Statistical Analysis and Data Mining: An ASA Data Science Journal

WILEY

RESEARCH ARTICLE OPEN ACCESS

Same/Other/All K-Fold Cross-Validation for Estimating Similarity of Patterns in Data Subsets

Toby Dylan Hocking[1] | Gabrielle Thibault[2] | Cameron Scott Bodine[3] | Paul Nelson Arellano[3] | Alexander F. Shenkin[3] | Olivia Jasmine Lindly[3]

[1]Université de Sherbrooke, Quebec, Canada | [2]Université Laval, Quebec, Canada | [3]Northern Arizona University, Flagstaff, Arizona, USA

Correspondence: Toby Dylan Hocking (toby.dylan.hocking@usherbrooke.ca)

Received: 15 March 2025 | **Revised:** 28 November 2025 | **Accepted:** 20 December 2025

Keywords: cross-validation | domain adaptation | machine learning | subsets | transfer learning

ABSTRACT

In many real-world applications of machine learning, we are interested to know if it is possible to train on the data that we have gathered so far, and obtain accurate predictions on a new test data subset that is qualitatively different in some respect (time period, geographic region, etc.). Another question is whether data subsets are similar enough so that it is beneficial to combine subsets during model training. We propose SOAK, Same/Other/All K-fold cross-validation, a new method which can be used to answer both questions. SOAK systematically compares models which are trained on different subsets of data, and then used for prediction on a fixed test subset, to estimate the similarity of learnable/predictable patterns in data subsets. We show results of using SOAK on six new real data sets (with geographic/temporal subsets, to check if predictions are accurate on new subsets), 3 image pair data sets (subsets are different image types, to check that we get smaller prediction error on similar images), and 11 benchmark data sets with predefined train/test splits (to check similarity of predefined splits).

1 | Introduction

A fundamental assumption in supervised learning is similarity between the train data (input to the learning algorithm) and test data (used to evaluate prediction accuracy of the learned model). This assumption is known as "independent and identically distributed" (i.i.d.) in statistics [1]. Although special modifications to supervised learning algorithms can guarantee accurate predictions in other scenarios such as covariate shift [2], this paper focuses on standard supervised learning algorithms, designed for i.i.d. data. Real-world applications of such supervised learning algorithms often involve training/predicting on data subsets which are qualitatively different in some respect (time period, geographic region, data source, etc.). The main contribution of this paper is a new method that allows us to quantify if these data subsets are similar enough for accurate learning and prediction.

Motivation for comparing models trained on Same/Other/All subset(s) as test subset. For example, in image segmentation, subsets are regions of the image. We would like to train on several labeled regions and predict accurately on a new region. In this paper, we use the name Other for this model because it is trained on other subsets of data (relative to the new/test region, see Figure 1). We would like to quantitatively determine if those Other data subsets are useful for training a model for accurate prediction on the new subset relative to a baseline Same model trained using labeled data from the new subset. Ideally, these Other model predictions on a new subset would be just as accurate as "Same" model predictions (if learnable/predictable patterns are similar

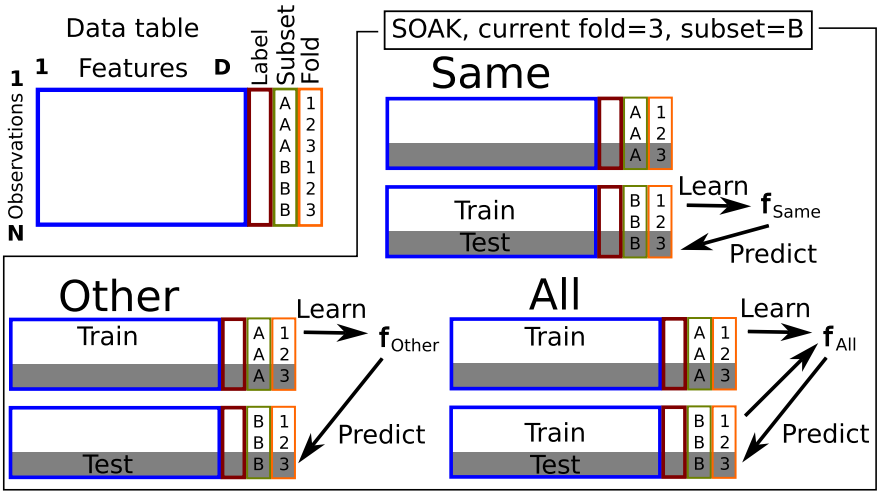


FIGURE 1 | SOAK (Same/Other/All K-fold CV) requires adding subset/fold columns to the data (upper left). For one iteration of SOAK train/test splits (black box, lower right), current test subset = B, so Same = B/Other = A/All = A + B are the values of subset which are used to define the train set, in combination with the current fold = 3, so test sets shown have subset = B and fold = 3, Same train set has subset = B and fold ∈ {1, 2}, and so forth.

in data subsets) or even more accurate (if the Other model has access to more training data than the Same model).

Another real-world example is in learning factors for predicting childhood autism, for which we propose a benchmark with two subsets, one for each year of survey data (2019 and 2020). We would like to know if it is beneficial to train using All data subsets (Figure 1), because if the subsets are similar enough, then we expect that the larger data set allows the learning algorithm to detect more subtle factors associated with autism (that were not possible to detect by training on any year/subset alone). Conversely, if subsets are very different, then we expect that training on All subsets is detrimental to prediction accuracy (relative to training using the Same subset as the test subset). Ideally, these All model predictions on any given subset would be more accurate than Same model predictions (if patterns in subsets are similar, and each subset is too small when used by itself for training).

Contributions and novelty. In this paper, we propose a new algorithm, Same/Other/All K-fold cross-validation (SOAK), which can be used to compare prediction accuracy of models trained on different data subsets. SOAK is a generalization of standard K-fold cross-validation, which corresponds to the special case of SOAK with only one subset, so only the "Same" subset as test is used for training. Although single train/test splits are commonly used for quantifying prediction error on a new subset of data (the "Other" model), this method is limited because it only yields a single measurement of prediction error. Our proposed SOAK algorithm is novel because it combines the idea of K-fold cross-validation with the idea of training/predicting on qualitatively different data subsets. SOAK can therefore be used with rigorous statistical tests of significance (e.g., T-test of prediction error rate between models trained on Same/Other), which are not possible to use with single train/test splits. There was no existing free software implementation of the proposed SOAK algorithm, so we provide one in the mlr3 framework: https://github.com/tdhock/mlr3resampling. After reviewing related work in Section 2, we describe the SOAK algorithm in Section 3 and details of 20 data sets (Table 1). In Section 4, we show results of using SOAK to estimate similarity/differences between subsets in 6 new real data sets (subsets are geographic/temporal), 11 benchmark data sets with predefined train and test subsets (which are treated as two SOAK subsets), and 3 image data sets (each has two subsets, MNIST and either EMNIST or FashionMNIST).

2 | Related Work

Cross-validation is a standard method in machine learning, that is widely used, and discussed in several textbooks [1, 3–5]. The first use of cross-validation was in the context of regularized linear models Larson [6], with the term "cross-validitory" appearing later [7], along with the idea of averaging several empirical estimators [8]. To clearly discuss the different data sets involved in cross-validation, we use the terms full, train, test, subtrain, and validation [9]. The full data set is split into train/test sets (for evaluation), and then the train set is further split into subtrain/validation sets (for hyper-parameter learning). K-fold cross-validation can be used for either type of split, and involves partitioning the data into K disjoint test (or validation) sets; for each, the train (or subtrain) set is defined as all the other data. A primary use of cross-validation is model selection (splitting the train set into subtrain/validation sets), to avoid overfitting during a learning algorithm [10, 11]. In contrast, our proposed method is primarily useful for splitting the full data set into train/test sets [12], in order to quantify the prediction error/accuracy on new/test data that were never seen during learning. Standard K-fold cross-validation can be used for that purpose, and yields K measurements of test error/accuracy that can be useful for comparing the prediction accuracy of different algorithms. An alternative is to use a single train/test split, with one subset for train, and another for test; while this approach is somewhat common in the machine learning literature, it only yields one test error/accuracy number, so it can be a misleading estimate of prediction error/accuracy, that tends to encourage overfitting [4, 13]. In contrast, our proposed SOAK method is based on K-fold CV,

TABLE 1 | Meta-data, one row per data set that we analyzed using the proposed SOAK algorithm.

	Subset type	Data	Rows	Features	Classes	Class imb.	Subsets	Subset imb.
1	ImagePair	IPair_E	140,000	784	10	1.1	2	1.0
2	ImagePair	IPair_E_rot	140,000	784	10	1.1	2	1.0
3	ImagePair	IPair_Fashion	140,000	784	10	1.1	2	1.0
4	time/space	CanadaFiresA	4827	46	2	2.0	4	7.0
5	time/space	CanadaFiresD	1491	46	2	1.5	4	1.6
6	time/space	FishSonar_river	2,815,744	81	2	3.2	4	1.2
7	time/space	NSCH_autism	46,010	364	2	31.8	2	1.5
8	time/space	aztrees3	5956	21	2	7.8	3	2.0
9	time/space	aztrees4	5956	21	2	7.8	4	4.9
10	train/test	CIFAR10	60,000	3072	10	1.0	2	5.0
11	train/test	EMNIST	70,000	784	10	1.0	2	6.0
12	train/test	FashionMNIST	70,000	784	10	1.0	2	6.0
13	train/test	KMNIST	70,000	784	10	1.0	2	6.0
14	train/test	MNIST	70,000	784	10	1.2	2	6.0
15	train/test	QMNIST	120,000	784	10	1.2	2	1.0
16	train/test	STL10	13,000	27,648	10	1.0	2	1.6
17	train/test	spam	4601	57	2	1.5	2	2.0
18	train/test	vowel	990	10	11	1.0	2	1.1
19	train/test	waveform	800	21	3	1.1	2	1.7
20	train/test	zipUSPS	9298	256	10	2.2	2	3.6

Note: Imb. = Imbalance ratio between largest/smallest class or subset (1 = balanced, larger = more imbalance).

so yields K test error/accuracy numbers, and can be used with statistical tests of significance.

