Received: 20 October 2021 | Revised: 6 June 2022 | Accepted: 6 June 2022

DOI: 10.1002/asmb.2701

SPECIAL ISSUE PAPER

WILEY

Self-supervised cross validation using data generation structure

Ron S. Kenett[1] | **Chris Gotwalt**[2] | **Laura Freeman**[3] | **Xinwei Deng**[3]

[1]The KPA Group, Raanna and the Samuel Neaman Institute, Technion, Haifa, Israel

[2]JMP Division, SAS, Research Triangle, Cary, North Carolina, USA

[3]Department of Statistics, Virginia Tech, Blacksburg, Virginia, USA

Correspondence
Ron S. Kenett, The KPA Group, Raanna and the Samuel Neaman Institute, Technion, Haifa, Israel.
Email: ron@kpa-group.com
Xinwei Deng, Department of Statistics, Virginia Tech, Blacksburg, VA, USA.
Email: xdeng@vt.edu

Abstract

Modern statistics and machine learning typically involve large amounts of data coupled with computationally intensive methods. In a predictive modeling context, one seeks models that achieve high predictive accuracy on new datasets. This is typically implemented by partitioning the data into training and hold-out data sets. The allocation is often conducted randomly, at the row level of the data matrix. In this work, we discuss an overlooked gap in machine learning and predictive modeling, the role of data structure and data generation process in the partitioning of observational data into training and hold-out datasets. Ignoring such structures can lead to deficiencies in model generalizability and operationalization. We highlight that explicitly embracing the data generation structure to partition the data for validating predictive model is essential to the success of data science projects. The proposed approach is called befitting cross validation (BCV). It relies on an information quality perspective of analytics. This requires an assessment with inputs from domain experts, in contrast to automated approaches that are purely data driven. BCV is motivated by the objective of generating information quality with data and modeling. Two case studies are illustrating the proposed approach. One is based on a 96-h burn-in process applied to electro-mechanical devices, implemented in order to reduce early failures at the customer site. The goal was to shorten the burn-in process with a predictive model applied at 20 h. The other case study is combining tablet dissolution profiles and designed mixture experiments. The goal there was to match the tablet under test dissolution profiles with a brand tablet reference profile. These case studies demonstrate the methodological points made with BCV, which are generic in nature. We suggest that BCV principles should be always considered in the development of data-driven predictive models.

KEYWORDS

bootstrapping, burn in, cross validation, information quality, splines

1 | INTRODUCTION

Modern statistics and machine learning typically involves large amounts of data coupled with computationally intensive methods such as neural networks[1] and Gaussian processes.[2] For many of these models, classical inference such as hypothesis testing, is replaced by an emphasis on understanding generalized (out of sample) performance, as measured

by prediction accuracy. Rather than fitting models to all the data, one partitions the observations into training and holdout sets. Models are fit (parameters are estimated) using the training data set. Many models of different types are then considered, and their prediction accuracy is assessed on the holdout set.

The allocation of data rows to the training and holdout partitions are often implemented using random sampling, not taking into account the structure of the data. However, data often has a complex underlying structure matching the problem at hand. Examples of structure in data include operational restrictions and time sequencing issues resulting from the data generation process itself. In the literature, some resampling methods consider the data structure, to some degree. For example, using the Euclidian distance between the samples, the Kennard-Stone algorithm[3] considers splitting the data into the training and calibration sets based on the distance to the points already selected. Hedayat et al.[4] discuss balanced resampling methods, with the exclusion of contiguous units. Other approaches for data partitioning, such as leverage-based sampling[5] and venetian-blinds method[6] are based on the ordering of the samples and have been developed under different contexts. However, these methods do not explicitly incorporate the data generation structure into the modeling and data partition.

To fill this gap, we focuse on the role of data structure in the partitioning of observational data into training and hold-out datasets for assessing the fit of predictive models. We propose embracing the data generation structure to partition the data for validating predictive model. This is essential to the success of data science projects. To meet this challenge, the framework of befitting cross validation (BCV) is developed here in connection to the Information Quality (InfoQ) perspective.[7] The goal is to maximize the information quality generated from data with sensible machine learning analytics. A key differentiator of BCV applications is preserving independence between the training set and the hold-out set. The data-driven supervised resampling schemes[3] use the information from the test set to resample the data, thus rendering the hold-out set not completely independent. Different from the data-driven supervised resampling method, the proposed BCV does not use the structure of the to-be-modeled data itself but its design structure, to supervise the resampling. Therefore, it maintains independence between the hold-out dataset from the training dataset.

The BCV framework can also be applied to bootstrapping methods used to empirically generate properties of statistical estimators.[2] In the context of designed experiments with replications in experimental runs, repeated sampling with replacement on the replicates, provides empirical estimates of model coefficient variability. This has been proposed as a model goodness of fit methodology.[8] In the case of un-replicated experiments, an approach based on fractionally weighted bootstrap called SVEM, provides estimates of uncertainty in model coefficients and prediction accuracy.[9] Such methods also apply in cases of heavily censored data due to rare event responses and insufficient mixing of successes and failures between explanatory variables and saturated designed experiments.

This work fills an overlooked gap in predictive modeling in a machine learning context. Not accounting for randomization restrictions can have underappreciated negative consequences, such as overestimated predictive accuracy and operationalization problems. We consider the task of splitting data into training and test sets from the InfoQ perspective. Partitioning the data accounting for its generative properties into training and test sets, prior to modeling, will save time and effort.[7] With such considerations, the derived models can be easier to deploy with a better understanding of the generalization performance of the model. The next section briefly reviews cross validation.

2 | REVIEW OF CROSS VALIDATION

Predictive models and machine learning enable decision makers to set up policies, procedures and interventions aimed at enhanced performance and safety. As mentioned in the introduction, the performance of predictive models is typically assessed by cross validation with a holdout data not used in the training phase. The holdout set is used to assess the model's performance by comparing validation data values with model predictions. This data partitioning can be conducted in several ways such as k-fold cross validation, which is technically presented below.

Consider a data set **(X, y)** where rows of **X** correspond to n observations of p predictor variables, and a response vector **y,** corresponds to the dependent variables. Cross validation is typically performed by randomly picking l observations as the hold-out dataset and fitting a model using $(n\text{-}l)$ data points. The predictions on the l singled out observations are then compared to their actual values. When the number of records is not large (e.g., less than 500), data partitioning might not be advisable as each partition will contain too few records for model estimation and performance evaluation. An alternative to data partitioning, is k-fold cross-validation which is especially useful with small samples. The k-fold cross validation is a procedure that starts with partitioning the data into "folds," or non-overlapping subsamples. Often, one choose $k = 5$ folds, meaning that the data are randomly partitioned into five equal parts, where each fold has 20% of the

observations. A model is then fit k times. Each time, one of the folds is used as the holdout set and the remaining $(k-1)$ folds serve as the training set. The result is that each fold is used once as the holdout set, thereby producing predictions for every observation in the dataset. We can then combine the model's predictions on each of the k holdout sets in order to evaluate the overall performance of the model.

Cross-validation has been used in estimating out-of-sample prediction error, comparing different statistical models, in fitting the same data. It is also widely used for choosing the tuning parameters in various machine learning algorithms.[10] Moreover, cross-validation is not restricted to supervised settings, such as regression and decision trees, it can be extended to unsupervised methods, such as covariance matrix estimation and graph models.[11,12] Various resampling techniques are also related to cross-validation. For example, analyzing the stability and the robustness of clustering methods[13] can be accomplished by applying resampling similar to cross-validation. For references see Donoho and Huber[14] and Hennig.[15]

By fitting a model to a training data set, and then evaluate it on a holdout set, over-optimism of using the same data to fit and validate a model is avoided. Craven and Wahba[16] and Seeger[17] use cross-validated objective functions for statistical inference by integrating out-of-sample prediction error estimation and model selection, into one step.

Here we consider cross-validation that befits the structure of the data generation context, so that it provides enhanced information quality. This acknowledges that the process of collecting data often imparts structure to data. For example, in manufacturing processes, products generated on the same assembly line may reasonably be assumed to have higher correlation than products generated on two separate production lines. In computer vision applications, the equipment used to capture the images may induce correlations between images. This is an example of restricted randomization in the data since each camera was not randomly assigned to capture an image. Moreover, data collection over time can be affected by autocorrelations. Consider, for example, time effects of new camera technologies in computer vision, equipment degradations or upgrades in manufacturing. By taking advantage of correlations due to the data generation process, we show that we can improve the statistical properties of the cross-validation set selection.

