Evaluation Protocols Under Extreme Class Imbalance: Evidence from a Newborn Screening Case Study

Nicole Sabourin[†,‡*], Paula Branco[†], Matthew Henderson[‡,◇]
[†] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario , Canada
[‡] Newborn Screening Ontario, Children's Hospital of Eastern Ontario, Ottawa, Ontario, Canada
[◇] Department of Pathology and Laboratory Medicine, University of Ottawa, Ottawa, Ontario, Canada

Abstract

Evaluation protocols, such as cross-validation and bootstrap, are extensively used when experimenting with machine learning and AI models to obtain reliable performance estimates. However, the choice of the specific configurations used, e.g., 5-fold versus 10-fold cross-validation, and the choice of strategy for hyper-parameter tuning are often arbitrary, with researchers relying on frequently used defaults. There is limited knowledge about how these selections influence the reported performance, particularly in scenarios characterized by extreme class imbalance. In such challenging scenarios, researchers often apply resampling strategies, such as random oversampling or smote, to improve the performance on the rare class. However, their effects on performance estimation under such extreme conditions also remain largely unexplored. This paper investigates the implications of multiple evaluation protocol choices in the context of extreme class imbalance, using a real-world case study in newborn screening to illustrate the practical impact on model assessment and reliability. Our findings show that some design choices critically influence the variability of the results, and different configurations can affect model evaluation metrics and robustness, sometimes leading to conflicting conclusions about the best-performing model.

Keywords: Performance Estimation, Hyper-parameter Tuning, Extreme Class Imbalance.

1. Introduction

Evaluation protocols play a central role in the experimental assessment of machine learning models. Cross-validation schemes or other data partitioning strategies, and hyper-parameter tuning procedures are routinely adopted to estimate predictive performance and support model selection. However, their specific configurations are often selected by convention or convenience, with limited understanding of their impact on the reported results. This issue becomes particularly critical in scenarios of extreme class imbalance, where the minority class is severely under-represented, and performance estimates are inherently fragile.

Resampling strategies are commonly employed to mitigate imbalance; however, the interaction between such techniques and evaluation protocol design remains largely unexplored under extreme imbalance conditions. In particular, little is known about how different validation and tuning configurations influence performance variability, robustness, and the stability of model rankings.

In this paper, we investigate these issues in the context of a real-world case study characterized by severe class imbalance. Our analysis reveals that certain protocol decisions have a substantial effect on performance variability and robustness, and can even lead to conflicting conclusions regarding the best-performing model. Additionally, we provide insights into how hyper-parameter tuning strategies interact with evaluation design in these challenging settings.

[*] nsabo068@uottawa.ca

This study compares evaluation methodologies under severely imbalanced datasets and is guided by the following research questions:

(1) RQ1: How do different model evaluation strategies and hyper-parameter tuning strategies affect model performance and robustness?
(2) RQ2: For cross-validation based approaches, how sensitive is model performance to the number of folds?
(3) RQ3: How does resampling influence model performance for different model evaluation and hyper-parameter tuning strategies?

Our main contributions are the following. We provide: i) a thorough set of experiments of evaluation protocols including performance estimation methods and hyper-parameter tuning strategies on multiple metrics for class imbalance problems; ii) experiments on the effectiveness of using resampling techniques in an extremely imbalanced scenario; and iii) a set of recommendations and guidelines for practitioners and end-users tackling extremely imbalanced problems.

This paper is structured as follows. Section 2 provides an overview of the state-of-the-art in performance estimation methods in extreme class imbalance scenarios. In Section 3 we present the dataset used in our case study, and in Section 4 we describe the experimental methodology of the paper. Section 5 presents and discusses the results obtained. Finally, Section 6 concludes the paper.

2. **Related Work**

This work relates to prior research in three main areas: model evaluation under class imbalance, resampling techniques for imbalanced classification, and hyper-parameter optimization and evaluation methodology.

Evaluating classification models under class imbalance is a well-studied challenge. However, this is not the case for extremely imbalanced scenarios. Evaluating classification models reliably is a long-standing challenge in machine learning, particularly when model selection and performance estimation are sensitive to data partitioning strategies. Prior work has highlighted the limitations of accuracy-based evaluation in such settings and emphasized the use of recall and precision-oriented metrics, including F-measures and geometric mean, to better reflect minority-class performance.

Early work by Kohavi [1] systematically compared cross-validation and bootstrap methods for accuracy estimation and model selection, demonstrating that the evaluation strategy can substantially influence both bias and variance of performance estimates. In particular, Kohavi showed that increasing the number of folds does not necessarily improve estimation quality and that ten-fold stratified cross-validation often provides a favourable trade-off between bias and variance for model selection [1]. Stratified cross-validation was highlighted as essential to ensure that minority classes are adequately represented in each fold, and therefore mitigate the bias in performance metrics [2]. Forman et al. [3] showed that increasing the number of folds in cross-validation can produce more granular estimates, but in small or imbalanced datasets it may inflate performance and yield unstable or undefined metrics (e.g., precision or recall) due to insufficient positive instances [3]. Still, these studies did not consider extreme class imbalanced domains.

Additionally, resampling methods are commonly employed to mitigate class imbalance during training. Random oversampling has been found to increase recall in models trained on imbalanced data, while lowering precision due to the increasing risk of overfitting [4]. The interaction between cross-validation and resampling techniques in imbalanced settings has been explicitly examined by Santos et al. [4], who showed that improper integration of

oversampling within cross-validation can lead to overoptimistic performance estimates, and that oversampling techniques that generate replicas of training data can lead to overfitting.

Hyper-parameter optimization methods are also a critical component of machine learning pipelines, and as such, prior works have studied the impact of different approaches on model performance. Recent work has noted that evaluation design choices can influence reported performance estimates [5].

While prior work has examined evaluation strategies, resampling techniques, and hyper-parameter optimization independently, fewer studies have systematically explored their interactions under severe class imbalance. This work contributes an empirical analysis of how evaluation strategy, hyper-parameter optimization, and resampling jointly influence performance estimates.