Distributional Homogeneity. The cross-validation method that we propose is related to the statistical concepts of independent and identically distributed (i.i.d.) random variables, and of homogeneity in meta-analyses, meaning that different subsets of the data follow the same distribution. Homogeneity/i.i.d. is a stronger condition than we are interested in measuring using our proposed cross-validation procedure (it may be beneficial to combine subsets when learning, even though they are heterogeneous). Classic examples of statistical tests for homogeneity are the Chi-Square test of Pearson [14] and the Q test of Cochran [15]. In meta-analysis, the goal is to provide a better estimate of a quantity measured in several different studies, and there are several methods available for estimating heterogeneity [16]. The ReDistribution test of homogeneity has been used for climate time series data [17], and is related to the study of change-point detection algorithms [18].

Fairness/bias. Our proposed SOAK method has an obvious application to determining if a trained model is fair or biased across demographic categories such as race, gender, or age (which could be used as SOAK subsets). However, most papers about algorithmic bias do not mention cross-validation explicitly. For example, West [19] examined how gender discrimination in AI workplaces has influenced AI systems' failures to see and hear women, and proposed that it may not be desirable to make AI systems "work better" for minorities. The author questions the definition of "algorithmic fairness" as a computational problem, which can be identified and mitigated, yielding prediction accuracy rates that are similar across demographic categories. There are many algorithms which can be used for predicting sex or gender [20]. Wilming et al. [21] proposed the GECO data set along with a framework for evaluating gender bias in explanations, in the context of large language models. Hall et al. [22] proposed the VisoGender data set for benchmarking gender bias in image-text pronoun resolution (690 images with paired captions, balanced gender characteristics). Currey et al. [23] proposed the MT-GenEval data set for evaluating gender bias in machine translation. Examples are counterfactual sentences, with subsets for Profession (Male, Female, Neutral), and for Person Gender (Male or Female). Other machine translation data for benchmarking gender bias include WinoMT [24], MuST-SHE [25] and others [26]. Existing software includes FairLearn [27, 28], which implements a reductions approach that is different from our proposed approach of comparing test error rates of Same and Other models.

Domain adaptation and transfer learning. Domain adaptation is the idea that the learning algorithm should adapt based on some known difference between the train and test sets. In this literature, SOAK subsets are known as "domains." Domain adaptation can be considered a special case of transfer learning (i.e., inductive transfer learning), where the data distributions are not the same, but the prediction task is the same, between the source/train and target/test subsets. A review of domain adaptation methods for machine translation refers to the Other model as "out-of-domain" prediction [29]. Farahani et al. [30] refer to the Other model as "unsupervised domain adaptation" because there are no labels available in the target domain.

Judge et al. [31] performed computational experiments using the Other model, but did compare with error rates from the Same model. In computer vision, domain adaptation has been called "out-of-distribution" prediction; Madan et al. [32] found that convolutional neural networks were able to do this to some extent (on new position/rotation/scale values), by increasing training data diversity. In that context, the SOAK Same model is referred to as "in-distribution" training, whereas the Other model is referred to as "out-of-distribution" (OOD) training. Spatial cross-validation is used in geographical data analysis [33], and can be viewed as the "Other" model in SOAK. Whereas these previous studies focus on the Other model (without necessarily comparing with the Same model), our proposed SOAK algorithm additionally trains the All model, and emphasizes a comparison with the Same model (using rigorous tests of statistical significance).

Novelty with respect to available software. A primary contribution of our paper is a free/open-source software framework which makes it easy to run SOAK with different data sets and learning algorithms. Although the Same/Other comparison is sometimes used in domain adaptation and computer vision research [29, 30, 32], there was no corresponding free software framework until our work on SOAK. There are many free/open-source implementations of cross-validation, including origami [34], splitTools [35], and mlr3 [36] in R, as well as scikit-learn in python [37]. All of these packages support standard K-fold cross-validation, and some support stratification and keeping groups of observations together when splitting (a concept/parameter called "group"). The proposed SOAK algorithm is based on the concept of data subsets, which was not previously supported in any free software machine learning framework, so we implemented it in the mlr3 framework [38] because it provides flexible support for parallelization over data sets, algorithms, and train/test splits (including parallelization over the proposed subsets).

Proposed subset concept is distinct from previous group concept. The "group" concept (observations/rows in the same group must be assigned the same fold ID) is present in both mlr3 and in scikit-learn, and the "group" concept is different from the SOAK "subset" concept (one subset is designated as test, and Same/Other/All subsets are designated as train), but actually the two concepts can be used together. For example, in image segmentation, the goal is to classify each pixel of an image. Labels are often created by drawing polygons on the image, and in each polygon there are several pixels which are assigned the same label. If each polygon is considered a group, then each pixel in a polygon gets the same group ID. Image(s) may be divided into regions, such that each pixel to classify is assigned to a distinct subset. For example, in satellite image analysis, as in the proposed aztrees data (Section 3.4), subsets may be defined as North/South/etc. In this example, we can use both concepts at the same time: let labeled pixels from polygons/groups in the South region/subset be the test set, and define the train set as labeled pixels from polygons/groups in the Same/Other/All region/subset. In datasets with imbalanced labels, stratification may also be used in combination with groups and subsets, in order to ensure that train and test sets are down-sampled proportionally. Our proposed free/open-source software implementation of SOAK (https://github.com/tdhock/mlr3resampling) provides support for using all three concepts together: strata, groups, subsets.

3 | Methods: Proposed SOAK Algorithm and Data Sets

In this section, we give details of the proposed SOAK algorithm, and then give details about the 20 data sets we analyzed (Table 1), which represent classification problems with 800–2,816,744 rows, 10–27,648 features, 2–11 classes, and 2–4 subsets.

3.1 | Proposed SOAK Algorithm

We propose SOAK (Same/Other/All K-fold CV), a new variant of cross-validation which is useful for determining similarity between learnable and predictable patterns in data subsets. As in standard K-fold cross-validation for supervised machine learning, we assume there is a data set with N observations/rows. Additionally, we assume that the rows can be partitioned into a certain number of subsets S, and we would like to estimate the similarity of learnable/predictable patterns these subsets. Each subset has an identifier, which we treat here as a category represented by an integer from 1 to S. For example, in Figure 1, there are $S = 2$ subsets, A and B, which can be visualized as a new column alongside the features and labels. Note that in our experiments, the subset column is was not included as a feature for learning and prediction (but it could be included, if desired). For each observation/row i, we assume there is a corresponding subset $s_i \in \{1, \ldots, S\}$ and fold ID $k_i \in \{1, \ldots, K\}$ (assigned randomly or using strata, such that each label/subset has about the same number of rows with a given fold ID). For example, in the NCSH_autism data set (Section 3.4), there are two temporal subsets, one for each year ($s_i = 1$ for 2019 and $s_i = 2$ for 2020).

The goal of SOAK is to estimate the prediction error or accuracy of a learning algorithm, when attempting to predict on a given subset, and training on that Same subset, or on different subsets (Others or All). Therefore, our method has a loop over each subset $\sigma \in \{1, \ldots, S\}$ and fold $\kappa \in \{1, \ldots, K\}$. For each, we define the train/test splits as in Figure 1.

Test: set is $\{i \in \{1, \ldots, N\} | k_i = \kappa \text{ and } s_i = \sigma\}$, all rows i in the current fold κ and subset σ.

Same: train set is $\{i \in \{1, \ldots, N\} | k_i \neq \kappa \text{ and } s_i = \sigma\}$, all rows i not in the current fold κ, but in the current subset σ.

Other: train set is $\{i \in \{1, \ldots, N\} | k_i \neq \kappa \text{ and } s_i \neq \sigma\}$, all rows i which are neither in the current fold κ, nor in the current subset σ.

All: train set is $\{i \in \{1, \ldots, N\} | k_i \neq \kappa\}$, all rows i not in the current fold κ.

SOAK runs learning algorithm(s) on Same/Other/All train sets, then computes the resulting predictions on the Test set for this subset σ and fold κ. This is repeated for each fold κ, yielding K measures of test error or accuracy for each train set (Same/Other/All), when predicting on a given subset σ. Although other significance tests have been proposed [39, 40], we propose for simplicity to quantitatively compare test accuracy or error values using a two-sided paired t-test with $K - 1$ degrees of freedom:

Same versus Other: if test accuracy for Other is comparable or better than Same, then we can conclude that the learning algorithm can be trained on other subsets and accurately predict on this subset. But if test accuracy for Other is worse than Same, then we can conclude that the learning algorithm is not able to train on other subsets and accurately predict on this subset.

Same versus All: if test accuracy for All is significantly better than Same, then we can conclude that combining subsets is beneficial to the learning algorithm when predicting in this subset. But if test accuracy for All is significantly worse than Same, then we can conclude that combining subsets is detrimental to the learning algorithm when predicting in this subset.

For each subset σ, the final output to analyze is the corresponding t-test result (mean error differences and p-values, for Same vs. Other and Same vs. All).

