

Variance-Based Stratified Undersampling with Capacity-Aware Sampling Control for Imbalanced Tabular Classification

Mohd Hafiz Zakaria¹, Jafreezal Jaafar², Irving V. Paputungan³, Said Jadid Abdulkadir⁴

Department of Computing-Faculty of Science and Information Technology,
Universiti Teknologi PETRONAS, Seri Iskandar, 32610 Perak, Malaysia^{1,2,3,4}

Department of Mathematical Sciences-Faculty of Intelligent Computing,
Universiti Malaysia Perlis, Arau, 02600 Perlis, Malaysia¹

Department of Informatics-Faculty of Industrial Technology,
Universitas Islam Indonesia, Sleman 55584, Yogyakarta, Indonesia³

Abstract—Minority class instances in imbalanced tabular data are routinely misclassified, and existing undersampling methods offer little principled basis for deciding which majority samples to discard, how many to remove, and when reduction should stop. We propose Variance-Based Undersampling (VBU), a tuning-free data-level method for binary imbalanced tabular classification that sets these three choices from the training data. Samples are placed on a supervised ridge-classifier projection and stratified into eight interquartile-range nodes, draws are allocated across the four core nodes by within-node variance, and a capacity check based on Yamane's finite-population formula stops reduction when the minority class is too small, with a diagnostic override for recall-first use. We evaluated VBU on 14 public tabular datasets against six undersampling baselines and a Base Model, using one stratified split and a Random Forest classifier. On the eight datasets where the capacity check allowed reduction, VBU raised median Recall by 0.24 over the Base Model (Holm-adjusted Wilcoxon $p = 0.039$; Nemenyi $p = 0.0002$) and had the best Recall mean rank (2.25) under a significant Friedman test ($\chi^2 = 37.95$, $p < 0.001$). Median Precision fell by 0.22 and AUC-PR by 0.04 (both adjusted $p = 0.039$), while F1 did not change significantly. Random undersampling reached a similar Recall and scored higher than VBU on G-Mean, F1, Precision and AUC-PR, so the Recall gain is better credited to rebalancing than to VBU's selection rule. On the six datasets where the check halted reduction, forcing balance, whether through the override or through random undersampling, lowered Precision on four. VBU is therefore best viewed as a reproducible, rule-based way to rebalance training data, with an explicit check on when not to undersample and with Recall on par with random undersampling rather than above it.

Keywords—Class imbalance; undersampling; variance-based sampling; stratified sampling; data-level resampling; tabular data classification; minority class detection; machine learning

I. INTRODUCTION

When a classifier is trained on data where one class dominates, the minority class usually carries the cases that matter most in practice. Fraudulent transactions, loan defaults, equipment faults and disease diagnoses are all rare events, and a model that misses them can still report high accuracy because the majority class inflates the score [1]. In other words, the headline metric hides exactly the failure the application

was built to prevent. This mismatch between what accuracy reports and what practitioners actually need has kept class imbalance an active problem in tabular classification [2], [3]. Algorithm-level remedies exist, but data-level resampling remains attractive because it corrects the training distribution before any model is involved.

Oversampling methods such as SMOTE (Synthetic Minority Over-sampling Technique) [4] address imbalance by synthesizing minority instances. Undersampling instead reduces the majority class. Random undersampling discards majority instances uniformly. Cleaning rules such as Tomek Links [5] and Edited Nearest Neighbors [6] remove instances that sit near the class boundary or disagree with their neighborhood. These methods are widely used and often effective. Even so, they share an uncomfortable property. They remove samples without a statistical rationale for the three decisions that define any undersampling procedure. The first is which samples to remove. The second is how many to remove. The third is when to stop removing. Each existing method answers at most one of these questions, answers it heuristically, and leaves the rest to the user or to convention. None of them ties the answers to the statistical structure of the data being reduced.

We propose Variance-Based Undersampling (VBU), a tuning-free method that sets all three decisions from the training data. VBU uses variance to decide how many samples to draw from each region of the majority-class distribution. The method first builds a supervised one-dimensional projection through recursive feature elimination with cross-validation (RFECV) and a class-balanced ridge classifier, then stratifies the projection into eight interquartile-range (IQR) nodes and restricts sampling to the inner-fence core, Nodes 3 to 6 (which samples). Sampling quotas are allocated across the core nodes in proportion to within-node variance, so regions with more spread retain more instances (how many). A capacity check based on the Yamane finite-population formula [7], which we call capacity-aware sampling control, stops the reduction when the minority class is too small to support balanced sampling, and a diagnostic override is available for recall-first use (when to stop). No hyperparameter search is involved. The node quotas follow from the projection, and the only random step, the

draw within each node, runs under a fixed seed. We evaluated VBU against six undersampling methods and a Base Model across 14 public tabular datasets spanning several domains and imbalance ratios from 1.51 to 129.44. On the eight datasets where the capacity check allowed reduction, VBU improved median Recall by 0.24 over the Base Model while the change in F1 was not significant (median -0.02). Precision fell by 0.22 at the median, and random undersampling (RUS) reached a similar Recall at a smaller cost in Precision. VBU is therefore best read as a transparent and reproducible way to rebalance training data. It suits users who put Recall ahead of Precision and want a stated rule for when undersampling should not be applied at all.

The remainder of this paper is organized as follows. Section II reviews data-level methods for class imbalance and the undersampling baselines used in this study. Section III describes the VBU procedure. Section IV details the datasets, baselines, and evaluation protocol. Section V presents the results, the statistical analysis and the limitations of the study. Section VI concludes the paper and outlines future work.

II. RELATED WORK

Class imbalance arises when one class substantially outnumbers another in the training data, causing standard classifiers to bias toward the majority class and suppress minority detection [1], [3], [8]–[10]. The condition is widespread across domains where the event of interest is rare: medical diagnosis, financial fraud, fault detection, and customer churn, among others [2], [11]–[13]. Left unaddressed, a classifier trained on imbalanced data typically achieves high accuracy by predicting the majority class almost exclusively, while the minority class, which is often the class of practical interest, goes largely undetected.

Approaches to handling class imbalance divide broadly into data-level methods, which modify the training set before the classifier sees it, and algorithm-level methods, which adjust the learner or its loss function [14], [15]. Data-level methods are classifier-agnostic and therefore more transferable across applications. Within data-level methods, oversampling augments the minority class and undersampling reduces the majority class. SMOTE [4] and its variants [16]–[18] have been extensively studied and shown to improve minority detection in many settings [19]. Oversampling is excluded from this study because adding synthetic samples constitutes a categorically different intervention from removing real ones, and mixing the two in the same comparison would attribute performance differences to the category of intervention rather than the sampling strategy.

Undersampling methods differ in how they select which majority samples to remove. Random Undersampling (RUS) removes majority samples uniformly at random with no selection criterion [20]. Tomek Links removes majority samples that form minimal-distance pairs with minority samples [5]. Edited Nearest Neighbors (ENN) removes a majority sample when at least two of its three nearest neighbors belong to the minority class [6], and the Neighborhood Cleaning Rule (NCR) adds a second pass that also removes the majority neighbors of minority samples misclassified by their own neighborhood [21], [22]. NearMiss-v2 keeps the majority samples whose average

distance to the three farthest minority samples is smallest [23]. Cluster Centroids replaces the majority class with k -means centroids, so the reduced majority consists of synthetic representative points [24]. A comparative analysis of several of these methods by Batista et al. [25] showed that cleaning-based approaches generally improve minority detection but that no single method dominates across all conditions. More recent empirical comparisons across undersampling and oversampling variants reach the same conclusion, finding no universally superior method across datasets and domains [26]–[29].

