Output Thresholding for Ensemble Learners and Imbalanced Big Data

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*—Class imbalance is a common problem in many real-world machine learning applications that has been shown to significantly degrade classification performance. This is especially true in the context of big data, where large volumes of data from the majority class dominate training processes and bias learning algorithms. Of the various methods for treating class imbalance, output thresholding is one technique that improves classification performance by tuning the decision threshold that is used to assign class labels to class probabilities. While thresholding techniques have been successful, systematic studies within big and imbalanced data applications are limited. In this study, we compare four popular thresholding strategies using two big and imbalanced fraud classification data sets. We focus specifically on tree-based ensemble learners, and employ four popular bagging and boosting ensemble learners that are well known for achieving state-of-the-art performance. Overall classification is measured using the Geometric Mean (G-Mean) and F-Measure metrics and class-wise performance tradeoffs are compared using the true positive rate (TPR) and true negative rate (TNR). The average threshold values of each strategy are compared, and statistical tests are provided to illustrate the importance of careful threshold tuning. Results show that the G-Mean and F-Measure metrics provide misleading results, and careful validation of TPR and TNR is necessary for selecting optimal thresholds. Furthermore, we show how small changes to decision thresholds yield significant changes to classification performance. Our comparison of popular thresholding techniques on both big and highly-imbalanced data makes this a unique contribution in the area of output thresholding with ensemble learners and big data.**

***Keywords*—Class Imbalance, Output Thresholding, Ensemble Learners, Big Data, Medicare, Fraud Detection**

1. Introduction

Class imbalance occurs when the total number of samples from one category, or class, is significantly greater than the other classes within the data set. In many problems [1], [2], [3], [4], the minority group is the class of interest, i.e., the positive class, and there is an abundance of less-interesting negative samples comprising the majority group. When data is highly-imbalanced and the positive class is $\leq 1\%$ of the data set, machine learning algorithms will usually become biased towards the majority group and have difficulty detecting the minority group [5].

Methods for treating class imbalance consist of data-level techniques, algorithm-level techniques, and hybrid approaches [6]. Data-level techniques attempt to reduce the level of imbalance through various data sampling methods. Examples of data-level techniques that have been shown to be successful include random oversampling the minority class (ROS), random undersampling the majority class (RUS), and intelligent sampling techniques that take class distributions into consideration when selecting data to sample [7], [8], [9], [10]. Algorithm-level methods for handling class imbalance, commonly implemented with a weight or cost schema, include modifying the underlying learner or its output in order to reduce bias towards the majority group. These include cost-sensitive methods [11], custom loss functions [12], [13], and output thresholding [14], [15].

The focus of this study is to utilize output thresholding to improve the binary classification of imbalanced and highly-imbalanced big data sets using ensemble learners. In the binary classification problem, there exists a data set $\mathcal{D} : \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$ where x_i and y_i correspond to the i^{th} feature vector and class label, respectively. A probabilistic classifier $\mathcal{M}$ is trained on $\mathcal{D}$ and used to estimate the posterior probability $\mathcal{P}(y_i = 1 \mid x_i)$. Predictions are then made by comparing the probability estimate to the decision threshold λ, i.e. $\hat{y}_i = 1$ if $\mathcal{P}(y_i = 1 \mid x_i) > \lambda$, else $\hat{y}_i = 0$. Most commonly, the default threshold of $\lambda = 0.5$ is used to assign class labels to test samples. Several studies have shown, however, that the default threshold is suboptimal, and that classification performance can be improved by using non-default thresholds [14], [15], [16]. Examples of thresholding techniques from related works include setting the threshold equal to the prior probability of the positive class and optimizing the threshold on the training set. Few studies have compared these thresholding techniques using big data or highly-imbalanced data, however, and none of these related works have systematically compared each of these thresholding techniques to each other.

We address this gap in the literature by comparing four thresholding strategies using four ensemble learners and two Medicare fraud classification data sets that contain millions

of negative class samples and positive class sizes as small as 0.8%. The prior threshold (λ_{prior}) sets the decision threshold equal to the prior probability of the positive class. The optimal-fmeasure threshold ($\lambda_{fmeasure}$) and optimal-gmean threshold (λ_{gmean}) strategies identify optimal thresholds by maximizing the F-Measure and Geometric Mean (G-Mean) on the training set, respectively. Thresholds are compared to the default threshold using the G-Mean, F-Measure, true positive rate (TPR), and true negative rate (TNR) using five-fold cross validation and four popular tree-based ensemble learners. The ensemble learners used in this study include the Random Forest (RF), Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LightGBM), and CatBoost learners. Finally, we average results across learners and data sets, and use Tukey's Honestly Significant Difference (HSD) test [17] to identify meaningful differences between each thresholding strategy.