Mathematical interpretation. A supervised learning algorithm inputs a train data set and produces a function f_{train}, which typically aims to output a predicted label $\hat{y} = f_{\text{train}}(x)$ with minimal loss L for the given input x. Classical K-fold cross-validation provides an empirical estimator of the expected loss L (or accuracy, AUC, etc.), assuming test samples (x, y) are drawn from the same distribution D as train: $E_{\text{test}\sim D}\left[L\left(y, f_{\text{train}\sim D}(x)\right)\right]$ The proposed SOAK algorithm involves a subset variable s, which is used to condition the train and test distributions D. For a fixed test subset σ,

- The Same train/test split estimates $E_{\text{test}\sim D|s=\sigma}\left[L\left(y, f_{\text{train}\sim D|s=\sigma}(x)\right)\right]$, the expected loss when training and predicting on subset σ.
- The Other train/test split estimates $E_{\text{test}\sim D|s=\sigma}\left[L\left(y, f_{\text{train}\sim D|s\neq\sigma}(x)\right)\right]$, the expected loss when predicting on subset σ, and training on other subsets.
- The All train/test split estimates $E_{\text{test}\sim D|s=\sigma}\left[L\left(y, f_{\text{train}\sim D}(x)\right)\right]$, the expected loss when predicting on subset σ, and training on all subsets.

Standard CV is a special case. Note that standard K-fold CV is the special case with $S = 1$ subset, which means that the Other train set is empty, the All train set is identical to Same, and so for each learning algorithm, we have only K test accuracy or error numbers (one for each fold/split).

Computational complexity. For each of S test subsets and K folds, we need to consider training on Same/Other/All subsets, so the number of train/test splits considered by SOAK is $3SK = O(SK)$, which is the number of times each learning algorithm needs to be run (training and prediction). Importantly, this is linear in the number of subsets S, so it is possible to run SOAK on data with a large number of subsets S.

Implementation Details. SOAK can be easily implemented in any programming language, by looping over all subsets $\sigma \in \{1, \ldots, S\}$ and fold IDs $\kappa \in \{1, \ldots, K\}$. We implemented the computational experiments in this paper using the mlr3 framework in R, which made it easy to compute results in parallel (over all algorithms, data sets, and train/test splits) using the mlr3batchmark package [41].

3.2 | Image Pairs: Train on MNIST, Predict on EMNIST or FashionMNIST

The MNIST data set consists of 70,000 images of handwritten digits, and the goal of learning is to accurately classify each image into one of 10 classes (0–9) [42]. EMNIST is another set of 70,000 images, also of handwritten digits, with balanced classes (see Table 1, column class Imb.), and a different pre-processing technique that attempts to scale images to fill the available space [43]. FashionMNIST is a set of 70,000 images, with 7000 examples of each of 10 classes of clothing [44]. Intuitively, we expect that we should be able to train on MNIST (images of digits) and get reasonable predictions on EMNIST (because images are also digits), but not on FashionMNIST (because images are clothing). Therefore, we created three data sets (Table 1), each with 14,000 images, by combining MNIST with one of the other variants.

IPair_E: MNIST combined with EMNIST. Note that the raw EMNIST images are not in the same orientation as MNIST (if MNIST is upright, then EMNIST appears to be rotated 90˚).

IPair_E_rot: MNIST combined with rotated EMNIST (all images in upright orientation).

IPair_Fashion: MNIST combined with FashionMNIST.

In each of the three data sets, there are two subsets (one from each source: MNIST and EMNIST or FashionMNIST).

3.3 | Benchmark Data With a Predefined Train/Test Split

There are many benchmark data sets in the machine learning literature that include a column to designate a predefined train/test split. In this paper, we consider several data sets which were downloaded using torchvision [45]: CIFAR10 [46], EMNIST [43], FashionMNIST [44], KMNIST [47], MNIST [42], QMNIST [48], STL10 [49]; several others that are included as supplementary materials in the textbook of Hastie et al. [1] (spam, vowel, waveform), and one data set that was included in both (USPS in torch, and zip in the textbook, which we call zipUSPS in this paper). Rather than using the predefined train/test split for its intended purpose, we instead use it as a subset ID, so each of these data sets has two subsets (Table 1). By using SOAK on these data, we seek to answer the question: are the predefined train and test subsets similar enough, so that if we train on one subset (either predefined = train or test), we can get predictions on the other subset that are just as accurate as if we had access to the same subset?

3.4 | Real Data From Various Application Domains With Spatial/Temporal Subsets

CanadaFires data consist of four satellite images of forest fires, in which we are primarily interested to answer the question: can we train on some images and accurately predict burned areas in a new image? (subsets are images of different forest fires) Characterizing the burning pattern using these high-resolution Skysat images is important because it is used by the Québec government to plan salvage operations. Skysat images of forest fires

that occurred in 2020–2021 were used. In addition, Landsat were obtained using Google Earth Engine [50]. Each row/observation in these data is a labeled pixel, and there are columns/features for Normalized Burn Ratio [51], delta Normalized Burn Ratio [52], Normalized Difference Vegetation Index [53], Green Chromatic Coordinates [54], and so forth, for a total of 46 features. Labels were created for individual pixels, each at least 100 m apart, and more than 6 m from the image borders, by manually assigning one of six burn classes, based on a classification that the Québec government uses to characterize fires. We then transformed the labels to a binary problem, burned (positive class) versus other (negative class). There are two versions of these data: A means All data, and D means Down-sampled to promote class balance, while retaining representative examples. In these data, we are interested to see if it is possible to train on a few images/fires and accurately predict on a new image/fire (there are four such images/subsets).

aztrees: data come from satellite images around Flagstaff, AZ, in which we are primarily interested to answer this question: can we train on some regions, and accurately predict presence of trees in new regions? (subsets are different regions in the satellite image) The goal is to predict presence/absence of trees throughout Arizona, in a project that seeks to determine the extent to which trees are under stress/drought/bark beetle infestation. Three sets of satellite images were retrieved from Google Earth Engine: Sentinel-1, Sentinel-2, and the NASA Shuttle Radar Topography Mission (SRTM) Global image [50, 55]. Data were converted so that each row is a pixel, and 20 features/columns were computed, including several spectral bands, Normalized Difference Vegetation Index-NDVI [56], Normalized Difference Infrared Index-NDII [57], Normalized Difference Water Index-NDWI [58], three Topographic Position-TPI Indexes [59]. Images were manually labeled using Quantum GIS software [60] by drawing polygons that indicate seven Land Use/Land Cover classes (trees, natural grass, planted grass, infrastructure, bare ground, and open water), then labels were converted into a binary problem (tree vs. other). There are two variants of these data, with either 3 (S/NE/NW) or 4 (SE/SW/NE/NW) geographic subsets, and we would like to know if it is possible to train on some subsets, and accurately predict on a new subset.

FishSonar_river: data come from sonar imagery of river bottoms, in which we are primarily interested to answer this question: can we train on a few rivers and accurately predict areas suitable for fish spawning on a new river? The goal is to accurately predict spawning areas of Gulf Sturgeon (*Acispenser oxyrinchus desotoi*) in a wildlife conservation project [61]. In detail, data come from Humminbird fish finder sonar from surveys in the Pearl and Pascagoula watersheds in Mississippi, USA over 2021–2023, then processed with PING-Mapper software [62–65]. Sonar data give imagery of the river bottom; the goal is to classify each pixel as either suitable (gravel, cobbles, boulders, and bedrock) or not (silt, mud, and sand) for Gulf Sturgeon spawning [66]. Data were converted so that each row is a pixel and each column is a mean over windows around that pixel (mean pooling, window size 9). Multiple class labels were first created using Doodler software [67] and then transformed to obtain a binary classification problem: hard bottom (suitable for spawning) versus anything else. Subsets to train/test on were defined using four different rivers (Bouie, Chickasawhay, Leaf, Pearl), and we are interested to see if models learned on a subset of rivers can be used for accurate prediction on a new river.

NSCH_autism data: consist of two subsets/years (2019 and 2020) from the National Survey of Children's Health (NCSH) [68–70]. The goal of this machine learning analysis is to accurately predict autism diagnosis, using an interpretable model which can identify a subset of questions/responses that are useful for prediction. Each row represents a child for whom an adult familiar with the child's health answered the survey in a given year, and each column represents a question/response with various categories such as mental health diagnoses, family income, health insurance status, healthcare needed and received, neighborhood characteristics, and so forth. There are two subsets/years in these data, and we would like to know if training with combined years results in a more accurate model.

4 | Results: Estimating Differences Between Subsets in 20 Data Sets

In this section, we show how SOAK can be used to quantify similarity/differences of learnable and predictable patterns of the subsets in 20 data sets. We ran experiments in parallel using a compute cluster: one CPU per data set, algorithm, and train/test split (64 GB RAM and 2 days computation time for each).

4.1 | Comparing Prediction Accuracy of 5 Learning Algorithms on 20 Data Sets

First, we used standard 10-fold cross-validation on 20 classification data sets (Table 1) to compare the prediction accuracy of 5 learning algorithms (with hyper-parameters tuned using internal 10-fold cross-validation): cv_glmnet (L1-regularized linear model); featureless (baseline always predicting most frequent class); rpart (default decision tree learner with no hyper-parameter tuning); nearest neighbors (tuned 1–20 neighbors); xgboost (gradient boosting, learning rate `eta` tuned over 5 values on log scale between 0.001 and 1, `nrounds` tuned over 5 values from 1 to 100). From Figure 2, we see that xgboost was always the slowest, and sometimes the most accurate; cv_glmnet had reasonable/intermediate computation time and prediction accuracy, so we used it for the other experiments with SOAK.

4.2 | SOAK Prediction Error Comparison Plots

Next, we used SOAK with 10-fold CV on each of the 20 data sets (Table 1) to compute prediction error on each subset after training cv_glmnet on Same/Other/All subset(s).

