



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 14      **Issue:** IX      **Month of publication:** September 2026

**DOI:** <https://doi.org/10.22214/ijraset.2026.84946>

**[www.ijraset.com](http://www.ijraset.com)**

**Call:** ☎ 08813907089

**E-mail ID:** [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Depletion or Contamination? Decomposing Minority Label Noise in the Evaluation of Imbalance-Handling Strategies

Louis Bratawidjaja<sup>1</sup>, Hans Philemon Limanza<sup>2</sup>, Taufik Hidayat<sup>3</sup>

School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China

**Abstract:** *Class imbalance and unreliable data are usually studied apart, and when they are studied together, minority label noise is treated as a single mechanism. It is not. Recording a genuine positive as a negative both depletes the positive class and contaminates the negative class with an example that contradicts the positives that remain, and a study that reports only the resulting performance drop cannot say which mechanism a method is failing to survive. We separate them with paired corruption arms that share a random draw of affected rows, relabelling the selected positives in one arm and deleting exactly the same rows in the other, so that the gap between the arms isolates the cost of the contradictory labels. Crossing six imbalance-handling strategies with three classifiers over 20 benchmark datasets and one extremely imbalanced fraud dataset, and summarising robustness by the normalised area under the corruption curve, we find that mislabelling a training positive costs 0.038 of clean PR-AUC more than deleting the same row, so a substantial part of the damage is the contradictory label rather than the lost example. Comparing methods with the Friedman test blocked on datasets — rather than on cross-validation folds, a common error that inflates the statistic in proportion to the number of folds — the imbalance strategy separates where the classifier family does not, and only 1 of five strategies separates from applying none at all. Resampling is not free: the most aggressive strategy degrades the Brier score to 4.5× that of the unmodified baseline, a cost invisible to threshold-free discrimination metrics. The decomposition costs one experimental arm, applies to any study that injects minority label noise, and turns a measurement of how much a method degrades into a statement about what it is failing to survive.*

**Keywords:** *class imbalance; label noise; robustness; SMOTE; cost-sensitive learning; calibration; fraud detection.*

## I. INTRODUCTION

Two difficulties recur in applied classification, and in practice they arrive together. The first is class imbalance: the class of interest — a fraudulent transaction, a defective component, a diseased patient — is rare, so a model can achieve a high overall score while missing most of the cases that matter. The second is imperfect data: labels may be wrong or delayed, and features may be distorted by sensor faults, logging errors, or privacy-preserving transformations. Each difficulty has a mature literature of its own. Imbalance is addressed by resampling methods such as SMOTE [1], or by cost-sensitive learning that reweights the classes [2]; unreliable data is addressed by a separate family of noise-tolerant and noise-filtering techniques [3].

Their intersection is less settled, though it is not unexplored. Van Hulse and Khoshgoftaar examined learners under simultaneous imbalance and class noise [4], Sáez and colleagues designed SMOTE-IPF specifically for noisy and borderline minority examples [5], and a recent systematic review consolidates the area [6]. What that body of work has largely left open is a question about mechanism rather than about method ranking. When a minority label is wrong in the direction that real labelling processes produce — a genuine positive recorded as a negative — two things happen at once. The positive class is *depleted*, losing an example the learner needed, and the negative class is *contaminated*, gaining an example that sits in positive feature space and contradicts the positives that remain. A study that injects minority label noise and reports the resulting drop measures the sum of these two effects. It cannot say which one a given imbalance-handling strategy is failing to survive, and therefore cannot say what a more robust strategy would have to do differently. The distinction matters because the two mechanisms imply opposite remedies. If degradation is dominated by depletion, the deficit is one of information, and synthesising additional minority examples from the survivors — which is what oversampling does — is at least the right shape of response. If degradation is dominated by contamination, then oversampling is being asked to interpolate within a class whose boundary has been corrupted by examples now carrying the opposite label, and the appropriate response is filtering or robust weighting rather than synthesis. Pooling the two mechanisms into a single label-noise axis makes these cases indistinguishable.

This paper separates them. We inject minority label noise in two paired arms that share the same random draw of affected examples: a *flip* arm that relabels the selected positives as negatives, reproducing the standard asymmetric noise model, and a *drop* arm that deletes exactly those same rows. Because the draw is shared, the arms differ in one respect only — whether the affected examples remain in the training set bearing the wrong label, or are absent — so the gap between them isolates the cost of contradictory labels net of the cost of the lost positives. We call that gap the contamination cost.

We embed this decomposition in a broader robustness evaluation. Six imbalance-handling strategies are crossed with three classifiers and evaluated under increasing corruption on twenty benchmark problems with imbalance ratios from roughly 9:1 to 67:1, and — where noted — on the Credit Card Fraud dataset [7] as an extreme case near 600:1. Robustness is summarised by severity curves and their normalised areas rather than by a single operating point, following the methodology of corruption benchmarks in vision [8], and strategies are compared with the Friedman test blocked on datasets, as that procedure requires [9]. We ask four questions.

RQ1. How does the performance of common imbalance-handling strategies degrade as training-data corruption increases, and is the ordering of strategies stable across datasets?

RQ2. Of the damage done by minority label noise, how much is attributable to depletion of the positive class and how much to contamination of the negative class, and does that split depend on the imbalance-handling strategy or on the classifier?

RQ3. Do resampling methods provide a robustness advantage over cost-sensitive class weighting, or over applying no imbalance handling at all?

RQ4. What do these strategies cost in probability calibration, and how much of the apparent degradation under corruption is drift in the operating point rather than loss of ranking quality?

The remainder of the paper reviews related work (Section 2), describes the datasets, corruption models and evaluation protocol (Section 3), presents results (Section 4), discusses their implications (Section 5), and concludes (Section 6).

## II. RELATED WORK

### A. Learning from imbalanced data

Class imbalance has been studied extensively, and broad reviews establish that it degrades standard learners because most algorithms implicitly assume comparable class priors and symmetric misclassification costs [10, 11, 12, 13, 14]. The proposed remedies divide into two families. *Data-level* methods rebalance the training set before learning. The Synthetic Minority Over-sampling Technique (SMOTE) [1] creates new minority examples by interpolating between a minority point and its nearest minority neighbours, and later variants concentrate synthesis near the decision boundary, as in Borderline-SMOTE [15] and the adaptive ADASYN scheme [16]. Because pure oversampling can blur the boundary, hybrid methods combine it with cleaning rules such as Tomek links [17] or the Edited Nearest Neighbour rule [18], giving the SMOTE+Tomek and SMOTE+ENN procedures studied by Batista and colleagues [19]. Random undersampling instead discards majority examples to achieve balance, which is simple but sacrifices information [11]. *Algorithm-level* methods leave the data unchanged and adjust the learner, most commonly through cost-sensitive learning, whose theoretical foundations were set out by Elkan [2] and which is realised in practice as class weights or, in gradient boosting, a positive-class scaling factor [20]. Empirical surveys consistently report that no single method dominates across datasets [11, 12], and large studies confirm that the picture persists in deep models [21].

A more sceptical line of work questions whether rebalancing helps at all once the base learner is strong. Blagus and Lusa found that SMOTE confers little benefit for high-dimensional data and can be actively harmful for some classifiers [22]. Elor and Averbuch-Elor evaluated balancing across a large collection of tabular datasets and concluded that while weak learners benefit, state-of-the-art classifiers do not — a result that depends on using threshold-free metrics and tuned hyperparameters [23]. Van den Goorbergh and colleagues made the complementary point from the clinical-risk literature: imbalance corrections leave discrimination essentially unchanged while systematically destroying calibration, so corrected models produce risk estimates biased strongly upward [24]. The present paper takes that scepticism as its starting point and asks whether the remaining argument for resampling — that it may earn its keep when the data is unreliable — survives the same scrutiny.

### B. Learning with noisy and corrupted data

A separate literature studies learning when the data itself is unreliable. Label noise, in which training labels are wrong, is surveyed comprehensively by Frénay and Verleysen [3]; confident learning estimates and corrects such errors directly [25], and Song and colleagues survey the corresponding deep-learning methods [26].

Feature noise and distribution shift are commonly studied through corruption benchmarks that report performance as a function of severity rather than at a single operating point, an approach popularised by ImageNet-C [8]. That severity-curve methodology motivates our use of corruption-level curves and their summary areas.