None of the six methods checks whether the minority class is large enough for balanced reduction to make sense, and the size of the reduced majority is either fixed by convention, usually at the minority count, or left to a cleaning rule. The methods also differ in what they keep. RUS preserves the majority distribution only in expectation, so a single draw can over- or under-represent parts of it. Distance-based cleaners (Tomek Links, ENN and NCR) act on proximity to the minority class without regard to the spread of the majority class itself. NearMiss-v2 selects by proximity alone, and Cluster Centroids replaces real samples with centroids, so the retained majority no longer contains observed cases. VBU differs from these methods in two design choices. It spreads the retained majority across IQR-defined strata with variance-weighted quotas, and it checks minority capacity before reducing anything. Whether the first choice preserves the majority distribution better than random selection is not measured in this study (Section V-G).

III. PROPOSED METHOD

A. Overview of the VBU Pipeline

At a high level, VBU passes the training set through four sequential components: the ridge projection, IQR stratification, capacity-aware sampling control, and variance-weighted allocation. Fig. 1 illustrates this pipeline.

The pipeline starts with standardization using StandardScaler, fitted on the training partition only to prevent information leakage to the validation and test sets. RFECV and a ridge classifier then produce v_i , a representative scalar for each sample, which serves as the axis for IQR stratification. Before allocation, the capacity check decides whether reduction can go ahead. If the minority class is too small, VBU stops and returns the original training set unchanged. Otherwise the pipeline proceeds to allocation, and the reduced majority class and the sampled minority class are combined into a balanced training set.

B. Supervised Projection and IQR Stratification

IQR stratification requires a one-dimensional input. However, real-world datasets are multidimensional, so those features need to be compressed into a single representative scalar before stratification can proceed. That scalar must be class-discriminative. Without this property, v_i would still be representative of the features it was computed from, but its representativeness would not be class-discriminative, since the input would include features that carry no class-separating information.

To obtain a class-discriminative projection axis, VBU first applies Recursive Feature Elimination with Cross-Validation

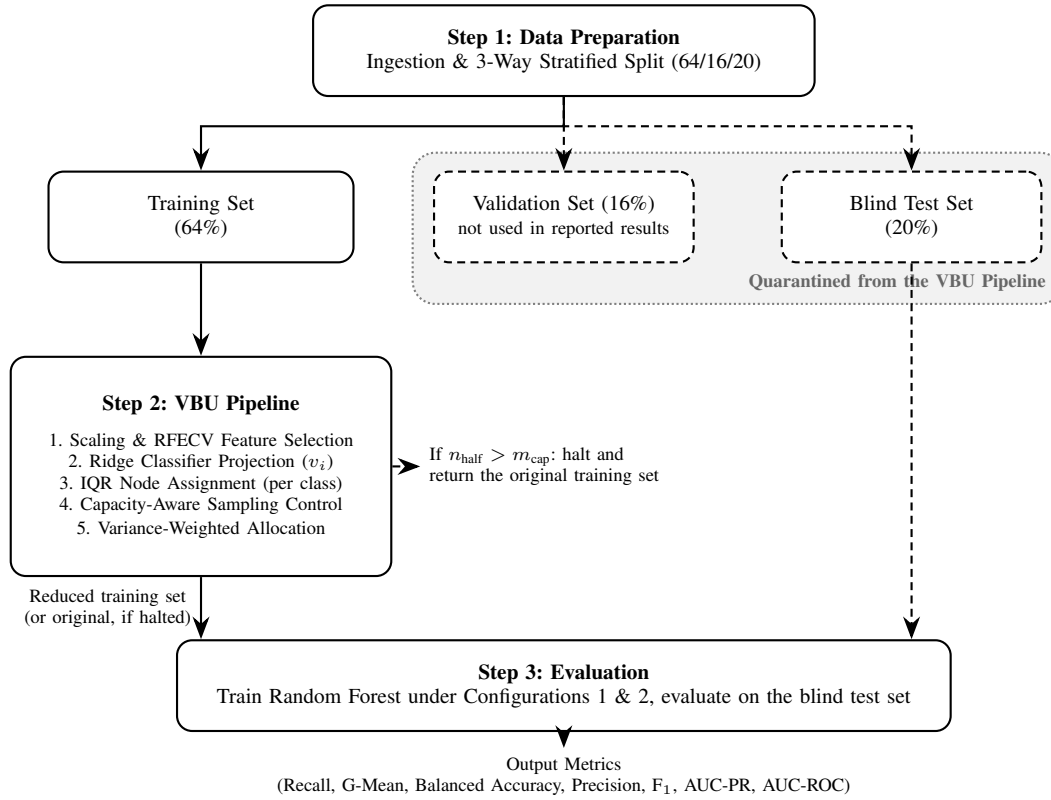


Fig. 1. High-level overview of the VBU pipeline.

(RFECV) [30] to identify the feature subset S that maximizes three-fold cross-validated F1. The internal estimator is a Random Forest with 50 trees and `max_depth=5`, chosen intentionally lighter than the downstream classifier because feature selection does not require full classifier capacity. The step size is set to 0.1, so each iteration removes 10% of the original number of features (at least one). F1 is used as the scoring metric specifically because accuracy misleads on imbalanced data. The output is the subset S of features that contribute to class separation.

Even so, after RFECV the data typically remains multidimensional. VBU therefore fits a class-balanced ridge classifier (scikit-learn `RidgeClassifier`, $\alpha = 1.0$) on the RFECV-selected subset S . A ridge classifier codes the two classes as -1 and $+1$ and fits a ridge regression [31] to these codes, so its decision function is the linear score

$$v_i = \beta_0 + \sum_{j \in S} \beta_j x_{ij}, \quad (1)$$

with β estimated by ridge regression on the ± 1 class codes. This score is taken as the projection v_i .

Class-balanced weighting reweights training samples inversely to class frequency, so the minority class contributes to the projection axis in proportion to its discriminative information rather than being dominated by the majority class. Ridge is preferred over Ordinary Least Squares (OLS) because the ℓ_2 penalty stabilizes coefficients under multicollinearity, preventing unstable values in v_i .

With v_i established, VBU partitions the training samples of each class into eight nodes. The 8-node structure originates from the standard boxplot, which naturally produces eight partitions through its inner and outer Tukey fence boundaries [32]. Quartile boundaries are computed separately per class, since each class has its own distribution and merging both into a single set risks collapsing one class into too few nodes. As a result, a given node covers a different score range in each class. The nodes act as within-class strata, and the two classes are later matched by sample count, not by score range. The eight nodes are defined as follows:

- Node 1: $v_i \geq Q_3 + 3 \cdot \text{IQR}$ (extreme high tail)
- Node 2: $Q_3 + 1.5 \cdot \text{IQR} \leq v_i < Q_3 + 3 \cdot \text{IQR}$ (mild high tail)
- Node 3: $Q_3 \leq v_i < Q_3 + 1.5 \cdot \text{IQR}$ (upper core edge)
- Node 4: $Q_2 \leq v_i < Q_3$ (upper middle)
- Node 5: $Q_1 \leq v_i < Q_2$ (lower middle)
- Node 6: $Q_1 - 1.5 \cdot \text{IQR} \leq v_i < Q_1$ (lower core edge)
- Node 7: $Q_1 - 3 \cdot \text{IQR} \leq v_i < Q_1 - 1.5 \cdot \text{IQR}$ (mild low tail)
- Node 8: $v_i < Q_1 - 3 \cdot \text{IQR}$ (extreme low tail)

Nodes 3 to 6 span Tukey's inner-fence region, and Nodes 1, 2, 7, and 8 correspond to the outlier regions. VBU restricts default sampling to Nodes 3–6 because the regions beyond the inner fences contain leverage points and outliers that can have

a large influence on classifier decisions. Unless the application relies heavily on edge cases, sampling from outlier nodes risks importing outlier bias into the reduced training set. The inference core also holds the bulk of each class, so the retained sample still covers most of the majority distribution. The same restriction applies to the minority class, which means that minority samples beyond the inner fences are not used for training under the default setting (Section V-G).