Expectation: similarity of predefined train/test subsets. We categorized each of the 20 data sets according to their subset types: ImagePair (Section 3.2), pre-defined train/test (Section 3.3), and time/space (Section 3.4). We expected that some data sets would have very similar subsets (All model better than Same), and others would have very different subsets (Same model better than All). In particular, since train/test splits are typically assigned randomly (due to the i.i.d. assumption), we expected to observe large similarity in the benchmark data sets with predefined

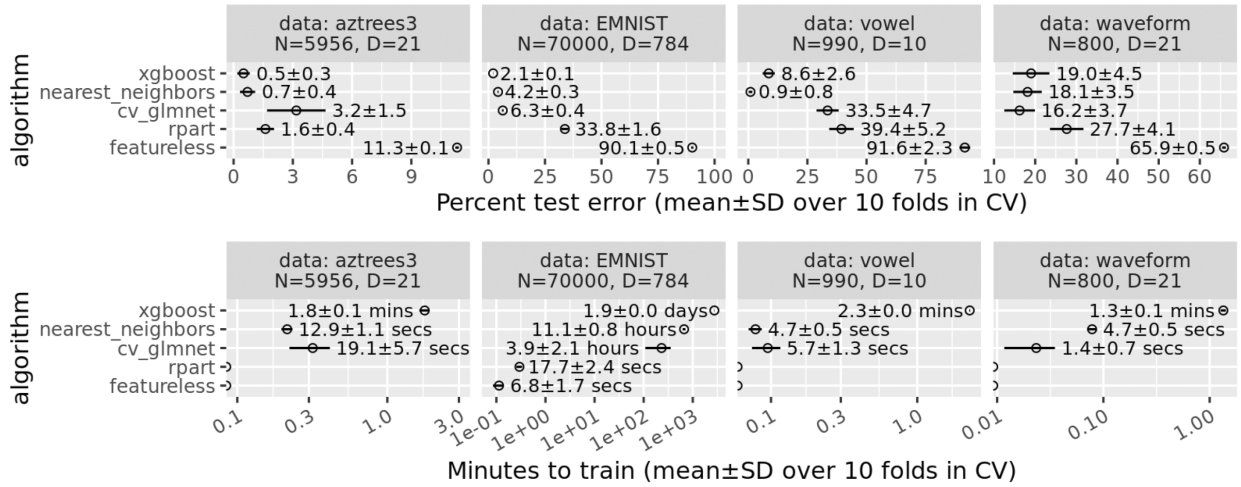


FIGURE 2 | Test error (top) and training time (bottom) of five algorithms on four data sets.

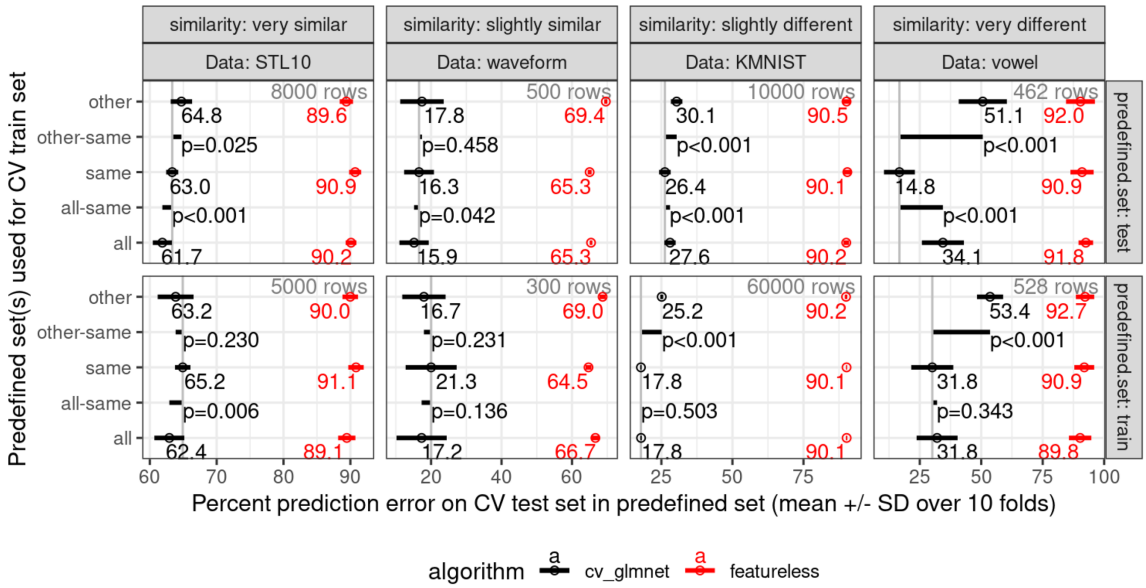


FIGURE 3 | SOAK was used to compute mean/SD of test error over 10 cross-validation folds, and *p*-values for differences (other-same and all-same), in each of four data sets in which there were two subsets (predefined train/test assignments in the data table). For data sets that have similar learnable/predictable patterns (left), training on all subsets has smaller test error than same, and training on other has either smaller or larger test error than same (depending on number of rows in subset). For data sets that have different learnable/predictable patterns (right), training on all subsets never has smaller test error than same, and training on other always has larger test error than same.

train/test subsets, such as in STL10, waveform, KMNIST, and vowel data sets (Figure 3).

Similarity observed between predefined train/test subsets in STL10 and waveform. Two of those data sets (STL10 and waveform) exhibited evidence of similarity between subsets: All test error was smaller than Same test error (STL10 significantly, $p < 0.05$ in two-sided paired t_9-tests; waveform slightly, $p = 0.042$ and 0.136). Additional evidence of similarity between subsets in STL10 was that Other test error was significantly larger or smaller than Same (depending on sample size, predefined train subset had 8000 rows, whereas predefined test subset had only 5000 rows). These data provide evidence that in the STL10 and waveform benchmark data, the predefined train/test splits were created by random assignment.

Differences observed between predefined train/test subsets in KMNIST and vowel. Surprisingly, there were two benchmark data sets (KMNIST and vowel) which exhibited differences between learnable/predictable patterns in predefined train/test subsets. Other test error was significantly larger than Same (mean difference of 5%–25%, $p < 0.001$), indicating that the linear model was not capable of learning on one subset and accurately predicting on the other. All test error was 1%–20% larger than Same when predicting on the smaller subset (predefined test subset, $p < 0.001$), and All test error was not significantly different from Same when predicting on the larger subset (predefined train subset, $p = 0.343$ to 0.503).

Synthesis and conclusions. Figure 3 suggests that two benchmark data sets have similar predefined train/test subsets (STL10

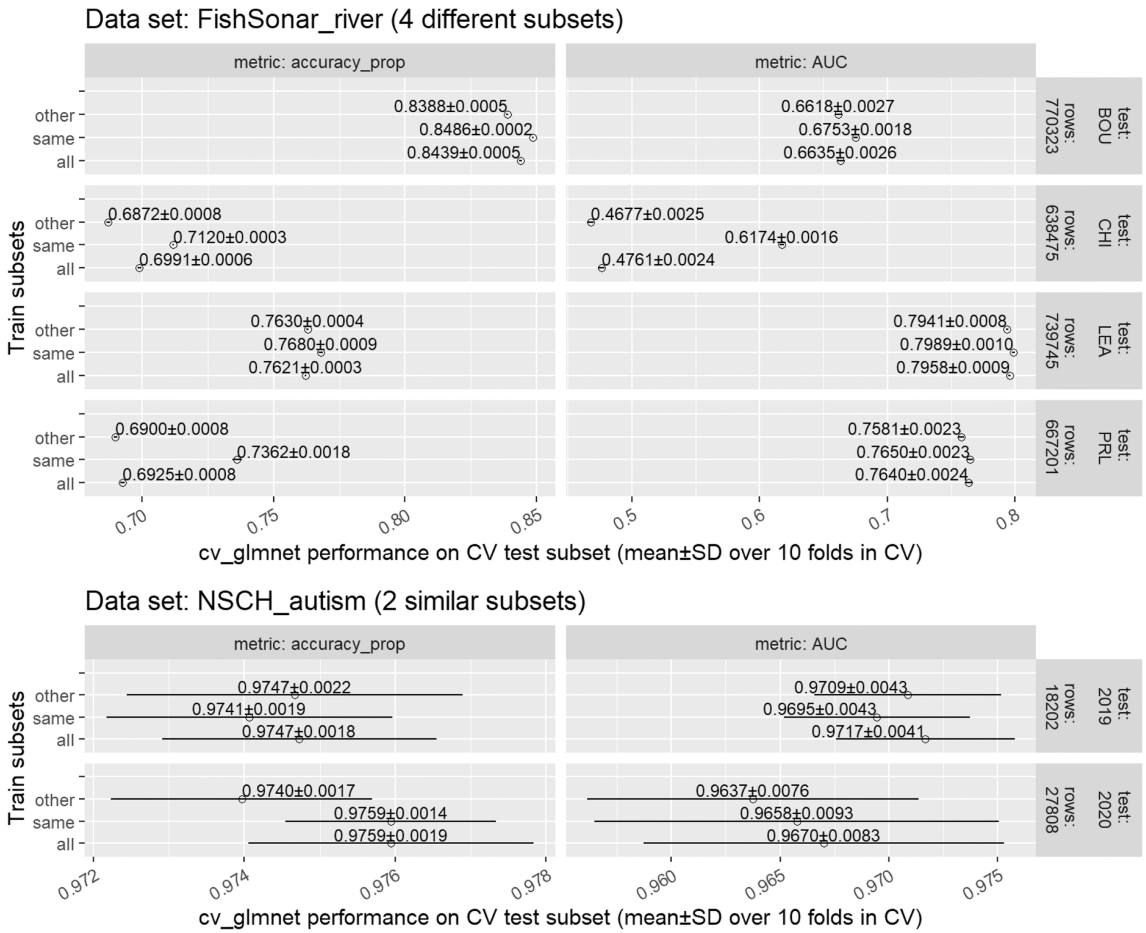


FIGURE 4 | SOAK yields identical conclusions about similarity of subsets, using either proportion accuracy (accuracy_prop, left) or Area Under the ROC Curve (AUC, right) as the performance metric. *Top*: In the FishSonar_river data, training on the Same subset always has larger accuracy/AUC than Other/All, indicating that the cv_glmnet algorithm is unable to generalize between subsets in these data. *Bottom*: In the NSCH_autism data, training on All never yields smaller accuracy/AUC than Same, indicating that the cv_glmnet algorithm is able to generalize between subsets in these data.

and waveform), whereas two others have different predefined train/test subsets (KMNIST and vowel). These results are consistent with the method used to construct the vowel data, which used different speakers for train and test subsets: "Four male and four female speakers were used to train the networks, and the other four male and three female speakers were used for testing the performance" [71]. Because the vowel subsets correspond to different speakers, our results indicate that these speakers produce signals that are too different for the linear model to benefit from combining different speakers when training. These results suggest that for optimal prediction accuracy in the vowel data, different linear models should be trained for each speaker. Overall, it is clear that by running SOAK then plotting Same/Other/All test error (Figure 3), it is possible to estimate the extent of similarity/differences between learnable and predictable patterns in data subsets.

