2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-6283-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMLA55696.2022.00225

Cost-Sensitive Ensemble Learning for Highly Imbalanced Classification

Justin M. Johnson and Taghi M. Khoshgoftaar
College of Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida 33431
jjohn273@fau.edu, khoshgof@fau.edu

***Abstract*—There are a variety of data-level and algorithm-level methods available for treating class imbalance. Data-level methods include data sampling strategies that pre-process training data to reduce levels of class imbalance. Algorithm-level methods modify the learning and inference processes to reduce bias towards the majority class. This study evaluates both data-level and algorithm-level methods for class imbalance using a highly imbalanced healthcare fraud data set. We approach the problem from a cost-sensitive learning perspective, and demonstrate how these direct and indirect cost-sensitive methods can be implemented using a common cost matrix. For each method, a wide range of costs are evaluated using three popular ensemble learning algorithms. Initial results show that random undersampling (RUS) and class weighting are both effective ways to improve classification when the default classification threshold is used. Further analysis using the area under the precision-recall curve, however, shows that both RUS and class weighting actually decrease the discriminative power of these learners. Through multiple complementary performance metrics and confidence interval analysis, we find that the best model performance is consistently obtained when RUS and class weighting are not applied, but when output thresholding is used to maximize the confusion matrix instead. Our contributions include various recommendations related to implementing cost-sensitive ensemble learning and effective model evaluation, as well as empirical evidence that contradicts popular beliefs about learning from imbalanced data.**

***Keywords*—Class Imbalance, Big Data, Ensemble Learning, Cost-Sensitive Learning, Data Sampling, Thresholding, Fraud Detection**

1. Introduction

The class imbalance problem arises when the total number of samples from one category, or class, is significantly greater than the other classes within the data set. In most cases, the positive class (class of interest) is the minority class, and there is an abundance of negative samples comprising the majority group [1], [2], [3]. This phenomenon causes predictive models to become biased towards the majority class and generally yields high minority class error rates. These challenges are often compounded by the presence of big data [4]. For example, it is not uncommon to encounter highly imbalanced big data sets with millions of negative samples and positive class sizes that comprise $\leq 1\%$ of all data [5], [6], [7]. Fortunately, there are a variety of data-level and algorithm-level techniques that have been introduced over the years to address the class imbalance problem.

Algorithm-level techniques for treating class imbalance modify the underlying optimization and inference processes to reduce bias towards the majority class. In this study, we evaluate the effect of class weighting and output thresholding on a highly imbalanced healthcare fraud detection problem. Class weighting is a direct algorithm-level method, because it directly modifies the underlying learning objective [8]. Output thresholding, on the other hand, is an indirect method that wraps existing learners with a strategy for addressing class imbalance during the post-processing stage. The class weighting method is a popular technique because it is built into many popular machine learning algorithms, and its effect is similar to that of random oversampling without the increased cost of duplicating data. Thresholding reduces the bias towards the majority class by tuning the classification decision threshold that is used to assign class labels to a model's probability estimates [9]. Hence, thresholding is very flexible, as it does not require modifying learners and it can be applied to any learner that produces probability estimates. We describe both of these methods in more detail in Section 3.2.

Data-level techniques for treating class imbalance include data sampling strategies that pre-process training data to reduce levels of class imbalance [10]. Hence, data-level techniques are also referred to as indirect techniques, since they do not require modifications to underlying learners. We employ random undersampling (RUS) in this study to reduce the level of class imbalance in the training data. We do not consider random oversampling (ROS) because our data set is very large already. Furthermore, the class weighting mechanism has been shown to have the same effect as ROS [11].

Cost-sensitive learning is a machine learning paradigm that assigns misclassification costs to each class and optimizes learners to minimize the total cost [12]. This is contrary to the typical model optimization processes that minimize an objective function where all classes have equal

978-1-6654-6283-9/22/$31.00 ©2022 IEEE
DOI 10.1109/ICMLA55696.2022.00225

TABLE 1. COST MATRIX

	Actual Negative	Actual Positive
Predict Negative	TN - $C(0,0)$	FN - $C(0,1)$
Predict Positive	FP - $C(1,0)$	TP - $C(1,1)$

weight. Table 1 defines a cost matrix for the binary classification problem, where $C(i,j)$ is the cost associated with assigning class label i to a sample that has a ground truth class j. The cost of a correct classification, i.e. when $i = j$, is typically zero, while the misclassification costs can be defined by a domain expert or optimized on a validation set. A common approach to cost-sensitive learning with imbalanced data is to fix the cost of false positive errors $C(1,0) = 1$ and increase the cost of false negative errors $C(0,1) > 1$. Many related works and open-source libraries recommend setting the false negative cost to $C(0,1) = N_{neg}/N_{pos}$, as this effectively balances the cumulative cost of all samples [13], [14]. We evaluate 10 different false negative costs in this study using the three class imbalance techniques: RUS, class weighting, and output thresholding. In Section 3, we demonstrate exactly how each of these class imbalance techniques are cast to the common cost matrix.

