Evaluating Resampling Strategies for Imbalanced Classification: A Comparative Study Across Varying Imbalance Ratios

Marwan Eslam Ouda
AI Department
Faculty of computer and artificial intelligence benha University
Benha, Egypt
marwan403465@fci.bu.edu.eg

ABSTRACT

Class imbalance remains a common problem in machine learning, especially when rare cases carry high value. In such settings, minority classes appear too infrequently during training. This work compares five resampling strategies: no resampling, random oversampling, random undersampling, SMOTE, and ADASYN. Experiments are conducted on three datasets with different imbalance levels: moderate (1.87:1), extreme (601:1), and nearly balanced (1.05:1). Random Forest and XGBoost models are evaluated using F1-score, precision, recall, AUC-ROC, and training time. Results show that random undersampling collapses under extreme imbalance, reaching F1=0.086. In contrast, baseline XGBoost achieves strong performance (F1=0.845) without resampling. On moderate imbalance, simple undersampling slightly outperforms synthetic methods. SMOTE and ADASYN increase runtime by 45–50× while offering only minor gains. These findings highlight when resampling helps and when it adds unnecessary cost.

***Index Terms*—imbalanced learning, resampling techniques, SMOTE, ADASYN, classification**

I. Introduction

Many real datasets show uneven class distributions. Fraud data often contains fewer than one percent positive cases. Medical datasets may include only a small number of rare diagnoses. Similar patterns appear in intrusion detection and quality inspection. Standard classifiers tend to favor the majority class. As a result, minority cases are often missed, even though they matter most.

A. Problem Statement

A wide range of methods exists to address class imbalance. Still, practitioners face a simple but unresolved issue: which resampling method should be used, and when? Prior studies report mixed results across datasets and models. In addition, most comparisons ignore training cost, even though it matters in real systems. The core question remains open. Do advanced resampling techniques justify their added complexity, or are simpler methods enough? This question becomes sharper when comparing moderate imbalance to extreme cases above 100:1.

B. Research Objectives

This study addresses four specific research questions:

- **RQ1**: How does resampling effectiveness vary across different imbalance severities (moderate vs. extreme)?
- **RQ2**: What is the computational cost-benefit trade-off for synthetic oversampling methods?
- **RQ3**: Under what conditions does random undersampling fail, and why?
- **RQ4**: Can modern gradient boosting classifiers handle imbalance without explicit resampling?

C. Contributions

Our work makes three primary contributions:

1) **Systematic evaluation across imbalance severities**: First, it evaluates resampling across a wide range of imbalance ratios
2) **Computational cost-benefit analysis**: it reports training time alongside predictive metrics.
3) **Failure mode identification**: it documents clear failure cases and defines safe usage limits..

II. Related Work

A. Resampling Approaches

Resampling methods modify the training distribution directly. Random oversampling repeats minority samples, while random undersampling removes majority cases. Despite their simplicity, these methods serve as strong baselines.

SMOTE generates synthetic minority samples by interpolating between neighbors. ADASYN extends this idea by focusing more on sparse minority regions. Both aim to expand decision boundaries without simple duplication.

B. Classifier Selection

Boosting methods such as XGBoost include built-in mechanisms for imbalance, including weighted loss terms and tree-based splitting. These features may reduce the need for external resampling, but their limits remain unclear.

C. Gaps in Existing Research

Most earlier studies focus on narrow imbalance ranges. Few report computational cost. Extreme imbalance cases are rarely explored. This study addresses all three gaps.

III. Methodology

A. Experimental Datasets

We selected three publicly available datasets representing distinct imbalance regimes:

B. Datasets

Three public datasets are used.

Diabetes: The Pima Indians Diabetes dataset includes 768 samples with eight features. Positive cases represent 34.9

Credit Card Fraud: A subset of 50,000 transactions is used. Only 83 cases are fraud, yielding a 601:1 imbalance.

Heart Disease: This dataset contains 1,025 records with near-equal class distribution (1.05:1), serving as a control.

C. Resampling Techniques

We evaluated five strategies:

1) **Baseline (No Resampling)**: Train directly on original class distribution.
2) **Random Oversampling**: Duplicate minority class instances with replacement until class balance is achieved.
3) **Random Undersampling**: Remove majority class instances randomly until class balance is achieved.
4) **SMOTE**: Generate synthetic minority instances using 5-nearest neighbors interpolation.
5) **ADASYN**: Generate adaptive synthetic samples with density-based weighting, using 5-nearest neighbors.