4.3 | Results Consistent Using Accuracy and AUC

Because most data sets we considered had more than two classes (Table 1), we performed most of our analyses using the evaluation metric defined as percent incorrectly predicted labels in the test set, which is applicable for data sets with any number of classes. For data sets with only two classes, we also performed analyses using the evaluation metric defined as the Area Under the ROC Curve (AUC), which is only applicable in data sets with two classes. Our hypothesis was that in data sets with large class imbalance, SOAK may yield different conclusions about whether or not subsets are similar, using different evaluation metrics. To test this hypothesis, we used SOAK with two evaluation metrics for the binary data sets: accuracy rate or AUC. Contrary to our hypothesis, we observed that the relative ranking of Same/All/Other models was preserved across the two evaluation metrics, for the vast majority of test subsets. For example, Figure 4 shows results on two representative data sets: NSCH_autism and FishSonar_river. In the FishSonar_river data, training on the Same subset always has larger accuracy/AUC than Other/All, indicating that the cv_glmnet algorithm is unable to generalize between subsets in these data. When predicting on 2020 in NSCH_autism data, we observed All = Same accuracy rates of 0.9759; AUC values were slightly larger for All (0.9670) compared to Same (0.9658). Results using both AUC and accuracy are therefore compatible with the hypothesis that in the NSCH_autism data, the cv_glmnet algorithm is able to generalize between subsets in these data (All accuracy/AUC is never smaller than Same). Overall, these results suggest that SOAK yields similar

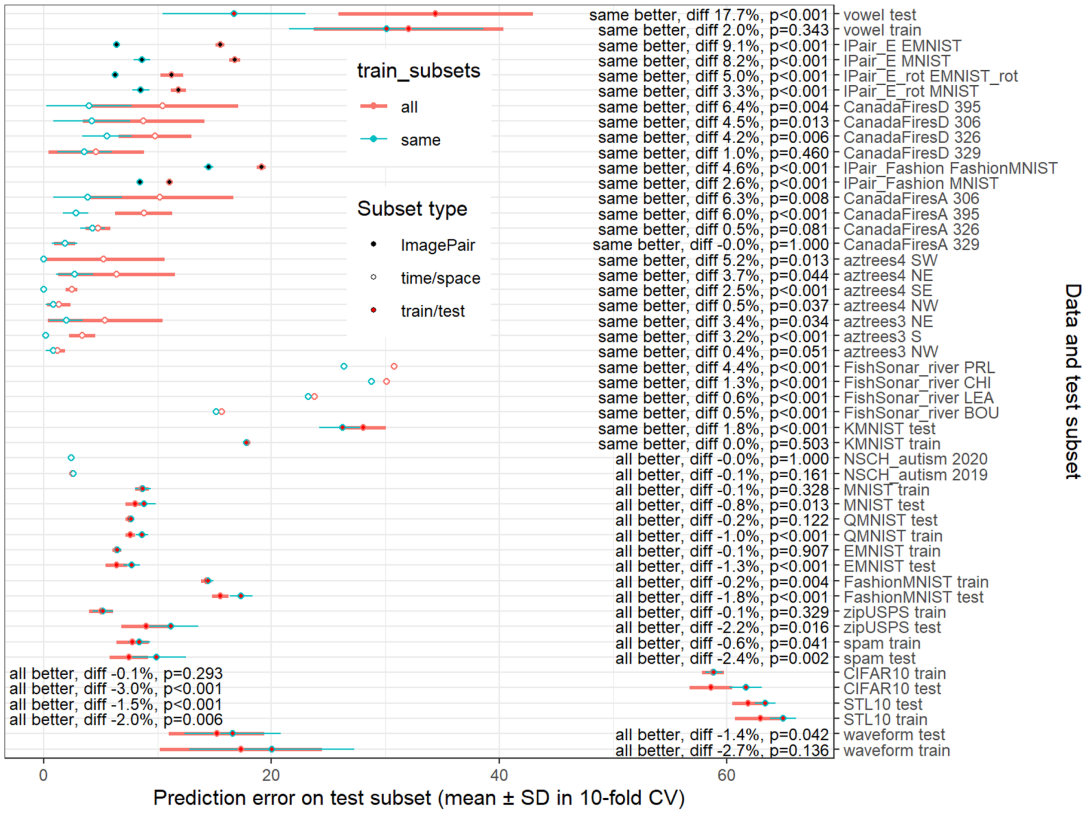


FIGURE 5 | Comparing test error when training on Same or All subsets. Mean and standard deviation of test error are shown as dots/segments, along with text showing difference between means (All-Same) and *p*-value in two-sided paired t_9-test, for each test subset (sorted by mean test error difference per data set).

conclusions about similarity of subsets, using either accuracy rate or AUC as the performance metric. In data sets with two classes, AUC should be the preferred evaluation metric in order to properly handle class imbalance.

4.4 | SOAK Summary Plots of Test Error Differences and *p*-Values

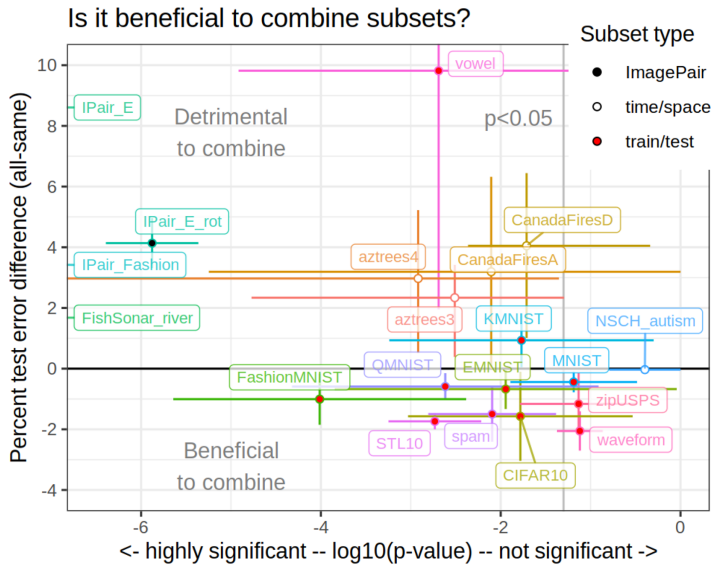
After running SOAK on each of the 20 data sets using the cv_glmnet regularized linear model, we wanted to create visual summaries of results across all data sets and test subsets. First, we plotted mean and standard deviation of test error over the 10 cross-validation folds, for each test subset (Figure 5). This plot compares the test error rates when training on Same and All subsets, and it is clear that Same is better in some data sets (top), whereas All is better in others (bottom). The data sets with the largest advantages for Same were vowel, image pair (IPair_*) and Canada Fires—these are the data sets for which training the linear model on All combined subsets was detrimental to prediction accuracy (with respect to training on only the Same subset). The data sets with the largest advantages for All were waveform, STL10, and CIFAR10—these are the data sets for which training the linear model on All combined subsets was beneficial to prediction accuracy (with respect to training on only the Same subset). Overall, this analysis shows that combining subsets when training the linear model is only beneficial in some data sets (and is detrimental in the others).

Next, we plotted test error differences (All-Same, Figure 6; Other-Same, Figure 7), and *p*-values (two-sided paired t_9-test because we used 10-fold CV). Each plot/table shows a line/row representing the min/max test error difference and *p*-value, over the 2–4 subsets in each data set. When analyzing All-Same test error differences (Figure 6), it is clear that the data sets can be divided into two categories: 10 data sets have similar subsets (negative test error differences, All-Same) and the other 10 have different subsets (zero or positive test error differences). Similarly when analyzing Other-Same test error differences (Figure 7), we obtain the same categorization of 10 data sets with similar subsets (min Other-Same test error difference negative, max positive), and 10 data sets with different subsets (min/max test error difference both positive). As expected, real data sets with space/time subsets, and ImagePair data, tended to have different subsets, whereas data with predefined train/test subsets were mostly similar (exceptions were NSCH_autism real data with slight similarity of subsets/years, and vowel/KMNIST with different predefined train/test subsets). Overall, these SOAK scatterplots/tables allow categorization of data sets as having subsets with either similar or different learnable/predictable patterns.