Cost-sensitive learning methods are evaluated on a highly imbalanced healthcare fraud data set using the XGBoost (XGB) [15], CatBoost [16], and Random Forest (RF) learners. Six iterations of five-fold cross-validation are used to measure the performance of each learner, cost-sensitive implementation, and cost matrix. Preliminary results show that RUS and class weighting are both effective ways to improve classification when the default classification threshold of 0.5 is used. In our analysis, we find that the RUS and class weighting methods only see an increase in performance because these methods shift each learner's average probability estimate closer to the default threshold of 0.5 When we consider a threshold-agnostic performance metric, i.e. the area under the precision-recall curve (AUPRC), we find that both RUS and class weighting actually decrease the discriminative power of all learners. An analysis of the true positive rate (TPR), true negative rate (TNR), and the geometric mean of TPR and TNR (G-Mean) results across a range of positive class costs show that threshold tuning alone is the preferred method for maximizing the fraud classification performance. Based on these results, we can conclude that model evaluation and hyperparameter tuning should begin with a threshold-agnostic performance metric like AUPRC. Once the optimal model, hyperparameters, sampling rates, etc., have been selected based on AUPRC, we can then tune the threshold to optimize the confusion matrix based on requirements. Our contributions include demonstrating how cost-sensitive learning can be implemented with a common cost matrix using RUS, class weighting, and output thresholding, and comparing these methods with a range of false negative costs and three popular enseble learners.

The remainder of the paper is outlined as follows. Section 2 discusses related works in the area of cost-sensitive learning. Section 3 describes the data sets used in this study, defines the cost-sensitive learning implementations used in this study, and outlines the experiment design. Finally, Section 4 presents the results of the experiment and Section 5 concludes with areas for future work.

2. Related Works

A number of related works have evaluated cost-sensitive ensemble learning with the direct class weighting method. Sandica and Fratila [17] compare four class weighted ensemble learners, including three boosting algorithms and the RF learner, on a credit risk classification data set. The data is not highly imbalanced, but the authors demonstrate how the cost matrix can be varied and associated with domain knowledge to meet business requirements, e.g. a risk averse model versus a risk tolerant model. Zhang et al. [18] evaluate the class weighted RF learner using class weighting on an imbalanced power system stability classification task. The class weighted RF learner is shown to significantly outperform the baseline RF learner and the support vector machine. Ghatasheh et al. [19] evaluate cost-sensitive AdaBoost and RF learners on a bankruptcy prediction problem. The authors vary the false negative cost, compare G-Mean performance and error rates across learners, and find that the RF learner performs best overall. In general, these studies all show that cost-sensitive learning is an effective way to improve the classification performance of ensemble learners. They do not, however, evaluate these methods on highly imbalanced data with positive class sizes $< 1\%$, and they do not compare class weighting to thresholding and data sampling.

Despite many algorithms having built-in methods for class weighting, several authors have proposed new techniques and open-source packages. Devi et al. [20] propose a cost-sensitive RF learner that uses a weighted voting strategy instead of the standard majority vote. Experiment results show that the proposed cost-sensitive RF learner outperforms the baseline RF on two moderately imbalanced credit card fraud data sets. Zhu et al. [21] propose a similar class-weighted voting strategy for the RF learner and evaluate it on five medical diagnosis data sets. In both cases, the weights of each tree in the ensemble are determined by a combination of the class-wise error rates. Wang et al. [22] provide an open-source Python package *imbalance-XGBoost* for cost-sensitive learning using either a class-weighted cross-entropy loss or the focal loss. They demonstrate how the weighted loss functions outperform a baseline XGB learner using five imbalanced data sets. Mienye and Sun [23] present cost-sensitive implementations for the logistic regression, decision tree, RF, and XGB learners. The authors demonstrate how the cost-sensitive implementations outperform their default baselines, and results from other publications, using four popular medical diagnosis data sets. Wu et al. [24] explore an imbalanced malicious code classification problem using a multi-stage system of XGB learners. Three layers of XGB learners are applied sequentially to detect the positive class, under the assumption that the level of imbalance will improve through each stage. These methods are shown to outperform baseline

methods on relatively small data sets with moderate levels of imbalance, e.g. $<$ 2,000 samples per data set. In our study, we focus on how cost-sensitive ensemble learning performs in the context of highly imbalanced big data, with millions of data points and a positive class size of just 0.046%.