Cody et al.[18] consider the structure of data in developing methods for a single systematic split between training and test sets. They leverage coverage across data features to partition data into a representative test and training split. In a representative split, the test set matches the training sets on all possible combinations of the features. They also develop out-of-distribution test sets where the test set is as different as possible from the training set to test model robustness. Their work demonstrates the value of including contextual information (such as information quality characteristics) on the structure of the data in cross-validation applications. Here we expand on the efforts in Cody et al.[18] in two ways. First we provide a general strategy for cross validation including multiple folds, and secondly we provide the theoretical basis for independence between the data sets, whereas Cody et al.[18] learn the features from the data.

Appropriate cross validation is connected to the concept of Information Quality concept (InfoQ). Kenett and Shmueli[7,19] present InfoQ as a general framework for planning, tracking and assessing information derived from a given data analysis effort. InfoQ is defined as the utility of a particular data set for achieving a given analysis goal by employing statistical analysis or machine learning. It consists of four components and eight dimensions that can be assessed qualitatively or, in some cases, quantitatively. The eight dimensions and information quality analysis workflow are shown in Figure 5. InfoQ has been applied in a wide range of application domains. For an automated application of InfoQ to the chemical processing industry see Reis and Kenett.[20] In Section 4, we implement InfoQ a to design befitting cross validation (BCV), a generic data holdout and cross validation strategy. The proposed BCV is introduced in the next section.

3 | BEFITTING CROSS VALIDATION

Machine learning often deals with observational data. This context meets a variety of data structures such as high-dimensional data with large numbers of variables relative to the number of observations or rectangular data with millions of observations. The data used for machine learning can include missing data points with random patterns such as missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).[21,22] From the data matrix perspective, data points in each row can be correlated or independent. Columns in the data matrix often contains certain dependency and collinearity that BCV accounts for. In Ghosh et al.,[23] five patterns of data structure in the crossed random effects model, $Y_{ij} = x_{ij}^T\beta + a_i + b_j + e_{ij}$, $1 \le i \le R$, $1 \le j \le C$, are summarized as in Figure 1: (1) balanced (2) row nested in column (3) column nested in row (4) independent and identically distributed (iid), and (5) arbitrary. For example, in the context of online experimentation, the data collected from web entries in search of a specific terms often

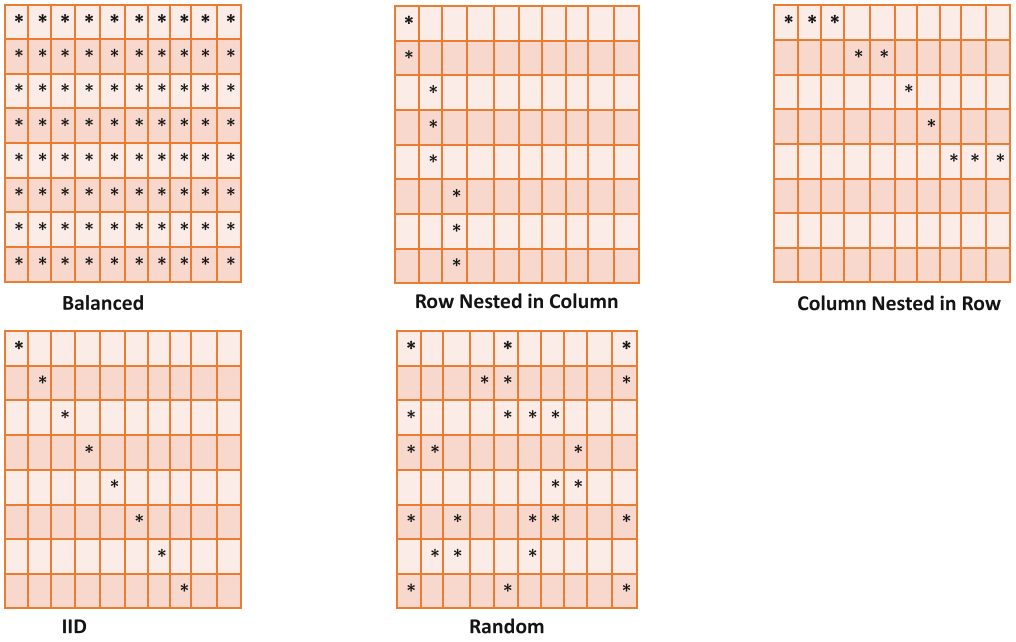

FIGURE 1 Illustration of five patterns of data structure.

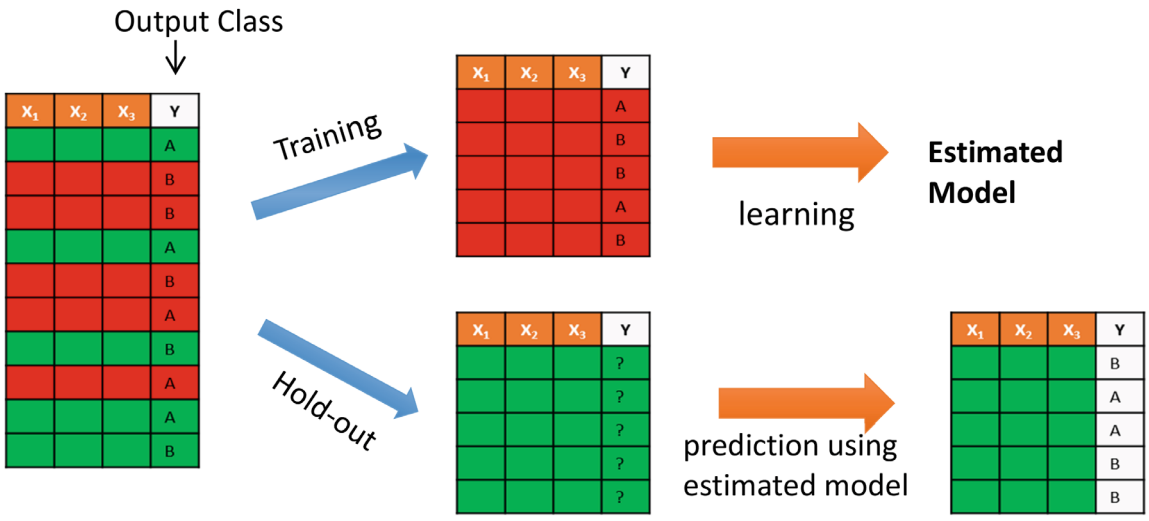

FIGURE 2 An illustration of conventional cross validation.

has the structure of row nested in column. Data collected from a sequence of web pages visited in unique surfing sessions often has the structure of column nested in row.[24]

Clearly, a complex data structure increases the challenge of machine learning to be sensible and accurate. In particular, the cross-validation approach, commonly used in machine learning, encounters the challenges of how to appropriately partition data into the training dataset and the hold-out dataset in complex data structures. Conventional cross validation applies a random partition of the data into training and hold-out datasets. As shown in Figure 2, such a random allocation cross validation approach does not necessarily account for the data structure in the whole dataset. In such cases the training dataset and the hold-out dataset do not have the same data structure as the whole data. Consequently, the machine learning method could result in misleading performance.

To address this challenge, we propose a befitting cross-validation (BCV) embracing the data generation structure, as illustrated in Table 1. BCV incorporates the data generation structure of the whole data into the partition of the training data and hold-out data with the following three principles.

TABLE 1 An illustration of a dataset with partitioning factors in the data generation structure.

Partitioning factors for data generation structure (z)						Predictor variables (x)			Response variables (y)			BCV based on units
Time tag (z_1)	*Unit ID* (z_2)	*Production line* (z_3)	*Product type* (z_4)	…	z_r	x_1	…	x_r	y_1	…	y_r	
1	1	A	X									Training
2	1	A	X									Training
3	1	A	Y									Training
4	1	B	Y									Training
5	1	B	Z									Training
6	1	B	Z									Training
7	2	A	X									Holdout
8	2	A	X									Holdout
9	2	A	Y									Holdout
10	2	B	Y									Holdout
11	2	B	Z									Holdout
12	2	B	Z									Holdout

BCV Principle 1: The formation of training and hold-out datasets should reflect the goal of the study.