5 | Discussion and Conclusions

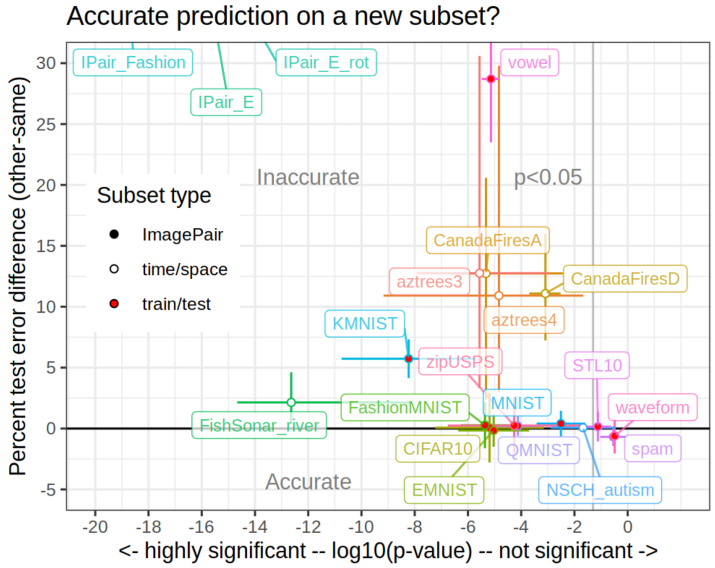
5.1 | Image Pairs: MNIST Data Combined With EMNIST/FashionMNIST

Our goal in analyzing these three data sets (explained in Section 3.2) was to quantify our intuition that MNIST/EMNIST



Data	ErrorDiff	log10(P)
● IPair_E	8.2, 9.1	-11.2,-10.2
● IPair_E_rot	3.3, 5.0	-6.4, -5.4
● IPair_Fashion	2.6, 4.6	-12.4, -8.6
● vowel	2.0, 17.7	-4.9, -0.5
○ CanadaFiresD	1.0, 6.4	-2.4, -0.3
○ aztrees4	0.5, 5.2	-7.0, -1.4
○ FishSonar_river	0.5, 4.4	-14.4, -8.6
○ aztrees3	0.4, 3.4	-4.8, -1.3
● KMNIST	0.0, 1.8	-3.2, -0.3
○ CanadaFiresA	-0.0, 6.3	-5.2, 0.0
○ NSCH_autism	-0.1, -0.0	-0.8, -0.0
● MNIST	-0.8, -0.1	-1.9, -0.5
● QMNIST	-1.0, -0.2	-4.3, -0.9
● EMNIST	-1.3, -0.0	-3.8, -0.0
● FashionMNIST	-1.9, -0.2	-5.6, -2.4
● STL10	-2.0, -1.5	-3.2, -2.2
● zipUSPS	-2.2, -0.1	-1.8, -0.5
● spam	-2.4, -0.6	-2.8, -1.4
● waveform	-2.7, -1.4	-1.4, -0.9
● CIFAR10	-3.0, -0.1	-3.0, -0.5

FIGURE 6 | SOAK was used to compute mean test error differences (All-Same) and *p*-values for each test subset, over 10 cross-validation folds. Line segments and table show min/max values over 2–4 test subsets in each data set; dot shows mean. Horizontal black line separates data sets by the degree of differences in learnable/predictable patterns: Top 10 for large differences (min/max ErrorDiff positive or zero: Never beneficial to combine subsets when training) and bottom 10 for small differences (min/max ErrorDiff negative or zero: Never detrimental to combine subsets).



Data	ErrorDiff	log10(P)
● IPair_Fashion	77.5, 84.7	-21.7,-17.8
● IPair_E	73.2, 74.7	-19.2,-18.3
● IPair_E_rot	36.7, 38.8	-15.3,-15.1
● vowel	23.5, 33.9	-5.5, -4.8
○ CanadaFiresD	7.2, 16.0	-3.7, -2.5
● KMNIST	4.1, 7.3	-10.7, -5.7
○ aztrees3	3.3, 30.6	-7.9, -2.9
○ aztrees4	3.0, 29.8	-9.2, -1.7
○ CanadaFiresA	2.4, 20.6	-6.7, -2.4
○ FishSonar_river	0.5, 4.6	-14.7, -8.1
○ NSCH_autism	-0.1, 0.2	-2.9, -0.5
● QMNIST	-0.6, 1.0	-4.9, -3.3
● MNIST	-0.6, 1.5	-3.4, -1.6
● STL10	-1.1, 1.4	-1.6, -0.6
● spam	-1.4, 0.1	-1.0, -0.0
● EMNIST	-1.5, 1.2	-6.4, -3.7
● FashionMNIST	-1.6, 2.2	-6.3, -4.5
● waveform	-2.0, 0.8	-0.6, -0.3
● zipUSPS	-2.1, 2.6	-6.8, -1.8
● CIFAR10	-2.8, 2.9	-7.2, -3.1

FIGURE 7 | SOAK was used to compute mean test error differences (Other-Same) and *p*-values for each test subset, over 10 cross-validation folds. Line segments and table show min/max values over 2–4 test subsets in each data set; dot shows mean. Horizontal black line separates data sets by the degree of differences in learnable/predictable patterns: Top 10 for large differences (min/max ErrorDiff both positive: Inaccurate prediction on all new subsets) and bottom 10 for small differences (min ErrorDiff negative, max positive: Accurate prediction on at least one new subset).

are more similar than MNIST/FashionMNIST. We expected that the worst prediction accuracy should be in IPair_Fashion, because the subsets are most different (images of digits/clothing), and that is what we observed (Figure 7, Other-Same test error differences of 77.5%–84.7%, $p < 10^{-17}$). We expected an intermediate prediction accuracy for IPair_E, because both subsets are images of digits (although not in the same orientation), and that is what we observed (Figure 7, Other-Same test error differences of 73.2%–74.7%, $p < 10^{-18}$). We expected best prediction accuracy for IPair_E_rot, because both subsets are images of digits (in the same orientation), and that is what we observed (Figure 7, Other-Same test error differences of 36.7%–38.8%, $p < 10^{-15}$). Overall, our SOAK analysis indicates that after training on MNIST, the linear model predictions on EMNIST are better than on FashionMNIST, but still not as good as prediction on MNIST (positive Other-Same differences in Figure 7), which is consistent with the different method used to construct the EMNIST data [43].

5.2 | A Neural Network Benefits From Combining EMNIST With MNIST

In the IPair_E_rot data, because both subsets are images of digits (in the same orientation, from MNIST and EMNIST data sets),

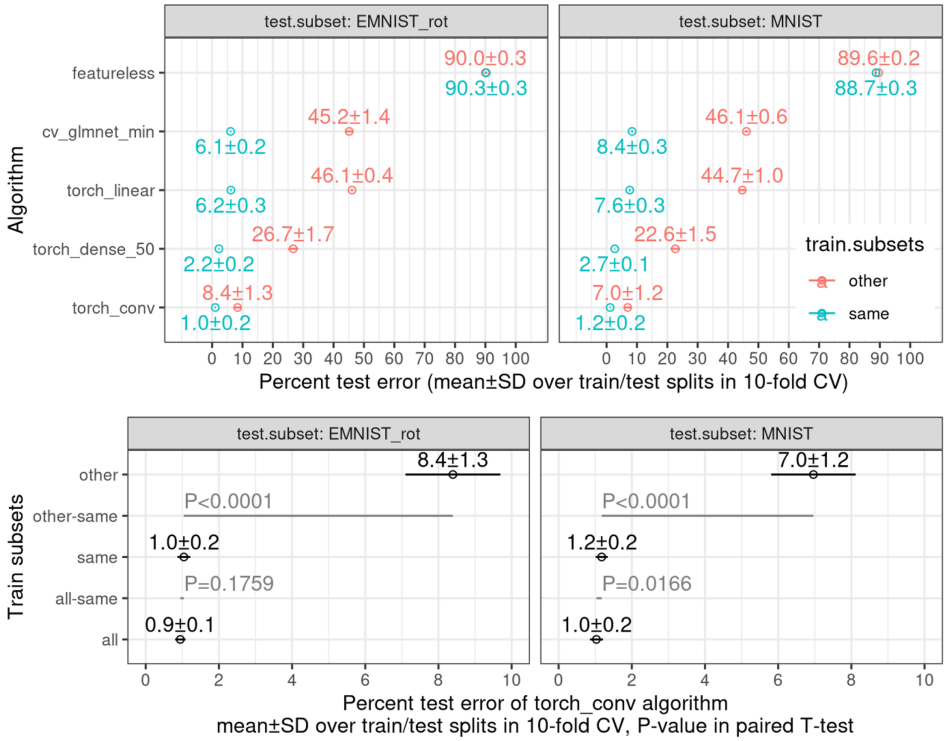


FIGURE 8 | SOAK was used on IPair_E_rot data, which has two subsets of images of digits (from MNIST and EMNIST, in the same orientation). *Top*: The convolutional neural network (torch_conv) is more accurate than the linear models (cv_glmnet_min and torch_linear), but training on the Same subset is still more accurate than Other, indicating that it is surprisingly difficult to generalize between MNIST and EMNIST images (due to their different pre-processing techniques). *Bottom*: For `torch_conv`, Other had significantly larger test error rates, whereas All had slightly (predict on EMNIST_rot) or significantly (predict on MNIST) smaller test error rates when compared to Same.

we expected that the Other model should provide as accurate predictions as Same, using a more powerful learning algorithm than the linear model. To test this hypothesis, we used SOAK on the IPair_E_rot data with two different neural networks implemented using the torch R package [72], and the mlr3 framework [38]. First, torch_conv is a convolutional neural network with architecture defined as follows: first layer 2d convolution with kernel size 6 and 20 output channels, ReLU activation, 2d max pooling with kernel size 4 and stride 4, fully connected layer with 50 hidden output units, ReLU activation, fully connected layer to the 10 outputs. Second, torch_dense_50 is a fully connected neural network with one hidden layer of 50 hidden units and ReLU activation. Stochastic gradient descent was used for training with cross-entropy loss, learning rate 0.1, and batch size 100. Early stopping regularization was used to determine the optimal number of epochs (from 1 to the max of 200): an internal validation split with 50% of training data was used, and the number of epochs with minimal validation loss was defined as optimal, and then the network was re-initialized and re-trained using the optimal number of epochs.