Few related works explore the effect of varying false negative costs on classification performance. Phankokkruad [25] evaluate the XGB learner with varying levels of positive class weight, but only present results from the optimal weight values. Nevertheless, the authors show that the cost-sensitive XGB learner outperforms the baseline on four breast cancer diagnosis data sets. Bauder and Khoshgoftaar [26] evaluated the RF learner on the Medicare fraud classification task with varying levels of RUS and found that sampling to a 90:10 class distribution performs best overall, but they do not take thresholding into consideration. Johnson and Khoshgoftaar [27] vary the levels of RUS and ROS on a similar Medicare fraud classification task, and find that a 99:1 class distribution performs best. He et al. [28] compare the cost-sensitive XGB learner to the synthetic minority oversampling technique (SMOTE) on a malicious URL classification problem. From this, the authors construct 18 data sets with different levels of class imbalance and then vary the false negative cost to determine its effect on G-Mean performance. As the cost of false negatives increases, there is an upward trend to G-Mean results until it reaches its optimal value, followed by a downward trend caused from over weighting the positive class. We believe this demonstrates the value of exploring a range of false negative costs. We expand on this related work by casting data sampling, class weighting, and output thresholding to the cost-sensitive cost matrix and evaluating a range of costs across multiple ensemble learners.

3. Methodology

This section begins by describing the publicly available Medicare fraud data sets that we use to evaluate cost-sensitive learning methods. Next, we discuss the implementation details of each cost-sensitive method as it relates to the common cost matrix. Finally, we outline the experiment design and performance metrics used to evaluate the methods.

3.1. Data Preparation

This study uses two of the latest available Medicare claims data sets from the CMS: the *Part B Summary by Provider* (SbP) [29] and *Part B Summary by Provider and Service* (SbPS) [30] data sets for years 2013–2019. This data set was made publicly available in 2021, and to the best of our knowledge we are the first research group to explore this data set for fraud classification. The Part B SbP data sets include provider-level, claims-level, and beneficiary-level statistics for providers over a given year. Examples of provider-level features include the provider's gender and specialty type, e.g. Cardiology, Gastroenterology, and Radiology. Examples of claims-level features include annual totals for the number of procedure codes billed, beneficiaries serviced, services provided, charges submitted to Medicare, and amounts paid by Medicare. Beneficiary-level features include the total number of beneficiaries serviced per age group, per gender, and per Medicare and Medicaid coverage, and another 16 beneficiary features that describe the percentage of patients with a particular chronic condition (CC). The Part B SbPS data set is aggregated by the CMS on: 1) NPI, 2) Healthcare Common Procedure Coding System (HCPCS) code, and 3) the place of service. As such, the remaining claims-level attributes summarize the provider's billing activity relative to a specific HCPCS code and place of service. Some examples of claims-level features include the total number of times a specific service was provided, the total number of beneficiaries receiving a given service, and the total amount paid by Medicare for a given service. We process the Part B SbPS data set following the steps outlined by Herland et al [31]. Finally, we join these data sets on the provider's national identifier (NPI), and attach real-world fraud labels from the publicly available List of Excluded Individuals and Entities (LEIE) [32] to create a feature-rich data set for supervised learning.

TABLE 2. MEDICARE PART B DATA SET SUMMARY

Features	Categorical Features	Rows	Pos. Count	Pos. Ratio
80	3	8,669,497	3,954	0.0456%

We join the Part B SbP and SbPS data sets on the provider NPI and year attributes, then apply data preprocessing steps to prepare the data for supervised learning. We exclude various provider-level features that are not related to general claims fraud, e.g. first and last name, address, and other geographic details. We also exclude beneficiary, race and suppression indicator features because they are missing values for more than 50% of the data set. Missing gender attributes are imputed with a third category Unknown (U), and the remaining numeric columns with missing values are imputed with 0. LEIE data sets are downloaded for the years 2013–2019 using an internet archive tool and concatenated to obtain a single, cumulative list of providers that have been excluded from participating in Medicare over the years. We select providers from the LEIE whose exclusion type is indicative of healthcare fraud and compute their last exclusion year to be the year of their exclusion plus the minimum exclusion period for the given offense. For the providers within the Medicare data sets whose NPI number matches those of the LEIE data set, claims that are dated prior to the provider's last exclusion year are labeled as fraudulent. A summary of the resulting data set is provided in Table 2. Note that this is a highly imbalanced data set with a positive class size of just 0.0456% of the entire data set.