3. Dataset

3.1. Background

Fatty acid oxidation disorders (FAODs) are a group of rare, serious, genetic disorders that manifest upon birth and can be life-threatening if not diagnosed and treated rapidly. FAODs affect an individual's ability to break down and metabolize fatty acids. As such, the individual cannot rely on fats as a vital source of energy, which leads to life-threatening symptoms such as hypoglycemia, liver dysfunction, cardiomyopathy, and others [6]. In Ontario, newborns undergo a simple screening process when they are born to test for many genetic conditions. A small blood sample is taken from the baby's heel and certain markers are measured within the sample to assess the probability of the child possessing a genetic disorder. The current NSO process uses these measurements to determine positive or negative screening determinations for a wide range of diseases. A positive screening determination indicates an increased likelihood of the disease. FAODs are part of the conditions screened for by Newborn Screening Ontario (NSO), and this case study's focus.

This study focused on four different types of FAODs: carnitine uptake disorder (CUD), Medium Chain Acyl CoA Dehydrogenase Deficiency (MCAD), Long Chain 3-Hydroxyacyl-CoA Dehydrogenase Deficiency (LCHAD), and Very Long Chain Acyl CoA Dehydrogenase Deficiency (VLCAD). These diseases have very low prevalence, ranging from 1 in 20,000 (MCAD) to 1 in 150,000 (LCHAD) [7]. This rarity is reflected in the source data for this case study, leading to a severely imbalanced dataset for experimentation.

3.2. Dataset Description

The dataset used in this study (the evaluation protocol experimental dataset) was built from NSO data collected between August 2019 and September 2024[1]. The NSO source data is divided into two initial datasets: the analyte and screening determinations, and the FAOD outcomes. The analyte and screening determinations set includes general patient information and analyte measurements. The evaluation protocol experimental dataset contains 7 categories of features extracted from, or derived using, the analyte and screening determinations data, for a total of 59 feature columns. These features and categories are listed in Table 1.

To ensure compatibility with decision tree models, categorical features from the analytes and screening determinations dataset were encoded as numerical values using one-hot encoding (sex_male, sex_female) or binary encoding (transfusion_status, HGB_Pattern).

Each record in the analyte and screening determinations dataset also includes NSO screening determinations for a wide range of diseases screened. The screening determinations of

[1]Ethical Approval: This study was approved, with the requirement to obtain consent waived, by the (undisclosed for double blind review) Hospital Research Ethics Board (REB)

Category	Features
Clinical Features	gestational_age, birth_weight, age_at_collection, transfusion_status, sex_female, sex_male
Acyl-carnitines	C0, C2, C3, C3DC, C4, C4DC, C4OH, C5, C5:1, C5OH, C5DC, C6, C6DC, C8, C8:1, C10, C10:1, C12, C12:1, C14, C14:1, C14:2, C14OH, C16, C16OH, C16:1OH, C18, C18:1, C18:2,C18OH, C18:10H
Amino Acids	ALA, ARG, CIT, GLY, LEU, MET, ORN, PHE, SUAC, TYR, VAL
Hemoglobins	A, F, F1, FAST, HGB_Pattern
Endocrine Markers	TSH, 17OHP
Enzyme Markers	BIOT, GALT
Immune Markers	TREC_QN

Table 1. Feature categories

interest to this study are only those for the following diseases, characterized as FAODs: CUD, MCAD, LCHAD, VLCAD. In the evaluation protocol experimental dataset, FAODs were not modelled separately. Instead, the task is formulated as a **binary classification problem** in which the presence of any FAOD is treated as a positive case.

Records with a positive screening determination for any of these four diseases contain a corresponding entry in the FAOD outcomes. Following a positive screening result, patients are reevaluated by a medical professional and undergo proper testing for the disease. The outcomes dataset contains the result of this evaluation, indicating either a positive or negative diagnosis. This ground truth is the basis for the target column of the evaluation protocol experimental dataset (definitive_diagnosis).

In this binary classification problem, positive diagnoses were given the label 1, and negative diagnoses were given the label 0. Individuals with negative FAOD screening determinations do not have corresponding outcome data entries and were treated as negative cases. Otherwise, the evaluation protocol experimental dataset's target variable was derived from the *Simplified True Positive Category* column of the outcomes dataset, as shown in Table 2.

Simplified True Positive Category	definitive_diagnosis Label
Absent label	(Row removed)
DERF PENDING	(Row removed)
INCIDENTAL	0
No	0
OTHER	0
VARIANT	0
Yes	1

Table 2. definitive_diagnosis label mappings from outcome data

To ensure consistent, high-quality data, samples that meet any of the following criteria were removed:

(1) Positive screenings with missing or pending diagnostic outcomes.
(2) Invalid values for any feature (e.g., negative values where not biologically plausible).
(3) Missing values or features marked as "Not Tested".
(4) An age at collection of less than 1 day or greater than 7 days.

The evaluation protocol experimental dataset contains 675,863 rows and 60 columns (59 features, 1 target). Additional characteristics and statistics of the dataset are shown in Appendix A.

This dataset exhibits a severe class imbalance, making it suitable for addressing the research questions posed in this study. Specifically, the dataset is comprised of 675,802 negative cases and 61 positive cases, for a positive-to-negative class imbalance ratio of approximately 0.009%. Table 3 shows the distribution of the positive cases across all four FAODs considered.

FAOD	Positive cases
CUD	12
MCAD	41
LCHAD	1
VLCAD	7
Total	61

Table 3. Distribution of positive cases across FAODs

4. **Experimental Setup**

To study the impact of model evaluation and hyper-parameter tuning methods under severe class imbalance, two evaluation strategies and two hyper-parameter tuning methods were considered. The model evaluation strategies considered were stratified cross-validation and bootstrap sampling. The bootstrap sampling method constructs training sets for each iteration by sampling with repetition. The corresponding test sets are made up of the samples that were not selected during the resampling process (out-of-bag samples). The sampling is stratified, i.e., it preserves the class imbalance of the dataset. Hyper-parameters were optimized using either Bayesian optimization within cross-validation (BayesSearchCV) [8], or a bootstrap-based Optuna [9] optimization procedure.

Each model evaluation strategy was combined with each hyper-parameter tuning method, resulting in four experimental configurations: Cross-validation with Bayesian Optimization (CV-CV), Bootstrap sampling with Bayesian Optimization (Bootstrap-CV), Cross-validation with Bootstrap-based Optuna (CV-Bootstrap), Bootstrap sampling with Bootstrap-based Optuna (Bootstrap-Bootstrap).