Results show that all three non-default thresholds significantly outperform the default threshold of 0.5. G-Mean and F-Measure results show that all three non-default thresholds perform statistically the same on average. A closer inspection of the TPR and TNR performance, however, shows that small changes to the classification threshold yields significant changes to positive and negative class performance. Overall, we find that λ_{prior} consistently obtains the best TPR, but the $\lambda_{fmeasure}$ and λ_{gmean} metrics obtain more balanced TPR and TNR scores. To the best of our knowledge, this is the first study to systematically compare these thresholding techniques across a range of ensemble learners using imbalanced big data sets.

The remainder of the paper is structured as follows. Section 2 introduces related works in the areas of Medicare fraud detection, classification with imbalanced data, and output thresholding. Section 3 describes the Medicare data sets, experiment design, and performance evaluation used in this study. Section 4 presents the results of our experiments and discusses key findings. Finally, Section 5 concludes with a summary of our results and suggestions for future works.

2. Related Work

The Medicare data sets made publicly available by the CMS [18] are characterized by big data and high class imbalance. As such, they have been used in a number of studies that evaluate techniques for classifying imbalanced data. Bauder and Khoshgoftaar [19] use data sampling to explore the effects of class rarity by comparing Medicare fraud classification performance across a range of imbalance levels. In a related work [20], we combine RUS and ROS to balance classes and maximize Medicare fraud detection using deep neural network models. Despite these efforts to mitigate the effects of class imbalance, these studies have not compared the various output thresholding strategies that we present in this study.

Output thresholding has been used in a number of domains to improve the classification performance of imbalanced data. Buda et al. [16] applied the prior probability thresholding strategy to image classification problems using Convolutional Neural Networks. Results showed significant improvements to classification performance, especially when thresholding is combined with ROS. Zou et al. [14] identify optimal thresholds for imbalanced protein sequence classification with the Random Forest learner by maximizing the F-Measure on the training set, and then tuning the threshold using the minimum and maximum probability estimates from the test set. The authors show that the proposed threshold strategy outperforms the default threshold and other uniform thresholds, and propose a method for scaling the thresholding selection process to big data. Xingfu et al. [15] propose a thresholding strategy using the Random Forest classifier that optimizes the threshold on the test set by selecting the threshold that minimizes the difference between the positive class frequency in the training set and the positive class frequency in the test set predictions. They extend this approach to the multi-label classification problem and show that it outperforms the default threshold. The authors use the TPR, TNR, G-Mean, and F-Measure metrics to show that the optimal thresholds significantly outperform the default threshold, and demonstrate that the popular AUC metric alone is insufficient for selecting the best model. In a previous study [21], we compare λ_{gmean} and λ_{prior} using deep neural networks for Medicare fraud detection and find that both outperform the default threshold. Chad and Khoshgoftaar [22] explored optimal thresholds λ_{gmean} and $\lambda_{fmeasure}$ for maximizing classification performance with imbalanced network security data. Results showed that both optimal thresholds outperform the default threshold and that multiple performance metrics are required to properly interpret performance.

Each of the thresholding studies support using non-default decision thresholds for maximizing classification performance with imbalanced data. None of these studies, however, directly compare the default, optimal-gmean, optimal-fmeasure, and prior thresholding strategies directly. In our study, we uniquely compare these four thresholding strategies directly using two highly-imbalanced big data sets. We do not compare our results to the studies that utilize the test set for tuning, as this is generally discouraged and does not lend itself to real-world machine learning applications [23]. We focus specifically on tree-based ensemble learners, as these are some of the most popular machine learning algorithms for heterogeneous data and they are well known for obtaining state-of-the-art performance [24].