3.2. Cost-Sensitive Learning

Cost-sensitive learning is implemented using three different techniques that have been defined in the literature:

RUS, output thresholding, and class weighting. Since we are primarily concerned with detecting fraudulent providers, and to simplify calculations, we fix the false positive misclassification cost to $C(1,0) = 1$ and only vary the false negative misclassification cost $C(0,1)$. A cost matrix with $C(0,1) = C(1,0) = 1$ is equivalent to no treatment, i.e. no cost-sensitive learning is applied. Given a dataset with N_{neg} negative samples and N_{pos} positive samples, we can balance the cumalative cost of each class by setting the false negative cost to be $C(0,1) = N_{neg}/N_{pos}$.

The RUS and output thresholding methods are indirect cost-sensitive techniques that address class imbalance through pre-processing and post-processing, respectively The RUS method modifies the distribution of training folds by taking a random sample without replacement from the majority class and combining it with the minority class. The size of the random sample $N_{neg}^{'}$ is defined by Eq. 1.

$$N_{neg}^{'} = N_{neg}\frac{C(1,0)}{C(0,1)} \tag{1}$$

Output thresholding addresses class imbalance by tuning the output decision threshold λ according to Eq. 2. When $C(1,0) = C(0,1) = 1$ we obtain the default threshold $\lambda = 0.5$. As the false negative cost $C(0,1)$ increases, the decision threshold approaches 0. For example, a cost of $C(0,1) = 2,192$ produces a threshold that is equal to the prior probability of the positive class, i.e. 0.00046, which has been recommended by related works [33], [34].

$$\lambda = \frac{C(1,0)}{C(1,0) + C(0,1)} = \frac{1}{1 + C(0,1)} \tag{2}$$

Table 3 demonstrates the effect of the cost matrix on both RUS and thresholding using a subset of values. Note that the negative class sample sizes are calculated using the approximate size of the negative class in the training set, i.e. 7 million negative samples.

TABLE 3. EFFECT OF COST $C(0,1)$ ON RUS AND THRESHOLD

Method	Factor	$C(0,1)$				
		1	50	500	1000	**2192**
RUS	$N_{neg}^{'}$	7M	140K	14K	7K	3.2K
Thresholding	λ	0.5	2e-2	2e-3	1e-3	4.6e-4

Finally, the class weighting technique is a direct cost-sensitive method that modifies the underlying learning process. This technique is especially popular because it is built into many popular machine learning algorithms, e.g. all three learners used in this study. For the RF learner, the class weighting parameter directly influences the tree construction process by modifying the tree splitting criterion. The Gini impurity measure of a tree node c is $i_c = 1 - \sum_i f_i^2$, where f_i is the fraction of samples in node c that belong to class i. In the cost-sensitive scenario, each f_i is updated by multiplying the number of samples in each class n_i by the class weight w_i. The XGB and CatBoost learners are popular implementations of the gradient boosted trees (GBT) algorithm that have proven to be effective for Medicare fraud detection [35]. These GBT learners use the class weight parameter to influence the loss calculation by multiplying the gradient components by the respective class weights. This in turn influences the residuals that are used to construct subsequent trees in the additive learning process. The official documentation for these learners recommends setting the positive class weight equal to N_{neg}/N_{pos} [13], [14]. For all three learners, increasing the weight of the positive class has the same effect as random oversampling, i.e. duplicating samples from the minority class.

3.3. Performance Evaluation

Cost-sensitive ensemble learning methods are evaluated on the Medicare Part B fraud classification problem using the XGB, CatBoost, and RF learners. The RF algorithm is trained using the Scikit-Learn package [36], while the XGB and CatBoost algorithms are trained using their respective Python implementations [13], [14]. Hyperparameters are identified for each learner during preliminary experiments. A maximum depth of 4 is used for the XGB and CatBoost learners, and a maximum depth of 16 is used for the RF learner. All remaining hyperparameters are left as their default values. Categorical features are encoded using a sparse one-hot representation due to the relatively high dimensionality of the provider type variable. We do not consider advanced encoding techniques because this has been studied previously [37], [38], and it is outside the scope of cost-sensitive learning.

Six runs of five-fold cross-validation are performed to produce a total of 30 results for each learner and data set combination. We use an adjusted k-fold cross-validation technique, where we partition data to ensure that providers do not coexist in both the train and test folds. We find this necessary because learners can overfit to specific providers and memorize specific characteristics of providers, instead of learning a general fraud distribution.

We report performance using the AUPRC, TPR, TNR, and G-Mean metrics over 30 repetitions. The AUPRC metric is a threshold-agnostic metric that summarizes the trade off between the TPR and precision, and it is a popular metric for class-imbalanced problems [39]. We prefer the AUPRC metric over the area under the receiver operating characteristic curve (AUC) because the AUPRC has been shown to be more informative when comparing models with highly imbalanced data [40]. This is primarily because the AUPRC is more sensitive to false positives than the AUC. Finally, we use the TPR, TNR, and G-Mean metrics to illustrate the tradeoffs between class-wise performance as the cost matrix is varied.

