Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Distinguishing Order Effects from Random Variation in Cleaning Pipelines for Tabular Data Classification

Muhammad Idrees

muhammadidrees9405@gmail.com

Institute of Management Sciences Peshawar

Zaheer Ullah

Institute of Management Sciences Peshawar

Thanh Thi Nguyen

University of the Sunshine Coast

Muhammad Taimour Hamid

Institute of Management Sciences Peshawar

Research Article

Keywords: Preprocessing order, Data cleaning, Label noise, Class imbalance, Permutation test, Tabular data

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11432555/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Distinguishing Order Effects from Random Variation in Cleaning Pipelines for Tabular Data Classification

Muhammad Idrees[1*], Zaheer Ullah[1], Thanh Thi Nguyen[2], Muhammad Taimour Hamid[1]

[1*]Institute of Management Sciences, Peshawar, Pakistan.
[2]University of the Sunshine Coast, Sippy Downs, Queensland, Australia.

*Corresponding author(s). E-mail(s): muhammadidrees9405@gmail.com;
Contributing authors: zaheerdawar13@gmail.com;
tnguyen5@usc.edu.au; taimourhamid4@gmail.com;

Abstract

Before training a classifier on tabular data, practitioners usually fill in missing values, remove outliers, correct wrong labels and balance classes. Each step changes the data the next one receives, so their order can change accuracy, yet it is usually chosen by convention. Studies that compare orders rank them by average accuracy over a few train–test splits, but with so few splits the best and worst orders differ by chance even when order has no effect. We propose the Order Sensitivity Index (OSI), the relative accuracy gap between the best and worst order, and a permutation test that estimates the gap expected if order did not matter. We evaluated all 24 orders of four steps on ten datasets with three classifiers and known injected errors, and under eight further conditions. In the main experiment, this expected gap exceeded half of every observed gap. Order still mattered beyond chance in 15 of 30 dataset–classifier pairs, changing balanced accuracy by 0.6 to 2.8 percentage points. The effect grew with label noise and was not detected when errors were spread evenly across classes. The most consistent pattern concerned balancing classes before or after label correction. The known errors revealed why: a label-correction model trained on imbalanced data changed many correct minority-class labels. Balancing classes within the correction model's training data prevented much of this, raising balanced accuracy by 1.4 to 1.9 points over correction on imbalanced data. Order comparisons should be tested against chance, and label correction should account for class imbalance.

Keywords: Preprocessing order, Data cleaning, Label noise, Class imbalance, Permutation test, Tabular data

1 Introduction

Tabular data in applications such as credit scoring, medical screening and spam filtering are rarely clean. Before a classifier is trained, the analyst typically fills in missing values, removes outliers, corrects suspect labels and rebalances the classes, often alongside encoding, scaling and feature selection. Each of these steps has a substantial literature of its own (Côté et al. 2024; Mohammed et al. 2025), and data-centric AI now treats data preparation as part of the system to be designed and evaluated (Malerba and Pasquadibisceglie 2024; Sancricca et al. 2024). The order in which the steps are applied has received far less attention: most pipelines inherit it from convention, and few studies treat it as a variable.

Order matters because each step changes the data that the next step receives. The synthetic minority oversampling technique (SMOTE) (Chawla et al. 2002) creates new minority examples from existing ones, so if it runs before label errors are corrected, it spreads those errors into the synthetic examples it creates. A label correction model trained on imbalanced data sees few minority examples and may wrongly "correct" the ones it does see. An outlier detector judges each point against a data distribution that imputation and oversampling may already have changed. Several questions therefore remain open: how much these interactions change results, under which conditions they matter, which of them matters most, and whether a single order can serve as a sensible default.

Earlier work shows that order can matter, but it does not answer these questions. On tabular data with injected errors, Sancricca and Cappiello (2022) found that applying preparation actions in a recommended order gave better results than applying them in other orders, and their DIANA framework recommends actions and their order from such evidence (Sancricca et al. 2024). Pipeline search systems such as the Tree-based Pipeline Optimization Tool (TPOT) (Olson et al. 2016) and Learn2Clean (Berti-Équille 2019) look for a good order for a single dataset, and Li et al. (2025) compared six selected orders of four preprocessing steps while checking whether a declared pipeline had been used. For noise handling and oversampling the literature even gives opposite answers: SMOTE with edited nearest neighbors (SMOTE-ENN) (Batista et al. 2004) and SMOTE with an iterative-partitioning filter (SMOTE-IPF) (Sáez et al. 2015) oversample first and filter afterwards, whereas Cheng et al. (2019) and Sağlam and Cengiz (2022) deal with noise first so that SMOTE does not spread label errors, and Szeghalmy and Fazekas (2024) recommend removing noise before synthetic examples are generated.

To our knowledge, no study has isolated the order of noise handling and oversampling with the rest of the pipeline held fixed. More generally, we found no study that compares every order of a fixed set of cleaning steps, with the learner held fixed, and tests the gap between the best and worst orders against the gap expected when order has no effect.

That expected gap is the central measurement problem. Each order is usually evaluated on a few random train–test splits, each fixed by a random seed, and its score varies from one split to the next; we call this seed-to-seed variation. When 24 orders are compared, the best and worst average scores therefore differ even if order has no effect at all. For XGBoost on Pima in our main experiment, the best and worst orders differ by 11.5% of the best score, and a gap of 8.1% would be expected from seed-to-seed variation alone. Comparing the best order with the worst using a paired test makes the problem worse, because both orders were selected from the same results that the test then uses. A fair comparison therefore needs a baseline for the differences that chance alone produces. Our aim is to provide this baseline, to measure how much order sensitivity remains once it is accounted for, and to explain the strongest interaction that remains.

We applied every order of imputation, outlier removal, label noise correction and class balancing to 12 binary datasets with three classifiers. Ten datasets form the main experiment, and on seven of them we also ran every shorter sequence of the four steps. So that every error was known, we injected three kinds of error into the training data (missing values, outliers and flipped labels) and recorded where they were; class balancing addresses the imbalance that every dataset already has. Beyond the main experiment, eight further conditions varied individual design factors or added stress settings: the cleaning methods, the label noise rate, how label errors are divided between the classes, the oversampling ratio, the degree of imbalance, the absence of injected errors, the use of shorter sequences and the number of seeds (41,185 runs in all). Because we knew which labels had been flipped, we could check every change made by the label correction step instead of inferring its behavior from accuracy.

The paper is structured around four research questions; the first three are primary, and the fourth, on transfer, is secondary:

RQ1 How large are order effects once seed-to-seed variation is taken into account?
RQ2 Under which data conditions do they appear?
RQ3 Which ordering decision matters most, and why?
RQ4 Does an order chosen on the other datasets transfer to a held-out one?

The paper makes four contributions:

- A procedure for judging order effects against chance. Its index, the Order Sensitivity Index (OSI), is a simple normalized range; the contribution lies in calibrating it with a within-replicate permutation test that estimates the OSI expected when order has no effect. Both apply to any set of cleaning steps and any non-negative performance measure for which higher is better (Section 3.1).
- An exhaustive evaluation of all 24 orders of the four steps under nine conditions, on ten datasets in the main experiment and 12 in all, and of all 41 shorter sequences on seven of them. In the main experiment, chance accounts for more than half of every observed gap, yet order effects beyond chance appear in half of the dataset–classifier pairs. They grow with label noise, are largest on the two most imbalanced datasets under an adapted noise model, and are not detected when label errors are spread evenly across classes (RQ1 and RQ2; Sections 4.1 and 4.2).

- Evidence on the decision that matters most, the order of class balancing and label correction. When label errors fall mostly on the minority class, balancing first is the better order in most conditions tested, although in the main condition the difference reaches nominal significance only for logistic regression. The known injected errors show why: a correction model trained on imbalanced data changes many correct minority labels. Oversampling inside the correction model's own cross-validation folds reduces this and raises balanced accuracy by 1.4 to 1.9 points over correcting on the imbalanced data, for all three classifiers. The advantage disappears when label errors are spread evenly or when only highly confident corrections are made, and on Credit Default the opposite order wins for both tree ensembles (RQ3; Sections 4.3–4.5 and 5).
- A leave-one-dataset-out analysis, reported descriptively. Measured against the best order on the held-out dataset, an order chosen on the other datasets loses 37–68% less balanced accuracy than a randomly chosen order. Every chosen order places balancing before label correction (RQ4; Section 4.6).

2 Related Work

2.1 Data quality and cleaning

A systematic review of 101 studies on data cleaning for machine learning covers label cleaning, outlier detection, imputation and related activities (Côté et al. 2024). Mohammed et al. (2025) deliberately degraded tabular datasets along several quality dimensions at controlled levels and measured the effect on many learning algorithms. Puri and Gupta (2021) handle noisy, imbalanced and incomplete binary data jointly. CleanML (Li et al. 2021) measured how cleaning methods for missing values, outliers, mislabels and other errors affect classification, and Mahmud Sujon et al. (2024) compared standardization and normalization across many classifiers. None of these studies varies the order in which corrections are applied. Nevin et al. (2025) showed that cleaning the training and test splits with different pipelines degrades regression models; that is a consistency problem between splits, whereas we study the order of steps within a single training pipeline.

2.2 The four operations

Imputation ranges from k-nearest-neighbor averaging (Troyanskaya et al. 2001) to chained equations (Van Buuren and Groothuis-Oudshoorn 2011); Emmanuel et al. (2021) survey the field, and Little and Rubin (2019) define the missingness mechanisms it assumes. Isolation Forest (Liu et al. 2008) and the Local Outlier Factor (Breunig et al. 2000) are standard outlier detectors. Label noise methods either relabel examples, as Confident Learning does from out-of-fold predicted probabilities with a separate threshold for each class (Northcutt et al. 2021), or remove them, as edited nearest neighbors does (Wilson 1972); Frénay and Verleysen (2014) survey both families. Oversampling either copies minority examples at random (Batista et al. 2004) or,

as SMOTE does, creates new ones between a minority example and one of its nearest minority neighbors (Chawla et al. 2002); He and Garcia (2009) review the wider imbalance problem.

These methods interact: imputation and oversampling reshape the density that an outlier detector scores, and a relabeling model depends on the class balance of the data it is fitted on.

2.3 Noise handling and oversampling: two opposite designs

Hybrid resampling methods have to place noise handling either before or after oversampling, and published methods do both. SMOTE-ENN and SMOTE-Tomek oversample first and then remove examples that look noisy (Batista et al. 2004). SMOTE-IPF does the same with an iterative ensemble filter, so that the filter also catches noisy synthetic examples (Sáez et al. 2015). Cheng et al. (2019) take the opposite position and filter noise before generating synthetic examples, to stop SMOTE from interpolating towards mislabeled points; Sağlam and Cengiz (2022) detect noise with a boosting procedure and use it to set the number of neighbors for each example inside SMOTE, for the same reason. Szeghalmy and Fazekas (2024) compared many noise filters on synthetic imbalanced data with symmetric label noise and a k-nearest-neighbor classifier, and found that removing noise before generating synthetic examples worked better than relying on the sampler to limit its spread; they compared samplers with and without a filter rather than swapping the order of the two steps. Each argument is plausible on its own, but testing them requires changing only this order while the rest of the pipeline stays fixed, across datasets, classifiers and noise levels. Section 5 reports that test.