BCV Principle 2: The training dataset and the hold-out dataset should have the same data generation structure as the whole dataset.

BCV Principle 3: The construction of the hold-out dataset should reflect the data generation structure needed for the predictive model.

Specifically, we use the concept of a blocking factor to elaborate the key idea of BCV for partitioning the training dataset and hold-out dataset. Under the context of BCV, we label these blocking factors, *partitioning factors*. Let us denote the whole data as $\mathbf{D} = \{(\boldsymbol{z}_1, \boldsymbol{x}_1, \boldsymbol{y}_1), \ \ldots, (\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n)\}$, which contains n data points. The $\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \ \ldots, \boldsymbol{x}_{ip})'$ are the p-dimensional vector of predictor variables for the ith observation, which are often used as the input variables in the machine learning. The $\boldsymbol{y}_i = (\boldsymbol{y}_{i1}, \ \ldots, \boldsymbol{y}_{iq})'$ are the q-dimensional vector of response variables for the ith observation, which are often used as the output variables in the machine learning. Here $\boldsymbol{z}_i = (\boldsymbol{z}_{i1}, \ \ldots, \boldsymbol{z}_{ir})'$ are the r-dimensional vector of partitioning factors for the ith observation. As illustrated in Table 1, these r-dimensional partitioning factors represent the information of the data generation structure, such as the multi-level structure, the hierarchal structure, and the nested structure. From the data matrix perspective, we denote $\mathbf{D} = (\boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{Y})$ with $\boldsymbol{Z}$ as the matrix of partitioning factors, $\boldsymbol{X}$ the matrix of input variables, and $\boldsymbol{Y}$ as the matrix of output variables. Note that the machine learning usually takes the data of input and output, $(\boldsymbol{X}, \boldsymbol{Y})$, to build prediction model. Thus, the conventional cross validation partitions the data based on $(\boldsymbol{X}, \boldsymbol{Y})$, that is, randomly select data points, to form the training dataset and hold-out dataset. Clearly, the conventional cross-validation does not take account the information of data $\boldsymbol{Z}$, that is, the data generation structure, for constructing the training and hold-out datasets. In contrast, BCV considers the full information of data $(\boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{Y})$ to compose the training and hold-out datasets, with the emphasis of using the information of the data generation structure $\boldsymbol{Z}$ to form an appropriate cross validation. Such a partition of training and holdout datasets can serve a better purpose of predictive modeling for the practical need.

For example as shown in Table 1, the whole data in a matrix format with the partitioning factors $\boldsymbol{z} = (z_1, \ldots z_r)$, predictor variables $\boldsymbol{x} = (x_1, \ldots x_p)$, and response variables $\boldsymbol{y} = (y_1, \ldots y_q)$. Suppose the engineer is interested in assessing the predictive model at the unit level. Randomly allocating rows to a training and holdout sets will give overly optimistic assessments of algorithm performance. The proposed BCV accounts for the partitioning factors $z_1, \ldots z_r$ in composing the training data and hold-out datasets. Specifically, we apply BCV to partition the data into training and holdout datasets by randomly allocating entire units into training or holdout sets with proper stratification to ensure that the proportion of failing units was identical in the training and holdout sets.

Figure 3 illustrates a data set with an underlying structure such as raw material batch, production line, and type of product.[25] Conventional cross validation does not account for the structure by singling out specific raw material

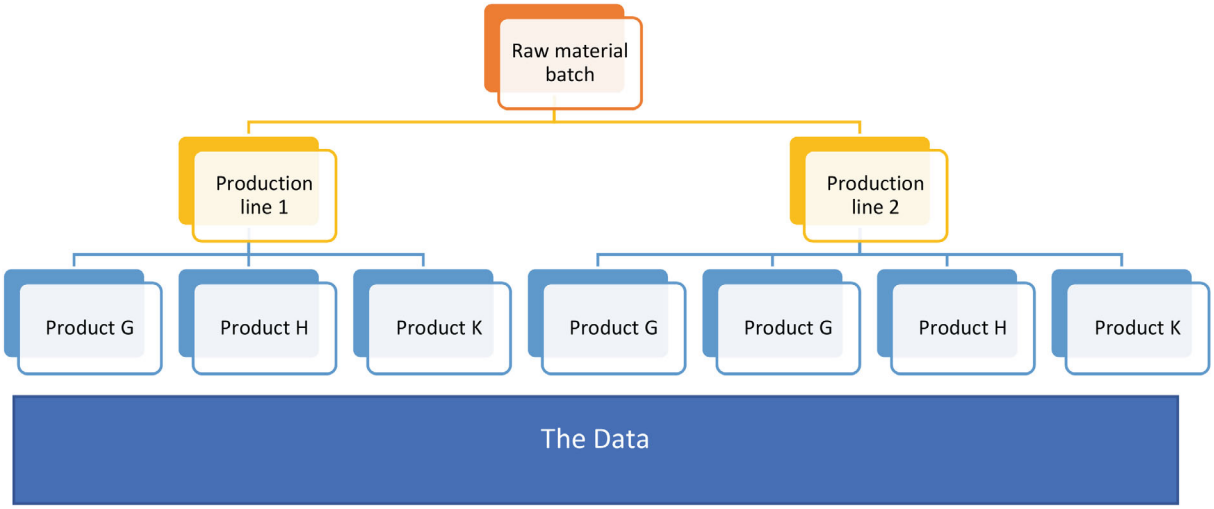

FIGURE 3 Illustration of the dataset with a hierarchical structure.

batches, production processes, or products. For such structured multilevel data, the use of cross-validation for estimating out-of-sample prediction error and model selection deserve close attention. BCV can be conducted over the batch index, production line, or product type.

In order to test a model accounting for such a structure, the holdout set cannot be a simple random sample of the data. It needs to have a multilevel structure where groups, as well as individual observations, are held out. An example in the context of the analysis of a survey is provided in Price et al.[26] A Bayesian context is presented in Vehtari et al.[27] In Section 4, we develop BCV in an industrial case study. In general, there are no specific guidelines for conducting cross validation in multilevel structured data.[28]

The BCV has close connections with attempts to improve cross validation. For example, Rabinowicz and Rosset[29] consider cross validation (CV) with correlated data. They introduce a criterion for suitability of standard CV in presence of correlations. When this criterion does not hold, they propose a bias corrected cross-validation estimator called CVc. It yields an unbiased estimate of prediction error in many settings where standard CV is invalid. The analysis in their work is focused on correlation due to latent objects, such as random effects used in generalized linear mixed models. In this context, the latent objects represent the data generation structure accounted for by cross validation based on the BCV principles.

Note that the data generation structure is typically considered beyond the partitioning factors. Yashchin[25] treats multistage process monitoring using time sliced data for enhanced diagnostics. In another context, Pawlowsky-Glahn[30] consider compositional data analysis (CoDa), where there is a distinction between missing data, due to a lack of observation or record, and structural zeroes, which are data points that cannot be observed. Compositional data has a structure of vectors of nonnegative elements, analyzed in relative terms. The properties in such data, need to be identified in considering the goals of the data analysis. The InfoQ framework mentioned in section 2 provides a structured approach linking domain expertise to such properties. This knowledge facilitates the incorporation of data generation structure into BCV.

4 | CASE STUDY I: ELECTRO-MECHANICAL DEVICE BURN-IN

In this section, we demonstrate BCV with a case study from an electronics manufacturer of electro-mechanical devices. The focus is an application of a burn in procedure for tackling early defects in products. Most units function properly after final assembly, but about 1 in 5 fail early in the life of the product. To meet warranty requirements, a burn-in procedure places each unit under temperature and humidity accelerated conditions for up to 96 h. The units that fail during the test are screened out, and the units that survive are shipped. The test equipment has sensors that record the activity in the devices. In addition to adding several days to the production time, the accelerated burn-in test damages and shorten the life of all the units, not just the faulty ones.