For model evaluation using bootstrap sampling with out-of-bag test set, 20 iterations were performed. For cross-validation, experiments were repeated while varying the number of folds from 2 to 9 in order to assess the sensitivity of model performance to the number of folds. The number of folds is capped at 9 in order to ensure that each test set contains a sufficient amount of positives. Further increasing the number of folds restricts the size of the positive test set to a level that compromises reliable performance.

A decision tree classifier [10] was used as the base model in all experiments. This model was chosen as it is appropriate for the amount of computational resources the research team has available while remaining interpretable. Hyper-parameters were optimized for every fold in the case of cross-validation, or for every iteration in the case of bootstrap sampling. This optimization was carried out with a new inner split of the training set being used. For Bayesian optimization within cross-validation, experiments were repeated across the same range of folds (2 to 9) to evaluate the impact of the number of folds on model performance. For the bootstrap-based Optuna procedure, hyper-parameters were optimized over 50 Optuna trials for 20 different iterations with different train-test sets. Each iteration created a different train-test combination from the full training set, while preserving class imbalance, and determined a unique set of hyper-parameters. The iteration with the best performance determined the final hyper-parameters used to train the model. Table 4 lists the hyper-parameters considered and their corresponding search spaces.

Hyper-parameter	Search space
criterion	{"gini", "entropy"}
max_depth	$\mathbb{Z} \in \{3, 10\}$
min_samples_leaf	$\mathbb{Z} \in \{1, 10\}$
min_samples_split	$\mathbb{Z} \in \{2, 10\}$

Table 4. Hyper-parameters and search spaces for optimization

All experiments were conducted both with and without random oversampling (ROS) applied to the training data, to assess the effectiveness of resampling for different model evaluation and hyper-parameter optimization strategies. SMOTE was also evaluated only in the cross-validation with Bayesian optimization (CV-CV) configuration and was excluded from subsequent experiments to maintain consistency within computational constraints.

Model performance metrics were recorded once per outer cross-validation fold or bootstrap iteration, respectively. The metrics used in this study are Precision, Recall, F1, F10, and geometric mean. In particular, F10 was included to place greater emphasis on recall, which is especially informative in the context of rare disease detection, where false negatives are most costly. The geometric mean was included to account for performance on both the minority and majority classes under severe class imbalance. All metrics used are defined in Table 5.

Metric	Equation
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1	$2 \times \frac{Precision \times Recall}{Precision+Recall}$
F10	$2 \times \frac{Precision \times Recall}{10 \times Precision+Recall}$
Geometric mean	$\sqrt{Recall \times \frac{TN}{TN+FP}}$

Table 5. Evaluation metrics used

All experiments were implemented in Python. Table 6 summarizes the external software components used and their respective functions.

Function	Components
Data manipulation	pandas [11]
Performance metric computation	sklearn.metrics.confusion_matrix
Base experimental model	sklearn.tree.DecisionTreeClassifier
Model evaluation through cross-validation	sklearn.model_selection.StratifiedKFold
Bootstrap sampling	sklearn.utils.resample [10]
Hyper-parameter optimization	skopt.BayesSearchCV, skopt.space [8], optuna [9]
Random oversampling	imblearn.over_sampling.RandomOverSampler [12]

Table 6. Python components used in the implementation and their functions

5. Results and Discussion

5.1. Experimental configurations performance overview

Performance metrics were computed on outer test sets for each cross-validation fold or bootstrap iteration. For bootstrap sampling, results were averaged across 20 iterations. For cross-validation, results were first averaged across outer folds for each fold setting (2–9), and subsequently aggregated across fold settings to obtain the reported values. Table 7 reports the mean of performance metrics for all four experimental configurations, with and without random oversampling.

		precision	recall	f1	f10	geometric mean
Baseline	CV-CV	0.685	0.564	0.596	0.564	0.741
	CV-Bootstrap	0.659	0.554	0.581	0.554	0.733
	Bootstrap-CV	0.603	0.596	0.592	0.596	0.769
	Bootstrap-Bootstrap	0.594	0.620	0.593	0.619	0.784
ROS	CV-CV	0.546	0.602	0.560	0.601	0.768
	CV-Bootstrap	0.564	0.594	0.561	0.593	0.761
	Bootstrap-CV	0.508	0.580	0.510	0.577	0.758
	Bootstrap-Bootstrap	0.441	0.584	0.470	0.579	0.761

Table 7. Average performance of experimental configurations

Across experimental configurations, performance metrics were broadly comparable, with no single configuration consistently outperforming all others on average across all metrics. In the baseline case, cross-validation showed higher precision, and bootstrap sampling showed higher recall, and consequently higher F10. Bootstrap sampling also demonstrated marginally higher geometric mean scores. When random oversampling was applied, cross-validation continued to exhibit higher precision, while recall-oriented metrics, including recall, F10, and geometric mean, showed little variation across configurations.

5.2. Impact of evaluation strategy and cross-validation fold settings

The performance of models trained using cross-validation was plotted across fold settings. Figure 1 shows the precision and recall metrics of the models for 2-fold to 9-fold cross-validation. The plotted metrics represent the average metrics across the outer folds for a particular fold setting. The plots for the other metrics recorded in this study are given in Figure 2 and Appendix B.1.

For cross-validation on this imbalanced dataset, increasing the number of cross-validation folds was associated with higher precision and lower recall. This extends to higher numbers of folds in Bayesian optimization. As the number of folds increases, test sets contain fewer positive samples, favouring more conservative prediction behaviour and resulting in fewer positive predictions. This increases precision, but the smaller number of test samples also increases the sensitivity of recall to missed detections, leading to greater variability and instability in recall estimates when positive samples are scarce [13].

The results of cross validation showed that the optimal hyper-parameter optimization strategy and resampling method for certain evaluation metrics vary between different fold settings. This is demonstrated for F1 in Figure 2. For example, the best performing combination for 3-fold cross-validation is BayesSearchCV with random oversampling, and the best performing combination for 4-fold cross-validation is baseline bootstrap-based optuna.

The number of folds also influenced the variability of performance and the impact of random oversampling, as shown in Figure 3. While random oversampling tended to decrease precision and increase recall, the magnitude of these effects varied across settings.