2.4 Searching over pipeline order

Automated machine learning (AutoML) systems differ in how much of the order they search. Auto-sklearn fixes the structure of its pipelines, data preprocessing, then at most one feature preprocessor, then a classifier, and searches over the components and their hyperparameters (Feurer et al. 2015). TPOT represents pipelines as trees of operators and evolves them by genetic programming, so the sequence of operators is part of what it searches (Olson et al. 2016). DiffPrep relaxes the choice of preprocessing operators, and in one variant their order, into a continuous space and searches it by gradient descent (Li et al. 2023). Learn2Clean targets the order directly: for a given dataset, model and quality metric, it uses Q-learning to choose a sequence of preparation tasks such as imputation and outlier detection (Berti-Équille 2019). Sancricca and Cappiello (2022) compared a recommended order of preparation actions with alternative orders on small tabular datasets with injected errors and found the recommended order better. Their later framework, DIANA (Sancricca et al. 2024), recommends data preparation actions and their order for a given dataset and analysis from a knowledge base of empirical results, and was evaluated in a case study on time series. Li et al. (2025) built a framework that checks whether a declared pipeline of duplicate removal, outlier removal, scaling and resampling was applied to a model;

besides omitting different subsets of these steps, they compared six of their 24 possible orders and found that order changed verification accuracy. They did not compare all orders or test the differences against chance. Table 1 summarizes how these studies treat the order of the steps.

Table 1 Comparison with related work. *Order isolated*: only the order of the steps changes. *Chance-calibrated*: order differences are tested against random variation. *Mechanism checked*: corrections are verified against known injected errors. ✓: yes; –: not reported; n/a: order not compared.

Study	Cleaning steps	How order is handled	Order isolated	Chance-calibrated	Mechanism checked
CleanML (Li et al. 2021)	Missing values, outliers, mislabels	Not varied	n/a	n/a	–
SMOTE-ENN (Batista et al. 2004)	Oversampling, noise filtering	Fixed: oversampling first	n/a	n/a	–
SMOTE-IPF (Sáez et al. 2015)	Oversampling, noise filtering	Fixed: oversampling first	n/a	n/a	–
Cheng et al. (2019)	Noise filtering, oversampling	Fixed: noise handled first	n/a	n/a	–
Sağlam and Cengiz (2022)	Noise detection, oversampling	Fixed: noise handled first	n/a	n/a	–
Szeghalmy and Fazekas (2024)	Noise filtering, oversampling	Filter first vs. no filter	n/a	n/a	–
TPOT (Olson et al. 2016)	Preprocessing operators, model	Searched with the steps	–	–	–
DiffPrep (Li et al. 2023)	Preprocessing operators	Searched with the steps	–	–	–
Learn2Clean (Berti-Équille 2019)	Preparation tasks	Searched per dataset	–	–	–
Sancricca and Cappiello (2022); Sancricca et al. (2024)	Preparation actions	Recommended vs. other orders	✓	–	–
Li et al. (2025)	Deduplication, outliers, scaling, resampling	Six selected orders; steps omitted	✓	–	–
This study	Imputation, outliers, label correction, balancing	All 24 orders and 41 shorter sequences	✓	✓	✓

Search and recommendation systems return a good order for the dataset at hand, from many evaluations or from prior evidence. They do not report whether the chosen order beats a default by more than chance, which is the question this paper addresses. We do not propose a search method.

3 Method and Experimental Design

Every run follows the same protocol: split the data, corrupt the training part, clean it with one pipeline, train a classifier, and score it on the untouched test part. The

experimental conditions differ in the fault model, the cleaning methods, the set of pipelines compared, and the datasets and seeds used.

3.1 Order Sensitivity Index

Let $\mathcal{D}$ be a training set and $\mathcal{O} = \{o_1, \ldots, o_k\}$ a set of cleaning operators, each mapping a training set to a training set. A pipeline $\pi = (o_{i_1}, \ldots, o_{i_m})$ is a sequence of distinct operators, and applying it gives

$$\mathcal{D}_\pi = (o_{i_m} \circ \cdots \circ o_{i_1})\,(\mathcal{D}). \tag{1}$$

We write Π for the set of pipelines being compared. In the main experiment Π holds all $k!$ orderings of $\mathcal{O}$; in Section 4 it also holds shorter subsequences. We evaluate every pipeline over S replicates; in our experiments a replicate is one random seed, which fixes the train–test split and the injected faults (Section 3.3). For replicate $s = 1, \ldots, S$, let $P_s(\pi)$ be the test score of a classifier trained on $\mathcal{D}_\pi^{(s)}$, under any non-negative metric for which higher is better, and let $\bar{P}(\pi) = S^{-1} \sum_s P_s(\pi)$. The Order Sensitivity Index is

$$\mathrm{OSI} = \frac{\max_{\pi \in \Pi} \bar{P}(\pi) - \min_{\pi \in \Pi} \bar{P}(\pi)}{\max_{\pi \in \Pi} \bar{P}(\pi)}, \tag{2}$$

the share of the best observed mean score that the worst pipeline loses. Dividing by the best mean makes OSI comparable across datasets whose attainable scores differ: a spread of two points matters more when the best pipeline reaches 0.60 than when it reaches 0.98. When $\max_\pi \bar{P}(\pi) > 0$, OSI lies in $[0, 1]$, and it equals zero exactly when all pipelines have the same mean score. The definition does not depend on the number of operators, the number of classes or the choice of metric, but its value under no order effect grows with the number of pipelines compared, which the baseline below accounts for. OSI is unchanged when every score is multiplied by a positive constant but not when a constant is added, so its value depends on the scale of the metric; we use balanced accuracy on its usual 0–1 scale throughout.

Equation (2) has one weakness. Each average $\bar{P}(\pi)$ varies from one set of replicates to another, so the gap between the highest and lowest of $|\Pi|$ averages is positive even when order has no effect. OSI is therefore biased upward, and more so with many pipelines and few replicates. We correct for this by comparing each observed OSI with the value expected under no order effect. Within one replicate, all pipelines share the same split and the same injected faults, so if order did not matter their scores would be interchangeable. Formally, the null hypothesis is that, within each replicate s, the scores $\{P_s(\pi) : \pi \in \Pi\}$ are exchangeable with respect to the pipeline labels. We draw $B = 10{,}000$ random reassignments of the scores to pipelines within each replicate and recompute OSI each time. Each reassignment keeps the scores of a replicate together, so it preserves the differences in difficulty between replicates and changes only which pipeline each score is attributed to. From these we report the null mean OSI_0, the excess $\Delta\mathrm{OSI} = \mathrm{OSI} - \mathrm{OSI}_0$, which is the sensitivity beyond this baseline rather than

a noise-free estimate of the order effect, and the permutation p-value

$$p = \frac{1 + \#\{b : \ \mathrm{OSI}^{(b)} \geq \mathrm{OSI}\}}{B + 1}. \tag{3}$$

Exchangeability is stronger than equal mean scores, but OSI is built from mean scores, so the test responds mainly to differences in mean score between pipelines. OSI_0 is a reference value for the observed OSI under this null, not a share of the observed gap that can be attributed to chance. This tests OSI itself, without first picking a best and a worst pipeline from the data (Ojala and Garriga 2010). Because every permuted statistic again takes the maximum and minimum over all $|\Pi|$ pipelines, the reference distribution includes the same selection of extremes as the observed OSI, and under the exchangeability null the test is valid at any number of replicates, up to Monte Carlo error from the finite B.

Our study instantiates (2) with $k = 4$ operators: missing value imputation (IMP), outlier removal (OUT), label noise correction (LNR) and class balancing (BAL). Three operators repair faults that the fault model below injects and records; BAL addresses the class imbalance that every dataset already has, which the oversampling-ratio and high-imbalance conditions vary. For brevity we refer to all four as cleaning steps, although class balancing is more often classed as data preparation than as cleaning. The metric is balanced accuracy, the mean of the true positive and true negative rates, which weights both classes equally; this matters because every dataset in the study is imbalanced to some degree.

3.2 Datasets

Table 2 lists the 12 binary classification datasets, taken from the UCI Machine Learning Repository (Kelly et al. 2023) and OpenML (Vanschoren et al. 2014). They span several application areas: medical diagnosis (Pima, Breast Cancer), credit and income prediction (German Credit, Credit Default, Adult), spam filtering (Spambase), chemistry (QSAR biodegradation), physical measurements (Ionosphere, MAGIC Telescope), banknote authentication, air-quality monitoring (Ozone) and software defect prediction (KC1). Ten enter the main experiment. The two most imbalanced, Ozone and KC1, cannot use the main fault model: its asymmetric label noise (Section 3.3) would flip all of Ozone's minority labels and 68% of KC1's. They appear in the natural-data condition and in a high-imbalance condition with an adapted noise model. We use all 768 rows of Pima and leave its zero entries, which encode missing measurements, as recorded: they enter every condition as observed values, so Pima has no missing entries in the natural-data condition. For Adult we start from the UCI training file and keep the 30,162 of its 32,561 rows that have no missing values, so that every missing value in the study is one the fault model injected and recorded. Nominal features in Adult, Credit Default and German Credit are integer-encoded. In Credit Default these are sex, education and marital status, which the source file already stores as numbers. Fault injection, imputation and oversampling treat these columns as nominal (Sections 3.3 and 3.4); outlier scoring, the correction forest and the classifiers see the integer codes. The eight OpenML datasets are identified in Table 2 by their

OpenML identifiers, and the code repository holds the dataset files used in the study or downloads them from OpenML (Data availability).

Table 2 Datasets. IR = $n_{\text{maj}}/n_{\text{min}}$. Conditions: M main, N natural data, B second method set, L label noise rates, R oversampling ratios, S step subsets, H high imbalance, Y symmetric noise, T 20 seeds (Table 5).

Dataset	Source	Samples	Features	IR	Minority (%)	Conditions
Pima Diabetes	OpenML 37	768	8	1.87	34.9	M N B L Y R S T
Breast Cancer (W.)	UCI	569	30	1.68	37.3	M N B L Y R S T
Ionosphere	OpenML 59	351	34	1.79	35.9	M N B L Y R S T
Banknote Auth.	OpenML 1462	1,372	4	1.25	44.5	M N B L Y R S T
German Credit	OpenML 31	1,000	20	2.33	30.0	M N B L Y R S T
QSAR Biodeg.	OpenML 1494	1,055	41	1.96	33.7	M N B L Y R S T
Spambase	UCI	4,601	57	1.54	39.4	M N B L Y R S T
MAGIC Telescope	OpenML 1120	19,020	10	1.84	35.2	M N B
Adult Income	UCI	30,162	14	3.02	24.9	M N B
Credit Default	UCI	30,000	23	3.52	22.1	M N B
Ozone (8 h)	OpenML 1487	2,534	72	14.84	6.3	N H
KC1	OpenML 1067	2,109	21	5.47	15.5	N H

3.3 Fault model

We call each injected data error a fault. Each seed $s \in \{42, \ldots, 46\}$ fixes a stratified 80/20 train–test split and the randomness of the injected faults. Faults go into the training part only, and the test part stays clean. The injector flips labels first, then deletes values, then inserts outliers, in the same order for every pipeline; this order only builds the corrupted training set and is not part of the order analysis. Each fault is large enough to give its cleaning step something to repair and small enough to leave the data learnable, although concentrating label flips in the minority class makes its contamination much heavier than the overall rate suggests.