3. Methodology

3.1. Data Preprocessing

This study uses two publicly available Medicare data sets: (1) 2012–2018 Medicare Provider Utilization and Payment Data: Physician and Other Supplier (Part B), and (2) 2013–2018 Medicare Provider Utilization and Payment Data: Part D Prescriber (Part D). These data sets are released annually and they can be downloaded in a tab delimited format from the CMS website [18]. For each data set, we utilize all available years on the CMS website. Each

TABLE 1. SUMMARY OF DATA SETS

Data Set	Years	# of Negative	# of Positive	% Positive
Part B	2012–2018	4,959,806	40,194	0.8039%
Part D	2013–2018	4,850,647	149,353	2.9871%

```
input : targets y, probability estimates p
output: optimal threshold
best_thresh ← curr_thresh ← max_gmean ← 0;
delta_thresh ← 0.0001;
while curr_thresh < 1.0 do
    ŷ ← ApplyThreshold(p, curr_thresh);
    tpr, tnr, gmean ← CalcPerformance(y, ŷ);
    if tpr < tnr then
        return best_thresh;
    end
    if gmean > max_gmean then
        max_gmean ← gmean;
        best_thresh ← curr_thresh;
    end
    curr_thresh ← curr_thresh + delta_thresh;
end
return best_thresh;
```

Algorithm 1: Optimal Threshold Procedure

year of data summarizes the final utilization and payments for procedures, services, and prescription drugs provided to Medicare beneficiaries by medical providers and healthcare professionals for the given year. Healthcare professionals are identified by their National Provider Identifier (NPI), a unique 10-digit identification number for healthcare providers.

The Part B data sets used in this study cover years 2012–2018 and contain approximately 67 million records that describe provider billing activity with respect to specific procedures. The Medicare Part D data sets span years 2013–2018 and includes approximately 122 million samples that describe provider billing activity with respect to prescription drugs. To reduce training complexity and enable experiments on multiple learners, we use a sample of five million Medicare observations from each data set that includes all fraudulent samples and a random sample from the non-fraudulent class. A summary of the Part B and Part D data sets, including their levels of class imbalance, are provided in Table 1. We refer readers to related works [25], [26] for additional details on these data sets and joining real-world fraud labels from the List of Excluded Individuals and Entities (LEIE).

3.2. Thresholding Strategies

We compare three thresholding strategies to the default classification threshold $\lambda_{default} = 0.5$. The prior threshold strategy (λ_{prior}) is defined by the prior probability of the positive class, i.e. 0.0080 and 0.0299 for the Part B and Part D data sets, respectively. These thresholds are orders of magnitude smaller than the default threshold.

The next two thresholding strategies are optimized on the training data after the ensemble learner has been trained. The optimal-gmean threshold (λ_{gmean}) is the threshold that maximizes the G-Mean metric on the training set. Similarly, the optimal-fmeasure threshold ($\lambda_{fmeasure}$) is the threshold that maximizes the F-Measure metric on the training set. Optimizing the threshold by the G-Mean metric seeks to approximately balance the TPR and TNR, while the F-Measure metric aims to maximize the harmonic mean of precision and recall. We add an additional constraint to both of these optimization processes, and require that TPR $\geq$ TNR, as we are mostly concerned with detecting fraudulent providers. This optimal-gmean threshold selection process is summarized by Algorithm 1. The optimal-fmeasure threshold is calculated following the same procedure as the optimal-gmean threshold, except the F-Measure is maximized instead of the G-Mean.

3.3. Learners and Performance Evaluation

Thresholding strategies are evaluated using four popular ensemble learners: Random Forest (RF) [23], Extreme Gradient Boosting (XGB) [24], CatBoost [27], and Light Gradient Boosting Machine (LightGBM) [28]. The RF algorithm is a popular decision tree bagging ensemble learner, and the XGB, CatBoost, and LightGBM are all variants of the gradient boosting tree ensemble family that have proven effective at Medicare fraud detection [29]. The RF algorithm is implemented using the scikit-learn package [30], and the remaining algorithms are implemented using their respective Python packages. A maximum depth of eight is used for the RF and XGB learners, and all remaining parameters are left to their defaults. We found that these hyperparameters performed well during preliminary experiments and did not require further hyperparameter tuning.

The performance for each learner and thresholding strategy is reported using five-fold cross-validation. For each iteration of cross-validation, the learner is first trained on $k-1$ training partitions. Next, thresholds λ_{gmean}, $\lambda_{fmeasure}$, and λ_{prior} are identified on the training partitions. Finally, the thresholds identified on the training partitions are used to make predictions on the k^{th} test partition. Five-fold cross-validation is repeated six times for each learner and data set, producing 30 results for every data set, learner, and thresholding strategy. We report the G-Mean and F-Measure metrics to measure overall classification performance and we report the TPR and TNR metrics to compare class-wise performance.