C. Capacity-Aware Sampling Control and Diagnostic Override

Before any reduction takes place, VBU runs a capacity check, which we refer to as capacity-aware sampling control. The check compares a minimum sample size, derived from the size of the majority core, with the number of minority samples available in the core nodes. If the minority count falls below this minimum, VBU does not reduce the majority class and returns the original training set.

The minimum comes from Yamane's finite-population formula [7],

$$N_{\text{yamane}} = \left\lceil \frac{M_{\text{cap}}}{1 + M_{\text{cap}} e^2} \right\rceil, \quad (2)$$

where M_{cap} is the number of majority samples in the core nodes and e is the margin of error. The formula contains no explicit proportion p or confidence coefficient Z because Yamane's simplification fixes both in advance. Cochran's sample size for estimating a proportion is $n_0 = Z^2 p(1-p)/e^2$ [33]. Setting $p = 0.5$, the worst case, and $Z \approx 2$, about 95% confidence, gives $n_0 = 1/e^2$, and a finite-population correction then gives (2) to a close approximation. Yamane's formula is therefore a special case of Cochran's, used here because the majority core is a finite population of known size.

VBU uses the formula as a floor on sample size, not as an estimate of how many majority cases can be removed safely. The rule asks that the reduced, balanced training set contain at least N_{yamane} samples, the size of a sample that would describe the majority core to within $\pm e$. Under a 1:1 design each class must supply half of this, so VBU computes $n_{\text{half}} = \lceil N_{\text{yamane}}/2 \rceil$ and compares it with m_{cap} , the number of minority samples in the core nodes. If $n_{\text{half}} > m_{\text{cap}}$, the dataset is too sparse for balanced reduction and VBU halts, returning the original training set unchanged.

The margin is set to $e = 0.03$, stricter than the conventional 0.05, and was fixed before any results were seen. Two properties of this rule follow directly from (2). The first is saturation. As M_{cap} grows, N_{yamane} approaches $1/e^2 \approx 1,111$, so n_{half} cannot exceed 556. For large majority classes the check therefore reduces to a minimum of about 556 minority samples in the core, and the size of the majority matters only when the core is small. Across our 14 datasets n_{half} ranged from 256 to 531. The second is sensitivity to e . With $e = 0.05$, n_{half} would have fallen to between 141 and 197, and three of the six datasets that halted (Iranian Churn, Obesity Risk and Page Blocks) would have been reduced.

The diagnostic override lets an expert relax the two conservative defaults, and Table I sets the default and override

TABLE I. DEFAULT AND DIAGNOSTIC-OVERRIDE SETTINGS OF VBU

Setting	Default	Diagnostic override
Capacity check	On: halt when $n_{\text{half}} > m_{\text{cap}}$	Can be bypassed: balance at m_{cap}
Sampling pool	Core Nodes 3–6	Core nodes, plus selected tail nodes if wanted
Use in this study	All 14 datasets	Cohort B only, halt bypassed, core nodes only

settings side by side. One switch bypasses the halt, so VBU balances the classes at about m_{cap} samples each even when $n_{\text{half}} > m_{\text{cap}}$. The other adds selected tail nodes (1, 2, 7 or 8) to the sampling pool for applications that depend on edge cases. The two switches are independent. In this study only the first was used, and only on the six datasets that halted, to measure what forcing balance would cost. The override is therefore a diagnostic tool here, not a recommended setting, and all Cohort A results use the default mode (Section V-D).

D. Variance-Weighted Allocation

The allocation rests on a working assumption. A node with high variance has its samples spread out along the projection axis, so a small draw from it is more likely to miss part of its range. A node with low variance has its samples packed close together, and a few of them can stand in for the rest. VBU therefore gives larger quotas to high-variance nodes. The intuition is the same as in optimal allocation for stratified sampling, where strata with more spread receive more draws. Neyman allocation, for example, sets the stratum sample size in proportion to $N_h S_h$ [33]. VBU's rule uses the variance alone and is not optimal in Neyman's sense, and the assumption that spread on a one-dimensional score reflects how much distinct information a node carries was not tested directly in this study.

VBU implements this through a variance-weighted allocation rule. For each active core node $i \in \{3, 4, 5, 6\}$, the within-node sample variance of the projected majority-class scores is computed (s_i^2). The allocation weight is:

$$w_i = \frac{s_i^2}{\sum_{j \in \mathcal{C}} s_j^2}, \quad \mathcal{C} = \{3, 4, 5, 6\}. \quad (3)$$

Nodes with higher variance receive proportionally larger quotas. If variance collapses to zero across all core nodes, the weighting defaults to raw capacity proportions, where each node receives a quota in proportion to its sample count alone.

The raw quota for node i is $q_i = w_i \cdot m_{\text{cap}}$, where m_{cap} is the total minority class count in the core nodes, and each quota is capped at the number of majority samples the node holds. Because q_i is continuous and sample counts must be integers, rounding introduces a discrepancy, and the caps can leave the majority total short of m_{cap} . VBU resolves this through an iterative adjustment loop. The loop alternates between classes. If $N_{\text{maj}} > N_{\text{min}}$, majority quotas are reallocated to match the current minority count. Otherwise the minority quotas are reallocated, using the minority's own variance weights, to match the majority count. The loop terminates

when $|N_{\text{maj}}/N_{\text{min}} - 1| \leq 0.05$, when either class falls below N_{yamane} , or after 25 iterations. If the tolerance is not reached within these bounds, VBU reverts to the first-pass quotas and proceeds. The $\pm 5\%$ tolerance is a practical choice. Exact 1:1 is not a necessity, since balanced data does not literally mean equal class counts, and exact convergence is hard to reach when rounding and capacity constraints are in play. On the eight Cohort A datasets the final majority-to-minority ratio ranged from 0.89 to 1.00, and VBU kept between 64% (Spambase) and 100% of the training minority.

Once the quotas are finalized, VBU draws q_i samples from each active node without replacement, for both the majority and minority classes, using a fixed random seed. The drawn samples from all core nodes across both classes are concatenated to form the reduced training set. Given the projection, the quotas are fixed, and the within-node draw is the only random step in the sampler. The full pipeline is reproducible for the same data, row order, seed and library versions. A change in any of these can alter the within-node draw, and a change in row order or library version can also alter the RFECV feature subset, which moves the projection and therefore the quotas. When we re-ran the pipeline with a different scikit-learn release (1.9.1), 12 of the 14 datasets reproduced the reported VBU results exactly, while NASA JM1 and Taiwanese Bankruptcy did not, so the library version should be recorded alongside the seed.

E. Algorithm and Complexity Analysis

Algorithm 1 consolidates the six-phase VBU pipeline into a unified pseudocode. Phases A through C prepare the sampling space, and Phases D through F carry out the reduction.

Phases B through F constitute the core VBU sampler. Phase B assigns samples to nodes through one-dimensional sorting, giving $\mathcal{O}(n \log n)$. Phase C computes scalar quantities only, giving $\mathcal{O}(1)$. Phases D through F each operate linearly on the retained sample pool, giving $\mathcal{O}(n)$, with Phase E bounded at 25 iterations. The overall core complexity is therefore $\mathcal{O}(n \log n)$, dominated by the sorting step in Phase B.