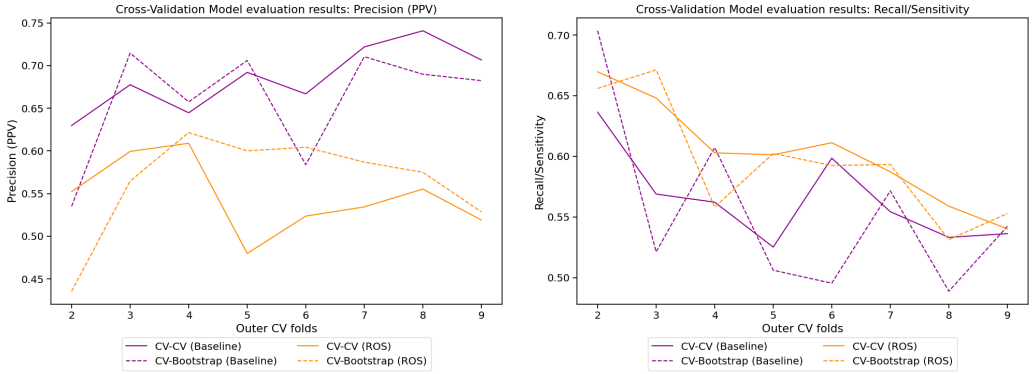

Figure 1. Cross-validation results over different fold settings

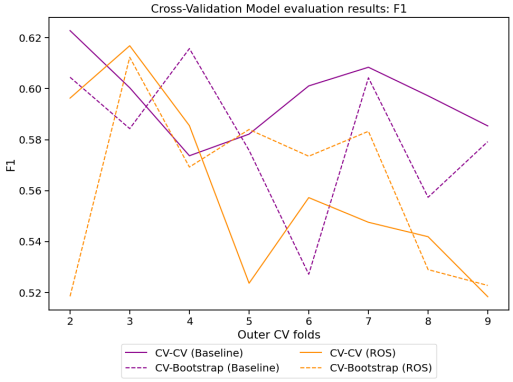

Figure 2. Cross-validation F1 results for different fold settings

In addition, variability differed across settings, with some exhibiting greater dispersion in performance metrics than others.

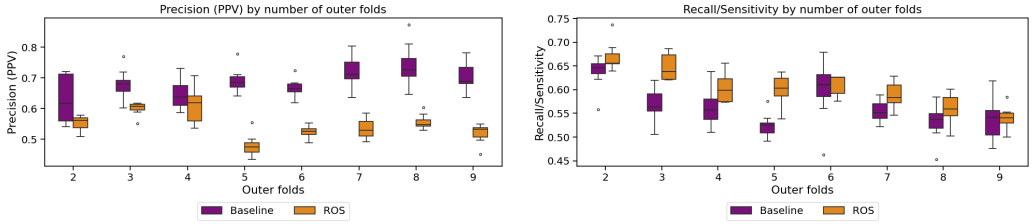

Figure 3. Distribution of precision and recall for varying outer folds configurations in cross-validation experiments.

The performance of models trained using bootstrap sampling was plotted across iterations. Figure 4 shows the precision and recall metrics of the models for the first 10 iterations. The plots for the other metrics recorded in this study are given in Appendix B.2.

For bootstrap sampling, metrics vary greatly across iterations, reflecting the sensitivity of models to the specific composition of the sampled training set and out-of-bag test set.

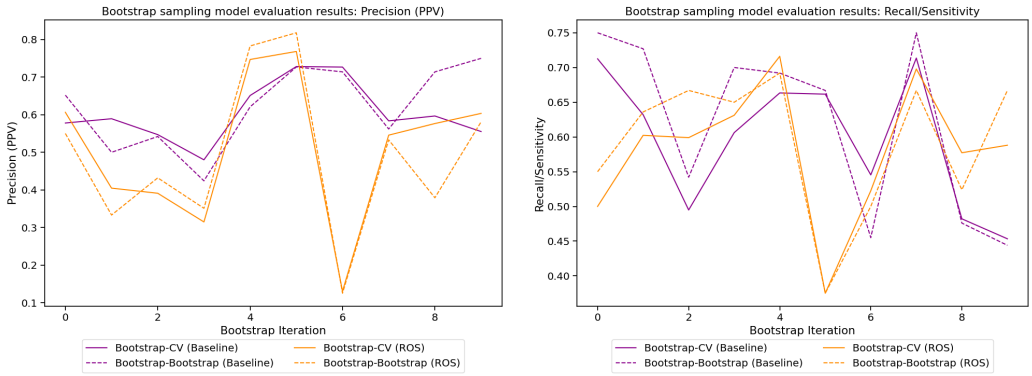

Figure 4. Bootstrap results over different iterations

5.3. **Impact of hyper-parameter optimization method**

For model evaluation with cross-validation, the choice of hyper-parameter optimization method used shows differences in performance metrics. While both optimization strategies produced broadly comparable performance under baseline and oversampled conditions, their relative performance varied across fold settings. For model evaluation with bootstrap sampling, the relative performance of hyper-parameter optimization methods also varied across iterations, although this variability was less pronounced than that observed for cross-validation.

The frequency of selected hyper-parameter values were also plotted for each experimental configuration. Figures 5 and 6 show the distribution of values selected for min_samples_leaf by both hyper-parameter optimization strategies, for cross-validation and bootstrap sampling respectively. Similar plots for other hyper-parameters are found in Appendix B.3.

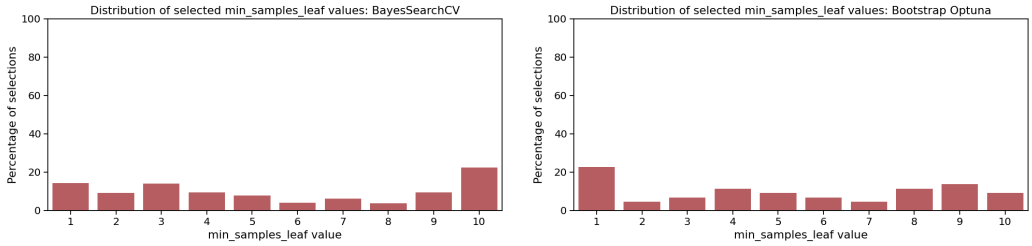

Figure 5. Frequency of selected values for min_samples_leaf for hyper-parameter optimization strategies within cross-validation

Within the same model evaluation strategy, the hyper-parameter values were selected at comparable frequencies by both hyper-parameter optimization methods. A greater difference in frequencies was observed between different model evaluation strategies. This pattern is consistent with differences in the composition of the training data induced by the evaluation strategy.