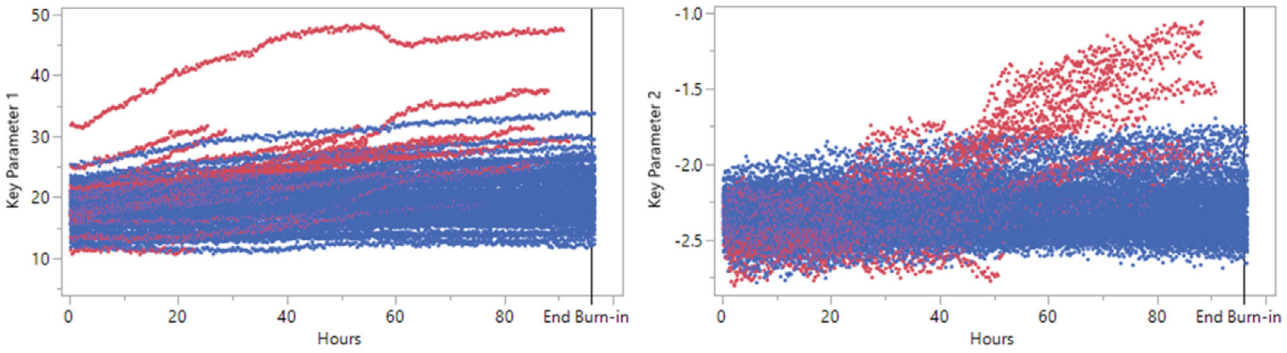

FIGURE 4 Traces of two sensors over the first 96 h.

The problem of interest is whether the sensor data can be used to shorten the burn-in test. The goal was defined as using sensor readings to predict, at 20 h, the units that would fail, had the test continued over the full 96 h. The data considered consists of sensor streams from 92 devices. There is a serial number indicating the device, which is coded as Unit ID. The units that did not fail had 96 h of measurements. The sensor readings had unequal times between measurements and the times of the measurements were different from unit to unit. Devices whose sensor series ended in less than 96 h were the ones that failed. Two columns of sensor readings in the data were identified as "key parameters," whose trajectories in time were thought to be predictive of failure. The case study is presented next using the 4 InfoQ components and the 8 InfoQ dimensions.

The *Goal* of the analysis is to develop a machine learning (ML) algorithm that takes as input only the early part of the series and accurately predicts the rest. The *Utility* of the project is measured by the proportion of flawed devices that escape detection by the algorithm, as well as the proportion of properly functioning devices that are incorrectly predicted to fail early. This can be viewed as a classification problem. There are quantifiable costs associated with each of these outcomes.

Figure 4 shows the complete, 96-h, sensor streams of the two key parameters. The units that failed are in red, and the ones that did not fail are in blue. The data were modeled using only the first 20 h. The color of the trace is determined retrospectively, once a system is determined as pass or fail.

The *Data Resolution* consists of pairs of continuous measurements for each of the 92 units, of which 26 failed. Despite there being many rows in the data, this is not a lot of information, since the objective is to predict a binary outcome (pass or fail). There are many more sensor measurements than needed. We subset the data by keeping only every 64th pair of sensor measurements, starting with the first measurement per unit. This subsampling did not alter or hide any important feature of the sensor streams, and our analysis of the complete data leads to the same results. However, there were other potentially crucial pieces of information missing. The data had no indicator of which machine the units were being tested on, and there was no batch level information about which units were made in the same lot or close to one another in time. Furthermore, actual test conditions, like temperature and humidity, were not present in the data. Despite the absence of this information, ultimately the project met its stated goal. From the visualization of Figure 4, it appears that the failed units have distinctly different parameter trajectories from those of the successful ones. The data are structured as sets of irregularly spaced pairs of time series of different lengths indexed by Unit ID. The trajectories of the failed units are missing from the data after the time of failure, since failure meant the device had shut down and become completely inoperable.

According to *Chronology of Data and Goal*, we only make predictions based on information that is available at the time of scoring. Accordingly, as a first step, we remove all the sensor measurements after 20 h. Figure 5 elaborates the information quality workflow for data preparation and analysis.

As mentioned in section 2, ML models are often assessed using a cross validation approach that partitions the data into a training set and a holdout set. Here, randomly selecting rows, without considering the unit trajectories, is problematic. For one, it is not emulating the structure of how new data arrives in practice. The assessment of how well models work is at the device level, and not at the row level. Randomly allocating rows to a training and holdout sets will give overly optimistic assessments of algorithm performance. We therefore apply BCV to partition the data into training and holdout datasets. Note that we need to make prediction of device failure, of which we have only 26 units of information. Thus, based on BCV, we randomly allocate entire devices into training or holdout sets. This was done in a stratified way to ensure

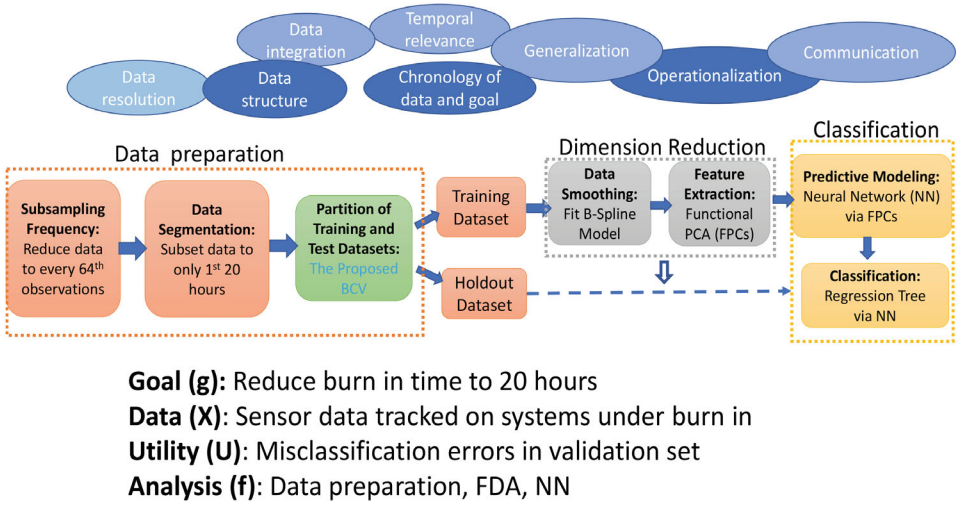

FIGURE 5 The data preparation and analysis workflow. The eight dimensions of InfoQ are shown at the top of the figure in parallel to the workflow.

that the proportion of failing units was identical in the training and holdout sets. This stratified and grouped partitioning is made easy with the Make Validation Column capability of JMP Pro 15.

Specifically, we use BCV to split the data using 2/3 of the units for training, and 1/3 for holdout. That is, the unit is used as the partitioning factor for BCV. This led to 61 devices being placed in the training set, of which 17 devices failed, while 31 devices were placed in the holdout, 9 of which failed. Preserving the unit structure in the training and holdout sets satisfies BCV Principles 1 and 2 in that the partitioning reflects the goal and generating structure. BCV Principle 3 is satisfied because only the first 20 h of data are used from units not used to train the model, reflecting the structure of new data will arrive when the model is put to use.

The two plots in Figure 6 show the stratified and grouped partitioning of the data on the top, and the erroneous naïve partitioning below. The key advantage of the stratified and grouped partitioning is that we are emulating the population we are predicting with whole functions from new devices. For instance, the BCV and random hold-out show a difference around 11 h and key parameter 1 at 20. The partitioning on the bottom ignores this structure and would be appropriate if our goal was fitting models that can interpolate functions that have random gaps in them, which is a different population from the one we intend to generalize the model to.

After the data partition, using BCV, we conduct a dimension reduction for the data. A B-spline model was used for data smoothing. A basis spline (or B-spline) is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. Any spline function of given degree can be expressed as a linear combination of B-splines of that degree. The term was coined by Schoenberg[31] who has been recognized as the "father of splines". B-splines of order n are basis functions for spline functions of the same order defined over the same knots. All possible spline functions can be built from a linear combination of B-splines, and there is only one unique combination for each spline function.[32]

The detail of using the B-spline smoothing can be found in the Appendix. Then a set of principal component scores were extracted for each function. Specifically, the B-spline model for smoothing of the sensor data is a linear mixed model, and the functional principal component (FPC) scores are a rotated version of the spline coefficient random effects, similar to the rotation that is applied to vector data in principal components analysis. The initial data was in a stacked format where the sequences of measurements were vertically stacked. These sequences have different lengths and unequal sampling times, and there is strong autocorrelation across the rows. Extracting and assembling the principal components scores, along with the pass/fail result for the device, moves this irregularly shaped dataset into a rectangular-shaped dataset shown below in Table 2, where each row is a separate device. This is the data used for the classification model.