Phase A dominates the cost of the full pipeline. With a step of 0.1, RFECV removes a tenth of the original feature count in each round, which gave between 4 and 14 elimination rounds R on our datasets. Across its $K = 3$ folds, RFECV therefore fits about KR forests of $T = 50$ trees. A depth-limited tree considers about $\sqrt{d}$ of the d features at each split and sorts the samples at each level, so one tree costs about $\mathcal{O}(\sqrt{d} n \log n)$ and Phase A costs about $\mathcal{O}(KRT\sqrt{d} n \log n)$, plus $\mathcal{O}(n|S|^2 + |S|^3)$ for the ridge fit on the $|S|$ selected features. This term grows with both n and d and dominates in practice. RFECV and the projection took between 4.09 and 18.62 seconds per dataset, while the sampler (Phases C through F) took under 11 milliseconds. Detailed runtime profiling is reported in Section V-F.

Neighborhood-based methods such as ENN and NearMiss-v2 rely on nearest-neighbor or pairwise distance computations, which cost up to $\mathcal{O}(n^2)$ in the number of samples, or $\mathcal{O}(n^2 \cdot k)$ depending on neighborhood size k . VBU's sampling step avoids pairwise distance computation by operating on the one-dimensional ridge projection, giving $\mathcal{O}(n \log n)$. The projection requires a feature-selection step, for which RFECV

Algorithm 1 Variance-Based Undersampling (VBU)

Require: $X_{\text{train}}, y_{\text{train}}, e = 0.03$, STOP_IF_OVERSAMPLE = True

Ensure: $X_{\text{reduced}}, y_{\text{reduced}}$

Phase A: Projection

- 1: Fit RFECV on $(X_{\text{train}}, y_{\text{train}})$; fit RidgeClassifier on selected features
- 2: $v_i \leftarrow \text{ridge.decision_function}(X_{\text{train,sel}})$ for each sample

Phase B: Node Segmentation

- 3: **for** each class $c \in \{\text{maj}, \text{min}\}$ **do**
- 4: Compute Q_1, Q_2, Q_3 , IQR from $v_{i,c}$; assign samples to Nodes 1–8
- 5: **end for**
- 6: $M_{\text{cap}} \leftarrow |\text{maj_kept}|$; $m_{\text{cap}} \leftarrow |\text{mi_kept}|$

Phase C: Capacity Check

- 7: $N_{\text{yamane}} \leftarrow \lceil M_{\text{cap}} / (1 + M_{\text{cap}} \cdot e^2) \rceil$; $n_{\text{half}} \leftarrow \lceil N_{\text{yamane}} / 2 \rceil$
- 8: **if** STOP_IF_OVERSAMPLE **and** $n_{\text{half}} > m_{\text{cap}}$ **then**
- 9: **return** $(X_{\text{train}}, y_{\text{train}})$ ▷ halt: original set unchanged
- 10: **end if**

Phase D: Variance-Weighted Allocation

- 11: **for** each core node $i \in \{3, 4, 5, 6\}$ **do**
- 12: Compute s_i^2 from majority scores in node i
- 13: **end for**
- 14: $w_i \leftarrow s_i^2 / \sum_{j \in C} s_j^2$ (fallback: count proportions if $\sum s^2 = 0$)
- 15: $q_i^{\text{maj}} \leftarrow \min(\lfloor w_i \cdot m_{\text{cap}} \rfloor, C_i^{\text{maj}})$

Phase E: Iterative Balancing

- 16: $N_{\text{maj}} \leftarrow \sum q_i^{\text{maj}}$; $N_{\text{min}} \leftarrow m_{\text{cap}}$; $\text{iter} \leftarrow 0$
- 17: **while** $|N_{\text{maj}}/N_{\text{min}} - 1| > 0.05$ **and** $N_{\text{maj}}, N_{\text{min}} \geq N_{\text{yamane}}$ **and** $\text{iter} < 25$ **do**
- 18: $\text{iter} \leftarrow \text{iter} + 1$
- 19: **if** $N_{\text{maj}} > N_{\text{min}}$ **then**
- 20: Reallocate q_i^{maj} targeting N_{min}
- 21: **else**
- 22: Reallocate q_i^{min} targeting N_{maj}
- 23: **end if**
- 24: **end while**
- 25: **if** tolerance not achieved **then**
- 26: Revert to initial Phase D quotas
- 27: **end if**

Phase F: Final Extraction

- 28: **for** each core node i **do**
- 29: Sample q_i^{maj} majority instances without replacement
- 30: Sample q_i^{min} minority instances without replacement
- 31: **end for**
- 32: **return** concatenate(sampled majority, sampled minority)

is the default. This step is modular, and a lighter selector that keeps the projection class-discriminative could reduce its cost, although we have not tested alternatives.

IV. EXPERIMENTAL SETUP

A. Datasets

The evaluation uses 14 publicly available tabular datasets, all exhibiting class imbalance as the defining characteristic. Datasets were selected to vary in domain, sample size, imbalance ratio and feature count so that VBU's behavior could be observed under different conditions. All datasets were

TABLE II. SUMMARY OF DATASET CHARACTERISTICS, ORDERED BY COHORT AND ASCENDING IMBALANCE RATIO

Dataset	Coh.	N	Maj.	Min.	Ft.	IR
<i>Cohort A: Parity-Achieved</i>						
Spambase	A	4210	2531	1679	57	1.51
Student Dropout	A	4424	3003	1421	36	2.11
Occupancy Detection	A	19119	14372	4747	5	3.03
AIDS ACTG175	A	2139	1618	521	23	3.11
NASA JMI	A	8907	6903	2004	21	3.44
Credit Card Default	A	29965	23335	6630	23	3.52
Online Shoppers	A	12205	10297	1908	28	5.40
Bank Marketing	A	45211	39922	5289	48	7.55
<i>Cohort B: Sparsity-Halted</i>						
Iranian Churn	B	2850	2404	446	13	5.39
Obesity Risk	B	2087	1763	324	23	5.44
Page Blocks	B	5403	4889	514	10	9.51
Taiwanese Bankruptcy	B	6819	6599	220	95	30.00
Ozone Level	B	1848	1791	57	72	31.42
Abalone19	B	4174	4142	32	9	129.44

N = total records; Maj./Min. = majority/minority counts; Ft. = raw feature count; IR = imbalance ratio (majority:minority). Stratification preserves the IR in all three partitions. Test-set minority counts range from 6 (Abalone19) to 1,326 (Credit Card Default), and the smallest in Cohort A is 104 (AIDS ACTG175).

sourced from the UCI Machine Learning Repository, with the exception of Abalone19, which was obtained from the KEEL Imbalanced Dataset Repository. Duplicate rows were removed before splitting.

Naturally binary datasets were used as-is. Multi-class datasets were converted to binary by taking one class, or a group of classes, as the positive target and pooling the remaining classes. In Student Dropout, for example, students who dropped out form the positive class against those still enrolled or graduated. AIDS ACTG175 was used with all of its variables, including the follow-up time (time), which is recorded after enrollment and on its own separates the two classes well (AUC-ROC of 0.89 on the full data). Scores on this dataset are therefore likely to be optimistic for every method, although all methods receive the same features. Table II summarizes the dataset characteristics. The partition into Cohort A (Parity-Achieved) and Cohort B (Sparsity-Halted) is set by the capacity check described in Section III-C, not chosen by us. Because the cohorts are defined by VBU's own rule, the Cohort A statistics describe VBU only on datasets where it reduces the majority class, and they should not be read as performance across all 14 datasets.