5.4. **Experimental configuration robustness and effects oversampling**

For these experiments, SMOTE was applied to the first experimental configuration (CV-CV) only. The average F1 of CV-CV trained models for different fold settings is shown

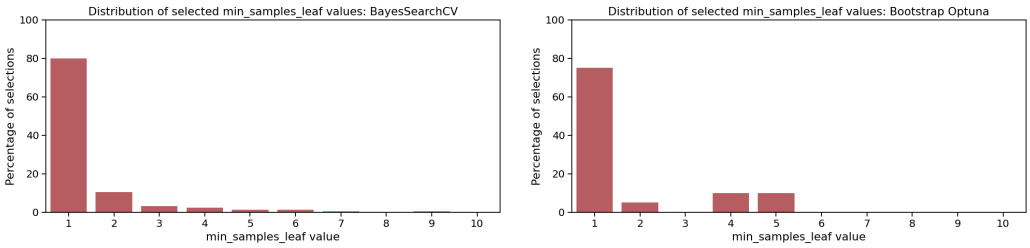

Figure 6. Frequency of selected values for min_samples_leaf for hyper-parameter optimization strategies within bootstrap sampling

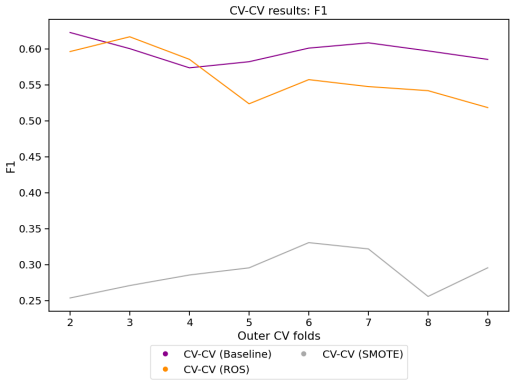

Figure 7. CV-CV F1 results for different fold settings

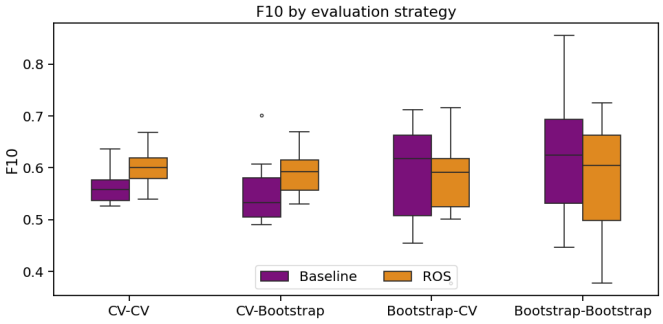

Figure 8. Distribution of F10 for different experimental configurations

in Figure 7. SMOTE was found to exhibit poor performance in very imbalanced datasets. This resampling technique was left out of subsequent experiments due to this conclusion and computational constraints. Similar plots for other metrics are found in Appendix B.4.

Each of the experimental configurations was also plotted against each other to assess variability and robustness, both for baseline and oversampled conditions. This is shown for the F10 metric in Figure 8. For cross-validation, variability is assessed over fold settings, and for bootstrap sampling, it is assessed over different iterations. Figures for the other evaluation metrics used in this study are shown in Figure 9 and Appendix B.5.

It was demonstrated that model evaluation using cross-validation produced more robust and consistent models overall. The performance metrics are comparable on average, but bootstrap sampling exhibits much more variability across different iterations, due to its sensitivity to specific sampled training sets and out-of-bag test sets.

It was also determined that the impact of random oversampling varied across experimental configurations. Figure 9 shows the impact of random oversampling on the precision and recall metrics for each experimental configuration.

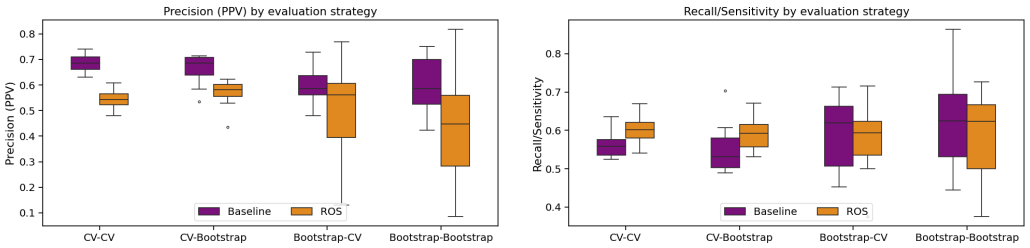

Figure 9. Distribution of precision and recall for different experimental configurations

For both model evaluation strategies, applying random oversampling reduced precision. For cross-validation, random oversampling improved recall by increasing minority class representation in structured training splits. In contrast, for bootstrap sampling, applying random oversampling amplified redundancy already introduced by sampling with replacement, leading to overfitting and reduced recall on out-of-bag test sets.

5.5. **Practical implications**

For highly imbalanced datasets, the choice of evaluation strategy and hyper-parameter optimization method warrants careful consideration, as performance metrics were sensitive to the evaluation protocols used. Under cross-validation, different numbers of outer folds influenced both performance variability and the effects of random oversampling. Generally, higher numbers of folds were associated with increased precision and decreased recall. Higher numbers of folds were also associated with decreased F10 and geometric mean. A moderate number of folds (4-5) was found optimal, by providing a moderate increase in precision without sacrificing too much recall.

Evaluation strategies and hyper-parameter optimization methods also interact differently with resampling methods. In particular, random oversampling, when combined with model evaluation using bootstrap sampling exhibited greater variability across iterations, and poorer performance for all metrics when compared with baseline models. When combined with cross-validation, random oversampling produced more stable estimates and increased performance for recall, F10 and geometric mean.

The trade-off observed between precision/recall changes associated to the number of folds used can be considered when training models with cross-validation on severely imbalanced datasets, along with the differences in variability and effects of oversampling to determine the optimal number of folds.

