The Undersampling Effects on RANDSHUFF Oversampling Algorithms

Tora Fahrudin
School of Applied Science
Telkom University
Bandung, Indonesia
torafahrudin@telkomuniversity.ac.id

***Abstract*—Randshuff (Random Shuffle Oversampling Techniques for Qualitative Data) is one of an oversampling algorithm which appropriate for nominal attributes. Randshuff uses IVDM (Interpolated Value Difference Metric) distance calculation and crossover with random shuffle technique. Although Randshuff can overcome the problems on minority data, but the problems on majority data are ignored. The problem arises where majority data contain distribution complexity problems such as small disjuncts, overlap and noise. There are two kinds of undersampling concepts: informed undersampling and simple random undersampling. Tomeks links, Edited Nearest neighbors (ENN) and Near Miss are informed undersampling state of the art methods. Meanwhile, Random Undersampling (RUS) is simple random undersampling method. So, evaluations of both undersampling concepts on Randshuff are needed to be conducted. The experiments were evaluated on five public datasets. The results show that RUS as simple random undersampling and Near Miss as informed under sampling improve recall, f-measure and g-mean performance on Randshuff algorithm.**

Keywords—randshuff, tomeks, edited nearst neighbors, random undersampling, near miss

I. INTRODUCTION

Overcome imbalance problems is still interesting topic which challenging attention in the community [1]. That is because many imbalanced learning need to tackle the presence of underrepresented data and severe class distribution skewed in the real word dataset [2]. Community detection [3], analyzing the cancer severity [4], software defect prediction [5], student failure detection [6], medical decision [7] are some samples of research in many real world imbalance data.

Basic strategies to combat imbalance learning problems can be divided into data-level methods and algorithmic-level methods [8]. Oversampling, Undersampling and hybrid of both methods [9] are samples of data-level methods which concern to balance the class distribution by resampling techniques. Algorithmic-level tries to adapt or design new classifiers directly considering the class imbalance such as AECID [10], Imbalanced Data Set CSVM Classification Method [11], and etc. Most of researchers focus on data level solution because of the simplicity of the methods and the independence of classifiers [12].

Synthetic Minority Over-sampling Technique (SMOTE) is one of the well-known oversampling algorithms. SMOTE tries to add additional minority data with synthetic data generated by using interpolation methods [13]. Randshuff is one of SMOTE enhancement algorithms which focus on solving imbalance problems in the qualitative data domain [14]. Randshuff uses Interpolated Value Difference Metric (IVDM) for distance calculation and crossover with random shuffle techniques to generate synthetic value. Randshuff provides competitive performance compared to other "state of the art" imbalance algorithms [14].

Oversampling methods only focus on minority data and neglect complexity distribution on majority data. Meanwhile, undersampling is an efficient method for classifying imbalance data by using a subset of the majority class [15]. Combining oversampling with undersampling methods led to the best results for smaller data while simple random oversampling was competitive to other methods for datasets containing a relatively high number of the minority examples [16]. So, in this study, the performance of Randshuff combine with several undersampling methods (Tomeks links, Edited Nearest neighbors, and Near Miss as informed undersampling and Random Undersampling as simple random sampling) will be evaluated on five qualitative datasets.

The remainder of this paper is organized as follows: Section 2 explains the related work or literature about Randshuff, Tomeks links, Edited Nearest neighbors, Random Undersampling and Near Miss. Section 3 describes experimental setting. Section 4 shows experimental results. Finally, section 5 provides conclusion.

II. RELATED WORK

A. Oversampling

Oversampling is an method that increases the size of the minority class [17]. Some of oversampling methods such as SMOTE [13], ADASYN [18], SMOTE-Borderline [19], Randshuff [14] had been proposed to tackle imbalance problems. They make synthetic data to make a larger decision region. So overfitting condition caused by replicating only original minority data can be avoided.

Randshuff is one of SMOTE enhancement which concentrates on qualitative imbalance data. IVDM was used in Randshuff to compute neighbors distance. IVDM for qualitative data can be obtained by using formula [20]:

978-1-7281-5118-2/19/$31.00 ©2019 IEEE

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right| \quad (1)$$

For generating synthetic data, Randshuff uses cross over techniques which can be seen in Table 1.

TABLE I. CROSS OVER TECHNIQUES IN RANDSHUFF ALGORITHM

Features				
Col 1	***Col 2***	***Col 3***	***Label***	***Notes***
10-19	**left**	high	minority	original
20-29	right	**medium**	minority	original
20-29	left	medium	minority	synthetic

RANDSHUFF algorithm requires 3 important parameters: the number of neighbors (k), the choice of neighborhoods (Who_NearestNeighbors) and is keeping correlated attribute (IKA). RANDSHUFF provides competitive performance compared to other "state of the art" imbalance algorithms [14].