Label noise. We flip 15% of the training labels, taking 70% of the flips from the minority class (capped at its size) and the rest from the majority. We concentrate flips in the minority class to model settings in which the rare label is the harder one to assign; this is a modeling choice, and Section 4 varies the overall rate from 5% to 20% and, in a symmetric condition, spreads the same 15% evenly across both classes. A consequence matters for the whole paper: with minority share m, the scheme flips a fraction $0.105/m$ of the minority labels, between 23% and 48% in our main datasets (Table 3). This is a stress test for label correction on the minority class rather than a typical noise level. In the high-imbalance condition we instead flip 15% of the minority labels and the same number of majority labels, so the minority class keeps 85% correct labels at any imbalance ratio.

Missing values. For half of the features, chosen at random, we pick a second feature at random and delete the value in every row where that second feature exceeds its 90th percentile; the percentile and the comparison use the training data as left by the deletions made so far, so rows in which the second feature is already missing are not deleted. Missingness therefore depends on the value of another feature, as under missing at random (Little and Rubin 2019); because that feature may itself have lost

Table 3 Realized fault rates in the main condition, means over five seeds.

Dataset	Training rows	Minority share (%)	Labels flipped (%)	Minority labels flipped (%)	Missing cells (%)
Pima	614	34.9	15.0	29.9	4.4
Breast Cancer	455	37.4	14.9	27.6	4.7
Ionosphere	280	36.1	15.0	28.7	2.7
Banknote	1,097	44.5	14.9	23.4	4.7
German Credit	800	30.0	15.0	35.0	1.2
QSAR	844	33.8	14.9	30.9	3.2
Spambase	3,680	39.4	15.0	26.6	3.9
Telescope	15,216	35.2	15.0	29.9	4.7
Adult	24,129	24.9	15.0	42.2	2.2
Credit Default	24,000	22.1	15.0	47.5	2.9

values to an earlier deletion, the mechanism is close to, rather than exactly, missing at random.

Outliers. In 5% of the rows we replace every numeric feature with $\mu \pm (3.5 + u)\,\sigma$, where μ and σ are the mean and standard deviation of that feature's observed training values after the deletions above, $u \sim U[0, 1]$, and the sign is random. Nominal features keep their values, since a value several standard deviations from the mean is not a category.

Outlier rows overwrite any numeric values deleted earlier, and ties at the 90th percentile shrink the deleted set for discrete features. The realized share of missing cells is therefore 1.2–4.7%, well below the nominal 5% (half of the features, each deleted in 10% of rows; Table 3); we report realized values throughout.

3.4 Cleaning methods

Method set A is used in every condition except the second-method-set condition, which uses set B (described at the end of this subsection). IMP is k-nearest-neighbor imputation with $k = 5$ (Troyanskaya et al. 2001), computed on features standardized from their observed values and transformed back afterwards. OUT is Isolation Forest with 100 trees and an expected outlier share of 5%, fitted on robust-scaled features; it removes the rows it flags (Liu et al. 2008). LNR computes out-of-fold class probabilities with a five-fold stratified cross-validation of a random forest (100 trees, maximum depth 10) and changes a label to the predicted class when the two disagree and the predicted class has probability at least 0.7. This follows the out-of-fold idea of Confident Learning (Northcutt et al. 2021), but with one fixed threshold instead of per-class thresholds, so we call it confidence-thresholded relabeling. It is one controlled instance of correction based on model-estimated label uncertainty, so our conclusions about label correction apply to this rule rather than to the whole family. The same correction forest serves all three classifiers, so differences between classifiers do not come from the correction step. BAL is SMOTE with five neighbors and a target minority-to-majority ratio of 0.8 (Chawla et al. 2002). On the three datasets with nominal features, BAL uses SMOTENC, the variant of SMOTE for mixed data, which interpolates numeric features and gives each nominal feature the value most common among the neighbors (Chawla et al. 2002), and IMP fills nominal features with their most frequent observed value instead of a neighbor average.

Outlier removal, label correction and SMOTE cannot work with missing values, so each fills them temporarily for its own computation and passes them on unchanged to the next step. OUT scores a copy filled with the median of each numeric feature and the most frequent value of each nominal one. LNR fits the same imputer inside each training fold and applies it to the held-out fold. BAL searches for neighbors in a copy filled the same way and keeps the original rows as they were, so only the synthetic rows are complete. BAL is skipped when the minority-to-majority ratio is already at least 95% of the target, when fewer than two minority examples remain, or when only one class remains. Whether these conditions hold depends on what earlier steps did, so we record every skip.

Method set B replaces each operator with one from a different family: median imputation, with the most frequent value for nominal features; the Local Outlier Factor with 20 neighbors and a 5% expected outlier share (Breunig et al. 2000); edited nearest neighbors, which removes a row when at least two of its three nearest neighbors carry another label (Wilson 1972); and random oversampling to the same 0.8 ratio, with the same skip rules (Batista et al. 2004). Set B therefore removes suspect labels instead of relabeling them, and duplicates minority rows instead of interpolating between them. Table 4 summarizes both sets.

Table 4 Cleaning operators and their settings in the two method sets.

Step	Set A (all conditions except one)	Set B (second-method-set condition)
IMP	k-nearest neighbors, $k = 5$	Median; most frequent value for nominal features
OUT	Isolation Forest, 100 trees, 5% outlier share	Local Outlier Factor, 20 neighbors, 5% outlier share
LNR	Out-of-fold random forest relabeling, threshold 0.7	Edited nearest neighbors (removes rows), 3 neighbors
BAL	SMOTE (SMOTENC with nominal features), ratio 0.8	Random oversampling, ratio 0.8

3.5 Classifiers

Three classifiers with fixed hyperparameters cover bagged trees, a linear model and boosted trees: random forest (RF; 100 trees, maximum depth 10), logistic regression (LR; L_2 penalty of strength 1, L-BFGS, at most 1000 iterations) and XGBoost (XGB; 100 boosting rounds, maximum depth 10, learning rate 0.1, logistic loss). We did not tune them per pipeline, so differences between pipelines cannot come from tuning. Logistic regression standardizes its inputs inside the model, using statistics from the cleaned training set, and every logistic regression fit in the study converged within its iteration limit.

3.6 Experimental conditions

Table 5 lists the nine conditions. Each varies one design factor of the main setting, except the high-imbalance condition, which also changes the noise model (Section 3.3). They fall into four groups: changes to the data or methods (natural data, second

method set), changes to the label noise (label noise rate, symmetric noise, high imbalance), a change to balancing (oversampling ratio), and extensions of the design (step subsets, 20 seeds). In the natural-data condition no dataset contains missing values (Pima's zero codes count as observed values; Section 3.2), so IMP does nothing, and the 24 orderings reduce to the 6 orderings of OUT, LNR and BAL, each run four times. We confirmed that the four copies gave identical results and analyze the 6 distinct orderings. The subset condition evaluates every ordered subsequence of zero to three steps, 41 in all, including the empty pipeline; together with the 24 full orderings of the main condition, this covers all 65 sequences. Because these sequences differ in which steps they include as well as in their order, this condition asks whether every step is needed, not how much order matters. Subsequences without IMP leave missing values in the training data, which random forests and XGBoost handle natively and logistic regression cannot, so logistic regression is evaluated only on subsequences that include IMP. The symmetric-noise condition flips the same 15% of training labels, drawn at random regardless of class, so that each class loses about the same share of its labels. The 20-seed condition repeats the main setting with seeds 42–61 on the seven datasets small enough for it (Pima, Breast Cancer, Ionosphere, Banknote, German Credit, QSAR and Spambase); its first five seeds reproduce the main runs.

Table 5 Experimental conditions. Every condition uses three classifiers and five seeds, except the 20-seed condition. A run is one pipeline, one classifier and one seed. In the step-subsets condition logistic regression runs only on the 25 sequences that include IMP, which gives $(41 \times 2 + 25) \times 5 \times 7 = 3{,}745$ runs. The natural-data total includes four identical copies of each of its 6 distinct orderings, and the 20-seed total includes the five seeds that reproduce the main runs.

Condition	Pipelines	Fault model	Methods	Datasets	Runs
Main	24 orderings	as Sec. 3.3	Set A	10	3,600
Natural data	24 orderings	none	Set A	12	4,320
Second method set	24 orderings	as main	Set B	10	3,600
Label noise rate	24 orderings	5, 10, 20% label noise	Set A	7	7,560
Symmetric noise	24 orderings	15% label noise, any class	Set A	7	2,520
Oversampling ratio	24 orderings	as main	Set A, ratio 0.5, 1.0	7	5,040
High imbalance	24 orderings	minority-anchored noise	Set A	2	720
Step subsets	41 subsequences	as main	Set A	7	3,745
20 seeds	24 orderings	as main, seeds 42–61	Set A	7	10,080
Total					41,185

3.7 Statistical analysis

The analyses answer three different questions: whether orders differ by more than chance (sensitivity, measured by OSI and its permutation test), which ordering decision is better (direction, measured by the planned comparison of balancing and label correction), and whether an order chosen on some datasets serves others (transfer). An answer to one does not imply an answer to another. For every dataset–classifier pair in a condition, we compute OSI from the per-ordering means over the seeds of that condition, together with OSI_0, ΔOSI and the permutation p-value of (3). We correct the p-values within each condition for the false discovery rate (Benjamini and Hochberg 1995) and call an order effect significant when the corrected value is below

0.05. The seeds are repeated resampling experiments on the same data, not independent replications of it, so conclusions about one dataset rest on the permutation test within it, and conclusions across datasets on tests that use one value per dataset. The main condition is the primary analysis. The other conditions are sensitivity analyses, each corrected within itself, and we read them for the direction and size of the effects rather than as separate confirmatory tests. Differences in balanced accuracy are reported in percentage points (pp).

The comparison of label correction and balancing is planned, not picked from the results. Each ordering in which LNR precedes BAL has exactly one partner ordering that swaps these two steps and leaves IMP and OUT where they are, which gives 12 pairs. For each seed we average the 12 paired differences in balanced accuracy, and for each dataset we report the mean over seeds and the number of seeds favoring LNR first. Five seeds cannot reach significance on their own, since the smallest two-sided p-value of a signed-rank test with five pairs is 0.0625, so we test across datasets with an exact Wilcoxon signed-rank test, one value per dataset (Demšar 2006), and report with it the Hodges–Lehmann estimate of the paired difference and its exact 95% interval (Hollander et al. 2013). Tables and figures show the difference as correction first minus balancing first, so negative values favor balancing first; the Hodges–Lehmann estimates in the text are given as balancing first minus correction first, so positive values favor balancing first.