### C. The intersection of imbalance and noise

Work at the intersection is smaller but substantial. Van Hulse and Khoshgoftaar studied eleven learners under jointly varied imbalance and class noise and concluded that simple sampling, particularly random undersampling, was generally the most effective response [4] — a conclusion drawn with the learners of its time, before gradient-boosted trees became a default tabular baseline. Seiffert and colleagues examined the same interaction in software-quality data and likewise found some sampling techniques more noise-robust than others [27]. Sáez and colleagues addressed the problem constructively with SMOTE-IPF, which adds an iterative partitioning filter to remove noisy and borderline examples before synthesis [5], and a family of related noise-filtering oversamplers has followed. A recent systematic review consolidates the area and identifies the separation of genuine minority patterns from corrupted ones as an open challenge [6].

Three gaps motivate the present study. First, the noise models used are typically symmetric or pairwise, whereas the error process in domains such as fraud detection is strongly asymmetric: delayed chargebacks and incomplete manual review produce genuine positives recorded as negatives far more often than the reverse. Second, degradation is usually reported at one or two noise levels rather than as a severity curve, which makes robustness hard to separate from baseline accuracy — a pipeline that starts low can retain a large fraction of very little. Third, and most importantly here, minority label noise is treated as a single mechanism, so depletion of the positive class and contamination of the negative class are never disentangled. We address all three.

### D. Performance evaluation under imbalance

Metric choice is critical under imbalance, because overall accuracy is dominated by the majority class. The precision-recall curve and its area are preferred to the ROC curve when positives are rare, since ROC analysis can present an overly optimistic view in that regime [28, 29].

Because discrimination and calibration can move independently, and resampling is known to move them in opposite directions [24], we report the Brier score alongside PR-AUC. For comparing many methods, Demšar recommends the non-parametric Friedman test followed by a Nemenyi post-hoc analysis and a critical-difference diagram, with datasets as the blocking factor [9]; Derrac and colleagues give a practical treatment of the same machinery [30]. Benavoli and colleagues note that mean-rank post-hoc tests depend on the pool of methods compared and recommend caution in their interpretation [31], which we observe by reporting pairwise signed-rank tests for the contrasts that carry our conclusions.

## III. METHODOLOGY

### A. Datasets

We evaluate on twenty-one binary imbalanced datasets, summarised in Table 1. Twenty are drawn from the KEEL imbalanced-classification collection [32], selected by three criteria applied before any modelling: an imbalance ratio of at least 8:1; at least thirty minority examples, so that removing 40% of them still leaves a usable positive class under five-fold cross-validation; and at most 15,000 rows and 250 features, so that the full factorial design remains computationally feasible. No dataset was added or removed after results were seen.

The twenty-first dataset is the Credit Card Fraud Detection dataset released by Dal Pozzolo and colleagues [7], included as an extreme case whose imbalance ratio is an order of magnitude beyond anything in the benchmark suite. It contains 284,807 European card transactions of which 492 are fraudulent. For confidentiality, twenty-eight of its features are anonymised principal components that are already centred and approximately scaled; the only raw features are the transaction amount and a time index. We retain the amount and standardise it, and we discard the time index, returning to that choice in the limitations. We remove 1,081 exactly duplicated rows before splitting, which also removes nineteen duplicated frauds and leaves 283,726 transactions of which 473 are fraudulent, a positive rate of 0.167%. Duplicate removal matters here because identical rows straddling a random split leak training data into the evaluation fold; this step is frequently omitted in published work on this dataset.

Table 1. Datasets used in the study, ordered by imbalance ratio (IR).  $n$  is the number of rows,  $p$  the number of features, and  $n_{\min}$  the number of minority examples.

| Dataset                        | $n$     | $p$ | $n_{\min}$ | IR    |
|--------------------------------|---------|-----|------------|-------|
| creditcard                     | 283,726 | 29  | 473        | 598.8 |
| shuttle_2_vs_5                 | 3,316   | 9   | 49         | 66.7  |
| abalone_19_vs_10_11_12_13      | 1,622   | 8   | 32         | 49.7  |
| abalone_17_vs_7_8_9_10         | 2,338   | 8   | 58         | 39.3  |
| kr_vs_k_three_vs_eleven        | 2,935   | 6   | 81         | 35.2  |
| winequality_red_4              | 1,599   | 11  | 53         | 29.2  |
| kr_vs_k_one_vs_fifteen         | 2,244   | 6   | 78         | 27.8  |
| kr_vs_k_zero_one_vs_draw       | 2,901   | 6   | 105        | 26.6  |
| car_vgood                      | 1,728   | 6   | 65         | 25.6  |
| car_good                       | 1,728   | 6   | 69         | 24.0  |
| flaref                         | 1,066   | 11  | 43         | 23.8  |
| abalone9_18                    | 731     | 8   | 42         | 16.4  |
| yeast_1_vs_7                   | 459     | 7   | 30         | 14.3  |
| shuttle_c0_vs_c4               | 1,829   | 9   | 123        | 13.9  |
| pc1                            | 1,109   | 21  | 77         | 13.4  |
| led7digit_0_2_4_5_6_7_8_9_vs_1 | 443     | 7   | 37         | 11.0  |
| vowel0                         | 988     | 13  | 90         | 10.0  |
| satimage                       | 6,435   | 36  | 626        | 9.3   |
| yeast_2_vs_4                   | 514     | 8   | 51         | 9.1   |
| page_blocks0                   | 5,472   | 10  | 559        | 8.8   |
| ecoli3                         | 336     | 7   | 35         | 8.6   |

All datasets are binary. The credit-card dataset is reported after duplicate removal.

## B. Corruption Models

We degrade only the training data, always on a copy, using three corruption processes at several intensity levels each. Level zero is the shared clean baseline. The conditions are listed in Table 2.

### 1) Minority label flipping (depletion and contamination)

We randomly select a fraction  $p$  of the training positives and relabel them as negatives, injecting false negatives only and never converting a negative into a positive. This asymmetry is deliberate. Real fraud labels are produced by delayed chargebacks and manual review, so the dominant error is a genuine fraud recorded as legitimate, which is exactly a minority-to-majority flip. A symmetric scheme flipping the same percentage of each class would be catastrophic at these imbalance ratios: on the most imbalanced dataset here, flipping even 5% of the majority would create spurious positives outnumbering the real ones by more than an order of magnitude, inverting the problem rather than corrupting it.

### 2) Minority deletion (depletion only)

We select a fraction  $p$  of the training positives using the same seeded draw as the flip arm and delete those rows entirely. The two arms therefore differ in exactly one respect: whether the affected examples remain in the training set bearing the wrong label, or are absent. Their difference at a given  $p$  isolates the cost of contradictory labels, net of the cost of the lost positives. To our knowledge this decomposition has not been reported for imbalance-handling strategies.

### 3) Feature noise (additive Gaussian)

Each continuous feature  $x$  receives independent Gaussian noise scaled by that feature's standard deviation  $\hat{s}$ , computed on the training fold only, so that  $x' = x + \epsilon$  with  $\epsilon \sim N(0, \sigma^2 \hat{s}^2)$ . Scaling by  $\hat{s}$  makes a given  $\sigma$  equally strong on every feature. This corresponds to a missing-completely-at-random style of measurement error, such as sensor drift, logging faults, or precision loss from anonymisation, and follows the severity-curve methodology of corruption benchmarks [8]. We extend  $\sigma$  well past the range at which the effect is negligible so that a breaking point can be located rather than merely bounded.