B. Baseline Methods

Seven comparators are used: one non-intervened control and six undersampling algorithms, all operating on the same datasets, split, and classifier as VBU.

The Base Model is the Random Forest classifier trained directly on the imbalanced training set with no resampling applied. It shows how the classifier performs when class imbalance is left untreated.

The six undersampling baselines, described in Section II, cover the main families of undersampling: random removal (RUS), boundary cleaning (Tomek Links, ENN and NCR),

distance-based selection (NearMiss-v2) and prototype generation (Cluster Centroids). RUS, NearMiss-v2 and Cluster Centroids reduce the majority class to the size of the training minority, so they balance the classes exactly and keep every minority sample. Tomek Links, ENN and NCR remove only the majority samples their rules flag, so the classes stay imbalanced after cleaning. ENN and NCR use the three nearest neighbors of each sample.

Oversampling is excluded because undersampling and oversampling differ in their core sampling procedure: one removes samples, the other adds them. Including oversampling baselines would mix two categorically different interventions in the same comparison, making it impossible to attribute any performance difference to the sampling strategy rather than the category of intervention. For the same reason, algorithm-level alternatives such as class weighting and threshold moving are not included. This keeps the comparison within one family of interventions, but it also means the study cannot show whether VBU's Recall gain could be matched without resampling (Section V-G). Comparison against oversampling and algorithm-level baselines is left for future work.

All six baselines are implemented using scikit-learn components with manually controlled parameters and no hyperparameter tuning beyond what is stated.

C. Evaluation Protocol

The classifier used throughout is the Random Forest [34], chosen for its robustness to feature scaling, its tolerance of mixed feature types, and its stability across moderate hyperparameter perturbations. The configuration is held constant: 100 trees, the Gini impurity criterion, unbounded tree depth, and a fixed random seed (SEED = 1). No grid search or per-dataset tuning is performed. The class_weight parameter is set to None. Setting it to balanced would activate scikit-learn's cost-sensitive adjustment, mixing two distinct interventions, sampling and cost-sensitive learning, in the same experiment and making it impossible to isolate the effect of the sampling strategy alone.

Each dataset is partitioned into training (64%), validation (16%) and test (20%) sets using a stratified split that preserves the original imbalance ratio in all three partitions. The validation partition was not used for any decision or result reported in this paper. A single stratified split with one seed is used rather than k -fold cross-validation, for two reasons. Several Cohort B datasets have minority counts too small for stable fold-level estimates. Abalone19 carries only 32 minority instances, which would leave fewer than seven per fold under a five-fold scheme. In addition, the capacity check depends on the training-partition minority count, and letting this count vary across folds would mix the halt decision with fold-to-fold variation. Neither reason rules out repeated stratified splits, which would give a measure of variance across partitions. We did not run them, so every result below comes from one partition per dataset (Section V-G). RFECV, the ridge projection and the sampler are fitted on the training partition only, and all reported metrics are computed on the test partition at the original imbalance ratio.

Two evaluation configurations are applied to all eight methods. Configuration 1 passes the full standardized feature

matrix to the classifier, and Configuration 2 passes the RFECV-selected subset. VBU's projection and its reduced sample are the same in both, so the two configurations differ only in the features seen by the classifier and by the distance-based baselines. Statistical comparisons reported in Section V use Configuration 2 test-set results on Cohort A unless stated otherwise, and Configuration 1 results are summarized at the end of Section V-A.

D. Evaluation Metrics and Statistical Tests

Global accuracy is not computed, as it is uninformative under class imbalance. Seven metrics that explicitly account for class structure are used [35], [36]. Recall is the primary indicator, measuring the proportion of actual minority instances correctly identified. G-Mean and Balanced Accuracy are the structural balance metrics [37], showing whether Recall gains are sustained without majority-class collapse. Precision and F1 are the trade-off indicators, making the precision tax visible and quantifiable. The Area Under the Precision-Recall Curve (AUC-PR) and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) provide threshold-independent diagnostic support. AUC-PR is the more relevant of the two under strong imbalance [38], [39], while AUC-ROC is retained for cross-study comparability.

Statistical validation uses non-parametric tests. The Friedman test [40] is applied as an omnibus check across all eight methods per metric. A significant result at $\alpha = 0.05$ justifies pairwise comparisons [41], [42]. The Nemenyi post-hoc test [43] then identifies which method pairs differ by comparing mean ranks against the critical difference. The exact two-sided Wilcoxon signed-rank test [44] compares VBU directly with the Base Model and with RUS, which has the strongest overall results among the baselines in Table III. The comparison with RUS was added after the main results were known and should be read as a check rather than a planned test. Each of the two comparisons covers five metrics (Recall, G-Mean, Precision, F1 and AUC-PR), and Holm's step-down correction is applied within each set of five. No zero differences occurred. With eight datasets the smallest attainable two-sided p -value is 0.0078, which becomes 0.039 after Holm correction across five metrics, so these tests have little power and a non-significant result should not be read as evidence that two methods perform alike.

E. Ablation Study Design

The ablation separates the effect of resampling from that of the ridge projection by testing two projection-only pipelines that keep all training samples and perform no majority-class reduction. Baseline 1 (AF+Ridge) passes the full standardized feature matrix through the ridge projection without feature selection. Baseline 2 (ANOVA- k +Ridge) applies the ANOVA F-test to select k features, where k matches the count selected by RFECV for that dataset, then applies the ridge projection. Matching k means that both pipelines use the same number of features as full VBU, so differences reflect resampling rather than feature count. Both pipelines are evaluated under the same protocol and classifier as all other methods. The design does not vary VBU's internal components, namely the variance weights, the core restriction, the number of nodes and the capacity check, so it tests whether resampling helps, not

which part of VBU's sampler helps. Results are reported in Section V-E.

V. RESULTS AND DISCUSSION

A. Cross-Dataset Performance

Table III reports median performance across the eight Cohort A datasets, and Recall shows the clearest change. VBU raises median Recall from 0.63 for the Base Model to 0.80, but RUS reaches a slightly higher 0.82 and is at or above VBU on every metric in the table, including G-Mean (0.84 against 0.82) and Precision (0.59 against 0.54). On G-Mean, VBU sits above the Base Model (0.78), Tomek Links (0.80) and ENN (0.81) but below RUS and NCR (0.83). NearMiss-v2 returns the highest Recall (0.87) but the lowest G-Mean (0.64) and Precision (0.34) of all methods. VBU's Precision (0.54) and F1 (0.66) are below those of the cleaning baselines, a trade-off examined in Section V-C.

The size of the Recall gain depends mainly on how many minority cases the Base Model was missing. The gain was largest where Base Recall was lowest, as in NASA JM1 (0.19 to 0.66), Bank Marketing (0.38 to 0.74) and Credit Card Default (0.38 to 0.67). Where the Base Model already found most minority cases, as in Occupancy Detection (0.99) and Spambase (0.93), the gain was below 0.03. RUS follows the same pattern, so this describes when rebalancing helps rather than a property of VBU alone. VBU adds one condition of its own. It reduces the majority class only when the minority core holds at least n_{half} samples, which ranged from 256 to 531 on these datasets (Section III-C).

Configuration 1, where the classifier sees all features, gives a similar picture. Median Recall is 0.85 for VBU, 0.88 for RUS and 0.61 for the Base Model, and RUS again leads VBU on G-Mean (0.86 against 0.84).