Neural network Other models are less accurate than Same models. Figure 8 shows the error rates with respect to the held-out test sets in SOAK. As expected, the convolutional neural network has smaller error rates than the linear models (cv_glmnet_min and torch_linear). Surprisingly, the error rates for Other were significantly larger than the Same, even for the convolutional network. For example, when predicting on MNIST, the convolutional network had $1.2\% \pm 0.2\%$ (mean $\pm$ SD) test error when training on MNIST (Same model, Figure 8, top), whereas it had $7.0\% \pm 1.2\%$ test error when training on EMNIST_rot (Other model). These data convincingly show that generalization between these two image data sets (MNIST and EMNIST) is surprisingly difficult, even for a convolutional neural network.

Convolutional network All models are more accurate than Same models. Interestingly, we observed that the convolutional network was able to benefit from combining subsets when training. In particular, we observed that All test error rate was smaller than Same, either slightly ($p = 0.1759$ in paired t_9-test, predict on EMNIST_rot, Figure 8, bottom) or significantly ($p = 0.0166$, predict on MNIST). These results show that the EMNIST/MNIST data are too different for training on one and predicting accurately on the other (at least with the models we examined), but are sufficiently similar so that the convolutional neural network benefits when it can train using data from both subsets.

5.3 | Analysis of Data With Predefined Train/Test Subsets

In analyzing these 11 data sets, our goal was to verify that most data sets have similar predefined train/test splits (when training and predicting using a linear model). In examining Figures 5–7, we observed that most data sets with predefined train/test subsets were in the similar category (MNIST, Fashion-MNIST, QMNIST, STL10, CIFAR10, EMNIST, waveform, spam, zipUSPS), and two had clearly different subsets (vowel and

KMNIST). Our SOAK analysis with a linear model suggests that the predefined train/test splits in vowel/KMNIST represent problems that may not be i.i.d.

6 | Conclusions and Future Work

We presented Same/Other/All K-fold cross-validation (SOAK), which is a new method that can be used to estimate the extent of similarity and differences between learnable and predictable patterns in data subsets. We showed how SOAK can be used to gain insights about 20 real-world and benchmark data sets. We quantitatively verified that a model trained on MNIST digits is not as accurate on FashionMNIST (clothing) as on EMNIST (digits). We were also able to verify that most benchmark data had similar predefined train/test subsets (except vowel/KMNIST). We observed significant differences between learnable and predictable patterns in space/time subsets of several new benchmarks based on segmentation problems (Canada forest fires, Arizona trees, Fish sonar river bottoms). Results for NSCH_autism indicate it would be slightly beneficial to train a model on the combined data from different years (Figures 4–6).

It is important to note that the results from a SOAK analysis are limited to the observed data subsets and may not be representative of new data subsets that could be collected in the future. For example, the NSCH_autism data consists of two subsets, 2019 and 2020. We observed that it was beneficial to combine these subsets when training the linear model. These results suggest that it may also be beneficial to add 2021 data to the training set when those data become available. However, if there is a very different pattern in the 2021 data, it may not be beneficial. To determine if the new 2021 subset is compatible with the previous subsets, we would recommend running SOAK on all three subsets when the new data become available.

We expect that the proposed SOAK algorithm will be very useful in practical applications of machine learning, in order to estimate how accurate predictions can be on qualitatively different subsets of data (time periods, hospitals in medical data, geographical regions in satellite data, characterizing fairness across age/sex/race subsets in demographic data, etc.). Future work includes algorithms for efficiently searching the space of subsets to use for training, which is a combinatorial problem, but may be tractable with submodular optimization [73]. We observed how sample size can affect prediction accuracy (Figure 3), and we will be interested to more thoroughly investigate that effect using SOAK with train sets that are down-sampled to equivalent sizes.

Limitations: SOAK is only applicable to data sets with subsets of interest (this may not be true of all data sets), for which there is enough time to use K-fold cross-validation (we used a linear model, which is fast enough, but this may not be appropriate for other learning algorithms, especially deep neural networks with lots of parameters).

Acknowledgments

T.D.H. thanks Université de Sherbrooke, Faculté des Sciences, for a start-up grant, and O.J.L. thanks National Institute of Mental Health grant 3R01MH134177-02S1.

Funding

This work was supported by the Université de Sherbrooke, Faculté des Sciences and National Institutes of Health (3R01MH134177-02S1).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

We provide an R package, published on the CRAN, which implements SOAK in the mlr3 framework: https://github.com/tdhock/mlr3resampling. Detailed documentation for the software, including example code, can be found in the vignettes published on CRAN: https://cloud.r-project.org/web/packages/mlr3resampling/. We also provide a GitHub repository with code used to make the figures in this paper: https://github.com/tdhock/cv-same-other-paper.

References

1. T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2 (Springer, 2009).

2. M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation* (MIT Press, 2012).

3. C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning* (Springer, 2006).

4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

5. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, fourth ed. (Pearson Education, Inc, 2021).

6. S. C. Larson, "The Shrinkage of the Coefficient of Multiple Correlation," *Journal of Educational Psychology* 22, no. 1 (1931): 45–55.

7. M. Stone, "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 36, no. 2 (1974): 111–133.

8. S. Geisser, "The Predictive Sample Reuse Method With Applications," *Journal of the American Statistical Association* 70, no. 350 (1975): 320–328.

9. T. D. Hocking, "Introduction to Machine Learning and Neural Networks," in *Land Carbon Cycle Modeling: Matrix Approach, Data Assimilation, and Ecological Forecasting: Chapter 36*, ed. Y. Luo (CRC Press, 2022), https://doi.org/10.1201/9780429155659.

10. S. Arlot and A. Celisse, "A Survey of Cross-Validation Procedures for Model Selection," *Statistical Surveys* 4 (2010): 40–79, https://doi.org/10.1214/09-SS054.

11. W. Stephenson, Z. Frangella, M. Udell, and T. Broderick, "Can We Globally Optimize Cross-Validation Loss? Quasiconvexity in Ridge Regression," in *Advances in Neural Information Processing Systems*, vol. 34 (MIT Press, 2021), 24352–24364.

12. S. Ghosh, W. Stephenson, T. D. Nguyen, S. Deshpande, and T. Broderick, "Approximate Cross-Validation for Structured Models," in *Advances in Neural Information Processing Systems*, vol. 33 (MIT Press, 2020), 8741–8752.

13. B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do Cifar-10 Classifiers Generalize to Cifar-10?," arXiv Preprint arXiv:1806.00451, 2018.

14. K. Pearson, "X. On the Criterion That a Given System of Deviations From the Probable in the Case of a Correlated System of Variables is Such That It Can Be Reasonably Supposed to Have Arisen From Random Sampling," *London, Edinburgh, and Dublin Philosophical Magazine and*

Journal of Science 50, no. 302 (1900): 157–175, https://doi.org/10.1080/14786440009463897.

15. W. G. Cochran, "The Combination of Estimates From Different Experiments," *Biometrics* 10, no. 1 (1954): 101–129.

16. A. A. Veroniki, D. Jackson, W. Viechtbauer, et al., "Methods to Estimate the Between-Study Variance and Its Uncertainty in Meta-Analysis," *Research Synthesis Methods* 7, no. 1 (2016): 55–79.

17. D. Romanić, M. Ćurić, I. Jovičić, and M. Lompar, "Long-Term Trends of the 'Koshava' Wind During the Period 1949–2010," *International Journal of Climatology* 35, no. 2 (2015): 288–302.

18. C. Truong, L. Oudre, and N. Vayatis, "Selective Review of Offline Change Point Detection Methods," *Signal Processing* 167 (2020): 107299.

19. S. M. West, "Redistribution and Rekognition: A Feminist Critique of Algorithmic Fairness," *Catalyst (Feminism, Theory, Technoscience)* 6, no. 2 (2020).

20. S. He, J. Song, Y. Ou, Y. Yuan, X. Zhang, and X. Xu, "Gard: Gender Difference Analysis and Recognition Based on Machine Learning," *Array* 14 (2022): 100140.

21. R. Wilming, A. Dox, H. Schulz, M. Oliveira, B. Clark, and S. Haufe, "Gecobench: A Gender-Controlled Text Dataset and Benchmark for Quantifying Biases in Explanations," arXiv Preprint arXiv:2406.11547, 2024.

22. S. M. Hall, F. Gonçalves Abrantes, H. Zhu, G. Sodunke, A. Shtedritski, and H. R. Kirk, "Visogender: A Dataset for Benchmarking Gender Bias in Image-Text Pronoun Resolution," in *Advances in Neural Information Processing Systems*, vol. 36 (MIT Press, 2024).

23. A. Currey, M. Nădejde, R. Pappagari, et al., "Mt-Geneval: A Counterfactual and Contextual Dataset for Evaluating Gender Accuracy in Machine Translation," 2022, https://www.amazon.science/publications/mt-geneval-a-counterfactual-and-contextual-dataset-for-evaluating-gender-accuracy-in-machine-translation.

24. G. Stanovsky, N. A. Smith, and L. Zettlemoyer, "Evaluating Gender Bias in Machine Translation," arXiv Preprint arXiv:1906.00591, 2019.