We use Tukey's HSD test ($\alpha = 0.01$) to estimate the significance of the thresholding results. Tukey's HSD test is a multiple comparison procedure that determines which method means are statistically different from each other by identifying differences that are greater than the expected standard error [17]. Result sets are assigned to alphabetic

TABLE 2. PART B THRESHOLDING RESULTS

Learner	Threshold Strategy	Average Threshold	Average G-Mean	Average F-Measure	Average TPR	Average TNR
CatBoost	$\lambda_{default}$	0.5000	0.0659	0.0087	0.0044	**0.9999**
	$\lambda_{fmeasure}$	0.0090	**0.7459**	**0.0443**	0.7534	0.7385
	λ_{gmean}	0.0090	**0.7459**	0.0442	0.7548	0.7372
	λ_{prior}	0.0080	0.7426	0.0404	**0.7920**	0.6964
LightGBM	$\lambda_{default}$	0.5000	0.1583	0.0466	0.0251	**0.9996**
	$\lambda_{fmeasure}$	0.0091	**0.7564**	**0.0476**	0.7563	0.7565
	λ_{gmean}	0.0090	**0.7564**	0.0475	0.7576	0.7554
	λ_{prior}	0.0080	0.7552	0.0439	**0.7887**	0.7230
RF	$\lambda_{default}$	0.5000	0.0000	0.0000	0.0000	**1.0000**
	$\lambda_{fmeasure}$	0.0094	**0.7228**	**0.0387**	0.7487	0.6987
	λ_{gmean}	0.0094	**0.7228**	**0.0387**	0.7487	0.6987
	λ_{prior}	0.0080	0.7037	0.0321	**0.8332**	0.5944
XGB	$\lambda_{default}$	0.5000	0.1688	0.0547	0.0285	**0.9999**
	$\lambda_{fmeasure}$	0.0105	0.7816	**0.0594**	0.7562	0.8078
	λ_{gmean}	0.0099	0.7832	0.0575	0.7692	0.7975
	λ_{prior}	0.0080	**0.7840**	0.0506	**0.8157**	0.7536

groups based on the statistical difference of performance means, e.g. group *a* performs significantly better than group *b*. We report the average performance and HSD group for each λ, averaged across all learners and data sets.

4. Results and Analysis

Table 2 and Table 3 include the average threshold values and classification performance for the Part B and Part D data sets, respectively. The maximum score for each learner and metric is highlighted in bold font. We begin by discussing the average G-Mean, F-Measure, TPR, and TNR performance for each learner and threshold strategy. Next, we review the average threshold values selected by each thresholding strategy, and use their values to explain the differences observed in TPR and TNR performance. Finally, Table 4 reports the average scores across all learners and data sets along with Tukey's HSD test results.

For both data sets, the non-default thresholds consistently outperform the default threshold in terms of G-Mean, F-Measure, and TPR metrics. In all examples, the default threshold obtains near-perfect TNR scores, but this is only because the default threshold is assigning all test samples to the negative class, which in turn yields a near-zero TPR score. This TNR ≈ 1.0 and TPR ≈ 0.0 is a clear indication that the default threshold of 0.5 is too high of a threshold, and these results allow us to quickly conclude that the default classification threshold is not appropriate for making predictions with ensemble learners and highly-imbalanced data sets.

Beginning with the Part B data set, the optimized thresholds ($\lambda_{gmean}, \lambda_{fmeasure}$) produce G-Mean and F-Measure scores of 0.7228–0.7832 and 0.0387–0.0594, respectively, and they outperform λ_{prior} for 4 out of 5 classifiers. While the optimized thresholds outperform the prior threshold for the majority of the classifiers, λ_{prior} still performs competitively with G-Mean and F-Measure scores of 0.7037–0.7840 and 0.0321–0.0506, respectively. Also, the prior threshold λ_{prior} obtains the best G-Mean score (0.7840) on the Part B data set using the XGB learner, outperforming all other learners and thresholds according to the G-Mean. The Part D results follow a similar pattern, where ($\lambda_{gmean}, \lambda_{fmeasure}$) outperform λ_{prior} for three of five learners according to G-Mean and for all learners according to F-Measure. Again, the best G-Mean performance obtained on the Part D data set (0.6671) uses the XGB learner with λ_{prior}. In general, these G-Mean and F-Measure results show that the optimized thresholds perform best on average, and the prior threshold obtains the highest G-Mean score on both data sets using the XGB learner.