B. Statistical Analysis

The Friedman test shows that mean ranks differ significantly across the eight methods for both Recall ($\chi^2 = 37.95$, $p < 0.001$) and G-Mean ($\chi^2 = 27.80$, $p < 0.001$), which justifies the post-hoc comparisons. The critical difference for the Nemenyi test is $CD = 3.712$ at $\alpha = 0.05$, computed with $k = 8$ methods and $n = 8$ datasets. Fig. 2 presents the critical difference diagrams for Recall (panel a) and G-Mean (panel b).

On Recall, the Nemenyi test does not separate VBU from RUS ($p = 0.9987$), NCR ($p = 0.8889$), NearMiss-v2 ($p = 0.9966$) or Cluster Centroids ($p = 0.9787$). VBU ranks significantly above the Base Model ($p = 0.0002$) and Tomek Links ($p = 0.0048$), while the difference from ENN ($p = 0.0708$) does not reach significance at $\alpha = 0.05$. VBU has the best mean Recall rank (2.25), ahead of RUS (3.00) and NearMiss-v2 (3.13), but the gaps among these three methods are well inside the critical difference.

On G-Mean, no Nemenyi comparison involving VBU reaches significance, including those with RUS ($p = 0.3869$), which has the best mean rank (2.00 against 4.625 for VBU), and ENN ($p = 0.9987$). This should not be read as equivalence. With $n = 8$ the critical difference is wide, and even NearMiss-v2, whose median G-Mean is only 0.64, is not separated

TABLE III. MEDIAN PERFORMANCE ACROSS COHORT A (8 DATASETS), CONFIGURATION 2, TEST SET. BEST VALUE PER METRIC IN BOLD. VBU MARKED WITH †

Method	Recall	G-Mean	Bal.Acc	Precision	F1	AUC-PR	AUC-ROC
Base Model	0.63	0.78	0.79	0.75	0.69	0.78	0.91
RUS	0.82	0.84	0.84	0.59	0.69	0.76	0.91
Tomek Links	0.68	0.80	0.81	0.73	0.71	0.77	0.91
ENN	0.72	0.81	0.82	0.71	0.71	0.77	0.91
NCR	0.79	0.83	0.83	0.66	0.72	0.76	0.91
NearMiss-v2	0.87	0.64	0.68	0.34	0.49	0.51	0.75
Cluster Centroids	0.81	0.79	0.79	0.52	0.62	0.71	0.87
VBU†	0.80	0.82	0.82	0.54	0.66	0.71	0.90

RUS=Random Undersampling; ENN=Edited Nearest Neighbors; NCR=Neighborhood Cleaning Rule. Bal.Acc=Balanced Accuracy. Cohort B results are reported in Section V-D.

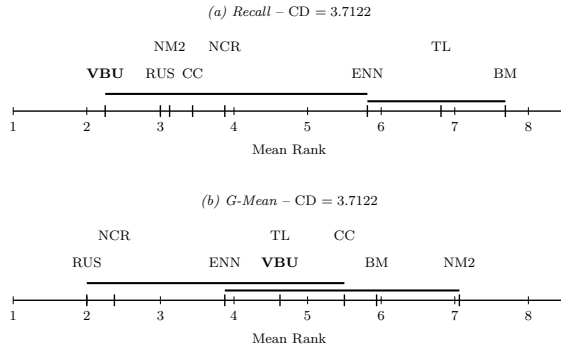


Fig. 2. Critical difference diagrams for Recall and G-Mean, Cohort A.

from VBU ($p = 0.4885$). RUS is the only method whose G-Mean rank differs significantly from that of the Base Model ($p = 0.0285$).

Because RUS is the strongest baseline in Table III, we also compared it with VBU directly (Table IV). The two methods do not differ on Recall. VBU is higher on six of the eight datasets, but the median difference is only $+0.004$ ($p = 0.547$). RUS scores higher on G-Mean and F1 on all eight datasets (Holm-adjusted $p = 0.039$ for both) and on Precision and AUC-PR on seven of the eight (adjusted $p = 0.047$ for both). In these experiments VBU's stratified selection did not improve on uniform random selection. Its Recall gain over the Base Model comes from rebalancing the training set, which RUS achieves with a smaller loss of Precision.

C. The Precision Trade-Off

Table IV also gives the paired comparison with the Base Model across the eight Cohort A datasets. VBU raises median Recall by 0.24 (Holm-adjusted $p = 0.039$). Median G-Mean rises by 0.05, but this does not survive Holm correction (adjusted $p = 0.078$). Precision falls by 0.22 and AUC-PR by 0.04, both on all eight datasets (adjusted $p = 0.039$).

The change in F1 is only -0.02 and is not significant ($p = 0.4609$). Because F1 is the harmonic mean of Precision and Recall, this small movement means that the Recall gain roughly offsets the Precision loss at the default decision threshold. AUC-PR gives a less favorable reading. It does not depend on the threshold, and its decline shows that the classifier trained

TABLE IV. PAIRED WILCOXON SIGNED-RANK COMPARISONS OF VBU WITH THE BASE MODEL AND WITH RUS ACROSS COHORT A (8 DATASETS), CONFIGURATION 2, TEST SET

Metric	VBU – Base Model			VBU – RUS		
	Med. Δ	p	p_{Holm}	Med. Δ	p	p_{Holm}
Recall	+0.240	0.0078	0.0391	+0.004	0.5469	0.5469
G-Mean	+0.051	0.0391	0.0781	-0.016	0.0078	0.0391
Precision	-0.219	0.0078	0.0391	-0.058	0.0234	0.0469
F1	-0.020	0.4609	0.4609	-0.037	0.0078	0.0391
AUC-PR	-0.044	0.0078	0.0391	-0.028	0.0156	0.0469

Positive Δ favors VBU. All tests are exact and two-sided. p_{Holm} is adjusted across the five metrics within each comparison. With $n = 8$ the smallest attainable p is 0.0078, reached when all eight datasets move in the same direction. VBU wins/losses against RUS: Recall 6/2, G-Mean 0/8, Precision 1/7, F1 0/8, AUC-PR 1/7.

on VBU's sample ranks minority cases slightly worse than the Base Model does. The Recall gain at the default threshold therefore comes from a shift in where the classifier draws its boundary rather than from better separation of the classes. This matches the known effect of undersampling on a classifier's posterior probabilities, which are pushed toward the minority class [45], and a similar shift could be obtained by lowering the threshold of the Base Model (Section V-G).

VBU's profile is therefore that of a recall-first method. It avoids the collapse seen with NearMiss-v2, which reaches a higher median Recall (0.87) but drops G-Mean to 0.64 and F1 to 0.49, whereas VBU holds them at 0.82 and 0.66. RUS, though, reaches a similar Recall at a smaller cost, with a median Precision of 0.59 and an F1 of 0.69.

The precision cost is not small. Where false positives are expensive, a 0.22 fall in Precision has to be weighed against the gain in minority detection.

D. Cohort Analysis and the Capacity Check

The six Cohort B datasets are those where the capacity check halted under the default setting. To see what the halt avoided, we engaged the diagnostic override on all six, bypassing the halt and balancing the classes within the core nodes. We also compared the results with RUS, which balances the classes on any dataset. Table V reports the results.

Forcing balance raised Recall and lowered Precision on most of these datasets, as it did in Cohort A. On Taiwanese Bankruptcy the override moved Recall from 0.16 to 0.73 while

Precision fell from 0.58 to 0.20, and RUS behaved in the same way (Recall 0.86, Precision 0.16). Iranian Churn and Page Blocks lost between 0.18 and 0.28 in Precision under both methods. On Abalone19 the Base Model detects no minority case. The override finds four of the six minority cases in the test set (Recall 0.67) at a Precision of 0.01, and RUS finds five at the same Precision.