Table 2. The corruption conditions. All corruption is applied to training folds only; validation and test data are always clean.

| Condition | Parameter                                                          | Levels                  |
|-----------|--------------------------------------------------------------------|-------------------------|
| clean     | none                                                               | 0                       |
| flip      | fraction of training positives relabelled as negative, $p$         | 0.10, 0.20, 0.40        |
| drop      | fraction of training positives deleted, $p$ (same draw as flip)    | 0.10, 0.20, 0.40        |
| feat      | Gaussian noise scale $\sigma$ , in per-feature standard deviations | 0.1, 0.2, 0.3, 0.5, 1.0 |

*The flip and drop arms share a seeded draw of affected rows, making their difference a paired contrast.*

### C. Imbalance-handling strategies and classifiers

We cross six imbalance-handling strategies with three classifiers, giving eighteen pipelines (Table 3). The strategy none trains directly on the imbalanced data and serves as the baseline. The three SMOTE strategies oversample the minority class [1], synthesising a new positive by picking a minority point, choosing one of its  $k$  nearest minority neighbours, and placing a new point at a random position on the segment between them; we vary the target ratio of minority to majority through 0.2, 0.5 and 1.0, so that smote\_1.0 fully balances the classes. The strategy rus performs random undersampling, discarding majority examples uniformly at random until the classes are balanced. The strategy class\_weight applies cost-sensitive learning instead of resampling: it leaves the data untouched and reweights the loss so that errors on the rare class count more [2]. Cost-sensitive weighting is enabled *only* for the class\_weight strategy; every other strategy uses an unweighted classifier, which separates the effect of resampling from the effect of reweighting.

Logistic regression is a linear baseline with an L2 penalty. The random forest [33] is a bagging ensemble that reduces variance by averaging many de-correlated trees. XGBoost [20] is a gradient-boosted tree ensemble that builds trees sequentially to correct the errors of the current model. Hyperparameters are fixed at broadly reasonable values rather than tuned per pipeline, which keeps the eighteen-way comparison uniform and the computational cost manageable across twenty-one datasets and twelve conditions; we return to this choice in the limitations, since Elor and Averbuch-Elor show that tuning interacts with the apparent benefit of balancing [23]. On the benchmark suite every pipeline is preceded by a standardisation step fit inside the training fold; on the credit-card data the features are already on comparable scales and no further scaling is applied. All pipelines are implemented with scikit-learn [34] and imbalanced-learn [35].

Table 3. The six imbalance-handling strategies and three classifiers. The last column gives the native cost-sensitive setting, which is used only by the class\_weight strategy.

| Strategy     | Type           | Mechanism                                    |
|--------------|----------------|----------------------------------------------|
| none         | baseline       | train directly on the imbalanced data        |
| smote_0.2    | oversampling   | SMOTE to a minority:majority ratio of 0.2    |
| smote_0.5    | oversampling   | SMOTE to a ratio of 0.5                      |
| smote_1.0    | oversampling   | SMOTE to a ratio of 1.0 (full balance)       |
| rus          | undersampling  | random majority removal to 50:50             |
| class_weight | cost-sensitive | reweight the loss by inverse class frequency |

Table 4. Classifiers and their fixed hyperparameters.

| Classifier    | Family   | Hyperparameters                                  | Cost-sensitive setting               |
|---------------|----------|--------------------------------------------------|--------------------------------------|
| logreg        | linear   | C = 1, L2, liblinear, max_iter 1000              | class_weight = balanced              |
| random_forest | bagging  | 100 trees, max depth 10, min split 5             | class_weight =<br>balanced_subsample |
| xgboost       | boosting | 100 trees, max depth 6, lr 0.1, subsample<br>0.8 | scale_pos_weight = neg/pos           |

#### D. Evaluation protocol

The protocol is designed to prevent information leaking from validation data into training. All resampling is performed inside an imbalanced-learn pipeline [35], so a sampler is fit only on training folds and never sees validation data. Feature-noise standard deviations are likewise computed on the training fold only.

For each dataset and pipeline we proceed in two stages. First, we calibrate a decision threshold on *clean* training data using five-fold cross-validated out-of-fold probabilities, selecting the threshold that maximises minority-class F1. This threshold is then frozen for all later evaluations, mirroring deployment, where an operating point is fixed on historical clean data and then applied as the data later degrades. Second, we run repeated stratified cross-validation: for every fold and every corruption condition we corrupt only the training portion, refit the pipeline, and score it on the clean validation fold. The benchmark suite uses 3 repeats of 5-fold cross-validation; the credit-card dataset, whose size makes the random forest the dominant computational cost, uses 3 folds. All randomness is seeded and corruption draws use a stable per-cell hash, so every run is reproducible.

#### E. Metrics and robustness summaries

Our primary metric is the area under the precision-recall curve (PR-AUC, also called average precision). PR-AUC is threshold-free and concentrates on the rare positive class, and it is preferred to ROC-AUC under severe imbalance because the large number of true negatives can make ROC analysis look optimistic [28, 29]. We report the Brier score as a calibration diagnostic, since discrimination and calibration can move in opposite directions under resampling [24].

From PR-AUC we derive two robustness summaries per corruption type. Retention at level  $\ell$  is the ratio of PR-AUC at that level to PR-AUC on clean data, and so expresses the fraction of clean performance kept, isolating relative degradation:

$$\text{Retention}(\ell) = \text{PR-AUC}(\ell) / \text{PR-AUC}(0) \quad (1)$$

The normalised area under the corruption curve is the trapezoidal integral of PR-AUC over the severity range divided by the width of that range, which is the average PR-AUC across the range:

$$\text{nAUCC} = (1/\ell_{\max}) \int_0^{\ell_{\max}} \text{PR-AUC}(\ell) d\ell \quad (2)$$

Normalising by the range matters. The label and feature axes span different intervals, so unnormalised integrals are not on a common scale and cannot meaningfully be averaged into an overall figure — an error that is easy to make when summarising several corruption types together. Because nAUCC rewards pipelines that are both accurate and stable, it also avoids the trap of crediting a weak model that merely starts low and stays flat, which retention alone falls into.

Recall, precision, F1 and balanced accuracy are reported as secondary, threshold-based diagnostics at the frozen operating point. We additionally report each of them at an *oracle* threshold re-optimised on the validation fold itself. The oracle threshold is not attainable in deployment; it is a diagnostic upper bound, and the gap between frozen and oracle performance separates degradation in the model's ranking of risk from drift in the calibrated operating point.

#### F. Statistical comparison

We compare methods with the Friedman test followed by a Nemenyi post-hoc analysis and a critical-difference diagram [9]. The blocking factor is the *dataset*: each dataset contributes one score per method, and methods are ranked within datasets. We stress this because a common misapplication uses cross-validation folds as blocks. Folds of a single dataset are not independent replicates; using them inflates the Friedman statistic roughly in proportion to the number of folds and yields p-values that certify a comparison the design cannot support. A  $\chi^2$  statistic in the hundreds with eighteen methods is a symptom of this error rather than a finding.

Because the Nemenyi critical difference widens with the number of methods compared, and because mean-rank post-hoc tests depend on the pool of methods included [31], we run the omnibus comparison at three levels: across the six strategies, across the three classifiers, and across all eighteen pipelines. For the specific contrasts that carry our conclusions — flip against drop, and each resampling strategy against none — we additionally report paired Wilcoxon signed-rank tests over datasets with Holm correction for multiplicity. We report the Iman-Davenport correction of the Friedman statistic alongside the raw  $\chi^2$  [30].

#### IV. RESULTS

##### A. Clean baseline

Before measuring robustness we establish what is being retained. Table 5 ranks the eighteen pipelines on clean data across the 20 benchmark datasets by mean within-dataset rank of PR-AUC. Contrary to what a focus on classifier families would suggest, the ordering is governed more by the imbalance strategy than by the classifier: averaged over classifiers, mean rank spans 5.4 rank positions across strategies against 3.0 across classifiers. By classifier the means are 8.4 for the random forest, 8.7 for XGBoost, 11.4 for logistic regression; by strategy they are 7.8 for smote\_0.2, 8.1 for none, 9.1 for smote\_0.5, 9.1 for smote\_1.0, 9.7 for class\_weight, 13.2 for rus. All seven of the best-ranked pipelines use an ensemble classifier. The single best pipeline is smote\_0.2+random\_forest (mean rank 6.10, median PR-AUC 0.829); the worst is rus+sgboost (mean rank 14.15, median PR-AUC 0.708).

