). Functional principal component regression and functional partial least-squares regression: An overview and a comparative study. *International Statistical Review*, 85(1): 61–83. https://doi.org/10.1111/insr.12116

Fisher A, Rudin C, Dominici F (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177): 1–81.

Friedman JH (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5). https://doi.org/10.1214/aos/1013203451

Gallegos IO, Dalton GM, Stohn AM, Koundinyan SP, Thompson KR, Jimenez ES (2019). High-fidelity calibration and characterization of a spectral computed tomography system. In: *Hard X-Ray, Gamma-Ray, and Neutron Detector Physics XXI*, volume 11114, 223–236. SPIE.

Gallegos IO, Koundinyan S, Suknot AN, Jimenez ES, Thompson KR, Goodner RN (2018). Unsupervised learning methods to perform material identification tasks on spectral computed tomography data. In: *Radiation Detectors in Medicine, Industry, and National Security XIX*, volume 10763, 91–104. SPIE.

Goldstein A, Kapelner A, Bleich J, Pitkin E (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1): 44–65. https://doi.org/10.1080/10618600.2014.907095

Goode K, Ries D, McClernon K (2024). Characterizing climate pathways using feature importance on echo state networks. *Statistical Analysis and Data Mining: An ASA Data Science Journal*. 17(4):e11706.

Goode K, Ries D, Zollweg J (2020). Explaining neural network predictions for functional data using principal component analysis and feature importance. In: *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Fall Symposium Series 2020: Artificial Intelligence in Government and Public Sector*, Geib C, Petrick R (Eds.).

Goode K, Tucker JD (2025). veesa: VEESA Pipeline for Explainable Machine Learning with Functional Data. R package version 0.1.7.

Ha W, Singh C, Lanusse F, Upadhyayula S, Yu B (2021). Adaptive wavelet distillation from neural networks through interpretations. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21*, Ranzato M et al. (Eds.), Curran Associates Inc., Red, Hook, NY, USA.

Hooker G, Mentch L, Zhou S (2021). Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31: 1–16. https://doi.org/10.1007/s11222-021-10057-z

Jimenez ES, Thompson KR, Stohn A, Goodner RN (2017). Leveraging multi-channel x-ray detector technology to improve quality metrics for industrial and security applications. In: *Radiation Detectors in Medicine, Industry, and National Security XVIII*, Grim F, Furenlid L, Barber H B (Eds.), volume 10393, 137–147. SPIE.

Joshi SH, Klassen E, Srivastava A, Jermyn I (2007). A novel representation for Riemannian analysis of elastic curves in Rn. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Lucey S, Chen T (Eds.), 1–7.

Lee S (2017). Integrative analysis of variation structure in high-dimensional multi-block data, Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA. Supervised by Sungkyu Jung.

Lee S, Jung S (2017). Combined analysis of amplitude and phase variations in functional data. arXiv preprint: https://arxiv.org/abs/1603.01775.

Li H, Xiao G, Xia T, Tang YY, Li L (2014). Hyperspectral image classification us-

ing functional data analysis. *IEEE Transactions on Cybernetics*, 44(9): 1544–1555. https://doi.org/10.1109/TCYB.2013.2289331

Liaw A, Wiener M (2002). Classification and regression by randomforest. *R News*, 2(3): 18–22.

Martin-Barragan B, Lillo R, Romo J (2014). Interpretable support vector machines for functional data. *European Journal of Operational Research*, 232(1): 146–155. https://doi.org/10.1016/j.ejor.2012.08.017

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Ramsay J, Silverman B (2005). *Functional Data Analysis*. Number 0172–7397 in *Springer Series in Statistics*. Springer - Verlag New York, Verlag, New York, 2 edition.

Ries D, Gabriel Huerta J (2023). Predicting fatigue from heart rate signatures using functional logistic regression. *Stat*, 12(1): e595. https://doi.org/10.1002/sta4.595

Rudin C, Chen C, Chen Z, Huang H, Semenova L, Zhong C (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16(none): 1–85. https://doi.org/10.1214/21-SS133

Sankaran K (2024). Data science principles for interpretable and explainable AI. *Journal of Data Science*, 1–27. https://doi.org/10.6339/24-JDS1150

Srivastava A, Klassen E, Joshi SH, Jermyn IH (2011). Shape analysis of elastic curves in Euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7): 1415–1428. https://doi.org/10.1109/TPAMI.2010.184

Srivastava A, Klassen EP (2016). *Functional Shape and Data Analysis*. Springer Nature, New York.

Strobl C, Boulesteix AL, Kneib T, Augustin T, Zeileis A (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1): 307. https://doi.org/10.1186/1471-2105-9-307

Thind B, Multani K, Cao J (2023). Deep learning with functional inputs. *Journal of Computational and Graphical Statistics*, 32(1): 171–180. https://doi.org/10.1080/10618600.2022.2097914

Tian TS (2010). Functional data analysis in brain imaging studies. *Frontiers in Psychology*, 1:35.

Tucker JD (2025a). fdasrsf. Python package version 2.6.3.

Tucker JD (2025b). fdasrvf: Elastic Functional Data Analysis. R package version 2.4.0.

Tucker JD, Lewis JR, King C, Kurtek S (2020). A geometric approach for computing tolerance bounds for elastic functional data. *Journal of Applied Statistics*, 47(3): 481–505. https://doi.org/10.1080/02664763.2019.1645818

Tucker JD, Lewis JR, Srivastava A (2019). Elastic functional principal component regression. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12(2): 101–115.

Tucker JD, Shand L, Chowdhary K (2021). Multimodal Bayesian registration of noisy functions using Hamiltonian Monte Carlo. *Computational Statistics & Data Analysis*, 163:107298.

Tucker JD, Wu W, Srivastava A (2013). Generative models for functional data using phase and amplitude separation. *Computational Statistics & Data Analysis*, 61: 50–66. https://doi.org/10.1016/j.csda.2012.12.001

Ullah S, Finch CF (2013). Applications of functional data analysis: A systematic review. *BMC Medical Research Methodology*, 13(1): 43. https://doi.org/10.1186/1471-2288-13-43