These results show what full balancing costs on sparse data, and the cost applies to RUS as much as to VBU. They do not show that halting is the better choice. Under the override, F1 fell on Iranian Churn (0.77 to 0.74) and Page Blocks (0.88 to 0.76) but rose slightly on Taiwanese Bankruptcy (0.25 to 0.31), Ozone Level (0.09 to 0.10) and Abalone19 (0.00 to 0.02). A drop in Precision is expected whenever Recall is pushed up, so the override results cannot by themselves confirm that the halt was correct. On these six datasets the halt avoided a Precision loss of up to 0.38, at the cost of Recall gains that a recall-first user may well want. The capacity check is best read as a conservative default, and the override exists for that user.

Obesity Risk is the exception. The Base Model already reaches Recall 1.00 and Precision 0.98, and neither the override nor RUS changes this. The halt there follows from the training data (207 minority samples in the core against $n_{\text{half}} = 280$), not from test-set difficulty.

E. Ablation Study

Two projection-only pipelines were evaluated to check whether the Recall gain comes from the resampling step or from the ridge projection alone: AF+Ridge (full feature matrix through the ridge projection, no sampling) and ANOVA- k +Ridge (ANOVA-selected features matched to the RFECV count, through the ridge projection, no sampling). Table VI reports the results.

Both projection-only pipelines return a median Recall of 0.64 against 0.80 for full VBU, a difference of 0.16. VBU's Recall spread across datasets is also narrower, 0.34 against 0.81 for both projection-only pipelines. G-Mean follows the same direction: full VBU returns 0.82 against 0.77 for both.

F1 moves the other way. AF+Ridge returns 0.69 and ANOVA- k +Ridge returns 0.67, both above VBU's 0.66, because keeping all training samples preserves more Precision. This is consistent with the precision cost described in Section V-C. The two projection-only pipelines differ by only 0.02 in median F1 and by less than 0.01 in median Recall, which suggests that the choice of feature selector matters little here.

The scope of this ablation is limited. It shows that adding a resampling step raises Recall, but RUS, which uses none of VBU's design choices, reaches a similar median Recall (0.82, Table III). The ablation does not vary the variance weights, the restriction to core nodes, the number of nodes or the capacity check, and it includes no measure of how closely the retained majority matches the original majority distribution. The claim that these components add value beyond random selection is therefore not supported by the present experiments (Section V-G).

F. Runtime

In practice, the full VBU pipeline runtime varies with dataset size, driven by the RFECV feature selection step.

Table VII reports the per-dataset breakdown across all 14 datasets. Total runtime ranges from 4.27 seconds on Occupancy Detection to 20.08 seconds on Bank Marketing, reflecting the cost of cross-validated Random Forest fitting inside RFECV across datasets of varying size and feature count. This is a feature selection cost, not a sampling cost, and any pipeline using RFECV would incur the same overhead regardless of the downstream sampler.

The sampler runtime is small on all 14 datasets, ranging from 0.0004 seconds on Ozone Level and Obesity Risk to 0.0106 seconds on Spambase. It stays under 11 milliseconds on every dataset tested, for training sets from under 1,200 instances to over 28,000. For the halted Cohort B datasets, the reported time covers only the capacity check.

A total of about 20 seconds is acceptable for a step that runs once during training, and prediction uses only the trained classifier. For larger datasets, users can keep RFECV or run a cheaper feature selector beforehand and pass the selected features to VBU. The sampler itself grows only as $\mathcal{O}(n \log n)$, but a replacement selector must still give a projection that separates the classes, and we have not tested alternatives to RFECV.

G. Limitations

All results come from one stratified split and one seed per dataset, with a single Random Forest configuration, so no variance across partitions is reported. Some test sets hold very few minority cases. Abalone19 has six and Ozone Level eleven, so a Recall of 0.67 on Abalone19 corresponds to four detections, and the smallest Cohort A test set (AIDS ACTG175) has 104. The Wilcoxon and Nemenyi tests rest on eight datasets and have low power. Repeated stratified splits, or cross-validation where the minority count permits, reported with confidence intervals, would address these points.

The comparison is restricted to undersampling methods. Class weighting and threshold moving were excluded by design, and because VBU's Recall gain comes with a lower AUC-PR, a similar gain could probably be reached by lowering the decision threshold of the Base Model. Only one classifier family was evaluated. The projection also depends on a Random Forest inside RFECV and on a linear ridge classifier, so VBU is not independent of the classifier in the way an unsupervised sampler would be, and its behavior with other classifiers is untested.

Some design choices carry costs of their own. VBU compresses all features into one linear score, which can hide structure that a multivariate method would keep. Node boundaries are set separately for each class, so the two classes are matched by count rather than by score range. VBU draws the minority class only from its core nodes and may subsample it during balancing, which works against the aim of protecting minority cases. On Spambase it kept 64% of the training minority, whereas RUS keeps all of it. The experiments also do not test the individual parts of the sampler. Variants with uniform quotas, with random selection inside the same core nodes, or without outlier exclusion were not run, and no direct measure, such as a distributional distance between the retained and original majority samples, was used to check whether variance-

TABLE V. COHORT B RESULTS ON THE TEST SET, CONFIGURATION 2: BASE MODEL, RUS, AND VBU WITH THE DIAGNOSTIC OVERRIDE ENGAGED (HALT BYPASSED, CORE NODES ONLY)

Dataset	Recall			Precision			F1			Test min.
	Base	RUS	VBU	Base	RUS	VBU	Base	RUS	VBU	
Iranian Churn	0.71	0.97	0.85	0.84	0.61	0.66	0.77	0.75	0.74	89
Abalone19	0.00	0.83	0.67	0.00	0.01	0.01	0.00	0.03	0.02	6
Obesity Risk	1.00	1.00	1.00	0.98	0.98	0.98	0.99	0.99	0.99	65
Ozone Level	0.09	0.45	0.64	0.08	0.05	0.06	0.09	0.10	0.10	11
Taiwanese Bankruptcy	0.16	0.86	0.73	0.58	0.16	0.20	0.25	0.27	0.31	44
Page Blocks	0.85	0.98	0.97	0.90	0.64	0.62	0.88	0.78	0.76	103

VBU columns report results with the diagnostic override engaged. The Base Model is the non-intervened control. Test min.=number of minority cases in the test set. Obesity Risk shows no change because the Base Model already reaches ceiling performance.

TABLE VI. ABLATION STUDY RESULTS, MEDIAN AND SPREAD ACROSS COHORT A (8 DATASETS), CONFIGURATION 2, TEST SET

Pipeline	Recall		F1		G-Mean	
	Med	Spr	Med	Spr	Med	Spr
AF+Ridge	0.64	0.81	0.69	0.72	0.77	0.57
ANOVA- k +Ridge	0.64	0.81	0.67	0.72	0.77	0.58
Full VBU	0.80	0.34	0.66	0.60	0.82	0.48

Med = median; Spr = spread (max – min) across 8 Cohort A datasets. Bold marks VBU's primary target metrics.