4. Results

Figure 1 illustrates the G-Mean, TPR, and TNR performance for each cost-sensitive method, averaged across all learners. We use the default threshold of 0.5 for the RUS and class weighting methods, and we have excluded

Figure 1. The Effect of Cost on Classification Performance

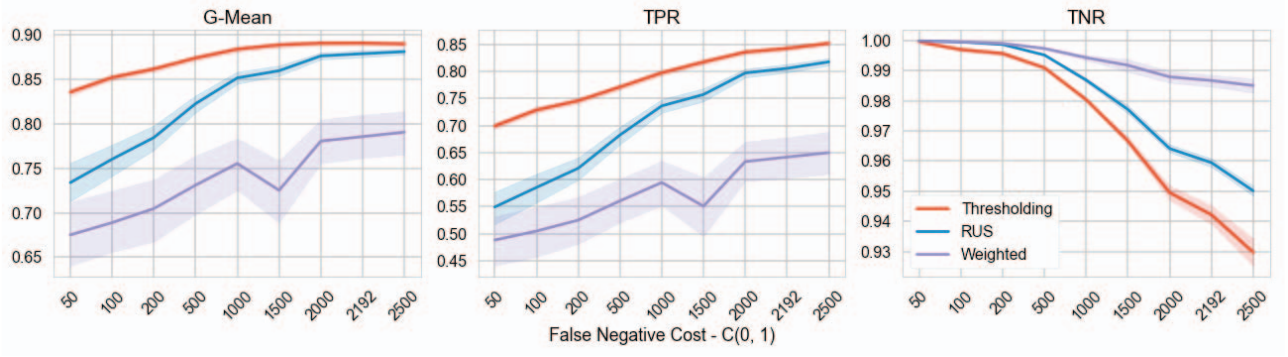
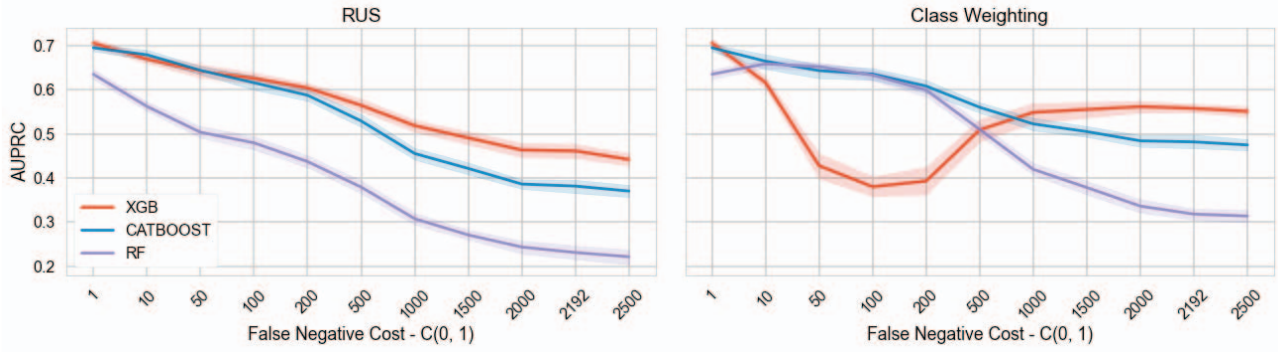


Figure 2. The Effect of RUS and Class Weight Costs on AUPRC



the baseline cost of $C(0,1) = 1$ to improve the quality of the visualization. As expected, G-Mean results show a general upward trend for all three methods as we increase the cost of the false negatives. This trend is also observed for the TPR metric, indicating that the increased cost of false negatives is allowing the model to better classify the minority class. For example, the TPR performance of the thresholding method increases from approximately 0.7 to 0.85 as the cost is increased from 50 to 2,192. As we increase the cost and the minority class performance, we also observe a decrease in the majority class performance. Continuing with the thresholding method, we see that the TNR decreases from approximately 1.0 down to 0.94 as we increase the cost to the recommended value of 2,192. Since the magnitude of the TPR increase is two times greater than that of the TNR decrease, we still observe a net positive G-Mean gain as the cost is increased. Overall, we can conclude from these results that all three methods improve the classification of our highly imbalanced data set.