Note: ADASYN was excluded from the Heart Disease dataset due to insufficient imbalance (ratio ¡ 1.5:1), where the algorithm cannot generate meaningful synthetic samples.

D. Classification Algorithms

Two widely-adopted classifiers were employed:

- **Random Forest**: Ensemble of 100 decision trees with maximum depth of 10, using bootstrap aggregation and random feature selection.
- **XGBoost**: Gradient boosting with 100 estimators, learning rate of 0.1, and maximum depth of 6, utilizing second-order optimization.

E. Evaluation Protocol

1) Performance Metrics: We measured:

- **F1-Score**: Harmonic mean of precision and recall, balancing false positives and false negatives
- **Precision**: Ratio of correctly predicted positive cases to total predicted positives
- **Recall**: Ratio of correctly predicted positive cases to actual positives
- **AUC-ROC**: Area under receiver operating characteristic curve, measuring discriminative ability across thresholds
- **Training Time**: Wall-clock time for complete model training

2) Validation Procedure: Five-fold stratified cross-validation was applied to ensure robust performance estimation. Stratification maintained original class proportions within each fold, critical for imbalanced datasets. All features were standardized using z-score normalization before modeling.

3) Implementation Details: Experiments utilized Python 3.12 with scikit-learn 1.3, imbalanced-learn 0.11, and XGBoost 2.0. Computations ran on Google Colab with Intel Xeon CPU (2.3 GHz) and 12.7 GB RAM. Random seeds were fixed at 42 for reproducibility.

IV. Experimental Results

A. Overall Performance Analysis

Table I presents average F1-scores across all datasets and classifiers. Baseline (no resampling) achieved the highest overall performance (0.816), followed by random oversampling (0.812) and SMOTE (0.796). ADASYN (0.676) and random undersampling (0.594) performed substantially worse in aggregate.

TABLE I
Average F1-Scores Across All Datasets and Classifiers

Method	Mean F1	Std Dev
Baseline	**0.8163**	0.192
Random Oversampling	0.8118	0.171
SMOTE	0.7960	0.174
ADASYN	0.6760	0.026
Random Undersampling	0.5944	0.348

However, aggregate metrics obscure critical dataset-specific patterns. The high standard deviation for random undersampling (0.348) signals inconsistent performance across imbalance regimes.

B. Dataset-Specific Results

1) Diabetes Dataset (Moderate Imbalance: 1.87:1): Table II shows performance on the diabetes dataset.

TABLE II
Diabetes Dataset Results (F1-Score)

Method + Classifier	F1	AUC-ROC
Random Under + RF	**0.681**	0.752
ADASYN + RF	0.680	0.752
Random Over + RF	0.674	0.749
Random Under + XGB	0.672	0.746
SMOTE + RF	0.662	0.738
SMOTE + XGB	0.648	0.727
Baseline + RF	0.647	0.731
ADASYN + XGB	0.636	0.717
Random Over + XGB	0.631	0.716
Baseline + XGB	0.623	0.714

Key Findings:

- Random undersampling achieved best F1-score (0.681), surprising given its simplicity.

- Performance differences were modest (0.681 vs 0.623), spanning only 0.058 F1 points.
- Random Forest consistently outperformed XGBoost (average 0.665 vs 0.642).
- ADASYN provided negligible benefit over simpler methods (0.680 vs 0.681).

2) Credit Card Fraud Dataset (Extreme Imbalance: 601:1): Table III presents results for the highly imbalanced fraud detection task.

TABLE III
CREDIT CARD FRAUD DATASET RESULTS (F1-SCORE)

Method + Classifier	F1	AUC-ROC
Baseline + XGB	**0.845**	0.879
Random Over + XGB	0.793	0.892
Baseline + RF	0.787	0.855
Random Over + RF	0.776	0.862
SMOTE + RF	0.755	0.880
SMOTE + XGB	0.711	0.904
ADASYN + XGB	0.701	0.909
ADASYN + RF	0.687	0.891
Catastrophic Failure:		
Random Under + RF	0.135	0.930
Random Under + XGB	0.086	0.935

Critical Findings:

1) **Baseline dominance**: XGBoost without resampling achieved highest F1 (0.845), suggesting modern boosting algorithms handle extreme imbalance effectively through internal weighting mechanisms.
2) **Undersampling catastrophe**: Random undersampling collapsed to F1=0.086-0.135, representing 90% performance degradation. With only 83 fraud cases, undersampling to 80 total samples destroys critical information.
3) **Synthetic methods underperform**: SMOTE (0.711) and ADASYN (0.687) both underperformed baseline, indicating that synthetic sample generation in extremely sparse minority regions produces low-quality instances.
4) **AUC-ROC divergence**: Interestingly, undersampling achieved highest AUC-ROC (0.930-0.935) despite lowest F1, revealing that it learns good ranking but poor decision boundaries due to insufficient minority examples.