Table 5. Clean-data performance of all eighteen pipelines across the 20 benchmark datasets, ordered by mean within-dataset rank of PR-AUC (1 = best of 18).

| Pipeline                   | Mean rank | Median PR-AUC |
|----------------------------|-----------|---------------|
| smote_0.2+random_forest    | 6.10      | 0.829         |
| none+random_forest         | 6.40      | 0.829         |
| smote_0.2+sgboost          | 6.65      | 0.830         |
| none+sgboost               | 7.00      | 0.824         |
| smote_1.0+sgboost          | 7.53      | 0.825         |
| smote_0.5+random_forest    | 7.85      | 0.822         |
| class_weight+sgboost       | 8.12      | 0.825         |
| smote_0.5+sgboost          | 8.70      | 0.826         |
| smote_1.0+random_forest    | 8.95      | 0.810         |
| class_weight+random_forest | 9.45      | 0.809         |
| smote_0.2+logreg           | 10.57     | 0.665         |
| smote_0.5+logreg           | 10.70     | 0.665         |
| smote_1.0+logreg           | 10.88     | 0.667         |
| none+logreg                | 10.88     | 0.656         |
| class_weight+logreg        | 11.65     | 0.647         |
| rus+random_forest          | 11.72     | 0.777         |
| rus+logreg                 | 13.70     | 0.579         |
| rus+sgboost                | 14.15     | 0.708         |

Ranks are computed within each dataset and then averaged, so they are not affected by differences in absolute difficulty between datasets.

## B. Minority label noise: depletion versus contamination

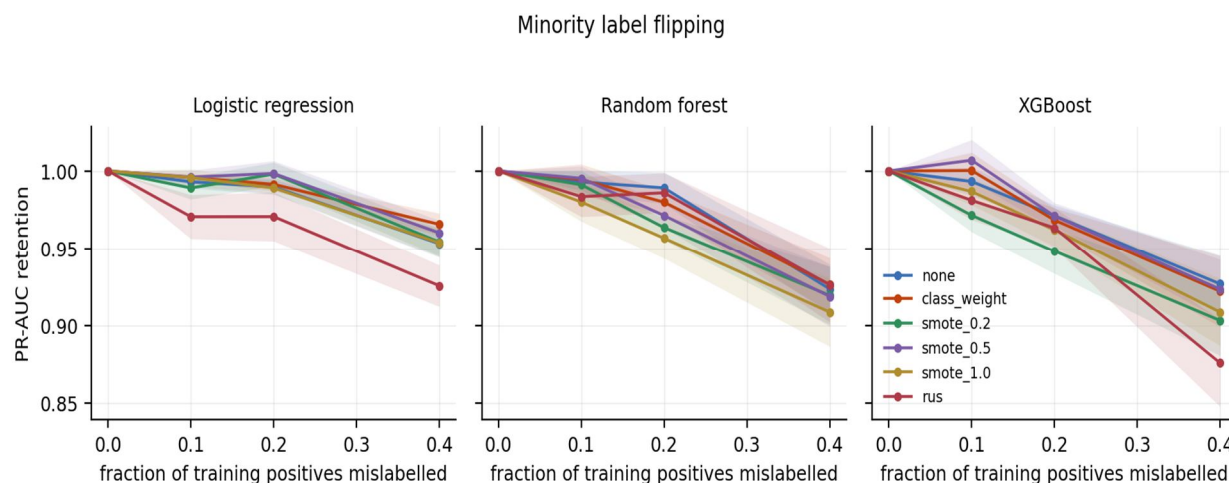


Figure 1. PR-AUC retention under minority label flipping, averaged across the benchmark datasets, one curve per imbalance-handling strategy and one panel per classifier. Shaded bands are one standard error across datasets.

Figure 1 shows the expected picture: performance declines as more training positives are mislabelled, and the decline is gentlest for logistic regression (retention 0.952 at the most severe level) and steepest for XGBoost (0.910). Figure 2 decomposes that decline. The solid curve is the flip arm, in which the affected positives are relabelled as negatives; the dashed curve is the drop arm, in which exactly the same rows are deleted. Because the two arms share their random draw, the vertical gap between them is the cost of the contradictory labels alone, with the loss of positives held constant.

Minority label noise decomposed: the gap between the curves is the cost of contradictory labels

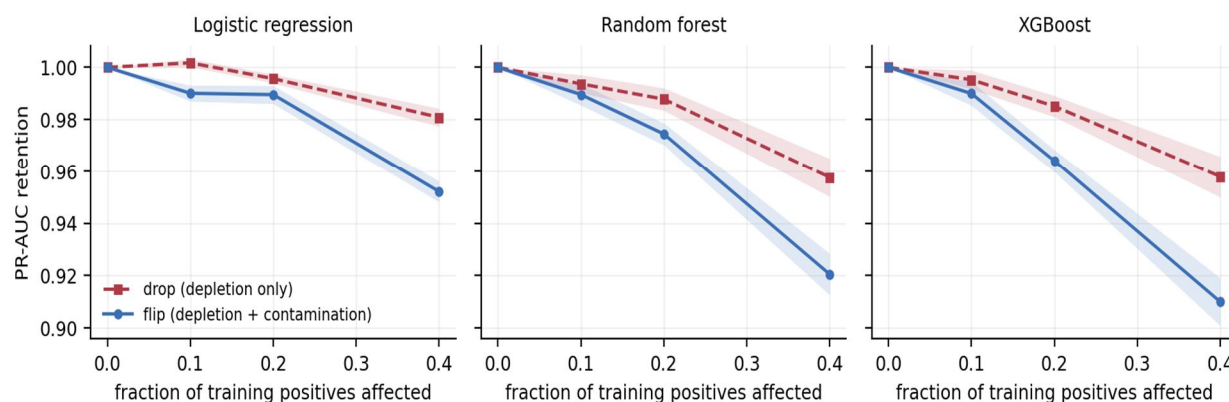


Figure 2. The decomposition. Solid curves relabel the selected training positives as negatives (depletion and contamination); dashed curves delete exactly the same rows (depletion only). The gap between the curves is the contamination cost. Averaged across the benchmark datasets; shaded bands are one standard error.

At the most severe level, with 40% of training positives affected, mislabelling retains 0.928 of clean PR-AUC while deleting the same rows retains 0.965. The contamination cost is therefore 0.038 of clean PR-AUC, which is significant by a paired Wilcoxon signed-rank test over the 20 datasets ( $p < 0.0001$ ). Keeping a mislabelled positive in the training set therefore costs more than losing them outright. Of the 0.072 of clean performance lost under mislabelling, 52% is attributable to contamination and the remainder to depletion.

Table 6. The depletion / contamination decomposition, averaged over pipelines within each dataset and then over the benchmark datasets. Retention is PR-AUC relative to clean data. The contamination cost is the difference between the two arms; a positive value means mislabelling hurts more than deleting.

| Positives affected | Retention (flip) | Retention (drop) | Contamination cost | Wilcoxon p |
|--------------------|------------------|------------------|--------------------|------------|
| 10%                | 0.990            | 0.997            | +0.007             | 0.0897     |
| 20%                | 0.976            | 0.990            | +0.014             | 0.0006     |
| 40%                | 0.928            | 0.965            | +0.038             | < 0.0001   |

*p-values are from a paired Wilcoxon signed-rank test blocked on datasets.*

The decomposition is not uniform across pipelines (Table 7). The contamination cost is largest for none+random\_forest (+0.072) and smallest for smote\_0.5+random\_forest (+0.020); 14 of the 18 pipelines show a contamination cost that survives Holm correction. Averaged over classifiers the ordering by strategy is none (+0.049), smote\_0.2 (+0.041), rus (+0.041), class\_weight (+0.038), smote\_1.0 (+0.033), smote\_0.5 (+0.026), and averaged over strategies the ordering by classifier is XGBoost (+0.048), the random forest (+0.037), logistic regression (+0.029).