6. **Conclusion**

This paper examined the impact of evaluation protocol design in the context of a real-world case study characterized by extreme class imbalance. While cross-validation schemes, resampling strategies, and hyper-parameter tuning procedures are routinely adopted in experimental studies, their specific configurations are often chosen by convention rather than

through principled analysis. Evidence from this case study demonstrates that such choices are critical. We showed that, for this dataset, protocol configurations can substantially influence performance estimates, variability, and robustness. In particular, certain combinations of validation design and resampling strategies led to markedly different model rankings, occasionally yielding conflicting conclusions regarding the best-performing approach. Moreover, hyper-parameter tuning strategies cannot be analyzed independently of the validation framework in which they are embedded. Although the conclusions of this case study are derived from a specific dataset, they may generalize to other severely imbalanced datasets. This warrants further investigation as part of future work.

Overall, our findings underscore the need for greater transparency and methodological rigor in the selection and reporting of evaluation protocols, especially in extremely imbalanced settings. For our future work, we plan to extend this analysis to additional extremely imbalanced domains and explore new methodologies for selecting evaluation configurations tailored to highly skewed data distributions.

References

[1] R. Kohavi et al. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai*. Vol. 14. 2. Montreal, Canada. 1995, pp. 1137–1145.

[2] S. Das, S. P. Nayak, B. Sahoo, and S. C. Nayak. "Evaluating ensemble models on imbalanced data sets: a comparative study across varied minority class ratios". In: *2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)*. IEEE. 2024, pp. 774–779.

[3] G. Forman and M. Scholz. "Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement". In: *Acm Sigkdd Explorations Newsletter* 12.1 (2010), pp. 49–57.

[4] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos. "Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier]". In: *ieee ComputatioNal iNtelligeNCe magaziNe* 13.4 (2018), pp. 59–76.

[5] S. Shekhar, A. Bansode, and A. Salim. "A Comparative study of Hyper-Parameter Optimization Tools". In: *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. 2021, pp. 1–6. DOI: 10.1109/CSDE53843.2021.9718485.

[6] J. L. Merritt, E. MacLeod, A. Jurecka, and B. Hainline. "Clinical manifestations and management of fatty acid oxidation disorders". In: *Reviews in Endocrine and Metabolic Disorders* 21.4 (2020), pp. 479–493. ISSN: 1573-2606. DOI: 10.1007/s11154-020-09568-3.

[7] J. L. Merritt, M. Norris, and S. Kanungo. "Fatty acid oxidation disorders". In: *Annals of Translational Medicine* 6.24 (2018), p. 473. DOI: 10.21037/atm.2018.10.57.

[8] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. *scikit-optimize/scikit-optimize*. 2021. DOI: 10.5281/zenodo.5565057.

[9] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, 2623–2631. ISBN: 9781450362016. DOI: 10.1145/3292500.3330701.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[11] W. McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[12] G. Lemaître, F. Nogueira, and C. K. Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.html.

[13] T. Abedin, H. Xu, and S. Uddin. "The impact of K selection in K-fold cross-validation on bias and variance in supervised learning models". In: *Scientific Reports* 16 (2026), p. 6084. DOI: 10.1038/s41598-026-37247-x.

Appendix A. Dataset Details

Table 8. Dataset characteristics and summary statistics

	Data type	Mean	Std	Min	Median	Max
gestational_age	int	272.760	12.857	141	274	344
birth_weight	int	3298.126	575.051	0	3328	34250
age_at_collection	int	27.441	9.646	24.020	24.700	167.980
transfusion_status	int $\in \{0, 1\}$	0.003	0.057	0	0	1
sex_female	int $\in \{0, 1\}$	0.487	0.500	0	0	1
sex_male	int $\in \{0, 1\}$	0.512	0.500	0	1	1
ALA	int	219.971	64.499	42	210	3986
ARG	int	13.853	10.012	0	12	3672
CIT	int	16.151	5.237	1	16	1206
GLY	int	482.322	110.767	127	469	17534
LEU	int	104.763	25.538	26	102	5559
MET	int	26.868	7.062	3	26	663
ORN	int	79.863	29.126	0	74	1141
PHE	int	61.221	12.617	14	60	1591
SUAC	float	0.760	0.425	0.000	0.780	50.360
TYR	int	86.617	31.772	5	81	788
VAL	int	119.552	29.701	36	115	3921
C0	float	34.919	13.001	3.500	32.500	210.200
C2	float	29.349	10.045	3.100	27.600	274.000
C3	float	2.155	0.925	0.040	1.970	22.380
C3DC	float	0.084	0.027	0.000	0.080	1.980
C4	float	0.276	0.163	0.000	0.240	69.190
C4DC	float	0.583	0.236	0.050	0.560	5.800
C4OH	float	0.493	0.221	0.000	0.450	4.040
C5	float	0.128	0.061	0.000	0.120	9.710
C5:1	float	0.022	0.011	0.000	0.020	0.630
C5OH	float	0.145	0.077	0.000	0.140	15.110
C5DC	float	0.091	0.032	0.000	0.090	7.830
C6	float	0.073	0.033	0.000	0.070	4.140
C6DC	float	0.102	0.042	0.000	0.100	2.930
C8	float	0.092	0.155	0.000	0.080	49.250
C8:1	float	0.341	0.171	0.000	0.310	2.900
C10	float	0.109	0.048	0.000	0.100	3.140
C10:1	float	0.105	0.047	0.000	0.100	3.700
C12	float	0.174	0.067	0.000	0.160	3.450
C12:1	float	0.084	0.047	0.000	0.080	0.680
C14	float	0.259	0.094	0.030	0.250	35.420
C14:1	float	0.163	0.071	0.000	0.150	9.270
C14:2	float	0.047	0.021	0.000	0.040	0.890
C14OH	float	0.045	0.020	0.000	0.040	0.370
C16	float	2.920	0.922	0.160	2.810	24.670
C16OH	float	0.036	0.017	0.000	0.030	1.130
C16:1OH	float	0.079	0.033	0.000	0.080	3.690
C18	float	0.839	0.276	0.060	0.800	17.410
C18:1	float	1.286	0.381	0.050	1.240	11.280
C18:2	float	0.150	0.077	0.000	0.130	5.870
C18OH	float	0.015	0.008	0.000	0.010	0.580
C18:1OH	float	0.022	0.010	0.000	0.020	0.720
BIOT	float	127.329	31.065	0.000	125.700	334.900
GALT	float	8.570	2.246	0.400	8.400	22.700
IRT	float	23.581	13.519	1.500	20.700	1508.100
TREC_QN	int	581.015	321.051	0	521	6612
TSH	float	5.052	4.987	0.000	4.600	568.900
17OHP	float	11.152	8.122	0.000	9.700	801.400
HGB_Pattern	int $\in \{0, 1\}$	0.974	0.158	0	1	1
A	float	15.513	5.851	0.200	14.700	86.200
F	float	58.671	5.450	1.200	59.100	78.700
F1	float	20.565	3.057	1.600	20.300	50.900
FAST	float	3.496	1.006	0.500	3.400	54.300
definitive_diagnosis	int $\in \{0, 1\}$	0.000	0.010	0	0	1