Based on the flowchart in Figure 5, the classification modeling component involves a sophisticated model and simple decision rule based on that model. The model is an FPCs analysis to the first 20 h of the two key parameters, and an extraction of the functional principal components scores (FPC scores). Then, a neural network is fit with the failure indicator as the response and the FPC scores as inputs. The neural network model predicts the probability of failure at 96 h using only the information available at 20 h. A final step in the classification modeling is to create an operationalizable

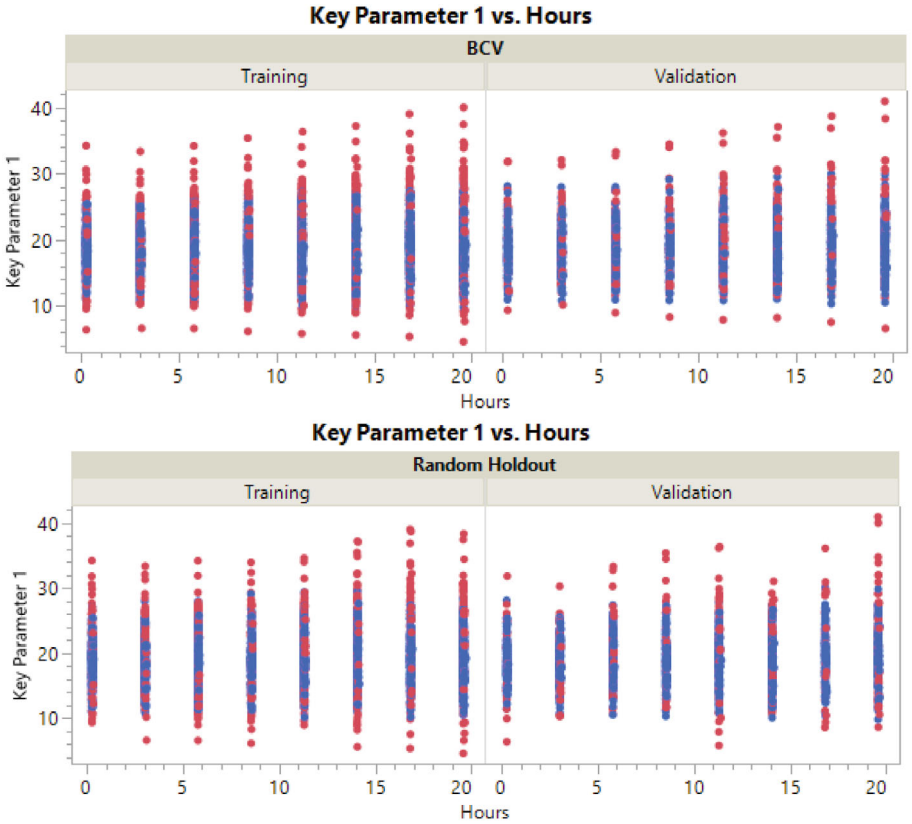

FIGURE 6 Partitioning by unit trajectories (top) and random partitioning (bottom).

TABLE 2 An illustration of the formed training dataset.

Row	Unit ID	Hours	Failed	Key parameter 1	Key parameter 2	BCV
1	1	0.274	0	17.6	−2.25	Training
2	1	0.541	0	17.9	−2.24	Training
3	1	0.819	0	18.1	−2.26	Training
4	1	1.10	0	17.6	−2.37	Training
5	1	1.37	0	18.3	−2.31	Training

decision rule for sorting the devices likely to fail from those less likely to fail. This is done by fitting a simple regression tree to the failure response using the probability of failure from the neural network as the only input. This simple tree model gives a data-derived probability threshold (e.g., a neural predicted failure probability >0.4) for deciding which units should be scrapped. The B-spline model that led to the FPC scores, the neural model, and the simple regression tree model were all fit using the BCV training set and tuned/evaluated using the BCV holdout set.

The initial structure of the data, when collected, often suggests the analysis approach. In this case, the time at which the sensors take measurements is unequally spaced within each unit and taken at different times for different units. This means that the data is in a "tall" format where all the measurements for a key parameter are contained in a single data column. This required us to think about unit structure explicitly and led to the model based on FPC scores. On the other hand, had the sensor measurements been taken at the same time for each unit, the data may have arrived in a "wide" format as one column per time point per key parameter, so that the data for each unit is contained in a single row. In that case, partitioning the data into training and holdout sets, by randomly allocating rows, would have naturally followed the BCV principles because the units correspond with the rows. We proceed with a second case study.

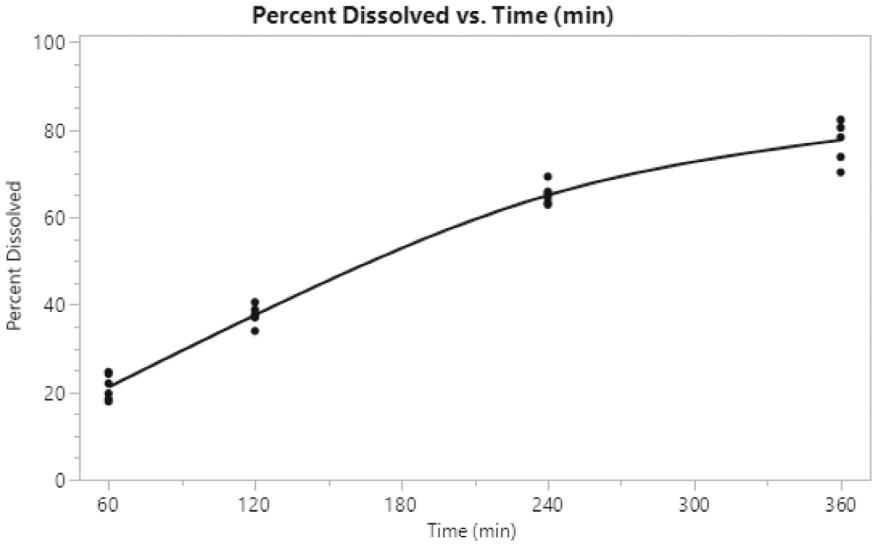

FIGURE 7 Dissolution measurement for the reference tablets.

5 | CASE STUDY II: DISSOLUTION FORMULATION EXPERIMENT

The profile of the percentage that a tablet-based drug delivery product typically dissolves over time is called a dissolution curve. Pharmaceutical industry manufacturers, operating in the drug development stage, optimize formulations of their products to match a specific dissolution curve trajectory. They typically use design of experiments techniques for planning the study and modeling the resulting data. Generic drug manufacturers must demonstrate to regulatory agencies that candidate formulations have a dissolution curve that matches that of a sample from the reference brand.[33] It saves time and resources for the manufacturer if the candidate formulations, proposed by a statistical model, generalize to new data. In this section, we describe an example of how a generic drug manufacturer applied BCV to the design and analysis of an experiment where the goal was to find a tablet formulation that closely matched that of a brand-name drug.

The generic drug manufacturer took six brand-name replicate tablets and measured their dissolution at 60, 120, 240, and 360 min. The average observed dissolution profile is shown in Figure 7.

They designed an experiment with four factors. Polymer A and Polymer B were the percentages of two possible polymer additives, Total Polymer, the total amount of both polymers added together, and Compression Force, the amount of force applied to the tablets during the printing step. There were 16 combinations of these four factors considered. For each combination of factors, two replicate batches of tablets were prepared and printed. From each batch, six tablets had their dissolution profiles measured at 60, 120, 240, and 360 min.