Table 7. Contamination cost by pipeline at 40% of training positives affected, ordered from largest to smallest. Asterisks mark contrasts significant at  $\alpha = 0.05$  after Holm correction across the eighteen pipelines.

| Pipeline                   | Ret. (flip) | Ret. (drop) | Contam. cost | p (Holm) |
|----------------------------|-------------|-------------|--------------|----------|
| none+random_forest         | 0.924       | 0.995       | +0.072       | 0.003*   |
| rus+xgboost                | 0.876       | 0.938       | +0.062       | 0.118    |
| class_weight+xgboost       | 0.922       | 0.977       | +0.055       | 0.019*   |
| smote_1.0+xgboost          | 0.909       | 0.955       | +0.047       | 0.023*   |
| smote_0.2+xgboost          | 0.903       | 0.948       | +0.045       | 0.023*   |
| none+xgboost               | 0.927       | 0.970       | +0.043       | 0.008*   |
| smote_0.2+random_forest    | 0.919       | 0.961       | +0.041       | 0.007*   |
| smote_0.2+logreg           | 0.954       | 0.991       | +0.037       | 0.013*   |
| smote_0.5+xgboost          | 0.923       | 0.959       | +0.035       | 0.020*   |
| class_weight+random_forest | 0.926       | 0.962       | +0.035       | 0.059    |
| none+logreg                | 0.953       | 0.987       | +0.034       | 0.020*   |
| rus+logreg                 | 0.926       | 0.958       | +0.033       | 0.023*   |
| rus+random_forest          | 0.926       | 0.954       | +0.027       | 0.118    |
| smote_1.0+random_forest    | 0.909       | 0.936       | +0.027       | 0.043*   |
| smote_1.0+logreg           | 0.954       | 0.978       | +0.024       | 0.016*   |
| class_weight+logreg        | 0.966       | 0.989       | +0.023       | 0.023*   |
| smote_0.5+logreg           | 0.960       | 0.982       | +0.022       | 0.023*   |
| smote_0.5+random_forest    | 0.919       | 0.939       | +0.020       | 0.118    |

### C. Feature Noise

Feature noise behaves quite differently. Across the benchmark suite, retention at  $\sigma = 0.1$  is 0.978 and remains at 0.955 or better throughout the range  $\sigma \leq 0.3$  that is commonly tested. Only when the noise scale reaches  $\sigma = 1$  — noise of the same order as the feature's own standard deviation — does retention fall to 0.830. Reporting this axis only up to  $\sigma = 0.3$ , as is common, would have shown a flat line and supported the conclusion that these models are simply insensitive to feature noise; extending the range locates where that ceases to be true.

Table 8. PR-AUC retention under additive Gaussian feature noise, averaged across the benchmark datasets and pipelines.

| $\sigma$ | All   | Logistic regression | Random forest | XGBoost |
|----------|-------|---------------------|---------------|---------|
| 0.1      | 0.978 | 0.994               | 0.966         | 0.973   |
| 0.2      | 0.976 | 0.982               | 0.966         | 0.979   |
| 0.3      | 0.955 | 0.972               | 0.939         | 0.953   |
| 0.5      | 0.912 | 0.948               | 0.893         | 0.896   |
| 1        | 0.830 | 0.890               | 0.801         | 0.798   |

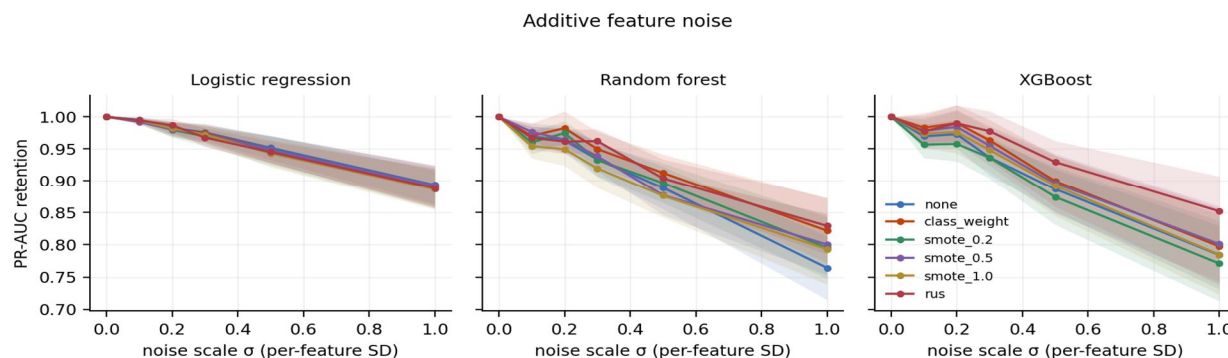


Figure 3. PR-AUC retention under additive feature noise, one curve per strategy and one panel per classifier, averaged across the benchmark datasets.

#### D. Robustness ranking and statistical comparison

We now rank pipelines by normalised area under the corruption curve (nAUCC), averaged over the three corruption types, and test the resulting ordering. Table 9 gives the ranking; Figures 4 to 6 give the critical-difference diagrams.

Table 9. Robustness ranking by normalised area under the corruption curve (nAUCC), averaged across the benchmark datasets. Higher is more robust. Because nAUCC is the mean PR-AUC across the severity range, it rewards pipelines that are both accurate and stable.

| Pipeline                   | flip  | drop  | feat  | Overall |
|----------------------------|-------|-------|-------|---------|
| none+random_forest         | 0.699 | 0.709 | 0.641 | 0.683   |
| smote_0.2+random_forest    | 0.693 | 0.703 | 0.653 | 0.683   |
| none+xgboost               | 0.694 | 0.705 | 0.646 | 0.681   |
| smote_0.2+xgboost          | 0.690 | 0.704 | 0.647 | 0.680   |
| smote_0.5+xgboost          | 0.688 | 0.700 | 0.647 | 0.678   |
| class_weight+xgboost       | 0.688 | 0.701 | 0.645 | 0.678   |
| smote_1.0+xgboost          | 0.687 | 0.702 | 0.645 | 0.678   |
| smote_0.5+random_forest    | 0.684 | 0.693 | 0.640 | 0.672   |
| class_weight+random_forest | 0.685 | 0.689 | 0.641 | 0.672   |
| smote_1.0+random_forest    | 0.675 | 0.685 | 0.629 | 0.663   |
| rus+random_forest          | 0.653 | 0.660 | 0.612 | 0.642   |
| rus+xgboost                | 0.597 | 0.606 | 0.590 | 0.598   |
| none+logreg                | 0.592 | 0.599 | 0.580 | 0.590   |

| Pipeline            | flip  | drop  | feat  | Overall |
|---------------------|-------|-------|-------|---------|
| smote_0.2+logreg    | 0.593 | 0.601 | 0.577 | 0.590   |
| smote_0.5+logreg    | 0.592 | 0.597 | 0.574 | 0.588   |
| smote_1.0+logreg    | 0.590 | 0.595 | 0.572 | 0.586   |
| class_weight+logreg | 0.588 | 0.593 | 0.570 | 0.584   |
| rus+logreg          | 0.558 | 0.564 | 0.549 | 0.557   |

Across the 20 datasets the Friedman test over 6 imbalance-handling strategies gives  $\chi^2 = 53.8$  ( $p < 0.0001$ ; Iman-Davenport  $F = 22.2$ ,  $p < 0.0001$ ), with a Nemenyi critical difference of 1.69 rank positions. The mean-rank ordering is smote\_0.2 (2.10), none (2.65), smote\_0.5 (2.85), class\_weight (3.40), smote\_1.0 (4.05), rus (5.95). Strategies separated by more than the critical difference are distinguishable at  $\alpha = 0.05$ ; those joined by a bar in Figure 4 are not.

Imbalance-handling strategies (Friedman  $p = 2.27\text{e-}10$ ,  $N = 20$  datasets)

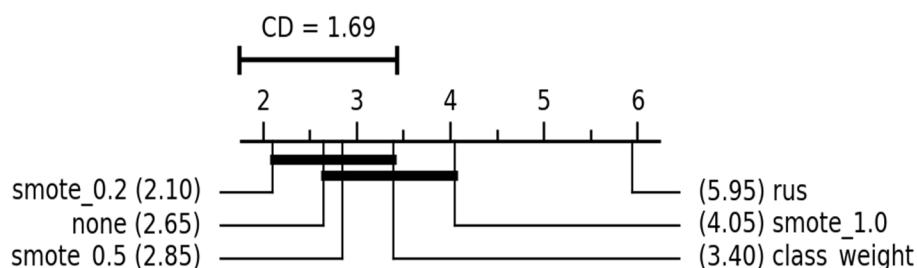


Figure 4. Critical-difference diagram for the six imbalance-handling strategies, ranked by nAUC across the benchmark datasets. Lower rank is better. Strategies joined by a bar are not significantly different at  $\alpha = 0.05$  by the Nemenyi test.

Across the 20 datasets the Friedman test over 3 classifiers gives  $\chi^2 = 3.7$  ( $p = 0.1572$ ; Iman-Davenport  $F = 1.9$ ,  $p = 0.1582$ ), with a Nemenyi critical difference of 0.74 rank positions. The ordering is the random forest (1.65), XGBoost (2.15), logistic regression (2.20).