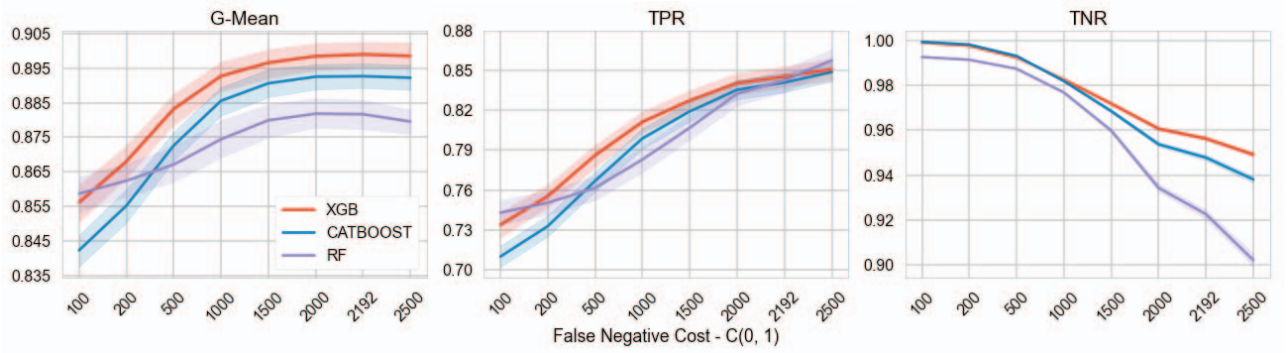
Table 4 provides the 95% confidence intervals (CI) for each cost-sensitive method's classification results, averaged across learners. The confidence intervals in bold font indicate the maximum non-overlapping C.I. for each performance metric. The non-overlapping confidence intervals show that thresholding provides statistically significant gains to TPR scores with an upper bound of 0.8510, compared to RUS and class weighting which have upper bounds of 0.8206 and 0.7083, respectively. On the other hand, thresholding also obtains the worst TNR score with a lower bound of 0.9369, while class weighting achieves the best TNR with a lower bound of 0.9831. We cannot consider the class weighting technique, however, as its TPR score is unacceptably low. In this fraud classification problem, our goal is to maximize the fraudulent provider recall and simultaneously minimize the false positive rate. Based on this criteria, we use the TPR and G-Mean metrics to conclude that thresholding and RUS perform best and second best, respectively.

TABLE 4. CLASSIFICATION PERFORMANCE FOR $C(0,1) = 2{,}192$

Method	G-Mean	TPR	TNR
RUS	(0.8718, 0.8861)	(0.7914, 0.8206)	(0.9570, 0.9620)
Thresholding	**(0.8864, 0.8958)**	**(0.8350, 0.8510)**	(0.9369, 0.9476)
Class Weighting	(0.7427, 0.8287)	(0.5754, 0.7083)	**(0.9831, 0.9903)**

Recall that the performance metrics presented thus far depend on the decision threshold, and for RUS and class weighting we have used a default threshold of 0.5. To better interpret model performance, we encourage using a threshold-agnostic performance metric like the AUC or AUPRC. We compare RUS and class weighting using the AUPRC metric in Figure 2 because it is more sensitive to false positives than the AUC, which is an important consideration given the large volume of negative samples in the Medicare data. We do not include thresholding AUPRC

Figure 3. The Effect of Cost-Senstive Thresholding



results because the AUPRC does not depend on a single threshold.

Surprisingly, we observe that increasing the false negative cost causes a decrease to AUPRC performance for both RUS and class weighting. We previously observed improvements to classification performance because the RUS and class weighting techniques have shifted the learner's probability estimates closer to the default threshold of 0.5, but the learner's discriminative power has actually declined. These results suggest that we should not apply RUS or class weighting. Instead, we can use thresholding to tune the operating point on the AUPRC curve and maximize the confusion matrix results.

We explore the effect that varying costs have on output thresholding in Figure 3. Again, we have excluded costs $C(0,1) < 100$ from these results to improve the quality of the visualization. Nevertheless, it is apparent that these low costs yield poor TPR scores and near perfect TNR scores that are indicative of a biased model overpredicting the majority class. The results show that a cost of $C(0,1) \geq 1000$ is required to obtain a minimum of 80% recall. Put another way, a decision threshold of $\lambda \leq 0.001$ is required to obtain at least 80% recall. We can further increase the cost to the recommended value of 2,192, or higher, to trade an increase in TPR for a decrease in TNR. The recommended cost of $C(0,1) = 2,192$ achieves a TPR of approximately 0.84 for all learners and satisfactory TNR scores between 0.92 and 0.96. When taking all performance metrics into consideration, we conclude that the best Medicare fraud classification performance is obtained using the GBT learners and output thresholding, and the XGB learner performs best overall.