Ultimately, the manufacturer needed to demonstrate that their product had a dissolution profile similar to that of the brand-label tablets. With this goal in mind, it was important that the dissolution curve of the formulation recommended by the model generalized to new batches. The data were partitioned into a training set and a holdout set using BCV. For each of the unique combinations of the factor settings, all the data from one of the two replicate batches was randomly placed into the training set. The data from the remaining replicate batch was placed into the holdout set. That is, the replicate batch is used as the partitioning factor in the BCV. In this way, the goal of predicting the dissolution profile at the batch level was preserved, satisfying BCV Principle 1. The procedure ensured that the set of formulations, randomization structure, and time sequencing of the measurement in the two partitions of the data were identical, which follows BCV Principle 2. This is necessary because randomly allocating batches without considering the replicates alters the design of experiments structure in the data, potentially leading to a training set that is unable to estimate all the interaction effects. Prediction is at the batch mean level, which is the generation structure the model will predict to, satisfying BCV Principle 3.

Table 3 shows how replicates were allocated randomly to the training and holdout sets, keeping in mind that data from six tablets correspond to each replicate within a batch.

The analysis proceeds by fitting Weibull growth curves, a common and recommended model for this type of data,[33] to each of the batches in the training and holdout sets. Weibull growth curves have three parameters: an asymptote,

TABLE 3 An illustration of the training/holdout partitioning in case study II.

Batch	DoE setting	Rep	Polymer A	Polymer B	Total polymer	Compression force	BCV
1	1	1	0.825	0.175	0.16	2500	Training
2	1	2	0.825	0.175	0.16	2500	Validation
3	2	1	0.775	0.225	0.14	2500	Training
4	2	2	0.775	0.225	0.14	2500	Validation
5	3	1	0.725	0.275	0.14	1500	Validation
6	3	2	0.725	0.275	0.14	1500	Training

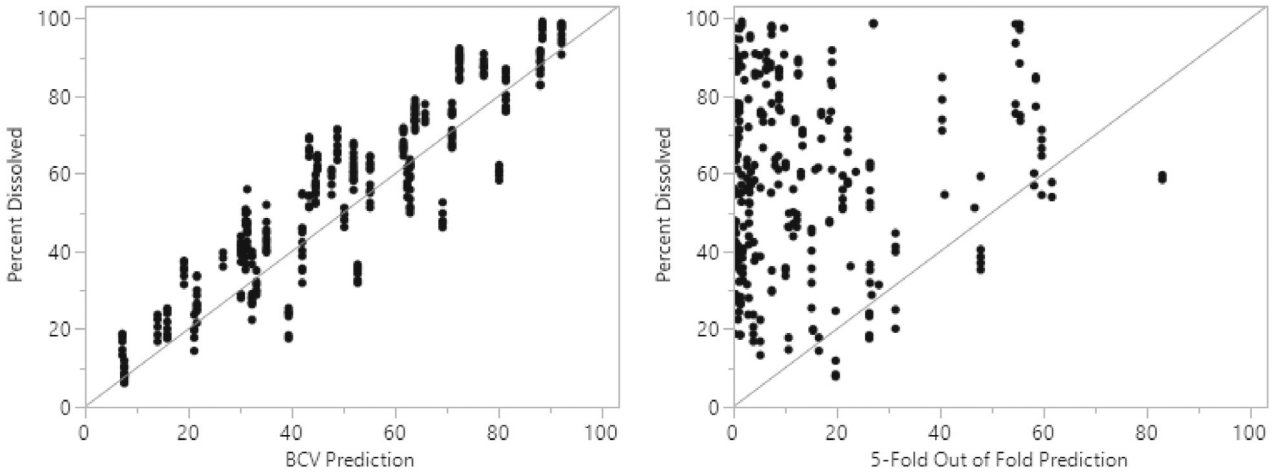

FIGURE 8 Prediction comparison for the BCV (the left panel) and five-fold random partition (the right panel).

an inflection point, and a growth rate. Best subsets regression models were fit to the extracted parameters using the training data, and the combination of the main effects and interactions of the four factors was chosen using the extracted parameters calculated from the holdout batches. The models for the three Weibull growth parameters were combined into an overall prediction of the mean dissolution curve trajectory as a function of both time and the design factors. The optimal tablet formulation, as measured by maximum relative difference between the predicted dissolution curve, was found to be 76% of polymer A, 24% of polymer B, 1.5% total polymer, and a compression force of 2000 lbs/sq. inch. The generic formulation was demonstrated to have a similar dissolution to the brand-name product using a new reference batch and a batch made using the optimal formulation recommended by the model.

We can compare the proposed BCV with other methods ignoring the generating structure of the data. The advantage of the BCV, over other methods, can vary widely from situation to situation and the extent to which structure is ignored. For example, a typical approach to analyzing data of the same size as the dissolution curve data, is to use random k-fold cross validation. As an alternative to BCV, we can randomly allocate each of the 792 measurements into one of five folds, ignoring the time sequencing and design of experiments structure of the data, and then repeat the analysis conducted to the BCV partitioning. Figure 8 reports the scatter plot of prediction from the BCV method against the true response in the holdout data (the left panel), and the out-of-fold prediction performance for the method based on random five-fold partitioning of the rows (the right panel).

One can see that the method based on five-fold cross-validation generalizes poorly as measured by out-of-fold performance because the time and design structure are critically important to maintain for this data. In practice, with the prediction results as in the right panel of Figure 8, analysts would begin rethinking their approach. It implies that the partitioning must be conduct in a way that preserves the structure of the data, essentially the key idea of BCV. Note that such practices with the need of BCV, is likely to occurr in many situations. Our purpose here is to draw attention to it and recommend thinking about goal and data structure at the data partitioning step of any modeling exercise.

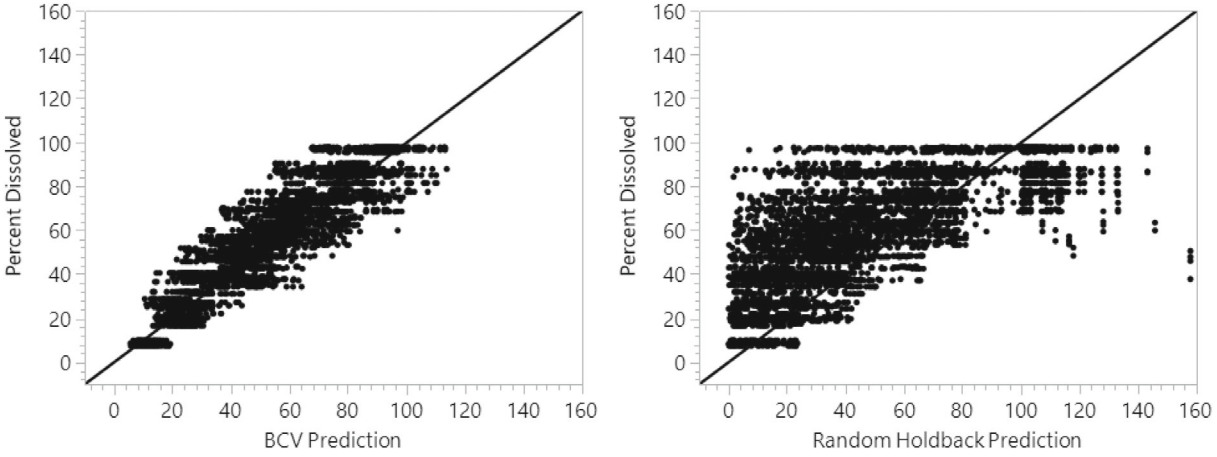

FIGURE 9 Prediction comparison for 1000 random partitions using both BCV (the left panel) and completely random partitions (the right panel).

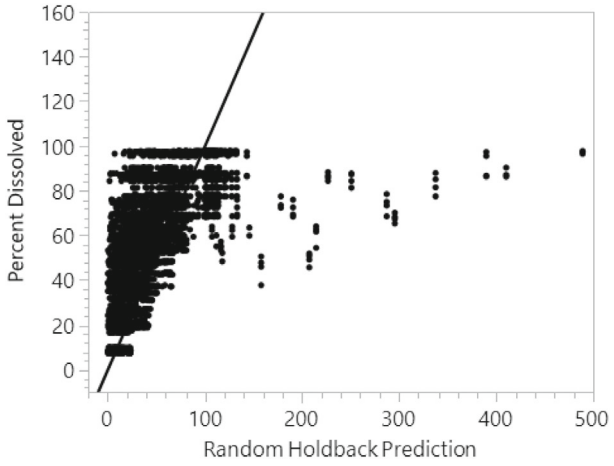

FIGURE 10 Prediction comparison for the complete range of the prediction from the completely random partitioning (the right panel).