The TPR and TNR scores are used to measure the trade-offs in class-wise performance as the threshold changes. The prior threshold λ_{prior} obtains the highest TPR score for all learners on the Part B data set and the highest TPR score for four of five learners on the Part D data set. This is not necessarily the best results, however, as λ_{prior} also obtains the lowest TNR scores across all learners. While λ_{prior} obtains the lowest TNR scores, we would argue that the TNR scores are not unsatisfactory, and in some cases they are acceptable as we would prefer to maximize the TPR. On the Part B data set, for example, the XGB learner with λ_{prior} obtains an average TPR of 0.8157 and an average TNR of 0.7536, respectively. Averaged across all learners, λ_{gmean} and $\lambda_{fmeasure}$ obtain more balanced TPR and TNR scores. The best performance for this fraud application, however, is obtained using the XGB learner with λ_{prior}. Due to these contradictions, cross-validation must be used to select the best threshold for each application and learner.

We provide the average threshold values in Table 2 and Table 3 to demonstrate how small changes to λ can yield significant changes to classification performance. The optimized thresholds ($\lambda_{gmean}, \lambda_{fmeasure}$) fall within the ranges 0.0087-0.0105 and 0.0298–0.0321 for the Part B and Part D data sets, respectively. These threshold ranges are orders

TABLE 3. PART D THRESHOLDING RESULTS

Learner	threshold-type	Average Threshold	Average G-Mean	Average F-Measure	Average TPR	Average TNR
CatBoost	$\lambda_{default}$	0.5000	0.0290	0.0017	0.0009	**1.0000**
	$\lambda_{fmeasure}$	0.0302	0.6461	**0.0979**	0.6506	0.6416
	λ_{gmean}	0.0300	0.6461	0.0974	0.6568	0.6357
	λ_{prior}	0.0299	**0.6462**	0.0972	**0.6592**	0.6335
LightGBM	$\lambda_{default}$	0.5000	0.0385	0.0030	0.0015	**1.0000**
	$\lambda_{fmeasure}$	0.0315	**0.6572**	**0.1028**	0.6566	0.6577
	λ_{gmean}	0.0312	**0.6572**	0.1023	0.6615	0.6530
	λ_{prior}	0.0299	0.6565	0.0996	**0.6879**	0.6265
RF	$\lambda_{default}$	0.5000	0.0000	0.0000	0.0000	**1.0000**
	$\lambda_{fmeasure}$	0.0302	**0.6152**	**0.0862**	0.6333	0.5977
	λ_{gmean}	0.0302	**0.6152**	**0.0862**	0.6333	0.5977
	λ_{prior}	0.0299	0.6137	0.0849	**0.6520**	0.5778
XGB	$\lambda_{default}$	0.5000	0.0843	0.0140	0.0071	**0.9998**
	$\lambda_{fmeasure}$	0.0321	0.6662	**0.1090**	0.6486	0.6842
	λ_{gmean}	0.0318	0.6664	0.1085	0.6526	0.6805
	λ_{prior}	0.0299	**0.6671**	0.1047	**0.6869**	0.6479

TABLE 4. AVERAGE THRESHOLD PERFORMANCE AND HSD GROUPS

Threshold	G-Mean		F-Measure		TPR		TNR	
$\lambda_{default}$	0.0681	b	0.0215	b	0.0084	c	0.9999	a
$\lambda_{fmeasure}$	0.6989	a	0.0732	a	0.7005	b	0.6979	b
λ_{gmean}	0.6992	a	0.0728	a	0.7043	b	0.6944	b
λ_{prior}	0.6961	a	0.0692	a	0.7395	a	0.6566	c

of magnitude smaller than the default threshold of 0.5, and they are all approximately equal to the prior probability of the positive class within each distribution. Using the XGB learner and Part B data set as an example, we can see significant changes to the TPR and TNR scores when we compare λ_{gmean} to λ_{prior}. For λ_{gmean}, we obtain a TPR score of 0.7692 and a TNR score of 0.7975 using an average threshold value of 0.0099. When we decrease the threshold by 0.0019, for $\lambda_{prior} = 0.0080$, the TPR score increases to 0.8157 and the TNR score decreases to 0.7536. This small decrease to the threshold, which some may consider negligible, effectively increases the total number of correctly classified fraudulent providers by almost 5%. Perhaps even more important, the small decrease to λ leads to more than 200,000 non-fraudulent providers being misclassified as fraudulent. This phenomenon is observed across all learners in this study, and it emphasizes the importance of careful threshold tuning for imbalanced and highly-imbalanced big data problems.