We measure position effects as balanced accuracy minus the mean over all orderings for the same dataset, classifier and seed, averaged by the position of each step; these averages are descriptive and are not tested. To check whether an ordering learned on some datasets helps on others, we use leave-one-dataset-out transfer: for each held-out dataset, we pick the order with the highest average centered accuracy on the remaining datasets (balanced accuracy minus the mean of all 24 orders for the same dataset, classifier and seed), then record its rank among the 24 orders and its regret, the gap to the best order, on the held-out dataset. The baseline is the expected regret of an ordering picked at random. Each fold chooses its ordering from datasets that are held out in other folds, so the ten held-out results are not independent; we report them descriptively, without a significance test.

Last, we use the injector's record of flipped labels to see what LNR does. For every main dataset and seed, we run IMP→LNR and IMP→BAL→LNR and classify each relabeled example as fixed (its label had been flipped) or broken (its label was correct). In IMP→BAL→LNR, only the original rows are graded: synthetic rows have no ground-truth label and are excluded from these counts. For IMP→BAL→OUT, we count how many removed rows are SMOTE examples. In the main experiment each step is applied to the data it receives, as a practitioner would apply it, so when BAL runs before LNR its synthetic rows enter the cross-validation of LNR; this dependence is part of the ordering being compared. To separate the effect of class balance from that dependence, we also run LNR after IMP with SMOTE applied inside each of its training folds (same target ratio of 0.8, five neighbors and skip rules as BAL), and with a correction forest using balanced class weights. Fold-internal SMOTE keeps synthetic examples built in one fold out of the predictions for another, and class weighting creates no synthetic examples at all, so we call both fold-safe. These two

variants were added after the relabeling counts below, to test the mechanism those counts suggest. For the end-to-end test, each variant is followed by BAL, OUT and the three classifiers, and compared with IMP→LNR→BAL→OUT on the same splits, faults and seeds. We also repeat the precision analysis, and the end-to-end comparison of IMP→BAL→LNR→OUT with IMP→LNR→BAL→OUT, at correction thresholds of 0.6, 0.8 and 0.9. The out-of-fold probabilities do not depend on the threshold, so only the relabeling and the steps after it change.

3.8 Implementation and reproducibility

The split happens before any fault injection, and every data-dependent step, from imputation and scaling to outlier scoring, label correction and oversampling, is fitted on training data only. Inside LNR, the temporary imputer is fitted on the training folds only. An earlier public release of our code fitted that imputer before the cross-validation split inside LNR; Section 6.2 describes the correction and shows that it leaves the direction of our main finding unchanged. Every stochastic component, from the split and the injected faults to the cleaning operators and the classifiers, takes its random state from the seed. Each operator receives the same random seed wherever it appears in the pipeline, so changing the order changes the data an operator sees but not its seed, and each cleaned training set is shared by the three classifiers. The implementation uses Python 3.12.2, scikit-learn 1.5.0, imbalanced-learn 0.14.1 and XGBoost 3.1.1, and no run in the study failed. ChatGPT (OpenAI) and Claude (Anthropic) were used to assist with coding and with editing the text; the authors reviewed and tested all code and text and take full responsibility for them.

4 Results

4.1 How large are order effects beyond seed-to-seed variation?

Table 6 gives OSI for every dataset and classifier in the main condition, next to its baseline OSI_0, the value expected when order has no effect. Raw OSI runs from 0.92% (Banknote, logistic regression) to 11.48% (Pima, XGBoost). The baseline runs from 1.00% to 8.13%, and in every pair it exceeds half of the raw value. After false discovery rate correction, 15 of the 30 pairs show an order effect beyond the baseline, spread over 8 of the 10 datasets. In the significant pairs the excess ranges from 0.7 to 3.6 points of OSI, or 0.6 to 2.8 points of balanced accuracy: small next to the 2.9 to 13.6 points that the tree ensembles gained from cleaning at all in the subset experiment (Section 4.5; a somewhat optimistic best-of-65 figure), but enough to change which pipeline a comparison would pick. With five seeds, Breast Cancer and Banknote show no detectable effect under any classifier, and for Breast Cancer the observed OSI lies below its baseline in all three cases; with more seeds, small effects on these datasets become detectable, as shown below.

Figure 1 shows why raw OSI misleads. Across the 30 pairs, raw OSI and its baseline have a rank correlation of 0.92, so ranking pairs by raw OSI would largely rank them by how variable their scores are. The normalization in (2) does not drive these calls: the same permutation test applied to the plain gap between the best and worst mean

gives the same significance call in all 30 pairs. Table A1 in Appendix A lists the best and worst ordering of every pair with its spread over seeds.

Because five seeds are few, we repeated the main condition with 20 seeds on the seven smaller datasets (Section 3.6). Thirteen of the 21 significance calls were unchanged. Seven pairs that were not significant with five seeds became significant with 20, including all three classifiers on Breast Cancer, and one pair lost significance. The 15 significant pairs of the main condition are therefore likely to understate how many pairs carry a small order effect.

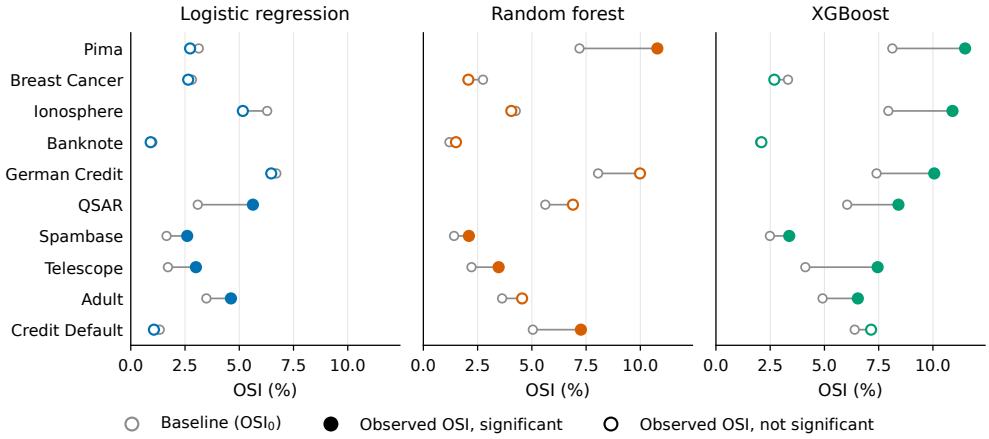


Fig. 1 Observed OSI against the baseline expected without an order effect (OSI_0, open markers), main condition. Colored markers: observed OSI (blue, logistic regression; orange, random forest; green, XGBoost); filled when the order effect is significant after false discovery rate correction. The segment between the two markers is ΔOSI.

4.2 When do order effects appear?

Table 7 repeats the analysis in five conditions; Fig. 2 covers the noise-rate and oversampling-ratio conditions. With the second method set, 15 of 30 pairs show an effect beyond the baseline and the median excess (0.71) is close to the main condition (0.94), so the effect is not specific to one set of methods. On natural data the picture changes. Median OSI falls to 1.1% and only 10 of 36 pairs are significant. Raw OSI here is a range over 6 distinct orderings rather than 24, which lowers it by itself, so this comparison is indicative. Under severe imbalance with minority-anchored noise, all six pairs are significant and the median excess reaches 9.2 points; with two datasets and a different noise model, this condition is a stress test that cannot separate the effect of imbalance from that of the noise rule. Spreading the label errors evenly has the opposite effect. With the same 15% of labels flipped regardless of class, no pair is significant and the median excess is -0.04, against 9 of 21 significant pairs on the same seven datasets in the main condition (Table A3).

Raising the label noise rate strengthens the effect (Fig. 2; exact values in Table A3, Appendix A). On the seven datasets used in every rate condition, no pair is significant

Table 6 Order effects in the main condition (the ten main datasets; Ozone and KC1 enter only the natural-data and high-imbalance conditions). Gap: best minus worst mean balanced accuracy (percentage points). OSI_0: OSI expected without an order effect. $\Delta OSI = OSI - OSI_0$. p_{BH}: permutation p-value after false discovery rate correction over the 30 pairs; bold marks $p_{BH} < 0.05$.

Dataset	Clf.	Gap (pp)	OSI (%)	OSI_0 (%)	ΔOSI	p_{BH}
Pima	LR	2.01	2.73	3.13	−0.40	0.889
	RF	7.61	10.78	7.20	3.59	**0.007**
	XGB	7.97	11.48	8.13	3.35	**0.013**
Breast Cancer	LR	2.54	2.65	2.82	−0.17	0.787
	RF	1.94	2.07	2.75	−0.68	0.938
	XGB	2.52	2.69	3.32	−0.63	0.938
Ionosphere	LR	4.31	5.17	6.29	−1.12	0.938
	RF	3.66	4.05	4.26	−0.20	0.787
	XGB	9.63	10.90	7.94	2.96	**0.017**
Banknote	LR	0.90	0.92	1.00	−0.08	0.841
	RF	1.48	1.50	1.20	0.30	0.119
	XGB	2.05	2.09	2.03	0.06	0.577
German Credit	LR	4.14	6.47	6.71	−0.24	0.787
	RF	6.21	9.99	8.05	1.93	0.085
	XGB	6.33	10.06	7.40	2.66	**0.028**
QSAR	LR	4.76	5.63	3.09	2.54	**<0.001**
	RF	5.54	6.89	5.62	1.27	0.096
	XGB	6.67	8.41	6.05	2.36	**0.013**
Spambase	LR	2.35	2.60	1.65	0.95	**<0.001**
	RF	1.92	2.10	1.41	0.68	**0.004**
	XGB	3.08	3.38	2.49	0.89	**0.028**
Telescope	LR	2.27	3.00	1.71	1.29	**<0.001**
	RF	2.89	3.47	2.22	1.25	**<0.001**
	XGB	6.19	7.45	4.12	3.33	**<0.001**
Adult	LR	3.41	4.62	3.48	1.13	**0.001**
	RF	3.40	4.55	3.63	0.92	0.058
	XGB	4.61	6.54	4.91	1.63	**0.010**
Credit Default	LR	0.71	1.07	1.33	−0.27	0.938
	RF	4.40	7.26	5.05	2.22	**<0.001**
	XGB	4.03	7.15	6.39	0.76	0.307

Table 7 Order effects by condition. Medians over dataset–classifier pairs, each taken separately, so the median ΔOSI need not equal the difference of the other two medians; significance after false discovery rate correction within each condition. ΔOSI is in percentage points of OSI. In the natural-data condition OSI is computed over 6 distinct orderings (Section 3.6).