Appendix B. Supplementary results

B.1. Cross-validation evaluation strategy results

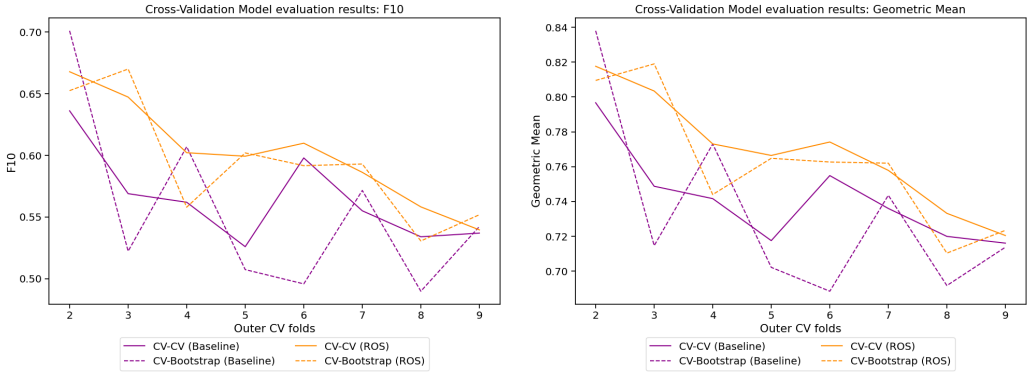

Figure 10. Cross-validation evaluation strategy supplementary results

B.2. Bootstrap sampling evaluation strategy results

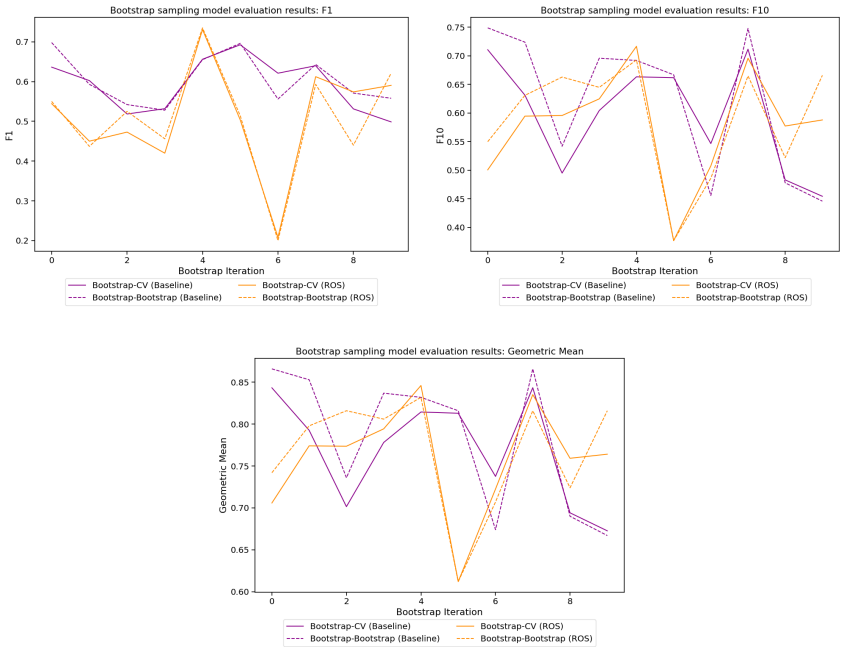

Figure 11. Bootstrap sampling evaluation strategy supplementary results

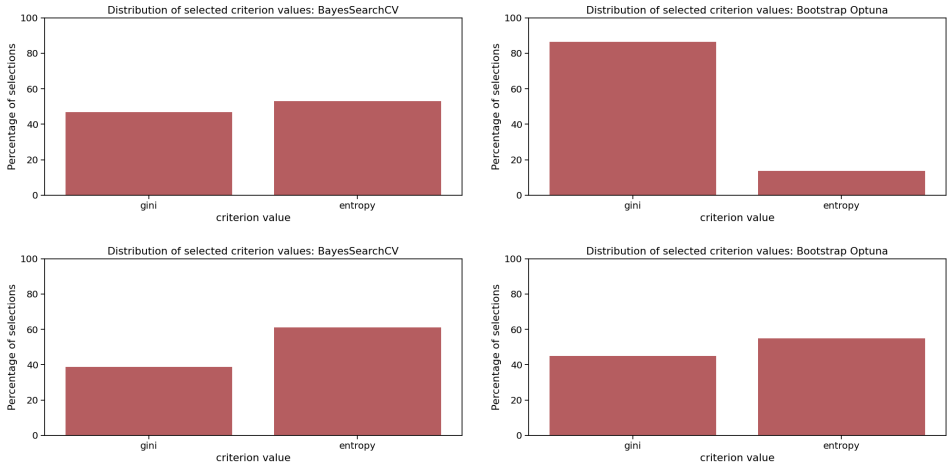

Figure 12. Frequency of selected values for criterion for hyper-parameter optimization strategies within cross-validation(top) and bootstrap sampling (bottom)