25. L. Bentivogli, B. Savoldi, M. Negri, M. A. Di Gangi, R. Cattoni, and M. Turchi, "Gender in Danger? Evaluating Speech Translation Technology on the Must-She Corpus," arXiv Preprint arXiv:2006.05754, 2020.

26. B. Savoldi, M. Gaido, L. Bentivogli, M. Negri, and M. Turchi, "Gender Bias in Machine Translation," *Transactions of the Association for Computational Linguistics* 9 (2021): 845–874.

27. A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A Reductions Approach to Fair Classification," in *International Conference on Machine Learning* (PMLR, 2018), 60–69.

28. H. Weerts, M. Dudík, R. Edgar, A. Jalali, R. Lutz, and M. Madaio, "Fairlearn: Assessing and Improving Fairness of Ai Systems," *Journal of Machine Learning Research* 24, no. 257 (2023): 1–8.

29. M. Costa-Jussà, "Domain Adaptation Strategies in Statistical Machine Translation: A Brief Overview," *Knowledge Engineering Review* 30, no. 5 (2015): 514–520.

30. A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A Brief Review of Domain Adaptation," in *Advances in Data Science and Information Engineering*, ed. R. Stahlbock, G. M. Weiss, M. Abou-Nasr, C.-Y. Yang, H. R. Arabnia, and L. Deligiannidis (Springer International Publishing, 2021), 877–894.

31. A. Judge, T. Judge, N. Duchateau, et al., "Domain Adaptation of Echocardiography Segmentation via Reinforcement Learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2024), 235–244.

32. S. Madan, T. Henry, J. Dozier, et al., "When and How Cnns Generalize to Out-of-Distribution Category-Viewpoint Combinations," arXiv Preprint arXiv:2007.08032, 2020.

33. P. Ploton, F. Mortier, M. Réjou-Méchain, et al., "Spatial Validation Reveals Poor Predictive Performance of Large-Scale Ecological Mapping Models," *Nature Communications* 11, no. 1 (2020): 4540.

34. J. R. Coyle and N. S. Hejazi, "Origami: A Generalized Framework for Cross-Validation in r," *Journal of Open Source Software* 3, no. 21 (2018): 512, https://doi.org/10.21105/joss.00512.

35. M. Mayer, "splitTools: Tools for Data Splitting," R Package Version 1.0.1, 2023, https://CRAN.R-project.org/package=splitTools.

36. M. Lang, M. Binder, J. Richter, et al., "mlr3: A Modern Object-Oriented Machine Learning Framework in R," *Journal of Open Source Software* 4 (2019): 1903, https://doi.org/10.21105/joss.01903.

37. F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research* 12 (2011): 2825–2830.

38. B. Bischl, R. Sonabend, L. Kotthoff, and M. Lang, eds., *Applied Machine Learning Using mlr3 in R* (CRC Press, 2024), https://mlr3book.mlr-org.com.

39. P. Bayle, A. Bayle, L. Janson, and L. Mackey, "Cross-Validation Confidence Intervals for Test Error," in *Advances in Neural Information Processing Systems*, vol. 33 (MIT Press, 2020), 16339–16350.

40. H. Schulz-Kümpel, S. Fischer, T. Nagler, A.-L. Boulesteix, B. Bischl, and R. Hornung, "Constructing Confidence Intervals for 'the' Generalization Error – A Comprehensive Benchmark Study," 2024, https://arxiv.org/abs/2409.18836.

41. M. Becker and M. Lang, "mlr3batchmark: Batch Experiments for 'mlr3'," 2024, https://mlr3batchmark.mlr-org.com, https://github.com/mlr-org/mlr3batchmark.

42. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE* 86, no. 11 (1998): 2278–2324.

43. G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending Mnist to Handwritten Letters," in *2017 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2017), 2921–2926.

44. H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-Mnist: A Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv Preprint arXiv:1708.07747, 2017.

45. S. Marcel and Y. Rodriguez, "Torchvision the Machine-Vision Package of Torch," in *Proceedings of the 18th ACM International Conference on Multimedia* (ACM, 2010), 1485–1488.

46. K. Alex, "Learning Multiple Layers of Features From Tiny Images," 2009, https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf.

47. T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep Learning for Classical Japanese Literature," arXiv Preprint arXiv:1812.01718, 2018.

48. C. Yadav and L. Bottou, "Cold Case: The Lost MNIST Digits," in *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc, 2019).

49. A. Coates, A. Ng, and H. Lee, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings* (JMLR, 2011), 215–223.

50. N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone," *Remote Sensing of Environment* 202 (2017): 18–27.

51. N. H. French, E. S. Kasischke, R. J. Hall, et al., "Using Landsat Data to Assess Fire and Burn Severity in the North American Boreal Forest Region: An Overview and Summary of Results," *International Journal of Wildland Fire* 17, no. 4 (2008): 443–462.

52. S. A. Parks, G. K. Dillon, and C. Miller, "A New Metric for Quantifying Burn Severity: The Relativized Burn Ratio," *Remote Sensing* 6, no. 3 (2014): 1827–1844.

53. N. Pettorelli, *The Normalized Difference Vegetation Index* (Oxford University Press, 2013).

54. B. Alberton, T. C. Martin, H. R. Da Rocha, et al., "Relationship Between Tropical Leaf Phenology and Ecosystem Productivity Using Phenocameras," *Frontiers in Environmental Science* 11 (2023): 1223219.

55. NASA JPL, *Nasa Shuttle Radar Topography Mission Global 1 Arc Second* (NASA Eosdis Land Processes Distributed Active Archive Center, 2013), https://doi.org/10.5067/MEaSUREs/SRTM/SRTMGL1.003.

56. J. Rouse, J. Schell, and D. Deering, "Monitoring Vegetation Systems in the Great Plains With Erts," in *Third ERTS Symposium* (IEEE, 1973), 309–317.

57. M. Hardisky, V. Klemas, and R. Smart, "The Influences of Soil Salinity, Growth Form, and Leaf Moisture on the Spectral Reflectance of *Spartina alterniflora* Canopies," *Photogrammetric Engineering and Remote Sensing* 49 (1983): 77–83.

58. B. Gao, "Normalized Difference Water Index for Remote Sensing of Vegetation Liquid Water From Space," in *Proceedings of SPIE*, vol. 2480 (SPIE, 1995), 225–236.

59. J. Gallant and J. Austin, "Topographic Position Index Derived From 1″ Srtm Dem-s," *CSIRO Data Collection* 6 (1983), https://doi.org/10.4225/08/5758CCC862AD5.

60. QGIS Development Team, "Qgis Geographic Information System," 2024, https://www.qgis.org.

61. U.S. Fish and Wildlife Service and National Marine Fisheries Service, *Gulf Sturgeon 5-Year Review. Technical Report* (U.S. Fish and Wildlife Service and National Marine Fisheries Service, 2022), https://www.fisheries.noaa.gov/resource/document/gulf-sturgeon-5-year-review.

62. C. S. Bodine and D. Buscombe, "PING-Mapper (v1.0.0) [Software]," 2022, https://doi.org/10.5281/zenodo.6604785.

63. C. S. Bodine and D. Buscombe, "PING-Mapper (v2.0.0) [Software]," 2023.

64. C. S. Bodine, D. Buscombe, R. J. Best, J. A. Redner, and A. J. Kaeser, "PING-Mapper: Open-Source Software for Automated Benthic Imaging and Mapping Using Recreation-Grade Sonar," *Earth and Space Science* 9, no. 9 (2022), https://doi.org/10.1029/2022EA002469.

65. C. S. Bodine, D. Buscombe, and T. D. Hocking, "Automated River Substrate Mapping From Sonar Imagery With Machine Learning," *Journal of Geophysical Research: Machine Learning and Computation* 1, no. 3 (2024): e2024JH000135, https://doi.org/10.1029/2024JH000135.

66. K. J. Sulak, F. Parauka, W. T. Slack, et al., "Status of Scientific Knowledge, Recovery Progress, and Future Research Directions for the Gulf Sturgeon, *Acipenser oxyrinchus desotoi* Vladykov, 1955," *Journal of Applied Ichthyology* 32, no. S1 (2016): 87–161, https://doi.org/10.1111/jai.13245.

67. D. Buscombe, E. B. Goldstein, C. R. Sherwood, et al., "Human-in-the-Loop Segmentation of Earth Surface Imagery," *Earth and Space Science* 9, no. 3 (2022): e2021EA002085, https://doi.org/10.1029/2021EA002085.

68. R. M. Ghandour, J. R. Jones, L. A. Lebrun-Harris, et al., "The Design and Implementation of the 2016 National Survey of Children's Health," *Maternal and Child Health Journal* 22 (2018): 1093–1102.

69. The United States Census Bureau, Associate Director of Demographic Programs, National Survey of Children's Health, "2019 NSCH Data Release," 2019 Topical Data and Input Files, September 2020, https://www.census.gov/programs-surveys/nsch/data/datasets/nsch2019.html.

70. The United States Census Bureau, Associate Director of Demographic Programs, National Survey of Children's Health, "2020 NSCH Data Release," 2020 Topical Data and Input Files, Oct 2021, https://www.census.gov/programs-surveys/nsch/data/datasets/nsch2020.html.

71. D. Deterding, M. Niranjan, and T. Robinson, *Connectionist Bench (Vowel Recognition — Deterding Data)* (UCI Machine Learning Repository, 1988), https://doi.org/10.24432/C58P4S.

72. D. Falbel and J. Luraschi, "torch: Tensors and Neural Networks With 'GPU' Acceleration," R Package Version 0.16.0, 2025, https://CRAN.R-project.org/package=torch.

73. F. Bach, "Learning With Submodular Functions: A Convex Optimization Perspective," *Foundations and Trends in Machine Learning* 6, no. 2–3 (2013): 145–373.