Condition	Pairs	Significant	OSI (%)	OSI_0 (%)	ΔOSI
Main	30	15	4.58	3.56	0.94
Second method set	30	15	6.15	4.91	0.71
Natural data	36	10	1.09	0.88	0.23
High imbalance	6	6	24.29	16.77	9.20
Symmetric noise	21	0	2.69	2.59	−0.04

at 5% noise; the count rises to 7, 9 and 11 of 21 at 10%, 15% and 20%, and the median excess rises with it, from 0.24 to 1.34. The oversampling ratio matters too. At a ratio of 0.5 no pair is significant. Part of this is mechanical: after noise injection the minority-to-majority ratio of Banknote already exceeds 0.5, so SMOTE was skipped in all 360 Banknote runs, and in half of the Spambase runs. In the main condition SMOTE was never skipped. At a ratio of 1.0, 6 of 21 pairs are significant, fewer than the 9 at 0.8.

Imbalance itself varies little in the main condition, where the ratio spans only 1.25 to 3.52, and a descriptive check links it to order sensitivity only for random forest. Across the ten datasets, its raw OSI rises with the imbalance ratio (Spearman $\rho = 0.79$,

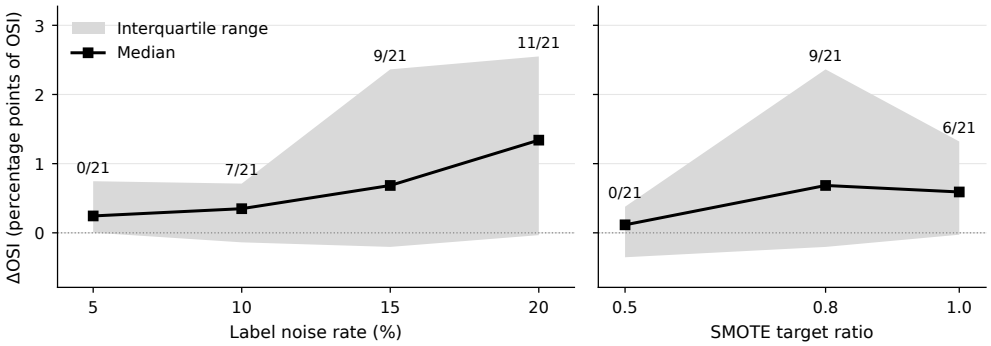


Fig. 2 Excess order sensitivity (ΔOSI) against the label noise rate (left) and the SMOTE target ratio (right), over the 21 dataset–classifier pairs of the seven datasets shared by these conditions. Squares: median; band: interquartile range; labels: significant pairs out of 21. The points at 15% and 0.8 are the main condition. At a ratio of 0.5, SMOTE was skipped in all Banknote runs and half of the Spambase runs.

unadjusted $p = 0.006$), and more weakly once the baseline is subtracted ($\rho = 0.70$, unadjusted $p = 0.025$). Logistic regression ($\rho = -0.03$) and XGBoost ($\rho = 0.25$) show no relation in the excess.

4.3 Label noise correction versus class balancing

Figure 3 gives the paired comparison for every dataset in the main condition; Table A2 in Appendix A lists the exact values and the number of seeds favoring each order. Negative values mean that balancing first is better. On Pima, QSAR, Spambase, Telescope and Adult, balancing first wins in every seed for every classifier, by up to 4.67 points. German Credit points the same way. The clear exception is Credit Default, where correcting labels first wins in all five seeds for random forest (+3.33) and XGBoost (+1.86); logistic regression on the same data shows almost no difference (−0.13). Ionosphere shows a small and unstable preference for correcting first, with seed-to-seed standard deviations of 2.8 to 5.3 points.

Table 8 tests the direction across datasets in every condition. In the main condition the median favors balancing first for all three classifiers, and the result is nominally significant for logistic regression ($p = 0.027$; after Holm correction across the three classifiers 0.081, against 0.210 for random forest and XGBoost; Hodges–Lehmann estimate of balancing first minus correction first 1.04 points, 95% interval 0.19 to 1.82). For random forest and XGBoost the estimates are larger, 1.40 and 1.72 points, but their intervals include zero (−0.41 to 3.38 and −0.42 to 3.59).

Several sensitivity conditions point the same way more consistently, although they are exploratory rather than confirmatory. With the second method set, balancing first wins on all ten datasets for random forest and XGBoost ($p = 0.002$). On natural data it wins on all twelve for XGBoost ($p < 0.001$). At every label noise rate it wins on at least five of seven datasets for every classifier, and on Ozone, under severe imbalance, by 9.7 to 16.5 points in every seed. The two conditions without a direction are the oversampling ratio of 0.5, where balancing changes the training set least, and symmetric label noise, where the minority class is no more contaminated than the

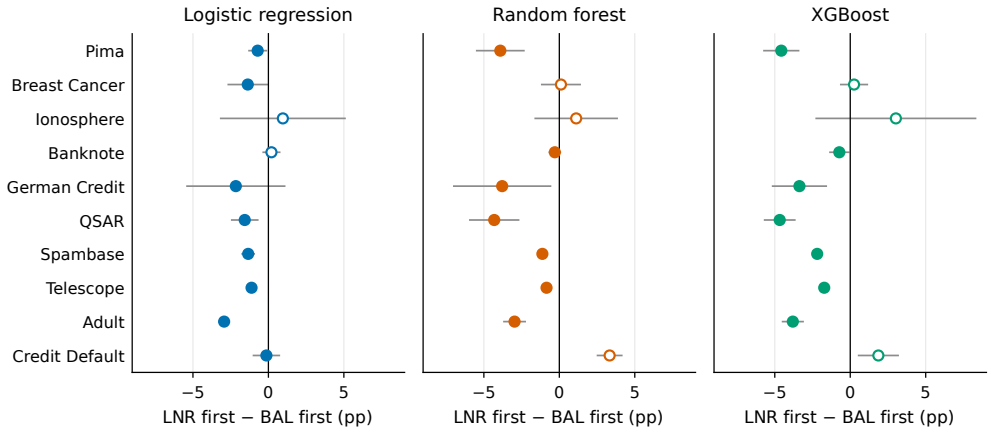


Fig. 3 Label noise correction first minus class balancing first, main condition. Points: mean over 12 paired orderings and five seeds; bars: ±1 standard deviation of the per-seed means. Filled markers: balancing first is better on average.

majority. At a ratio of 1.0 the medians favor balancing first, but none of the three tests is significant. As Section 3.7 explains, this directional test asks a different question from the OSI test: a small but consistent difference between two groups of orders can be detected even when the spread among all 24 is not, which is why balancing first wins at 5% label noise although no pair shows a significant order effect there.

Table 8 Label noise correction first versus class balancing first, across datasets. Each cell: datasets favoring correction first / datasets, median difference in pp, and the unadjusted exact Wilcoxon signed-rank p-value. Negative medians favor balancing first. For the main condition the text also gives Holm-adjusted values across the three classifiers; the other conditions are sensitivity analyses. With two datasets, the high-imbalance condition has no test. The text reports Hodges–Lehmann estimates, which can differ from these sample medians.

Condition	LR	RF	XGB
Main	2/10, −1.23, 0.027	3/10, −0.99, 0.131	3/10, −1.96, 0.105
Second method set	3/10, −0.87, 0.160	0/10, −4.48, 0.002	0/10, −5.55, 0.002
Natural data	4/12, −0.23, 0.083	2/12, −0.32, 0.007	0/12, −0.50, <0.001
Label noise 5%	1/7, −0.40, 0.031	1/7, −0.56, 0.031	2/7, −0.50, 0.156
Label noise 10%	1/7, −0.49, 0.031	1/7, −0.49, 0.047	2/7, −1.05, 0.156
Label noise 20%	1/7, −1.09, 0.031	1/7, −1.58, 0.031	0/7, −1.70, 0.016
Symmetric noise	5/7, +0.23, 0.078	3/7, −0.20, 0.297	3/7, −0.06, 0.469
Ratio 0.5	3/7, −0.01, 0.578	5/7, +0.05, 0.938	4/7, +0.12, 0.938
Ratio 1.0	2/7, −0.39, 0.375	1/7, −1.26, 0.109	2/7, −2.56, 0.156
High imbalance	1/2, −6.48, –	0/2, −8.32, –	0/2, −4.94, –

4.4 Where each step belongs

Figure 4 gives the average effect of each step's position, as a descriptive summary. Balancing does best first and worst last (+0.33 to +0.90 against −0.50 to −0.71 points across classifiers), label correction shows the reverse (−0.53 to −0.82 first, +0.59 to +0.65 last), imputation slightly prefers early placement, and outlier removal shifts by

at most 0.27 points. Section 5 explains the opposite patterns of balancing and label correction.

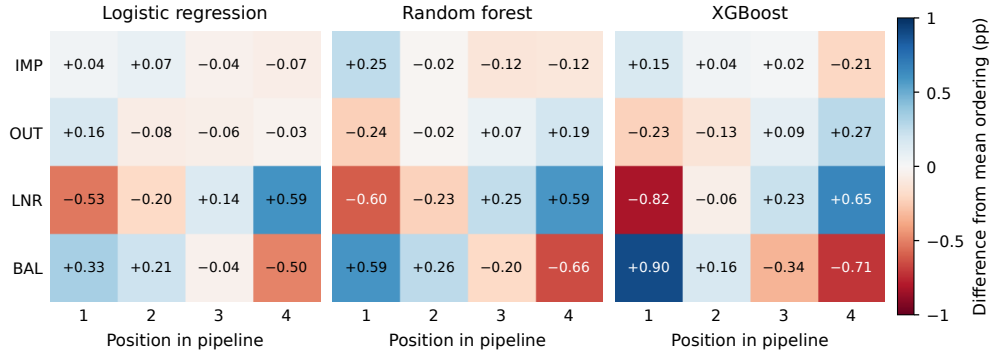


Fig. 4 Balanced accuracy relative to the mean over all 24 orderings, by the position of each step (1 = first), averaged over the ten main datasets and five seeds (pp). Positive values mean above-average balanced accuracy for orderings that put the step in that position; these averages are descriptive and untested.

4.5 Do all four steps help?

Table 9 compares the best of all 65 sequences with the best full ordering. A shorter sequence comes out ahead in 13 of 21 cases, mostly by less than one point, which may partly reflect selecting the best of 65 sequences; the larger gains are on German Credit (1.36 points for random forest, without label correction) and Pima (0.94 and 1.46 points). Balancing appears in 20 of the 21 best sequences. For the two tree ensembles, whether to clean matters more than the order: the best sequence beats the uncleaned pipeline by 2.9 to 13.6 points, against 0.6 to 2.8 points of excess from order in the main experiment, although the former is a somewhat optimistic best-of-65 comparison.