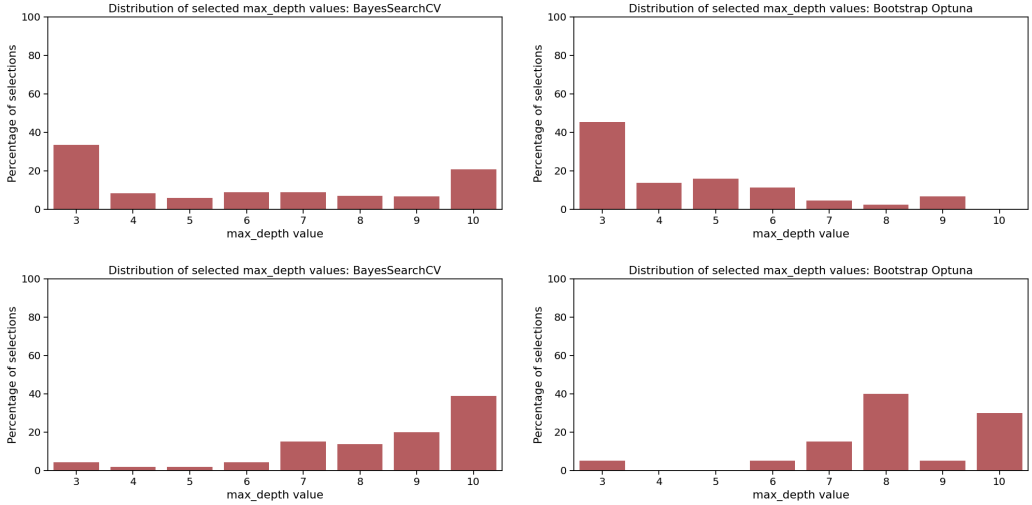

Figure 13. Frequency of selected values for max_depth for hyper-parameter optimization strategies within cross-validation (top) and bootstrap sampling (bottom)

B.3. **Hyper-parameter frequencies for evaluation strategies**

B.4. **CV-CV results of resampling**

B.5. **Experimental configuration result comparisons**

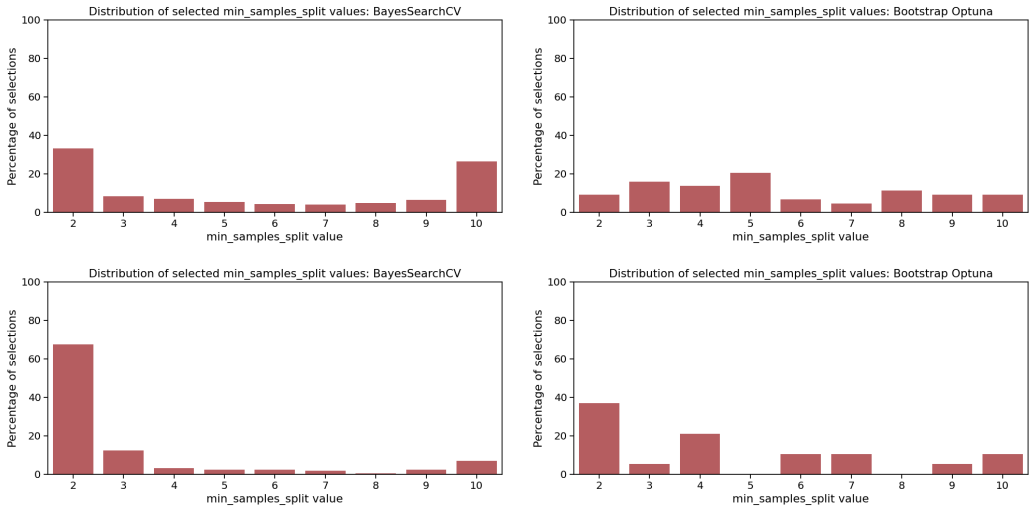

Figure 14. Frequency of selected values for min_samples_split for hyper-parameter optimization strategies within cross-validation (top) and bootstrap sampling (bottom)

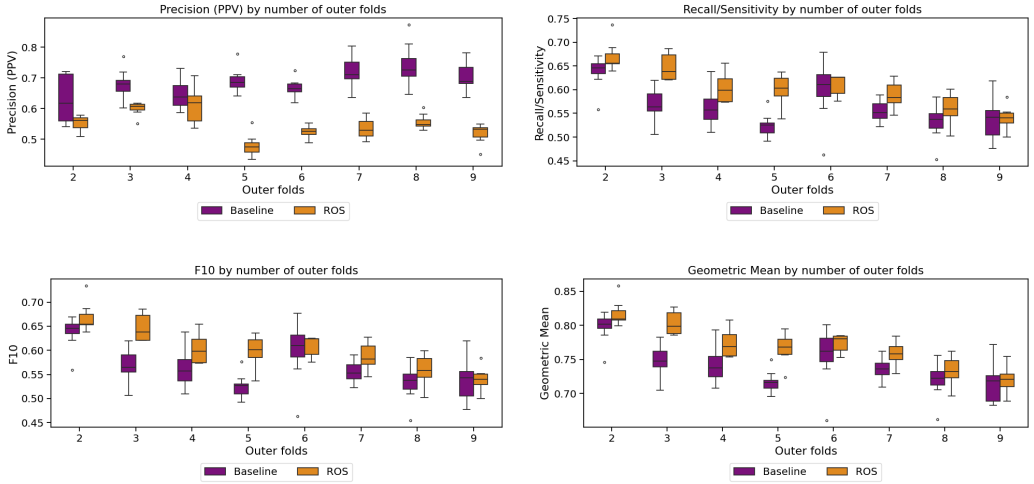

Figure 15. CV-CV with resampling supplementary results

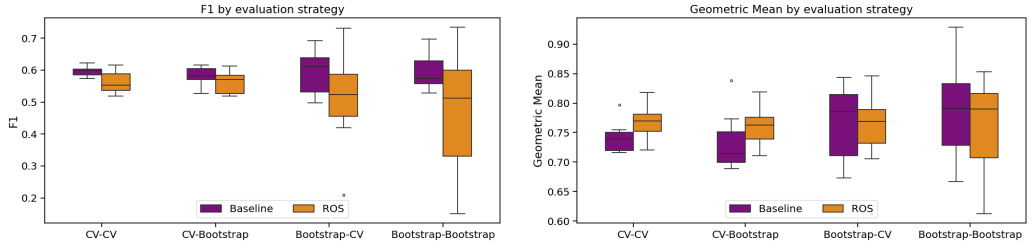

Figure 16. Experimental configuration supplementary result comparisons