A rigorous comparison of BCV to naïve approaches of partitioning the data is to perform a sensitivity analysis that repeats the random aspects of BCV, in comparison to a second approach, not accounting for the data structure. Here we repeat the BCV partitioning by randomly placing all the data from one of the two replicates from each of the 16 formulations studied into a training set and the other replicate into a holdout set, and repeat the analysis process. We compare this to the results of a naïve approach where we randomly place 50% of the original measurements into a training set, completely ignoring the design, tablet, and time structure in the data and place the remaining rows into a holdout set, and repeat the modeling process. This random repartitioning was done 1000 times for each approach, respectively. Note that although a 50%–50% training/holdout partitioning is less common, for BCV this was essential to preserve the structure of the designed experiment. We also use the same number of training rows in the naïve approach for a fair comparison.

Figure 9 reports the scatter plots of actual responses against predicted values from this sensitivity study. The plot on the left shows that randomly repeating the BCV approach leads to more stable predictions than the plot on the right of the naïve partitioning approach. While the *x*-axis and *y*-axis of Figure 9 have the same range, Figure 10 shows the complete range of the naïve approach predictions on the *x*-axis. It is seen that in Figure 10 there were many non-physically meaningful predictions far above 100% dissolution. The naïve approach also had predictions with the starting time dissolution exceeding 100% and runs with the dissolution at the final time point less than 1%. Neither of which is physically viable. This kind of extreme behavior did not happen in the data or any of the BCV-based predictions. Ignoring the data generating structure produced worse predictions.

6 | EXTENSION TO BOOTSTRAPPING

In this section, we briefly discuss the extension of BCV to resampling techniques such as bootstrapping.

The bootstrap was originally motivated as a general tool, when the goal is more inferential in nature where uncertainty quantification via confidence intervals is needed.[34] There are many variations of the bootstrap that can be placed into two groups, parametric and nonparametric bootstrap methods. Both families of methods are simulation-based and make it straightforward to obtain confidence intervals with excellent statistical properties for quantities that would otherwise be difficult to derive analytical results for. The bootstrap simulation creates a population of parameter estimates that is treated as an approximate sample from the population of parameter estimates.[2] Because uncertainty quantification is often more difficult than prediction, it is reasonable to assume that considerations of unit structure and randomization restrictions are even more consequential than the prediction.

The nonparametric bootstrap methods make use of resampling, or more generally, reweighting of the original data. A significant advantage of this is that data reuse leads to confidence intervals that are robust to model misspecification, such as the generating distribution of the data. With the consideration of BCV, an important but often unstated assumption, is that unit structure or randomization restrictions must be reflected in the resampling scheme. This is characteristic in designed experiments,[8] but also needs to be carefully accounted for in the observational data.[35] If correlation is present across the rows of a dataset, there is no reason to think that naïvely applying a resampling of the rows of the data will lead to confidence intervals with a controlled Type I error rate.

The parametric bootstrap takes a different approach to the simulation. In this case, one uses the model structure that was fit to the original data, along with the parameter estimates, to generate new values of the responses. Once one has the new response values, the procedure that was fit to the initial dataset is applied to the simulated data in much the same way as the nonparametric bootstrap. With the consideration of BCV, if data have correlation across the rows, then one crucial step is to check whether the fitted model can correctly capture the generating distribution (or model) of the original data as well as the underlying mechanism. This implies that one can treat the resulting simulation sample of parameter estimates as being approximately from the sampling distribution of the estimates. This means that one can obtain confidence intervals in situations that otherwise would be difficult to resample from, such as time series and spatial data. Note that this comes at the expense of requiring the assumption of model fitting the data. This restricts the set of models that can be used to probabilistic models that directly incorporate correlation between measurements, such as generalized linear mixed models, and excludes many machine learning models, such as neural networks, that do not.

7 | DISCUSSION

In this work, we address issues with hold out sets and cross validation in the context of data splitting.[36] We propose the BCV method to handle such conditions and thereby provide for enhanced information quality in predictive analytics modeling. BCV focuses on cross validation for predictive models, whose goal is prognostic with the InfoQ utility being prediction accuracy. The proposed BCV key concepts and principles can be extended to bootstrapping, whose goal is diagnostic. In this case the InfoQ utility is error rate and estimates of variability.

Rabinowicz and Rosset[29] discuss such issues in a generic context. They state that: "When it comes to prediction, the question of whether there is correlation between the observations from the training set and the prediction set—the sample that is used for the model's parameter estimation, and the set of points whose response is predicted based on the trained model, respectively—plays an important role." They propose a technical correction to the correlation.

Here we take an information quality perspective by first studying the structure in the data and then, design a cross validation approach that matches that structure. This approach provides enhanced generalizability of the predictive model.[7,19]

Growing area of application of BCV consist of physical and computer experiments,[2,37-39] advanced manufacturing,[40-42] and digital twins.[43,44] These platforms provide monitoring capabilities, diagnostic and prognostic analytics. A special feature of digital twins is that they are fed, online, by sensor data. In assessing the predictive performance of models used in prognostics, one often uses holdout sets or cross validation. As shown here, the hierarchical, stratified structure in the data, needs to be accounted for.

This paper deals with a general methodological issue requiring an information quality assessment, so that the analytics associated with data generates information quality. From such an perspective, one can derive proper holdout and cross

validation strategies that enable proper generalization of predictive analytic models. BCV extensions are also possible to situations where the data generation process can be learned from the data itself.

ACKNOWLEDGMENTS

The authors thank the Editor and referees for their insightful and constructive comments that have greatly improved the original manuscript.

DATA AVAILABILITY STATEMENT

Embargo on data due to commercial restrictions

ORCID

Ron S. Kenett https://orcid.org/0000-0003-2315-0477

REFERENCES

1. Efron B, Hastie T. *Computer Age Statistical Inference: Algorithms, Evidence and Data Science*. Cambridge University Press; 2016.
2. Kenett RS, Zacks S. *Modern Industrial Statistics: With Applications in R, MINITAB, and JMP*. 3rd ed. John Wiley and Sons; 2021.
3. Kennard RW, Stone LA. Computer aided design of experiments. *Dent Tech*. 1969;11:137-148.
4. Hedayat AS, Rao CR, Stufken J. Sampling plans excluding contiguous units. *J Stat Plan Infer*. 1988;19:159-170.
5. Wang Y, Veltkamp DJ, Kowalski BR. Multivariate instrument standardization. *Anal Chem*. 1991;63(23):2750-2756.
6. Rácz A, Bajusz D, Héberger K. Modelling methods and cross-validation variants in QSAR: a multi-level analysis. *SAR QSAR Environ Res*. 2018;29(9):661-674.
7. Kenett RS, Shmueli G. *Information Quality: The Potential of Data and Analytics to Generate Knowledge*. John Wiley and Sons; 2016.
8. Kenett RS, Rahav E, Steinberg D. Bootstrap analysis of designed experiments. *Qual Reliab Eng Int*. 2006;22:659-667.
9. Yu L, Gotwalt C, Hong Y, King C, Meeker WQ. Applications of the fractional random-weight bootstrap. *Am Stat*. 2020;74(4):345-358.
10. Morgan JP, Deng X. Experimental design. *WIREs Data Mining Knowl Discov*. 2011;2:164-172.
11. Deng X, Tsui KW. Penalized covariance matrix estimation using a matrix-logarithm transformation. *J Comput Graph Stat*. 2013;22(2):494-512.
12. Deng X, Yuan M. Large Gaussian covariance matrix estimation with Markov structure. *J Comput Graph Stat*. 2009;18(3):640-657.
13. Chu S, Jiang H, Xue Z, Deng X. Adaptive convex clustering of generalized linear models with application in purchase likelihood prediction. *Dent Tech*. 2020;63(2):171-183.
14. Donoho, D.L. and Huber, P.J. (1983) The notion of breakdown point, in: Bickel, P. J., Doksum, K. and Hodges J. L. Jr. (eds.), *A Festschrift for Erich L. Lehmann* Wadsworth, p. 157-184.
15. Hennig C. Cluster validation bootstrap robustness clustering with noise Jaccard coefficient. *Comput Stat Data Anal*. 2007;52(1):258-271.
16. Craven P, Wahba G. Smoothing noisy data with spline functions. *Numer Math*. 1978;31:377-403.
17. Seeger MW. Cross-validation optimization for large scale structured classification kernel methods. *J Mach Learn Res*. 2008;9:1147-1178.
18. Cody T, Lanus E, Doyle D, Freeman L. Systematic training and testing for machine learning using combinatorial interaction testing. Submitted to ICSE; 2022.
19. Kenett RS, Shmueli G. On information quality. *J Royal Stat Soc Ser A*. 2014;177(1):3-38.
20. Reis M, Kenett RS. Assessing the value of information of data-centric activities in the chemical processing industry 4.0, AIcHe. *Process Syst Eng*. 2018;64(11):3868-3881.
21. Fuchs C, Kenett RS. Missing data and imputation. In: Ruggeri F, Kenett RS, Faltin F, eds. *Encyclopedia of Statistics in Quality and Reliability*. John Wiley and Sons; 2007.
22. Rubin DB. Inference and missing data. *Biometrika*. 1976;63:581-592.
23. Ghosh S, Hastie T, Owen A. Backfitting for large scale crossed random effects regressions; 2020. https://arxiv.org/abs/2007.10612
24. Kohavi R, Thomke S. The surprising power of online experiments. *Harvard Bus Rev*. 2017;95(5):74-82. https://hbr.org/2017/09/the-surprising-power-of-online-experiments
25. Yashchin E. Statistical monitoring of multi-stage processes. In: Knoth S, Schmid W, eds. *Frontiers in Statistical Quality Control*. Vol 12. Springer; 2018. doi:10.1007/978-3-319-75295-2_11
26. Price PN, Nero AV, Gelman A. Bayesian prediction of mean indoor radon concentrations for minnesota counties. *Health Phys*. 1996;71:922-936.
27. Vehtari A, Gelman A, Gabry J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC; 2015. https://arxiv.org/abs/1507.04544
28. Wang W, Gelman A. Difficulty of selecting among multilevel models using predictive accuracy. *Stat Interf*. 2015;8:153-160.
29. Rabinowicz A, Rosset S. Cross-validation for correlated data; 2019. https://arxiv.org/abs/1904.02438
30. Pawlowsky-Glahn V, Egozcue JJ, Tolosana-Delgado R. *Modeling and Analysis of Compositional Data*. John Wiley & Sons; 2015.