3) Heart Disease Dataset (Balanced: 1.05:1): All methods achieved near-perfect performance (F1 0.996, AUC-ROC 0.996), confirming that resampling provides no benefit when classes are naturally balanced. This validates our experimental design by demonstrating that observed differences on imbalanced datasets result from class distribution rather than inherent data difficulty.

C. Computational Cost Analysis

Table IV presents average training times across datasets.

Computational Insights:

- Random undersampling was 20× faster (1.0s vs 14.8s) due to reduced dataset size, but produced worst overall F1 (0.594).

TABLE IV
AVERAGE TRAINING TIME PER METHOD (SECONDS)

Method	Avg Time	vs Baseline	F1/Time
Random Under	1.0	0.05×	0.620
Baseline	14.8	1.00×	0.055
Random Over	13.1	0.88×	0.062
SMOTE	27.3	1.84×	0.029
ADASYN	26.2	1.77×	0.026

- SMOTE and ADASYN incurred 84% and 77% overhead respectively, requiring 27-26 seconds versus 15 seconds for baseline.
- Cost-benefit metric (F1/Time) reveals baseline provides best performance per computational unit (0.055), followed by random oversampling (0.062).
- On the fraud dataset, SMOTE took 153 seconds (45× longer than baseline's 3.4s XGBoost) while achieving 16% worse F1 (0.711 vs 0.845).

D. Classifier Comparison

XGBoost demonstrated superior handling of imbalance without resampling, achieving:

- 7% higher F1 than Random Forest on extreme imbalance (0.845 vs 0.787)
- 8× faster training (3.4s vs 84.1s on fraud dataset)
- Consistent performance across resampling methods

Random Forest benefited more from resampling on moderate imbalance, showing 5% F1 improvement with random undersampling (0.681 vs 0.647).

V. DISCUSSION

A. Imbalance Severity Drives Method Selection

Our results reveal a clear pattern: optimal resampling strategy depends critically on imbalance ratio.

Moderate Imbalance (2:1 to 10:1): All resampling methods perform comparably, with random undersampling surprisingly competitive (F1=0.681). Computational efficiency favors simple methods over sophisticated alternatives.

Extreme Imbalance (¿100:1): Baseline modern classifiers (XGBoost) outperform all resampling approaches. Synthetic methods fail to produce quality samples in extremely sparse minority regions, while undersampling discards critical information.

Balanced Data: Resampling provides no benefit, as expected.

B. The Random Undersampling Paradox

Random undersampling exhibits schizophrenic behavior: excellent on moderate imbalance (F1=0.681, best performer), catastrophic on extreme imbalance (F1=0.086, worst performer by far).

The failure mechanism is straightforward: with only 83 fraud cases, undersampling to 80 total samples reduces the effective training set to 60 minority examples after 5-fold cross-validation. This sample size is insufficient to learn

fraud patterns, despite producing good ranking scores (AUC-ROC=0.935).

This finding establishes a critical safety boundary: *random undersampling should be avoided when minority class contains fewer than 500 samples*.

C. Questioning SMOTE's Necessity

SMOTE and ADASYN have become de facto standards in imbalanced learning research. However, our results challenge their necessity:

- **Marginal gains**: On diabetes dataset, SMOTE achieved F1=0.662 versus baseline 0.647 only 0.015 improvement (2.3%).
- **Extreme imbalance failure**: On fraud dataset, SMOTE (0.711) underperformed baseline (0.845) by 15.9%.
- **Computational cost**: 45-84% longer training time for marginal or negative returns.

These findings suggest SMOTE provides value primarily in a narrow regime (moderate imbalance with tree-based models), contrary to its widespread application.

D. XGBoost's Built-in Imbalance Handling

XGBoost consistently achieved strong baseline performance without resampling, particularly on extreme imbalance (F1=0.845, 7% better than best resampling method). This effectiveness likely stems from:

1) Weighted loss functions that penalize minority errors more heavily
2) Leaf-wise growth that can identify small minority pockets
3) Second-order gradient information providing better boundary refinement

For practitioners, this suggests: *try modern gradient boosting before applying resampling*.