Classifiers (Friedman  $p = 1.57\text{e-}01$ ,  $N = 20$  datasets)

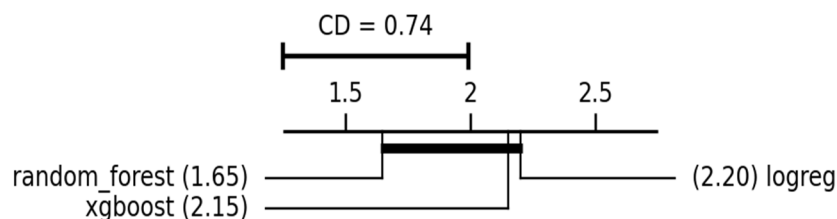


Figure 5. Critical-difference diagram for the three classifiers, ranked by nAUC across the benchmark datasets.

Across the 20 datasets the Friedman test over 18 pipelines gives  $\chi^2 = 74.0$  ( $p < 0.0001$ ; Iman-Davenport  $F = 5.3$ ,  $p < 0.0001$ ), with a Nemenyi critical difference of 5.89 rank positions. With eighteen methods and 20 datasets the critical difference is 5.89 rank positions, which is wide relative to the spread of the ranking; this is the expected behaviour of the Nemenyi test as the pool of compared methods grows [31], and is the reason we report the strategy-level and classifier-level comparisons separately.

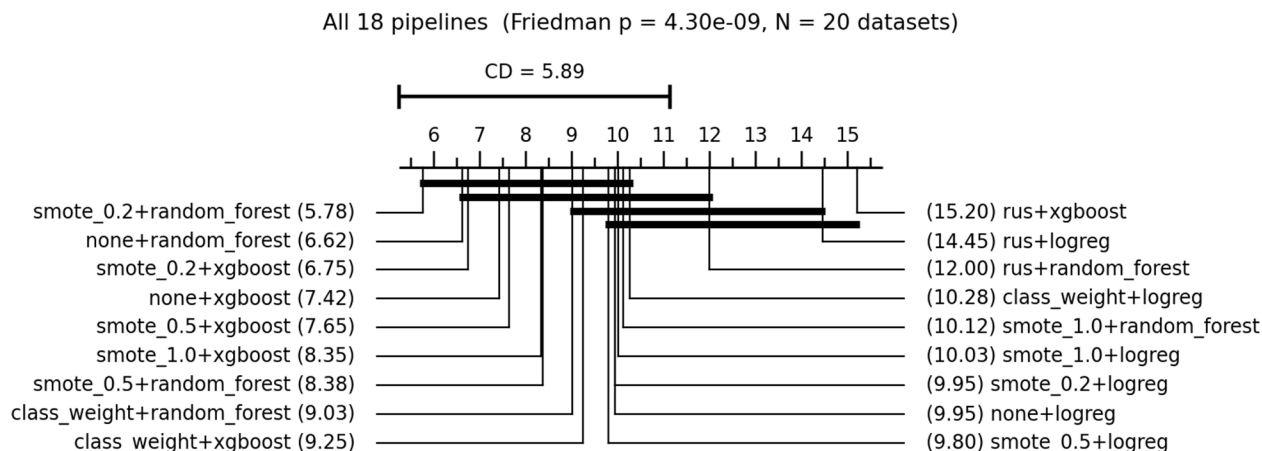


Figure 6. Critical-difference diagram for all eighteen pipelines, ranked by nAUCC across the benchmark datasets.

Table 10 tests each strategy directly against applying no imbalance handling at all, pairing on datasets. 1 of the five differences survive Holm correction: rus (-0.053,  $p < 0.0001$ ). The largest point difference in favour of any alternative is smote\_0.2 at -0.000 nAUCC.

Table 10. Each imbalance-handling strategy against none, paired on datasets, using nAUCC averaged over corruption types and classifiers. Asterisks mark differences significant at  $\alpha = 0.05$  after Holm correction.

| Strategy     | nAUCC | nAUCC (none) | Difference | p (Holm) |
|--------------|-------|--------------|------------|----------|
| class_weight | 0.644 | 0.652        | -0.007     | 0.160    |
| smote_0.2    | 0.651 | 0.652        | -0.000     | 0.596    |
| smote_0.5    | 0.646 | 0.652        | -0.005     | 0.405    |
| smote_1.0    | 0.642 | 0.652        | -0.010     | 0.054    |
| rus          | 0.599 | 0.652        | -0.053     | < 0.001* |

### E. Calibration

Discrimination is not the only thing these strategies affect. Table 11 reports the Brier score on clean data relative to none, paired within dataset and classifier. A ratio above one means worse calibration than doing nothing. The largest degradation is rus, whose median Brier score is 4.5× that of none ( $p < 0.0001$ ). The ordering follows the aggressiveness of the rebalancing: the more the training distribution is altered, the further the predicted probabilities drift from the observed event rate. This reproduces, for resampling on imbalanced tabular data, the effect that van den Goorbergh and colleagues document for clinical risk models [24], and it is invisible to PR-AUC, which depends only on the ranking of scores.

Table 11. Clean-data Brier score relative to the none baseline, paired within dataset and classifier across the benchmark suite. Values above 1 indicate worse calibration than applying no imbalance handling.

| Strategy     | Median ratio | Mean ratio | Wilcoxon p |
|--------------|--------------|------------|------------|
| rus          | 4.47×        | 36.71×     | < 0.001    |
| smote_1.0    | 1.42×        | 1.82×      | < 0.001    |
| smote_0.5    | 1.24×        | 1.46×      | < 0.001    |
| class_weight | 1.19×        | 1.89×      | < 0.001    |
| smote_0.2    | 1.03×        | 1.14×      | < 0.001    |

*The Brier score is a proper scoring rule, so it rewards calibrated probabilities rather than merely well-ordered ones.*

### F. Frozen against oracle thresholds

Finally we separate loss of ranking quality from drift in the operating point. Under 40% mislabelling, recall at the frozen threshold changes by -0.238 and precision by -0.042: the calibrated rule becomes more conservative, firing less often but being right more often when it does. Under an oracle threshold re-optimised on the evaluation fold, the same models lose only 0.014 of recall, so roughly 94% of the apparent recall loss is operating-point drift rather than degraded ranking. This matters in practice: it is the failure mode a fraud team experiences as missed cases, it is invisible in PR-AUC, and it is partly recoverable by re-calibrating the threshold rather than by changing the imbalance strategy.

Table 12. Threshold-based performance at the frozen operating point and at an oracle threshold re-optimised on the evaluation fold, averaged over pipelines and benchmark datasets. The oracle is an unattainable upper bound reported for diagnosis only.

| Condition | PR-AUC | Recall | Precision | F1    | Recall (oracle) | F1 (oracle) |
|-----------|--------|--------|-----------|-------|-----------------|-------------|
| clean     | 0.662  | 0.703  | 0.630     | 0.640 | 0.754           | 0.694       |
| flip 40%  | 0.627  | 0.465  | 0.588     | 0.475 | 0.740           | 0.662       |
| drop 40%  | 0.647  | 0.623  | 0.621     | 0.596 | 0.747           | 0.679       |

### G. Extreme imbalance: the credit-card case study

The benchmark suite tops out near 67:1. The credit-card dataset is roughly 600:1, and serves as a check on whether the preceding conclusions survive an order of magnitude more imbalance. On clean data the best pipeline is none+*xgboost* with PR-AUC 0.844, and the clean ordering is driven more by the imbalance strategy than by the classifier (a spread of 0.144 PR-AUC across strategies against 0.105 across classifiers) (Table 13).