31. Schoenberg J. Spline functions, convex curves and mechanical quadratures. *Bull Am Math Soc*. 1958;64:352-357.
32. Karlin S, Pinkus A. Interpolation by splines with mixed boundary conditions. In: Karlin S, Micchelli CA, Pinkus A, Schoenberg IJ, eds. *Studies in Spline Functions and Approximation Theory*. Academic Press; 1976:305-325.
33. FDE Guidance for Industry. Dissolution testing of immediate release solid oral dosage forms; 1997.
34. Efron B, Tibshirani T. *An Introduction to the Bootstrap*. Chapman & Hall; 1993.
35. Ganju J, Lucas JM. Analysis of unbalanced data from an experiment with random block effects and unequally spaced factor levels. *Am Stat*. 2000;54(1):5-11.
36. Cox DR. A note on data-splitting for the evaluation of significance levels. *Biometrika*. 1975;62(2):441-444.
37. Deng X, Lin CD, Liu K-W, Rowe RK. Additive Gaussian process for computer models with qualitative and quantitative factors. *Dent Tech*. 2017;59(3):283-292.
38. Wu CJ, Hamada MS. *Experiments: Planning, Analysis, and Optimization*. 2nd ed. John Wiley & Sons; 2011.
39. Zhang Q, Qian PZ. Designs for cross-validating approximation models. *Biometrika*. 2013;100(4):997-1004.
40. Kang S, Jin R, Deng X, Kenett RS. Challenges of modeling and analysis in cybermanufacturing: a machine learning and computation perspective. *J Intell Manuf*. 2021;1-14. doi:10.1007/s10845-021-01817-9
41. Wang L, Chen X, Kang S, Deng X, Jin R. Meta-modeling of high-fidelity FEA simulation for efficient product and process design in additive manufacturing. *Addit Manuf*. 2020;35:101211.
42. Zhang Q, Deng X, Qian PZ, Wang X. Spatial modeling for refining and predicting surface potential mapping with enhanced resolution. *Nanoscale*. 2013;5(3):921-926.
43. Jiang HJ, Deng X, Lopez V, Hamann H. Online updating of computer model output using real-time sensor data. *Dent Tech*. 2016;58(4):472-482.
44. Kenett RS, Bortman J. The digital twin in industry 4.0: a wide-angle perspective. *Qual Reliab Eng Int*. 2021;38(3):1357-1366. doi:10.1002/qre.2948

How to cite this article: Kenett RS, Gotwalt C, Freeman L, Deng X. Self-supervised cross validation using data generation structure. *Appl Stochastic Models Bus Ind*. 2022;1-16. doi: 10.1002/asmb.2701

APPENDIX A. THE B-SPLINE MODEL

The B-spline model introduced in Section 4 for smoothing of the sensor data is a linear mixed model. The application of linear mixed models is standard practice in classical statistical modeling situations where the sampling structure of the data is more complicated than a completely random design (CRD) but are often overlooked as part of the data scientist's toolkit in machine learning situations. In contrast to standard CRD-type machine learning scoring, which is essentially identical to scoring on the training set, the scoring of random effects on the validation set must be done in a way that combines training set estimates and non-training set data carefully.

$$y_i\left(t_{i,j}\right) = \sum_k \beta_k b\left(t_{i,j}\right) + \sum_k \gamma_{i,k} b\left(t_{i,j}\right) + \varepsilon_{i,j,k}.$$

Above is the linear mixed model that JMP Pro uses for fitting a functional response model using B-splines. $b\left(t_{i,j}\right)$ are the B-spline basis functions, β is the vector of fixed effects that characterize the mean function trajectory, the $\gamma_{i,k}$ are random effects that are assumed distributed $N\left(0, \sigma_k^2\right)$. These random effects characterize how individual functions deviate from the mean function, and each basis function has its own variance component, σ_k^2, that must be estimated from the data. There are also residual errors, $\varepsilon_{i,j,k} \sim N(0, \sigma_\varepsilon^2)$, whose variance must be estimated.

The elements of the fixed effects design matrix, $\mathbf{X}_{Tr}$, are the values of the B-spline basis functions, $b\left(t_{i,j}\right)$. The random effects design matrix, $\mathbf{Z}_{Tr}$, is $\mathbf{X}_{Tr}$ nested within Unit ID. The Henderson mixed model equations are solved jointly for the training set estimates and the best linear unbiased prediction (BLUPs), γ_{Tr}, which are then rotated using a weighted singular value decomposition. When scoring the validation set random effects, γ_{Val}, one might be tempted to combine the training and validation data into one large dataset and score the random effects using the Henderson equations after plugging in the training set variance component estimates. This is problematic in several ways. One computational disadvantage is that this can be slow and unwieldy because the training set response values and design matrices must be stored indefinitely. These computational costs would negatively impact the operationalizability of the scoring algorithm, possibly even disrupting the Chronology of Data and Goal if the matrices become big enough. It is also philosophically inelegant

because this would in essence give us a new set of training set fixed and random effect estimates that are incorporating information from the validation set.

The approach in JMP is to subtract the training set mean function from the validation set response values, and then apply the Henderson equations. This assumes that the response values have mean zero and there are no fixed effects to be estimated,

$$\hat{\gamma}_{\text{Val}} = \left(Z_{\text{Val}}^T Z_{\text{Val}} + G\left(\hat{\sigma}_{\text{Tr}}^2\right)\right)^{-1} Z_{\text{Val}}^T \left(y_{\text{Val}} - X_{\text{Val}}\hat{\beta}_{\text{Tr}}\right).$$

The validation set random effects estimates, $\hat{\gamma}_{Val}$, are then rotated using the loadings from the FPCs analysis of the training set. In this way, when we are applying machine learning algorithms to the validation set getting FPC scores, we are mimicking how the model would be scored on new data, and our independent assessment of goodness of fit is preserved by not leaking data from the training set into the scoring of the validation data, except via the estimated model parameters. This functional model scoring approach was developed with an eye towards keeping the generalizability of the conclusions drawn from the validation set as high as possible.