B. Undersampling

Undersampling is an method which create a subset of the original data-set by eliminating instances (usually majority class instances) [21]. Data to be eliminated can be selected in a random way or based on particular criteria such as, data which lying on the external regions of the input space [22]. Some undersampling methods such as Random undersampling [23], Tomek Links [24], Edited Nearest neighbors [25] and Near Miss [26].

Random undersampling tries to remove majority data randomly. Tomek links is two nearest examples which have different labels. The majority data which identified as tomek links are removed. Edited Nearest neighbors are majority data or minority data (e_i) which have different label against all three nearest neighbors. If (e_i) belongs to majority class, then (e_i) is removed, and if (e_i) belongs to minority class, then three nearest neighbors of (e_i) are removed. Near Miss method selects those majorities by using average distance to the three closest / farthest minority class based on near-miss version.

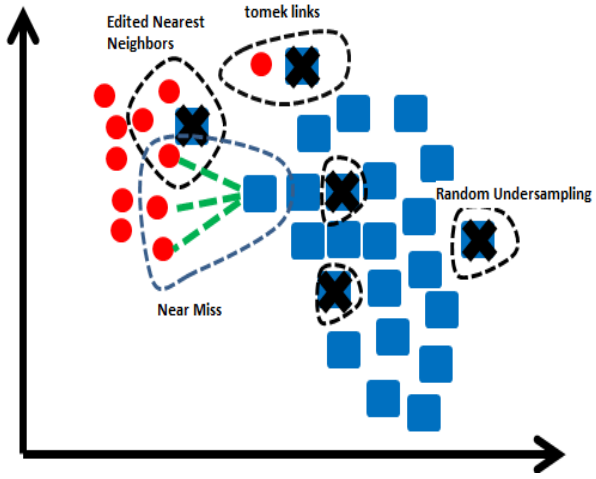

Fig. 1. Illustrations of tomeks, edited nearest neighbors, etc.

Figure 1 shows tomek links, edited nearest neighbors, near miss and random undersampling illustrations. Cross sign shows removed majority data.

III. EXPERIMENTAL SETTING

In this section, we discuss how to design experiments involve dataset description, parameter setting, testing scenario and assessment metric.

A. Data Set Description

Five qualitative datasets from keel data repository [27] which have different imbalance ratios ranging from 0.19 to 0.35 were used to evaluate the performance of the undersampling method on Randshuff algorithm. A brief summary of these five datasets is provided in Table 2 (R/I/N means Real/Integer/Nominal).

TABLE II. DESCRIPTION OF THE DATASETS

No	Datasets				
	Name	***Features (R/I/N)***	***Imbalance Ratios***	***Data***	***Fold***
1	Zoo-3	0/0/16	19.2	101	5
2	flare-F	(0/0/11)	23.79	1066	10
3	car-good	(0/0/6)	24.04	1728	10
4	car-vgood	(0/0/6)	25.58	1728	10
5	kr-vs-k-three_vs_eleven	(0/0/6)	35.23	2935	10

B. Parameter Setting

- We used k = 5 for number of neighbors parameter as a default value for Randshuff algorithm [14] and k-NN classifier algorithm.
- Who_NearestNeighbors was set to "Minor_Only" because it consistently shows better results on Recall, Precision and F-Measure than "Average_on_MajorMinor" [14].
- Parameter "is keeping correlated attribute (IKA)" was set to False.
- Balance ratio for oversampling with undersampling was set to 50 and 50. That means Randshuff oversampling will create 50% synthetic data from 100% majority data and then random undersampling was set to removes 50% majority data randomly. Other undersampling methods: Tomeks links, Edited Nearest neighbors and Near Miss will remove majority data according to their algorithm criteria.
- Balance ratio for oversampling without undersampling was set to 50:50. That means Randshuff oversampling will add minority data with synthetic data until the amount of original and synthetic data reach the same number to majority data.

C. Testing Scenarios

Figure 2 provides testing scenarios for this paper. Dataset is tested by two different scenarios: undersampling and without undersampling. Furthermore, oversampling by using Randshuff algorithm is conducted. Finally, balance dataset will be evaluated by using 5X 5/10 cross fold validation on j48 and Naïve Bayes. All classifiers are used in this paper is done by using weka [28].

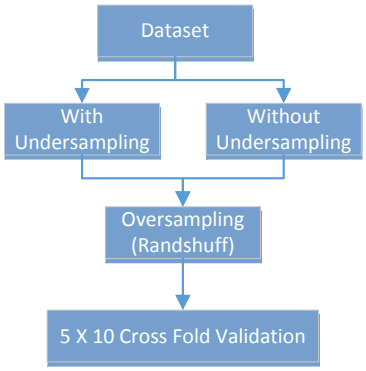

Fig. 2. Testing scenario.

D. Assessment Metric

For imbalanced datasets, the overall accuracy metric (TP+TN+FP+FN) can't be used as a measure of evaluation models in Imbalance problems. To assess the classifier performance, we need to count the number of TP, TN, FP, and FN which can be seen from confusion matrix on Table III. The Precision, Recall and F-Measure are appropriated parameter when we concern only to minority class [29]. Meanwhile, G-Mean and AUC are appropriated when we want to concern of both classes (majority and minority).