Based on these results, we provide several recommendations for training predictive models on highly imbalanced data. Despite some related works and open-source tools recommending data sampling and class weighting, we have shown that these cost-sensitive techniques are not always helpful. Instead, we recommend using a threshold-agnostic performance metric to identify optimal values for data sampling, class weighting, and any additional hyperparameters. We use the AUPRC metric because it is more sensitive to false positives than AUC, and it has been shown to be more informative than the AUC metric when data is highly imbalanced [39]. Once these optimal parameters have been identified, the confusion matrix can be adjusted to meet requirements by tuning the output threshold that is used to assign class labels to model probability estimates. Eq. 2 provides a good starting point for threshold selection. The output threshold can be further tuned using a cost matrix, as presented in this study, or it can be optimized on a holdout partition as shown in related works [33].

5. Conclusion

This study evaluates direct and indirect methods for cost-sensitive ensemble learning with highly imbalanced Medicare fraud data. Class weighting is a popular direct cost-sensitive technique because it is built into many popular machine learning algorithms, and its effect is similar to that of random oversampling without the increased cost of duplicating data. RUS and output thresholding are indirect cost-sensitive methods that do not modify the underlying learning objective, but instead wrap the learner with preprocessing and post-processing strategies for combatting class imbalance. We show that all three cost-sensitive methods can be cast to a common cost matrix that represents the cost of false positives and false negatives.

Output thresholding, class weighting, and RUS are evaluated using three popular ensemble learning techniques, multiple complementary performance metrics, and 95% confidence intervals. Initial results show that all three methods are effective ways to improve classification performance. Despite these initial improvements, further analysis using the AUPRC metric shows that the RUS and class weighting techniques actually decrease the discriminative power of the learners. Instead, AUPRC results suggest that the best model performance is consistently obtained when the RUS and class weighting are not applied. An analysis of G-Mean, TPR, and TNR results across a range of positive class costs show that threshold tuning alone is the preferred method for maximizing Medicare fraud classification performance. The classification performance of the XGB and CatBoost learners is optimal when the output threshold is set to $\lambda = 0.00046$, when the positive class cost $C(0,1) = N_{neg}/N_{pos}$ = 2,192. In future works, we plan to evaluate these cost-

sensitive methods on additional data sets with varying levels of high-to-severe class imbalance.

References

[1] R. B. Rao, S. Krishnan, and R. S. Niculescu, "Data mining for improved cardiac care," *SIGKDD Explor. Newsl.*, vol. 8, no. 1, pp. 3–10, Jun. 2006. [Online]. Available: http://doi.acm.org/10.1145/1147234.1147236

[2] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, Jul 2013. [Online]. Available: https://doi.org/10.1007/s11280-012-0178-0

[3] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "Mining data with rare events: A case study," in *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, vol. 2, 2007, pp. 132–139.

[4] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *Journal of Big Data*, vol. 5, no. 1, p. 42, 2018. [Online]. Available: https://doi.org/10.1186/s40537-018-0151-6

[5] I. Triguero, S. del Río, V. López, J. Bacardit, J. M. Benítez, and F. Herrera, "Rosefw-rf: The winner algorithm for the ecbdl'14 big data competition: An extremely imbalanced big data bioinformatics problem," *Knowledge-Based Systems*, vol. 87, pp. 69–79, 2015, computational Intelligence Applications for Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705115002130

[6] A. Maurya, "Bayesian optimization for predicting rare internal failures in manufacturing processes," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 2036–2045.

[7] N. G. Marchant and B. I. P. Rubinstein, "In search of an entity resolution oasis: Optimal asymptotic sequential importance sampling," *Proc. VLDB Endow.*, vol. 10, no. 11, p. 1322–1333, aug 2017. [Online]. Available: https://doi.org/10.14778/3137628.3137642

[8] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: a review," *International Journal of Computing and Business Research (IJCBR)*, vol. 5, no. 4, pp. 1–29, 2014.

[9] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519310497

[10] S. M. Abd Elrahman and A. Abraham, "A review of class imbalance problem," *Journal of Network and Innovative Computing*, vol. 1, no. 2013, pp. 332–340, 2013.

[11] K. M. Ting, "Inducing cost-sensitive trees via instance weighting," in *Principles of Data Mining and Knowledge Discovery*, J. M. Żytkow and M. Quafafou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 139–147.

[12] C. Ling and V. Sheng, "Cost-sensitive learning and the class imbalance problem," *Encyclopedia of Machine Learning*, 01 2010.