4.6 Transfer of model-specific default orderings

If a practitioner cannot afford to try every ordering, the useful question is whether an ordering chosen on other datasets holds up on a new one. Table 10 and Fig. 5 answer it by leaving one dataset out at a time. For random forest, all ten folds choose the same ordering, IMP→BAL→LNR→OUT. For logistic regression, all ten choose IMP→OUT→BAL→LNR. For XGBoost the folds split between IMP→BAL→LNR→OUT (6) and BAL→LNR→IMP→OUT (4). Every chosen ordering places balancing before label correction. On the held-out dataset, the chosen ordering for logistic regression, random forest and XGBoost has lower regret than the expected regret of a random ordering on 8, 9 and 8 of the 10 datasets, and its mean regret is 65%, 68% and 37% lower. Because the folds share datasets, these results are descriptive (Section 3.7). For random forest the default does worse than a random ordering would on average only on Credit Default, the dataset on which correcting labels first

Table 9 Best sequence among all 65 (subsets and full orderings) against the best full ordering, main fault model, mean balanced accuracy over five seeds. Gain: best sequence minus best full ordering (pp). Logistic regression cannot run without imputation, so it has no uncleaned baseline.

Dataset	Clf.	Best sequence	Its score	Best full ordering	Gain	Uncleaned
Banknote	LR	IMP→LNR→OUT→BAL	0.9822	0.9822	0.00	–
	RF	IMP→LNR	0.9825	0.9816	0.08	0.9538
	XGB	IMP→BAL→LNR→OUT	0.9784	0.9784	0.00	0.9205
Breast Cancer	LR	IMP→OUT→BAL→LNR	0.9599	0.9599	0.00	–
	RF	OUT→LNR→BAL	0.9383	0.9377	0.06	0.9048
	XGB	BAL→LNR	0.9433	0.9369	0.63	0.8877
German Credit	LR	BAL→IMP→OUT	0.6429	0.6400	0.29	–
	RF	IMP→BAL→OUT	0.6360	0.6224	1.36	0.5286
	XGB	OUT→BAL→LNR	0.6333	0.6295	0.38	0.5810
Ionosphere	LR	LNR→OUT→IMP→BAL	0.8350	0.8350	0.00	–
	RF	LNR→BAL→IMP→OUT	0.9033	0.9033	0.00	0.8411
	XGB	LNR→IMP→BAL→OUT	0.8830	0.8830	0.00	0.7470
Pima	LR	BAL→IMP→OUT	0.7384	0.7371	0.13	–
	RF	BAL→LNR	0.7152	0.7058	0.94	0.5820
	XGB	BAL→LNR→OUT	0.7087	0.6941	1.46	0.5996
QSAR	LR	IMP→OUT→BAL→LNR	0.8464	0.8464	0.00	–
	RF	BAL→LNR→OUT	0.8115	0.8036	0.78	0.7125
	XGB	BAL→OUT→LNR	0.7952	0.7932	0.21	0.7060
Spambase	LR	IMP→BAL→LNR→OUT	0.9024	0.9024	0.00	–
	RF	OUT→BAL	0.9177	0.9136	0.41	0.8798
	XGB	BAL→IMP→LNR	0.9129	0.9113	0.16	0.8584

wins (Section 4.3); XGBoost does worse there and on Ionosphere, which also leans towards correcting first.

Table 10 Leave-one-dataset-out transfer of the ordering chosen for each classifier, main condition. Regret: best ordering on the held-out dataset minus the chosen ordering, in pp, averaged over the ten held-out datasets. Median rank: rank of the chosen ordering among the 24 on the held-out dataset (1 = best). Below random: held-out datasets on which the chosen ordering's regret is below the expected regret of a random ordering.

Clf.	Ordering chosen (folds)	Median rank	Regret	Random regret	Below random
LR	IMP→OUT→BAL→LNR (10)	4.5	0.50	1.46	8/10
RF	IMP→BAL→LNR→OUT (10)	5.0	0.60	1.86	9/10
XGB	IMP→BAL→LNR→OUT (6), BAL→LNR→IMP→OUT (4)	8.5	1.58	2.50	8/10

5 How class balancing changes label correction

The injector records which labels it flipped, so we can grade every decision the correction step makes. Table 11 compares LNR run directly after imputation with LNR run after imputation and balancing. Run before balancing, the correction step makes many wrong changes. On Adult it changes 2,451 labels, and 1,435 of those were correct before. All 1,435 broken labels belonged to genuine minority examples that it moved to the majority class. Credit Default (1,947 of 1,947 broken labels), Telescope (819 of 828) and Pima (43 of 46) follow the same pattern.

The reason lies in what the correction model sees. After asymmetric noise the observed minority class is small and contaminated, so the out-of-fold random forest

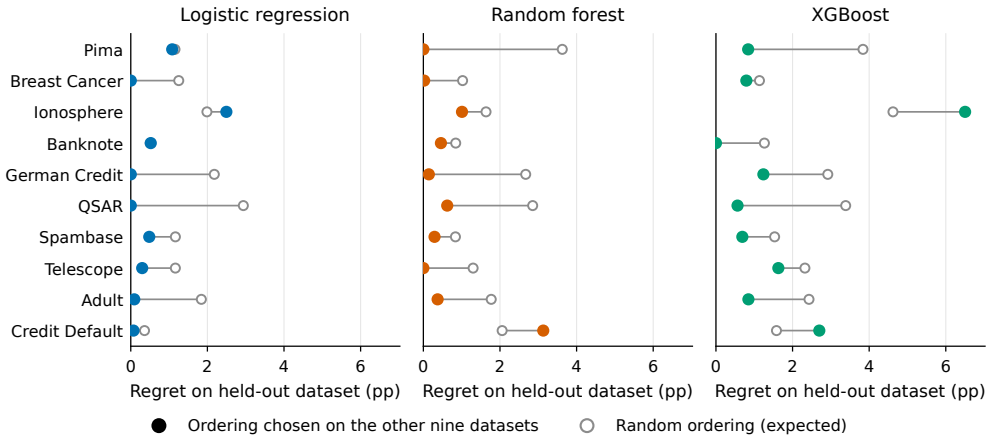


Fig. 5 Regret on each held-out dataset for the ordering chosen on the other nine (filled) and the expected regret of a random ordering (open), main condition, in pp.

leans towards the majority class. It then assigns high majority probability to genuine minority examples and "corrects" them. Balancing first raises the minority share to the target ratio before the correction model is fitted. With more minority examples to learn from, the model is less biased towards the majority class, and most of the harmful changes stop: on Adult, broken labels fall from 1,435 to 686, and correct fixes rise from 1,015 to 1,272. Precision rises on 9 of the 10 datasets (Fig. 6), from 0.41 to 0.65 on Adult and from 0.49 to 0.75 on Telescope, and changes by less than 0.01 on Banknote. On Banknote and Breast Cancer, the two datasets without a detectable order effect at five seeds (Table 6), precision is 0.90 or higher either way.

A reader could object that balancing first helps only because of a leak between folds. When balancing runs first, the cross-validation inside LNR contains synthetic examples interpolated from rows in other folds, so its out-of-fold predictions are not fully independent of the rows they score. We therefore reran the correction step after imputation with the two fold-safe variants of Section 3.7: SMOTE inside each training fold, and balanced class weights. Fold-internal SMOTE raised precision over IMP→LNR on 9 of the 10 datasets, almost as much as balancing first (0.64 against 0.65 on Adult, 0.72 against 0.75 on Telescope). Class weighting raised it on 7 of 10, most on Adult (0.41 to 0.73), Telescope (0.49 to 0.73) and Credit Default (0.35 to 0.54), and did not help on Banknote, Breast Cancer and Ionosphere, where precision was already 0.74 or higher. Dependence between folds therefore does not explain the gain, and the class balance the correction model sees is a major contributor to it.

The fold-safe variants also help the final classifier, in the end-to-end test of Section 3.7. Fold-internal SMOTE improved balanced accuracy on 8, 9 and 9 of the 10 datasets for logistic regression, random forest and XGBoost (Hodges–Lehmann estimates of 1.49, 1.39 and 1.87 points, with 95% intervals of 0.54 to 2.36, 0.58 to 2.46 and 0.61 to 2.94; exact Wilcoxon $p = 0.020$, 0.006 and 0.010), more consistently than the gain from placing balancing before correction (IMP→BAL→LNR→OUT), which was nominally significant only for logistic regression. The fold-safe gains remain significant

after Holm correction across the three classifiers. Class weighting also improved balanced accuracy on 7, 8 and 10 of the 10 datasets for logistic regression, random forest and XGBoost, respectively. Across the 30 dataset–classifier pairs, the gain from the fold-safe correction tracked the gain from balancing first (Spearman $\rho = 0.68$).

How much the order matters also depends on how aggressively labels are corrected; the threshold comparisons below are sensitivity analyses, not adjusted for multiplicity. At confidence thresholds of 0.6, 0.8 and 0.9, balancing first raised the precision of the correction step at every threshold, on 10, 10 and 7 of the 10 datasets (9 at the main threshold of 0.7), so the bias of a correction model fitted on imbalanced data is not an artifact of one threshold. The end-to-end gain, however, follows how many labels the step changes. At 0.6, balancing first won on 8, 10 and 10 of the 10 datasets for logistic regression, random forest and XGBoost (Hodges–Lehmann estimates of 2.04, 3.98 and 4.19 points, all intervals above zero). At 0.8 the order made no detectable difference (estimates of +0.25, −0.38 and −0.55 points in the same classifier order, all intervals including zero). At 0.9, where the correction step changes fewer labels and its precision is high either way (median 0.86 when it runs first), correcting first was slightly better for XGBoost (−0.92 points, 95% interval −1.63 to −0.50), with no detectable difference for the other two classifiers. Balancing first therefore matters most when the correction step is aggressive enough to break many correct labels.

Table 11 What label noise correction does, graded against the known injected flips: means over five seeds. Fixed: a flipped label restored. Broken: a correct label changed. Precision: fixed / relabeled. Only original rows are counted; synthetic rows have no ground-truth label.

	IMP→LNR				IMP→BAL→LNR			
Dataset	Relabeled	Fixed	Broken	Precision	Relabeled	Fixed	Broken	Precision
Pima	66.6	20.8	45.8	0.31	44.2	18.0	26.2	0.40
Breast Cancer	45.6	41.8	3.8	0.92	49.6	46.6	3.0	0.94
Ionosphere	22.4	16.4	6.0	0.74	16.2	13.4	2.8	0.83
Banknote	130.2	117.4	12.8	0.90	138.8	124.6	14.2	0.90
German Credit	101.2	28.8	72.4	0.28	42.8	16.6	26.2	0.38
QSAR	82.8	43.6	39.2	0.53	79.0	52.6	26.4	0.67
Spambase	368.2	288.8	79.4	0.78	406.6	349.8	56.8	0.86
Telescope	1609.8	782.0	827.8	0.49	1432.6	1073.6	359.0	0.75
Adult	2450.6	1015.2	1435.4	0.41	1958.8	1272.4	686.4	0.65
Credit Default	2986.2	1039.0	1947.2	0.35	1749.8	774.0	975.8	0.44

On three datasets, Pima, German Credit and Credit Default, the correction step breaks more labels than it fixes even after balancing, so at this threshold it leaves more wrong labels than it found. This fits the subset results: on German Credit the best pipeline for random forest leaves label correction out (Table 9). Credit Default is the one dataset where correcting labels first wins for both tree ensembles. There the correction step fixes more flipped labels when it runs before balancing (1,039 against 774). The end-to-end test suggests that this exception reflects how correction interacts with synthetic rows added before it, rather than evidence against balance itself: a correction step fitted on balanced folds and followed by balancing improved random forest and XGBoost on Credit Default by 1.93 and 0.73 points.