E. Practical Decision Framework

Based on our findings, we propose the following guidelines:

TABLE V
RESAMPLING STRATEGY SELECTION GUIDE

Scenario	Recommended Approach
Imbalance ¡ 5:1	Baseline (no resampling)
Imbalance 5:1-20:1	Random Over or Undersampling
Imbalance ¿ 20:1	XGBoost baseline; avoid undersampling
Minority ¡ 500 samples	Never use random undersampling
Time-constrained	Random oversampling; avoid SMOTE
Research/academic	SMOTE for comparability

F. Limitations

This study has several constraints:

1) **Limited dataset diversity**: Three datasets, though spanning three imbalance orders, may not generalize to all domains.
2) **Hyperparameter constraints**: Computational limits restricted extensive tuning. Optimal parameters may shift performance rankings.
3) **Classifier selection**: Results may differ for neural networks or support vector machines.
4) **Sampling strategy variants**: We evaluated canonical SMOTE and ADASYN; borderline-SMOTE and other variants remain unexplored.
5) **Multi-class problems**: This study focused on binary classification.

G. Broader Implications

Our findings have three implications for research and practice:

1. Algorithmic progress reduces preprocessing needs: Modern classifiers increasingly handle distributional challenges internally, reducing reliance on data-level fixes. This trend mirrors developments in other domains (e.g., batch normalization reducing need for careful initialization in deep learning).

2. Computational cost deserves equal consideration: Academic papers rarely report training time, focusing solely on predictive metrics. For production deployment, a method 2× slower with 1% better F1 may be inferior to a faster baseline.

3. Method selection requires regime awareness: No universal "best" resampling method exists. Practitioners must match technique to problem characteristics rather than applying sophisticated methods by default.

VI. CONCLUSION AND FUTURE DIRECTIONS

This study compares resampling strategies across multiple imbalance regimes. Modern boosting methods often outperform resampling, especially under extreme skew. Random undersampling shows sharp failure boundaries. Overall, simpler baselines remain strong choices.

A. Key Findings

Three primary conclusions emerge:

1) **Baseline competitiveness**: Modern gradient boosting (XGBoost) without resampling achieved best performance on extreme imbalance (F1=0.845), outperforming all resampling methods including sophisticated SMOTE variants.
2) **Undersampling's dual nature**: Random undersampling excels on moderate imbalance (F1=0.681, best method) but fails catastrophically on extreme imbalance (F1=0.086, 90% degradation), establishing clear safety boundaries.
3) **Cost-benefit reality**: SMOTE and ADASYN impose 45-84% computational overhead while providing minimal gains on moderate imbalance (2.3% F1 improvement) and negative returns on extreme imbalance (-15.9% F1 degradation).

B. Practical Recommendations

For practitioners facing imbalanced classification:

- Start with XGBoost or LightGBM baseline before attempting resampling
- Apply random oversampling for quick wins on moderate imbalance
- Avoid random undersampling when minority class contains fewer than 500 samples
- Reserve SMOTE for academic comparisons or specific niches where proven effective
- Always measure computational cost alongside predictive performance

C. Future Research Directions

Four promising avenues warrant investigation:

1) **Hybrid approaches**: Combine algorithmic methods (cost-sensitive learning) with selective resampling to maximize benefits while minimizing overhead.
2) **Deep learning adaptation**: Evaluate whether findings generalize to neural networks, which have different inductive biases than tree-based models.
3) **Multi-class extension**: Extend analysis to multi-class imbalanced problems where complexity increases combinatorially.
4) **Theoretical foundations**: Develop theory explaining when and why SMOTE succeeds or fails, moving beyond empirical observation to principled understanding.

D. Closing Perspective

The quest for sophisticated solutions to imbalanced learning may have overlooked a simpler truth: modern algorithms increasingly handle distributional challenges without explicit preprocessing. Rather than reflexively applying complex resampling techniques, practitioners should empirically evaluate whether simpler baselines suffice for their specific problem—a principle that extends beyond imbalanced learning to machine learning practice generally.

ACKNOWLEDGMENTS

Special acknowledgment to the developers of scikit-learn, imbalanced-learn, and XGBoost for their invaluable tools.

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*, 2002.

[2] 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*, 2008.

[3] 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*, 2004.