TABLE VII. PER-DATASET PIPELINE RUNTIME BREAKDOWN ACROSS ALL 14 DATASETS

Dataset	Coh.	RFECV (s)	Core (s)	Total (s)
<i>Cohort A: Parity-Achieved</i>				
AIDS ACTG175	A	5.90	0.0077	6.00
Student Dropout	A	9.55	0.0097	10.02
Bank Marketing	A	18.62	0.0076	20.08
Credit Card Default	A	18.55	0.0075	19.05
NASA JM1	A	6.50	0.0088	6.88
Occupancy Detection	A	4.09	0.0072	4.27
Online Shoppers	A	10.24	0.0073	10.68
Spambase	A	6.43	0.0106	6.60
<i>Cohort B: Sparsity-Halted</i>				
Iranian Churn	B	5.68	0.0005	5.80
Obesity Risk	B	4.57	0.0004	4.61
Page Blocks	B	9.99	0.0005	10.09
Taiwanese Bankruptcy	B	18.20	0.0008	19.26
Ozone Level	B	5.76	0.0004	5.84
Abalone19	B	6.26	0.0007	6.33

The RFECV column covers RFECV, the ridge projection and node assignment. The Core column covers the capacity check, quota allocation and the within-node draw. For Cohort B the Core column reports the capacity check only, since allocation does not run after a halt.

weighted allocation preserves the majority distribution better than random selection.

VI. CONCLUSION

Class imbalance remains a persistent obstacle in tabular classification because standard classifiers favor the majority class and overlook the minority cases that matter most in practice. This paper presented Variance-Based Undersampling (VBU), a tuning-free data-level method built around three questions. Which majority samples to retain is answered by a supervised one-dimensional projection with IQR-core stratification. How many to draw from each region is answered

through variance-weighted allocation. When to stop is answered by capacity-aware sampling control, a check based on Yamane's formula that halts reduction when the minority class is too small, with a diagnostic override for recall-first use. The evidence comes from a fourteen-dataset evaluation against six undersampling methods and a Base Model.

On the eight datasets where the capacity check allowed reduction, VBU raised median Recall by 0.24 over the Base Model (Holm-adjusted Wilcoxon $p = 0.039$) and had the best mean Recall rank (2.25), with the Nemenyi test separating it from the Base Model ($p = 0.0002$) and Tomek Links ($p = 0.0048$). Precision fell by 0.22 and AUC-PR by 0.04, while the F1 change of -0.02 was not significant. These gains have to be read against RUS, which matched VBU's Recall and scored higher on G-Mean, F1, Precision and AUC-PR. The Recall gain comes from rebalancing the training set, and in these experiments the variance-guided selection did not add to it. On the six halted datasets, forcing balance lowered Precision on four of them under both the override and RUS, which supports treating the halt as a conservative default rather than as a proven optimum.

What VBU offers, then, is a documented and reproducible procedure rather than a performance advantage, with quotas that follow from the data and an explicit rule for when not to undersample. Its main limitations, discussed in Section V-G, are the reliance on a single projected variable and an evaluation built on one split and one classifier.

The next step is to test VBU under repeated stratified splits with confidence intervals, and to compare it with class weighting and threshold moving using more than one classifier. A component-level ablation with uniform quotas, random selection within the core nodes and no outlier exclusion, together with a distributional distance between the retained and original majority samples, would show whether the variance-weighted design adds anything beyond random selection. Longer term, multivariate stratification could replace the single projected score, and a variance-based oversampling component could address the sparse datasets that the capacity check currently leaves untouched. Extending VBU to non-tabular data through tabular representations of other inputs remains an open question.

ACKNOWLEDGMENT

This work was supported by YUTP-FRG 015LC0-584.

DECLARATION ON GENERATIVE AI

The authors used generative AI tools in preparing this manuscript. Claude (Anthropic) was used to assist with drafting and language editing across all sections, and with the revision prepared in response to the reviewers' comments, including re-checking the reported statistics against the experiment output files. Gemini (Google) was used for outline generation and structural planning. All AI-assisted text was reviewed, revised and approved by the authors. The research design, the method, the experiments, the numerical results and their interpretation are the authors' own. The authors take full responsibility for the content of this manuscript.

REFERENCES

- [1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert systems with applications*, vol. 73, pp. 220–239, 2017.
- [3] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in artificial intelligence*, vol. 5, no. 4, pp. 221–232, 2016.
- [4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [5] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976.
- [6] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.
- [7] T. Yamane, *Statistics: An Introductory Analysis*, 3rd ed. New York: Harper and Row, 1973.
- [8] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artificial Intelligence Review*, vol. 57, no. 6, p. 137, 2024.
- [9] G. Aguiar, B. Krawczyk, and A. Cano, "A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework," *Machine learning*, vol. 113, no. 7, pp. 4165–4243, 2024.
- [10] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, "The class imbalance problem in deep learning," *Machine Learning*, vol. 113, no. 7, pp. 4845–4901, 2024.
- [11] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of big data*, vol. 6, no. 1, pp. 1–54, 2019.
- [12] M. Salmi, D. Atif, D. Oliva, A. Abraham, and S. Ventura, "Handling imbalanced medical datasets: review of a decade of research," *Artificial intelligence review*, vol. 57, no. 10, p. 273, 2024.
- [13] D. Breskuvienė and G. Dzemyda, "Enhancing credit card fraud detection: highly imbalanced data case," *Journal of Big Data*, vol. 11, no. 1, p. 182, 2024.
- [14] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.
- [15] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM computing surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.
- [16] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.
- [17] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. Ieee, 2008, pp. 1322–1328.
- [18] S. Matharaarachchi, M. Domaratzki, and S. Muthukumarana, "Enhancing smote for imbalanced data with abnormal minority instances," *Machine Learning with Applications*, vol. 18, p. 100597, 2024.
- [19] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.
- [20] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, vol. 40, no. 1, pp. 185–197, 2009.
- [21] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 2001, pp. 63–66.
- [22] 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*, vol. 18, no. 17, pp. 1–5, 2017.
- [23] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126. ICML United States, 2003, pp. 1–7.
- [24] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert systems with applications*, vol. 36, no. 3, pp. 5718–5727, 2009.
- [25] G. E. 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*, vol. 6, no. 1, pp. 20–29, 2004.
- [26] M. Kim and K.-B. Hwang, "An empirical evaluation of sampling methods for the classification of imbalanced data," *PLoS One*, vol. 17, no. 7, p. e0271260, 2022.
- [27] T. Wongvorachan, S. He, and O. Bulut, "A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining," *Information*, vol. 14, no. 1, p. 54, 2023.
- [28] J. Wainer, "An empirical evaluation of imbalanced data strategies from a practitioner's point of view," *Expert Systems with Applications*, vol. 256, p. 124863, 2024.
- [29] S. Szeghalmy and A. Fazekas, "A comparative study on noise filtering of imbalanced data sets," *Knowledge-Based Systems*, vol. 301, p. 112236, 2024.
- [30] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.
- [31] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [32] J. W. Tukey, *Exploratory Data Analysis*. Reading, Massachusetts: Addison-Wesley, 1977.
- [33] W. G. Cochran, *Sampling Techniques*, 3rd ed. New York: John Wiley and Sons, 1977.
- [34] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
- [35] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: classification evaluation," *Nature methods*, vol. 13, no. 8, pp. 603–604, 2016.
- [36] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.
- [37] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 3121–3124.
- [38] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.
- [39] 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*, vol. 10, no. 3, p. e0118432, 2015.

- [40] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
- [41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. [Online]. Available: <https://jmlr.org/papers/v7/demsar06a.html>
- [42] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *Journal of Machine Learning Research*, vol. 17, no. 5, pp. 1–10, 2016. [Online]. Available: <https://jmlr.org/papers/v17/benavoli16a.html>
- [43] P. B. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, 1963.
- [44] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [45] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 2015, pp. 159–166.