The symmetric-noise condition fits this account. When label errors fall on both classes at the same rate, the observed minority class is no more contaminated than

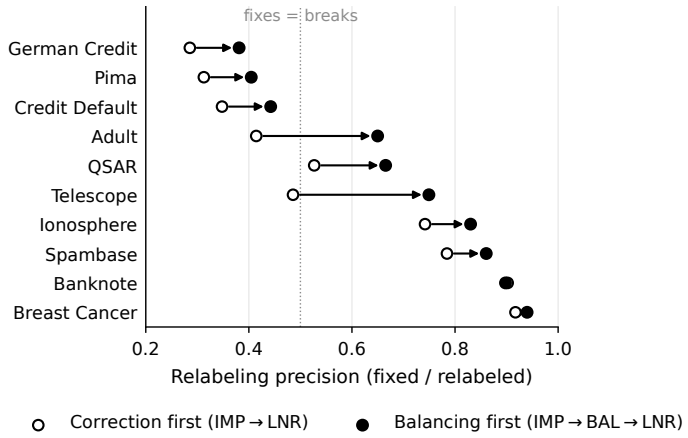


Fig. 6 Precision of label noise correction (restored flips divided by all relabels), graded against the known injected flips, when it runs directly after imputation (open) and after imputation and balancing (filled). Below 0.5 the step breaks more labels than it fixes.

the majority, the correction model has less reason to lean towards the majority, and balancing first no longer helps (Table 8).

A different interaction, between balancing and outlier removal, does not appear: Isolation Forest does not single out SMOTE examples once balancing has run. In IMP→BAL→OUT, synthetic examples make up a share of the removed rows equal to or below their share of the training set on all ten datasets. On Breast Cancer (0.19 of the data, 0.10 of the removals), Ionosphere (0.20 against 0.09) and QSAR (0.23 against 0.15), the detector spares them.

6 Discussion

6.1 Implications

The raw gap between the best and worst order overstates the evidence for order effects, because seed-to-seed variation alone accounts for a large part of it. The effects that remain beyond this baseline depend on the data: they grow with the rate of label noise concentrated in the minority class and are not detected when the same errors are spread evenly. Among the ordering decisions, the position of class balancing relative to label correction is the most consistent.

The mechanism of Section 5 bears on the disagreement described in Section 2.3. Balancing before correction has two opposing effects: it spreads label errors in the minority class into synthetic examples, and it removes the class-prior bias of the correction model. For the correction methods we tested, under minority-concentrated noise, the second effect outweighed the first. It also fits the filter-first advice of Szeghalmy and Fazekas (2024): their label noise was symmetric, the setting in which our symmetric-noise condition found no advantage for balancing first (Table 8), so which order is better depends on where the label errors fall. The result is consistent with the oversample-then-filter design of SMOTE-ENN and SMOTE-IPF (Batista et al. 2004;

Sáez et al. 2015), although we did not evaluate those methods themselves, nor the filters of Cheng et al. (2019) or Sağlam and Cengiz (2022), which may handle imbalance inside the filter.

For practice, four recommendations follow. First, state the order of the cleaning steps, since a pipeline cannot be reproduced without it. Second, judge order effects against the spread that seed-to-seed variation alone produces, and do not test a best ordering against a worst one selected from the same runs. Third, when correcting labels on imbalanced data whose errors may concentrate in the minority class, account for the imbalance in the correction model, by oversampling inside its training folds or with class weights, and report its confidence threshold. Fourth, a search over orderings is most likely to pay off when label errors are likely to fall on the minority class, and possibly when the data are strongly imbalanced, although our evidence for imbalance alone is limited; elsewhere, much of the spread it finds may be within chance.

The need for a chance baseline is not specific to these four steps: whenever many pipeline configurations are compared on a few replicates, the best and worst differ by chance, and more so as the number of configurations grows. The check requires no additional model fitting, because the baseline only reshuffles scores that such a comparison has already produced. Any set of interchangeable operators, including column-level choices such as imputing before or after discretizing a feature, can be assessed in the same way. Pipeline search and recommendation systems such as those in Section 2.4 could apply the same check before reporting one order as better than another. In applied machine learning, the order of preparation steps is part of the model's specification: it changes the data the learner is trained on, and it deserves the same reporting and validation as the choice of model and hyperparameters.

6.2 Limitations

Our claims operate at three levels that should not be conflated. The permutation calibration is methodological and does not depend on the four steps studied. The sizes of the order effects are specific to binary tabular data and the injected fault models. The advantage of balancing before correction is supported for minority-concentrated label noise and the correction methods tested; we do not claim that it holds universally, or that our stress settings represent typical real-world noise. Injected faults are a deliberate trade-off: only known errors let us grade each correction in Section 5, and the natural-data condition, with a median excess of 0.23 points of OSI against 0.94 in the main condition, suggests that the effects are smaller without them, although it compares 6 orderings rather than 24 and includes two more datasets. The asymmetric noise model flips 23–48% of minority labels in the main condition (Table 3), a stress test by design. The conclusions do not rest on that single noise level: the noise-rate conditions show the effect fading from 20% to 5%, and with the same rate of errors spread evenly across classes no order effect or consistent direction is detected.

The study covers four cleaning steps; encoding and scaling were held fixed, and steps such as feature selection and deduplication were not studied, so their interaction with the four steps remains open. We tested two methods for each step, not the full range of each family, and three classifier families; label noise was class-dependent in every condition, and instance-dependent noise, which concentrates errors near the class

boundary, or other learners such as support vector machines, neighbor-based classifiers and neural networks may behave differently. The correction step uses a fixed confidence threshold, 0.7 in all main results; the threshold analysis of Section 5 shows that the end-to-end benefit of balancing first depends on it, and a per-class threshold, as in the original Confident Learning, could respond to imbalance differently. Both outlier detectors are told the true outlier share of 5%, which favors them. The injected outliers also shift every numeric feature by at least 3.5 standard deviations, which makes them easy to detect and may explain why the position of outlier removal mattered little (Section 4.4). Nominal features are integer-encoded for outlier scoring, the correction forest and the classifiers, which imposes an order on their values; only fault injection, imputation and oversampling treat them as nominal. Pima keeps its zero codes for missing measurements.

All datasets are tabular and binary. In the main condition the imbalance ratio ranges from 1.25 to 3.52; stronger imbalance appears only in two datasets under an adapted noise model. Five seeds limit what can be tested within one dataset: the main significance counts likely understate small effects (Section 4.1), single paired comparisons are tested across datasets rather than within them, and ΔOSI is reported with permutation p-values rather than bootstrap intervals.

Two corrections were made during the study. An earlier version applied plain SMOTE and neighbor averaging to integer-encoded nominal features; after switching to the nominal-aware versions of Sections 3.3 and 3.4 and rerunning the three affected datasets, random forest on Adult moved from favoring correction first to favoring balancing first. An earlier public release of the code also fitted the temporary imputer inside label correction before its cross-validation split, and its oversampling filled the missing entries of original rows. Rerunning the seven smaller datasets with the original, partly fixed and final code gave identical results for orderings that neither change can affect. Affected runs moved by about one point of balanced accuracy, with mean signed shifts below 0.1 points, and the direction of the balancing–correction effect did not change.

7 Conclusion

The order of data cleaning steps matters, but less than raw comparisons suggest. In our main setting, seed-to-seed variation alone would produce an order sensitivity of 1 to 8%, and order effects beyond it appeared in half of the dataset–classifier pairs, worth 0.6 to 2.8 points of balanced accuracy. They grew with label noise, were largest in a two-dataset high-imbalance stress test with an adapted noise model, and were not detected when label errors were spread evenly across classes. For the methods and noise settings we evaluated, the most consistent practical lesson is to account for class imbalance in the correction model when label errors concentrate in the minority class; how much this helps end to end depends on the dataset and the correction threshold. A correction model fitted on imbalanced data relabels genuine minority examples, and fitting it on balanced folds removes much of that damage without relying on synthetic rows shared between folds. The advantage of balancing first is largest when correction is aggressive, shrinks at stricter thresholds and reverses slightly for XGBoost

at 0.9, and Credit Default shows that a single default will not fit every dataset. Two questions remain open: whether the picture holds under instance-dependent noise and in multiclass problems, and whether per-class correction thresholds, as in Confident Learning, remove the need to balance first.

Statements and Declarations

Funding. No funding was received for conducting this study.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent to participate. Not applicable. The study uses only publicly available, anonymized benchmark datasets.

Data availability. All datasets are public and are listed with their sources in Table 2. The code, the dataset files it uses and the summary results behind every table and figure are available at https://github.com/idrees118/data-cleaning-order-sensitivity. The per-run results are available from the authors on reasonable request.

Author contributions. **Muhammad Idrees:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft. **Zaheer Ullah:** Methodology, Validation, Investigation, Writing – review and editing. **Thanh Thi Nguyen:** Supervision, Methodology, Writing – review and editing. **Muhammad Taimour Hamid:** Validation, Writing – review and editing.

Appendix A Detailed results

Table A1 lists the best and worst ordering for every dataset and classifier in the main condition, with the spread of each over seeds; it underlies Fig. 1 and Table 6. Table A2 gives the values plotted in Fig. 3, with the number of seeds favoring correction first, and Table A3 the values behind Fig. 2.

Table A1 Best and worst ordering per dataset and classifier in the main condition: mean balanced accuracy ± standard deviation over five seeds.