Table 4 summarizes the thresholding results using the average performance across all learners and data sets. Also included are the HSD groups that indicate significantly different means. The default threshold consistently performs the worst, apart from the misleading high TNR score. According to the G-Mean and F-Measure, all three non-default thresholds belong to the same HSD group, and there is no significant difference between their results. The only significant difference is that λ_{prior} obtains a significantly higher TPR and lower TNR, when compared to $\lambda_{fmeasure}$ and λ_{gmean}. Based on these results, it is clear that non-default thresholds should be employed when using ensemble learners to classify highly-imbalanced Medicare data. Furthermore, results suggest that small changes to λ can yield significant changes to TPR and TNR. Therefore, we recommend validating thresholding strategies using a hold-out data set to ensure class-wise metrics meet application requirements.

5. Conclusion

This study compares four output thresholding techniques for improving the classification performance of big and highly-imbalanced data. The prior thresholding strategy sets the decision threshold equal to the prior probability of the positive class from each data set. The optimal-gmean and optimal-fmeasure thresholds optimize the threshold on training data using the G-Mean and F-Measure metrics, respectively. We compare these thresholding strategies to the default threshold using a popular family of tree-based ensemble learners and two Medicare fraud detection data sets that are characterized by big data and high class imbalance.

Contradicting results have shown that careful validation using TPR and TNR metrics is a critical step in the threshold selection process. For example, the prior threshold consistently performs the best in terms of TPR, but the optimal-gmean and optimal-fmeasure generally obtain more balanced TPR and TNR scores. Results have also shown that small changes to the decision threshold can yield significant changes to class performance, leading to hundreds of thousands of non-fraudulent providers being misclassified as fraudulent. Finally, results have shown that all three non-default thresholds significantly outperform the default threshold of 0.5 across all learners and data sets. In

future works, we will continue to explore these thresholding techniques across a range of imbalanced data domains and probabilistic classifiers.

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.

[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.

[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] T. M. Khoshgoftaar, C. Seiffert, J. V. Hulse, A. Napolitano, and A. Folleco, "Learning with limited minority class data," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, 2007, pp. 348–353.

[5] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 935–942.

[6] 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

[7] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 40–49, Jun. 2004.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622407.1622416

[9] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Structural, Syntactic, and Statistical Pattern Recognition*, A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 806–814.

[10] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887.

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

[12] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 4368–4374.

[13] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[14] Q. Zou, S. Xie, Z. Lin, M. Wu, and Y. Ju, "Finding the best classification threshold in imbalanced classification," *Big Data Research*, vol. 5, pp. 2–8, 2016, big data analytics and applications.

[15] X. Zhang, H. Gweon, and S. Provost, "Threshold moving approaches for addressing the class imbalance problem and their application to multi-label classification," in *2020 4th International Conference on Advances in Image Processing*, ser. ICAIP 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 72–77. [Online]. Available: https://doi.org/10.1145/3441250.3441274

[16] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249 – 259, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608018302107

[17] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949. [Online]. Available: http://www.jstor.org/stable/3001913

[18] Centers For Medicare & Medicaid Services. (2019) Medicare provider utilization and payment data. [Online]. Available: https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data

[19] R. A. Bauder and T. M. Khoshgoftaar, "A study on rare fraud predictions with big medicare claims fraud data," in *Intelligent Data Analysis*, vol. 24. IOS Press, 2020, pp. 141–161.

[20] J. M. Johnson and T. M. Khoshgoftaar, "The effects of data sampling with deep learning and highly imbalanced big data," *Information Systems Frontiers*, 2020. [Online]. Available: https://doi.org/10.1007/s10796-020-10022-7

[21] J. M. Johnson and T. M. Khoshgoftaar, "Thresholding strategies for deep learning with highly imbalanced big data," *Deep Learning Applications, Volume 2*, pp. 199–227, 2021. [Online]. Available: https://doi.org/10.1007/978-981-15-6759-9_9

[22] C. L. Calvert and T. M. Khoshgoftaar, "Threshold based optimization of performance metrics with severely imbalanced big security data," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 1328–1334.

[23] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*, 4th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.

[24] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[25] 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

[26] 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

[27] 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

[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 3149–3157.

[29] 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

[30] 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.