Table 13. Clean-data PR-AUC for all eighteen pipelines on the credit-card dataset, with the standard deviation across cross-validation folds.

| Pipeline                           | PR-AUC | SD across folds |
|------------------------------------|--------|-----------------|
| none+ <i>xgboost</i>               | 0.844  | 0.019           |
| class_weight+ <i>xgboost</i>       | 0.834  | 0.017           |
| none+ <i>random_forest</i>         | 0.831  | 0.024           |
| smote_0.2+ <i>xgboost</i>          | 0.822  | 0.027           |
| smote_0.5+ <i>xgboost</i>          | 0.821  | 0.031           |
| smote_1.0+ <i>xgboost</i>          | 0.818  | 0.030           |
| smote_0.2+ <i>random_forest</i>    | 0.817  | 0.036           |
| smote_0.5+ <i>random_forest</i>    | 0.808  | 0.022           |
| class_weight+ <i>random_forest</i> | 0.805  | 0.025           |
| smote_1.0+ <i>random_forest</i>    | 0.794  | 0.035           |
| none+ <i>logreg</i>                | 0.739  | 0.031           |
| smote_0.2+ <i>logreg</i>           | 0.726  | 0.066           |
| rus+ <i>random_forest</i>          | 0.725  | 0.049           |
| class_weight+ <i>logreg</i>        | 0.718  | 0.053           |
| smote_0.5+ <i>logreg</i>           | 0.718  | 0.071           |
| smote_1.0+ <i>logreg</i>           | 0.717  | 0.060           |
| rus+ <i>xgboost</i>                | 0.684  | 0.051           |
| rus+ <i>logreg</i>                 | 0.574  | 0.203           |

The decomposition reproduces the benchmark result: at 40% mislabelling, retention is 0.935 under flipping against 0.975 under deletion, a contamination cost of +0.040. Its magnitude is comparable to the benchmark suite, where the same contrast gives +0.038. With a single dataset this is a descriptive comparison rather than a test, since there is no blocking factor over which to compute a signed-rank statistic.

Table 14. The depletion / contamination decomposition on the credit-card dataset, averaged over the eighteen pipelines.

| Positives affected | Retention (flip) | Retention (drop) | Contamination cost |
|--------------------|------------------|------------------|--------------------|
| 10%                | 0.976            | 0.998            | +0.022             |
| 20%                | 0.976            | 0.987            | +0.011             |
| 40%                | 0.935            | 0.975            | +0.040             |

Estimated from three cross-validation folds on 473 positives, so the per-level figures are less stable than the benchmark-suite values, which average 15 folds over 20 datasets. The intermediate levels are not monotone, which is the expected consequence of that variance and should not be read as structure.

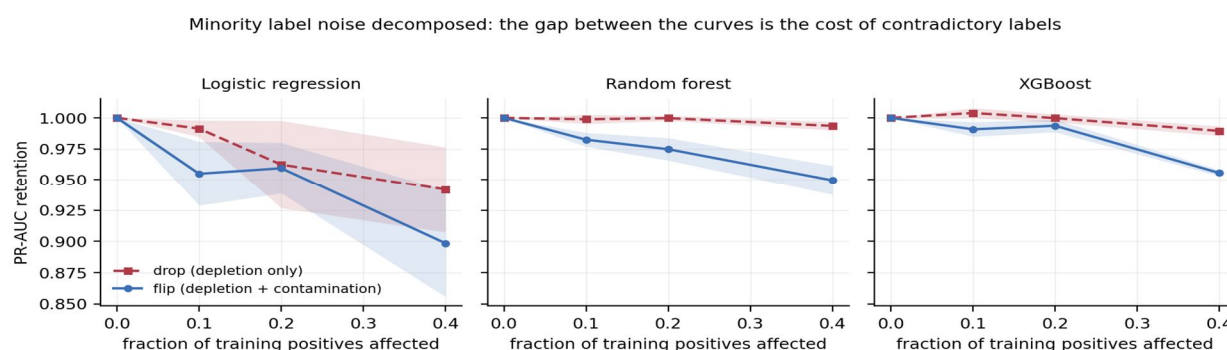


Figure 7. PR-AUC retention on the credit-card dataset under minority label flipping (solid) and deletion (dashed), one panel per classifier.

## V. DISCUSSION

### A. What minority label noise actually does

The decomposition gives a sharper answer than a single label-noise axis can. At 40% corruption the two arms differ by 0.038 of clean PR-AUC, so a mislabelled positive left in the training set is materially worse than the same positive removed. Contamination accounts for roughly 52% of the total degradation and depletion for the rest. The implication is practical. If the dominant mechanism were depletion, the deficit would be one of information, and synthesising more minority examples from the survivors would be the right kind of response. Because a substantial share is contamination, oversampling is being asked to interpolate within a class whose boundary is being actively contradicted by examples now labelled as their opposite, and adding more synthetic points on that boundary cannot repair it. This is the mechanism that noise-filtering oversamplers such as SMOTE-IPF [5] target, and our measurement supports the premise behind them: the filtering step, not the synthesis step, is where the leverage is.

The split is not the same for every learner. Contamination cost is largest for XGBoost (+0.048) and smallest for logistic regression (+0.029). The ordering runs with model capacity rather than against it, which is the reverse of what one might expect if contradictory points were simply dragging a decision boundary around. A linear model cannot represent a pocket of positives labelled negative: it is too rigid to be moved far by a handful of contradictory points, and it effectively averages over them. A boosted ensemble has exactly the capacity required to carve those points out and fit them, so the contradiction is absorbed into the model rather than smoothed away. This is the tabular counterpart of the memorisation of noisy labels documented in deep networks [26], and it carries an uncomfortable implication: the learners that are most accurate on clean data are also the ones that most faithfully learn the labelling errors. This design can measure the effect but not establish the mechanism behind it; separating capacity from inductive bias — by varying tree depth and regularisation strength while holding the corruption fixed — is the natural next experiment.

### B. Does imbalance handling buy robustness?

The answer is qualified. *rus* differs from none by  $-0.053$  nAUCC ( $p < 0.0001$ ). The remaining strategies are indistinguishable from applying no imbalance handling at all. Even where a difference is detectable, its size should be weighed against the calibration cost reported in Section 4.5 and the additional computation resampling requires. The strategy ranking spans 3.85 rank positions against a critical difference of 1.69 (omnibus  $p < 0.0001$ ), and the classifier ranking spans 0.55 against a critical difference of 0.74 (omnibus  $p = 0.1572$ ). The comparison that separates is therefore between imbalance strategies rather than between classifiers. This is worth stating plainly because it runs against the intuition — and against what a single-dataset study of this kind would suggest — that the learner is what matters. Across these datasets the three classifier families are not distinguishable by robustness, while the strategies are, and the strategy that separates most clearly is the one that discards data.

### C. Reconciliation with earlier findings

Van Hulse and Khoshgoftaar concluded that random undersampling was generally the most effective response to combined imbalance and class noise [4], and Seiffert and colleagues reached a similar conclusion on software-quality data [27]. Our ranking does not reproduce that, and the discrepancy is worth being explicit about rather than passing over. Here the best *rus* pipeline ranks 11 of eighteen by nAUCC. Two differences in design explain most of it. First, those studies predate gradient-boosted trees as a default tabular baseline; undersampling helps most when the learner is weak or cannot be made cost-sensitive, and much less when it is a strong ensemble that already handles skew. Second, and more importantly, undersampling scores well on *relative* retention precisely because it starts from a depressed clean baseline — discarding most of the negatives throws away information before any corruption is applied. A metric that normalises by clean performance will reward that, which is why we rank by nAUCC, an absolute quantity, and treat retention as diagnostic rather than as the criterion. Robustness that is purchased by being bad to begin with is not useful.

### D. The cost of the statistics

One methodological point deserves emphasis because it is easy to get wrong and hard to see afterwards. Demšar's procedure blocks on datasets [9]. If cross-validation folds are used as blocks instead, the Friedman statistic is inflated roughly in proportion to the number of folds, because folds of one dataset are correlated replicates rather than independent problems. The resulting p-values can be astronomically small while the design supports no population-level conclusion at all — the test is answering 'do these methods differ on this one dataset', dressed as 'do these methods differ in general'. A  $\chi^2$  in the hundreds over eighteen methods should be read as a diagnostic of this error. We report the dataset-blocked test throughout, which is why our critical differences are wide and several comparisons come back inconclusive. That is the honest resolution the design can support.