[13] CatBoost. (2022) Training parameters. [Online]. Available: https://catboost.ai/en/docs/references/training-parameters/

[14] XGBoost. (2022) Xgboost parameters. [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html

[15] J. T. Hancock and T. M. Khoshgoftaar, "Gradient boosted decision tree algorithms for medicare fraud detection," *SN Computer Science*, vol. 2, no. 4, p. 268, 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00655-z

[16] J. T. Hancock and T. M. Khoshgoftaar, "Catboost for big data: an interdisciplinary review," *Journal of Big Data*, vol. 7, no. 1, p. 94, 2020. [Online]. Available: https://doi.org/10.1186/s40537-020-00369-8

[17] A.-M. Sandica and A. F. (Adam), "Implications of macroeconomic conditions on romanian portfolio credit risk. a cost-sensitive ensemble learning methods comparison," *Economic Research-Ekonomska Istraživanja*, vol. 0, no. 0, pp. 1–20, 2021. [Online]. Available: https://doi.org/10.1080/1331677X.2021.1997625

[18] C. Zhang, Y. Li, Z. Yu, and F. Tian, "A weighted random forest approach to improve predictive performance for power system transient stability assessment," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2016, pp. 1259–1263.

[19] N. Ghatasheh, H. Faris, R. Abukhurma, P. A. Castillo, N. Al-Madi, A. M. Mora, A. Al-Zoubi, and A. Hassanat, "Cost-sensitive ensemble methods for bankruptcy prediction in a highly imbalanced data distribution: a real case from the spanish market," *Progress in Artificial Intelligence*, vol. 9, no. 4, pp. 361–375, 2020. [Online]. Available: https://doi.org/10.1007/s13748-020-00219-x

[20] D. Devi, S. K. Biswas, and B. Purkayastha, "A cost-sensitive weighted random forest technique for credit card fraud detection," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.

[21] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, and G. Ning, "Class weights random forest algorithm for processing class imbalanced medical data," *IEEE Access*, vol. 6, pp. 4641–4652, 2018.

[22] C. Wang, C. Deng, and S. Wang, "Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost," *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865520302129

[23] I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data," *Informatics in Medicine Unlocked*, vol. 25, p. 100690, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235291482100174X

[24] D. Wu, P. Guo, and P. Wang, "Malware detection based on cascading xgboost and cost sensitive," in *2020 International Conference on Computer Communication and Network Security (CCNS)*, 2020, pp. 201–205.

[25] M. Phankokkruad, "Cost-sensitive extreme gradient boosting for imbalanced classification of breast cancer diagnosis," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2020, pp. 46–51.

[26] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using random forest with class imbalanced big data," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, 2018, pp. 80–87.

[27] J. M. Johnson and T. M. Khoshgoftaar, "Medicare fraud detection using neural networks," *Journal of Big Data*, vol. 6, no. 1, p. 63, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0225-0

[28] S. He, B. Li, H. Peng, J. Xin, and E. Zhang, "An effective cost-sensitive xgboost method for malicious urls detection in imbalanced dataset," *IEEE Access*, vol. 9, pp. 93 089–93 096, 2021.

[29] Centers For Medicare & Medicaid Services. (2022) Medicare physician & other practitioners - by provider. [Online]. Available: https://data.cms.gov/provider-summary-by-type-of-service/medicare-physician-other-practitioners/medicare-physician-other-practitioners-by-provider

[30] Centers For Medicare & Medicaid Services. (2022) Medicare physician & other practitioners - by provider and service. [Online]. Available: https://data.cms.gov/provider-summary-by-type-of-service/medicare-physician-other-practitioners/medicare-physician-other-practitioners-by-provider-and-service

[31] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big data fraud detection using multiple medicare data sources," *Journal of Big Data*, vol. 5, no. 1, p. 29, Sep 2018. [Online]. Available: https://doi.org/10.1186/s40537-018-0138-3

[32] U.S. Department of Health and Human Services Office of Inspector General. (2022) Leie downloadable databases. [Online]. Available: https://oig.hhs.gov/exclusions/exclusions_list.asp

[33] J. M. Johnson and T. M. Khoshgoftaar, *Thresholding Strategies for Deep Learning with Highly Imbalanced Big Data*. Singapore: Springer Singapore, 2021, pp. 199–227. [Online]. Available: https://doi.org/10.1007/978-981-15-6759-9_9

[34] J. M. Johnson and T. M. Khoshgoftaar, "Output thresholding for ensemble learners and imbalanced big data," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021, pp. 1449–1454.

[35] J. T. Hancock and T. M. Khoshgoftaar, "Medicare fraud detection using catboost," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, 2020, pp. 97–103.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] J. M. Johnson and T. M. Khoshgoftaar, "Medical provider embeddings for healthcare fraud detection," *SN Computer Science*, vol. 2, no. 4, p. 276, 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00656-y

[38] J. M. Johnson and T. M. Khoshgoftaar, "Encoding high-dimensional procedure codes for healthcare fraud detection," *SN Computer Science*, vol. 3, no. 5, p. 362, 2022. [Online]. Available: https://doi.org/10.1007/s42979-022-01252-4

[39] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Comput. Surv.*, vol. 49, no. 2, aug 2016. [Online]. Available: https://doi.org/10.1145/2907070

[40] 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.