Dataset	Clf.	Best ordering	Mean ± SD	Worst ordering	Mean ± SD
Pima	LR	IMP→BAL→LNR→OUT	0.7371 ± 0.0380	LNR→OUT→IMP→BAL	0.7169 ± 0.0492
	RF	IMP→BAL→LNR→OUT	0.7058 ± 0.0507	LNR→BAL→OUT→IMP	0.6297 ± 0.0321
	XGB	IMP→BAL→LNR→OUT	0.6941 ± 0.0415	LNR→BAL→IMP→OUT	0.6144 ± 0.0339
Breast Cancer	LR	IMP→OUT→BAL→LNR	0.9599 ± 0.0155	OUT→LNR→IMP→BAL	0.9345 ± 0.0206
	RF	LNR→IMP→BAL→OUT	0.9377 ± 0.0261	OUT→IMP→LNR→BAL	0.9183 ± 0.0320
	XGB	LNR→IMP→BAL→OUT	0.9369 ± 0.0157	IMP→OUT→LNR→BAL	0.9117 ± 0.0324
Ionosphere	LR	LNR→OUT→IMP→BAL	0.8350 ± 0.0475	BAL→OUT→LNR→IMP	0.7918 ± 0.0784
	RF	LNR→BAL→IMP→OUT	0.9033 ± 0.0236	OUT→BAL→IMP→LNR	0.8667 ± 0.0387
	XGB	LNR→IMP→BAL→OUT	0.8830 ± 0.0457	BAL→IMP→LNR→OUT	0.7867 ± 0.0664
Banknote	LR	IMP→LNR→OUT→BAL	0.9822 ± 0.0050	LNR→BAL→IMP→OUT	0.9732 ± 0.0137
	RF	IMP→LNR→BAL→OUT	0.9816 ± 0.0105	OUT→LNR→IMP→BAL	0.9669 ± 0.0156
	XGB	IMP→BAL→LNR→OUT	0.9784 ± 0.0116	LNR→BAL→OUT→IMP	0.9579 ± 0.0191
German Credit	LR	IMP→OUT→BAL→LNR	0.6400 ± 0.0469	LNR→IMP→BAL→OUT	0.5986 ± 0.0684
	RF	BAL→IMP→OUT→LNR	0.6224 ± 0.0258	OUT→LNR→BAL→IMP	0.5602 ± 0.0451
	XGB	BAL→OUT→IMP→LNR	0.6295 ± 0.0418	LNR→OUT→BAL→IMP	0.5662 ± 0.0330
QSAR	LR	IMP→OUT→BAL→LNR	0.8464 ± 0.0168	IMP→LNR→OUT→BAL	0.7988 ± 0.0074
	RF	BAL→OUT→LNR→IMP	0.8036 ± 0.0110	OUT→IMP→LNR→BAL	0.7482 ± 0.0107
	XGB	BAL→OUT→IMP→LNR	0.7932 ± 0.0207	IMP→OUT→LNR→BAL	0.7265 ± 0.0234
Spambase	LR	IMP→BAL→LNR→OUT	0.9024 ± 0.0068	IMP→LNR→BAL→OUT	0.8790 ± 0.0106
	RF	BAL→OUT→IMP→LNR	0.9136 ± 0.0072	LNR→OUT→IMP→BAL	0.8944 ± 0.0043
	XGB	BAL→OUT→IMP→LNR	0.9113 ± 0.0093	LNR→OUT→IMP→BAL	0.8805 ± 0.0114
Telescope	LR	IMP→BAL→LNR→OUT	0.7560 ± 0.0123	LNR→OUT→BAL→IMP	0.7333 ± 0.0117
	RF	IMP→BAL→LNR→OUT	0.8323 ± 0.0077	OUT→LNR→BAL→IMP	0.8034 ± 0.0226
	XGB	IMP→BAL→OUT→LNR	0.8309 ± 0.0053	OUT→LNR→BAL→IMP	0.7690 ± 0.0179
Adult	LR	OUT→BAL→IMP→LNR	0.7387 ± 0.0046	LNR→BAL→IMP→OUT	0.7046 ± 0.0068
	RF	BAL→IMP→OUT→LNR	0.7472 ± 0.0056	LNR→OUT→IMP→BAL	0.7132 ± 0.0073
	XGB	IMP→BAL→OUT→LNR	0.7056 ± 0.0052	LNR→OUT→BAL→IMP	0.6594 ± 0.0143
Credit Default	LR	IMP→BAL→LNR→OUT	0.6619 ± 0.0124	IMP→LNR→OUT→BAL	0.6548 ± 0.0089
	RF	IMP→LNR→BAL→OUT	0.6060 ± 0.0078	BAL→IMP→OUT→LNR	0.5620 ± 0.0081
	XGB	IMP→LNR→BAL→OUT	0.5640 ± 0.0105	OUT→IMP→BAL→LNR	0.5236 ± 0.0456

Table A2 Label noise correction first minus class balancing first, main condition: mean difference in balanced accuracy over 12 paired orderings and five seeds (pp), with the number of seeds (of 5) favoring correction first in brackets. Negative values favor balancing first.

Dataset	LR	RF	XGB
Pima	−0.71 (0)	−3.91 (0)	−4.57 (0)
Breast Cancer	−1.37 (1)	+0.11 (3)	+0.25 (4)
Ionosphere	+0.96 (2)	+1.12 (3)	+3.02 (4)
Banknote	+0.20 (4)	−0.30 (1)	−0.72 (0)
German Credit	−2.16 (2)	−3.79 (1)	−3.37 (0)
QSAR	−1.57 (0)	−4.31 (0)	−4.67 (0)
Spambase	−1.34 (0)	−1.12 (0)	−2.19 (0)
Telescope	−1.12 (0)	−0.85 (0)	−1.72 (0)
Adult	−2.93 (0)	−2.97 (0)	−3.81 (0)
Credit Default	−0.13 (2)	+3.33 (5)	+1.86 (5)

Table A3 Order effects against label noise rate and oversampling ratio, on the seven datasets common to these conditions (21 pairs each). The 15% / 0.8 row is the main condition restricted to the same datasets, with the correction recomputed over those 21 pairs.

Setting	Significant	Median OSI (%)	Median ΔOSI
Label noise 5%	0	2.93	0.24
Label noise 10%	7	2.97	0.35
Label noise 15%, ratio 0.8	9	4.05	0.68
Label noise 20%	11	6.54	1.34
Symmetric noise 15%	0	2.69	−0.04
Oversampling ratio 0.5	0	3.42	0.12
Oversampling ratio 1.0	6	4.22	0.59

References

Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6(1):20–29

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B 57(1):289–300

Berti-Équille L (2019) Learn2Clean: Optimizing the sequence of tasks for web data preparation. In: Proceedings of The Web Conference (WWW), pp 2580–2586, https://doi.org/10.1145/3308558.3313602

Breunig MM, Kriegel HP, Ng RT, et al (2000) LOF: Identifying density-based local outliers. In: Proceedings of ACM SIGMOD, pp 93–104

Chawla NV, Bowyer KW, Hall LO, et al (2002) SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16:321–357. https://doi.org/10.1613/jair.953

Cheng K, Zhang C, Yu H, et al (2019) Grouped SMOTE with noise filtering mechanism for classifying imbalanced data. IEEE Access 7:170668–170681. https://doi.org/10.1109/ACCESS.2019.2955086

Côté PO, Nikanjam A, Ahmed N, et al (2024) Data cleaning and machine learning: a systematic literature review. Automated Software Engineering 31(2):54. https://doi.org/10.1007/s10515-024-00453-w

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1–30

Emmanuel T, Maupong T, Mpoeleng D, et al (2021) A survey on missing data in machine learning. Journal of Big Data 8:140. https://doi.org/10.1186/s40537-021-00516-9

Feurer M, Klein A, Eggensperger K, et al (2015) Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems (NeurIPS), pp 2962–2970

Frénay B, Verleysen M (2014) Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems 25(5):845–869

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9):1263–1284

Hollander M, Wolfe DA, Chicken E (2013) Nonparametric Statistical Methods, 3rd edn. Wiley, Hoboken, NJ

Kelly M, Longjohn R, Nottingham K (2023) The UCI machine learning repository. https://archive.ics.uci.edu

Li P, Rao X, Blase J, et al (2021) CleanML: A study for evaluating the impact of data cleaning on ML classification tasks. In: Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE), pp 13–24

Li P, Chen Z, Chu X, et al (2023) DiffPrep: Differentiable data preprocessing pipeline search for learning over tabular data. Proceedings of the ACM on Management of Data 1(2):1–26

Li W, Halimi A, Vaidya J, et al (2025) Privacy-preserving verification of ML preprocessing via model behavior indicators. IEEE Transactions on Privacy 2:144–158. https://doi.org/10.1109/TP.2025.3628998

Little RJ, Rubin DB (2019) Statistical Analysis with Missing Data, 3rd edn. John Wiley & Sons, Hoboken, NJ

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp 413–422

Mahmud Sujon K, Binti Hassan R, Tusnia Towshi Z, et al (2024) When to use standardization and normalization: Empirical evidence from machine learning models and XAI. IEEE Access 12:135300–135314. https://doi.org/10.1109/ACCESS.2024.3462434

Malerba D, Pasquadibisceglie V (2024) Data-centric AI. Journal of Intelligent Information Systems 62(6):1493–1502. https://doi.org/10.1007/s10844-024-00901-9

Mohammed S, Budach L, Feuerpfeil M, et al (2025) The effects of data quality on machine learning performance on tabular data. Information Systems 132:102549. https://doi.org/10.1016/j.is.2025.102549

Nevin J, Lees M, Groth P (2025) The effects of mismatched train and test data cleaning pipelines on regression models: Lessons for practice. PeerJ Computer Science 11:e2793. https://doi.org/10.7717/peerj-cs.2793

Northcutt CG, Jiang L, Chuang IL (2021) Confident learning: Estimating uncertainty in dataset labels. Journal of Artificial Intelligence Research 70:1373–1411

Ojala M, Garriga GC (2010) Permutation tests for studying classifier performance. Journal of Machine Learning Research 11:1833–1863

Olson RS, Bartley N, Urbanowicz RJ, et al (2016) Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp 485–492, https://doi.org/10.1145/2908812.2908918

Puri A, Gupta MK (2021) Knowledge discovery from noisy imbalanced and incomplete binary class data. Expert Systems with Applications 181:115179. https://doi.org/10.1016/j.eswa.2021.115179

Sáez JA, Luengo J, Stefanowski J, et al (2015) SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences 291:184–203. https://doi.org/10.1016/j.ins.2014.08.051

Sağlam F, Cengiz MA (2022) A novel SMOTE-based resampling technique trough noise detection and the boosting procedure. Expert Systems with Applications 200:117023. https://doi.org/10.1016/j.eswa.2022.117023

Sancricca C, Cappiello C (2022) Supporting the design of data preparation pipelines. In: Proceedings of the 30th Italian Symposium on Advanced Database Systems (SEBD 2022), Tirrenia, Italy, pp 149–158

Sancricca C, Siracusa G, Cappiello C (2024) Enhancing data preparation: insights from a time series case study. Journal of Intelligent Information Systems 62(6):1503–1530. https://doi.org/10.1007/s10844-024-00867-8

Szeghalmy S, Fazekas A (2024) A comparative study on noise filtering of imbalanced data sets. Knowledge-Based Systems 301:112236. https://doi.org/10.1016/j.knosys.2024.112236

Troyanskaya O, Cantor M, Sherlock G, et al (2001) Missing value estimation methods for DNA microarrays. Bioinformatics 17(6):520–525

Van Buuren S, Groothuis-Oudshoorn K (2011) mice: Multivariate imputation by chained equations in R. Journal of Statistical Software 45(3):1–67

Vanschoren J, van Rijn JN, Bischl B, et al (2014) OpenML: Networked science in machine learning. ACM SIGKDD Explorations Newsletter 15(2):49–60

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics SMC-2(3):408–421