### E. Limitations

The corruptions are synthetic. Gaussian feature noise is of the missing-completely-at-random type and the label noise is one-directional; neither covers not-missing-at-random mechanisms, covariate shift, or adversarial perturbation, and real labelling errors are unlikely to be uniformly distributed over the positives. Hyperparameters are fixed rather than tuned per pipeline, which keeps the comparison uniform but leaves open whether tuning would change the strategy ordering; Elor and Averbuch-Elor find that it interacts with the value of balancing [23]. The study is restricted to binary classification and to three classifier families, and omits hybrid cleaning methods such as SMOTE+Tomek and SMOTE+ENN [19], and the noise-aware oversamplers [5] that our mechanism result most directly bears on — testing those against the decomposition is the obvious next step.

Two limitations are specific to the credit-card dataset. Its transactions are ordered in time over two days, and we discard the time index and split at random; a temporal split would be the more realistic protocol and is generally less optimistic, so the absolute figures on that dataset should be read as an upper bound. And because it is a single dataset, the case study in Section 4.7 is descriptive: it can show whether the suite conclusions survive at 600:1 imbalance, but it cannot furnish a test.

## VI. CONCLUSION

We evaluated eighteen imbalance-handling pipelines under controlled training-data corruption on 20 benchmark datasets and one extremely imbalanced fraud dataset, summarising robustness by normalised area under the corruption curve and comparing methods with the Friedman test blocked on datasets. Three conclusions follow. First, minority label noise is not one mechanism but two, and they can be separated: deleting a training positive and mislabelling it differ by  $+0.038$  of clean PR-AUC at 40% corruption, so the contradictory label carries a cost distinct from the lost example.

Second, the choice of imbalance strategy separates for robustness where the classifier family does not, and only some strategies separate from applying none at all. Third, resampling is not free: it degrades probability calibration even where it leaves ranking untouched, which a PR-AUC-only evaluation cannot see.

The decomposition is the part we would most like to see reused. It costs one additional experimental arm, it is applicable to any study that injects minority label noise, and it converts a statement about how much a method degrades into a statement about what it is failing to survive. The natural continuations are to test noise-aware oversamplers against it, to extend it to instance-dependent noise where contamination is concentrated near the boundary rather than spread uniformly, and to ask whether the classifier-dependence of the contamination cost reflects the locality of the decision boundary, as Section 5.1 conjectures.

#### A. Data and code availability

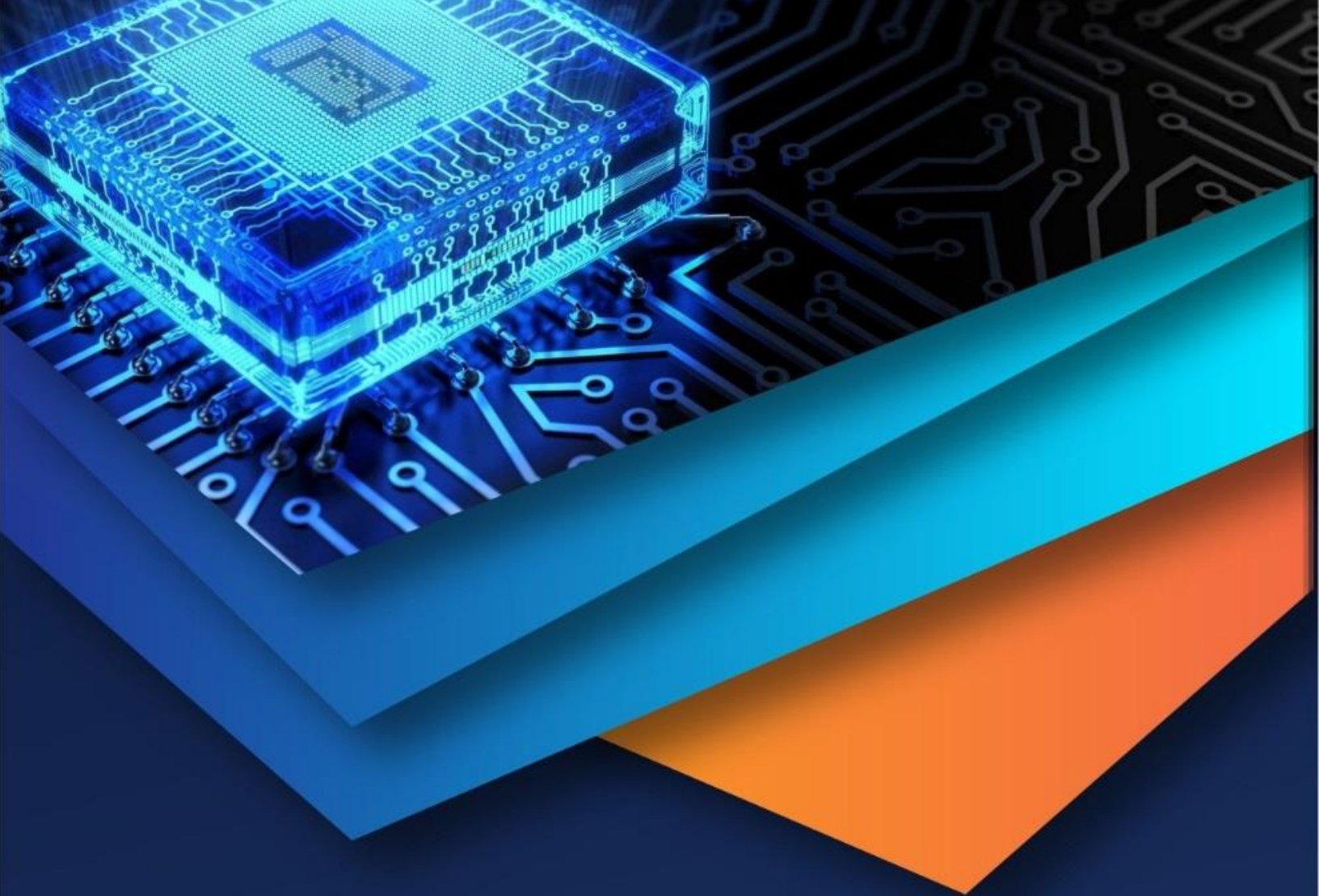
All datasets used are public. The twenty benchmark problems are from the KEEL repository [32] as distributed in the `imbalanced_databases` package, and the credit-card dataset is the ULB release [7]. The experimental harness, the analysis code, the complete per-fold result tables and the scripts that generate every figure and table in this paper are available from the authors on request. All randomness is seeded and corruption draws use a stable per-cell hash, so the reported numbers are reproducible from the harness without further configuration.

## REFERENCES

- [1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
- [2] C. Elkan. The foundations of cost-sensitive learning. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 973–978, 2001.
- [3] B. Frénay and M. Verleysen. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.
- [4] J. Van Hulse and T. M. Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513–1542, 2009.
- [5] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera. SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.
- [6] F. Brishti, F. Zhang, S. Mohammed, L. Bai, F. Wu, and B. Chen. Imbalanced classification with label noise: A systematic review and comparative analysis. *ICT Express*, 11(6):1127–1145, 2025.
- [7] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 159–166, 2015.
- [8] D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.
- [9] J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [10] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.
- [11] H. He and E. A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- [12] Y. Sun, A. K. C. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4):687–719, 2009.
- [13] B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.
- [14] P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2):31:1–31:50, 2016.
- [15] H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Proc. International Conference on Intelligent Computing (ICIC)*, pages 878–887, 2005.
- [16] H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328, 2008.
- [17] I. Tomek. Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.
- [18] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972.
- [19] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.
- [20] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016.
- [21] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.
- [22] R. Blagus and L. Lusa. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14:106, 2013.
- [23] Y. Elor and H. Averbuch-Elor. To SMOTE, or not to SMOTE? *arXiv:2201.08528*, 2022.
- [24] R. van den Goorbergh, M. van Smeden, D. Timmerman, and B. Van Calster. The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *Journal of the American Medical Informatics Association*, 29(9):1525–1534, 2022.
- [25] C. G. Northcutt, L. Jiang, and I. L. Chuang. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.



- [26] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from Noisy Labels with Deep Neural Networks: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153, 2023.
- [27] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571–595, 2014.
- [28] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proc. International Conference on Machine Learning (ICML)*, pages 233–240, 2006.
- [29] T. Saito and M. Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):e0118432, 2015.
- [30] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
- [31] A. Benavoli, G. Corani, and F. Mangili. Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, 17(5):1–10, 2016.
- [32] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2–3):255–287, 2011.
- [33] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
- [34] F. Pedregosa et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [35] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)