TABLE III. CONFUSION MATRIX

Actual	Predicted	
	(+)	(-)
(+)	TP	FN
(-)	FP	TN

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

$$F - Measure = \frac{(1+\beta^2).Recall.Precision}{(\beta^2.Precision+Recall)} \quad (4)$$

$$G - Mean = \sqrt{\frac{TP}{(TP+FN)}.\frac{TN}{(TN+FP)}} \quad (5)$$

$$AUC = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} 1_{p_i>p_j} \quad (6)$$

Where m data points will be iterated using variable i with true label = TP, and j runs over all n data points with true label = FN; p_i and p_j denote the probability score assigned by the classifier i-th and j-th data point respectively. 1 is the indicator function which gives outputs 1 if the condition is satisfied

IV. EXPERIMENTAL RESULTS

A. Without Undersampling

Table IV and Table V show performance of Randshuff Algorithm on j48 and Naïve Bayes. Those performance values would be used as based line for sub section B (With Undersampling) performance.

TABLE IV. RANDSHUFF WITHOUT UNDERSAMPLING ON J48

No	j48					
	Name	*Recall*	*Precision*	*F-M*	*G-Mean*	*AUC*
1	Zoo-3	0.6	0.37	0.43	0.58	0.78
2	flare-F	0.62	0.17	0.25	0.69	0.74
3	car-good	0.93	0.42	0.43	0.75	0.86
4	car-vgood	0.99	0.60	0.69	0.96	0.97
5	kr-vs-k-three_vs_eleven	1	0.78	0.83	0.99	0.99

TABLE V. RANDSHUFF WITHOUT UNDERSAMPLING ON NAÏVE BAYES

No	Naïve Bayes					
	Name	*Recall*	*Precision*	*F-M*	*G-Mean*	*AUC*
1	Zoo-3	0.6	0.23	0.32	0.57	0.91
2	flare-F	0.84	0.19	0.32	0.84	0.92
3	car-good	1	0.71	0.74	0.95	0.97
4	car-vgood	1	0.74	0.83	0.98	0.99
5	kr-vs-k-three_vs_eleven	0.94	0.87	0.87	0.96	1

B. With Undersampling

Table VI shows performance of each under sampling methods on five different datasets. Grey color for ENN and tomek means that those methods didn't work on particular data. All majority data in flare-F, car-vgood and kr-vs-k-three_vs_eleven did not comply with ENN removing criteria and also for tomek link removing criteria in kr-vs-k-three_vs_eleven.

TABLE VI. RANDSHUFF WITH UNDERSAMPLING ON J48

Data	Parameter	j48				NaiveBayes			
		RUS	*Tomek*	*ENN*	*NearMiss*	*RUS*	*Tomek*	*ENN*	*NearMiss*
Zoo-3	Recall	0.80	0.60	0.60	0.80	0.88	0.60	0.60	0.80
	Precision	0.11	0.37	0.37	0.05	0.11	0.18	0.21	0.05
	F-M	0.20	0.43	0.43	0.09	0.20	0.27	0.31	0.09

	G-Mean	0.64	0.58	0.86	0.86	0.69	0.56	0.57	0.35
	AUC	0.75	0.78	0.78	0.78	0.86	0.94	0.93	0.37
flare-F	Recall	0.82	0.63		0.66	0.82	0.85		0.85
	Precision	0.20	0.17		0.18	0.18	0.19		0.20
	F-M	0.31	0.27		0.27	0.30	0.31		0.31
	G-Mean	0.82	0.73		0.73	0.83	0.84		0.84
	AUC	0.85	0.78		0.77	0.92	0.92		0.92
car-good	Recall	0.91	0.95	0.94	0.95	1.00	1.00	1.00	1.00
	Precision	0.61	0.53	0.42	0.42	0.70	0.70	0.70	0.70
	F-M	0.67	0.54	0.44	0.44	0.75	0.74	0.74	0.74
	G-Mean	0.89	0.81	0.76	0.76	0.95	0.95	0.95	0.95
	AUC	0.93	0.88	0.87	0.88	0.98	0.97	0.97	0.97
car-vgood	Recall	1.00	0.99		1.00	1.00	1.00		1.00
	Precision	0.66	0.60		0.60	0.75	0.75		0.75
	F-M	0.74	0.69		0.69	0.83	0.83		0.83
	G-Mean	0.97	0.96		0.97	0.99	0.99		0.99
	AUC	0.98	0.97		0.97	1.00	0.99		0.99
kr-vs-k-three_vs_eleven	Recall	0.96			1.00	0.94			0.94
	Precision	0.74			0.78	0.84			0.87
	F-M	0.79			0.84	0.86			0.87
	G-Mean	0.97			0.99	0.96			0.96
	AUC	0.99			0.99	1.00			1.00

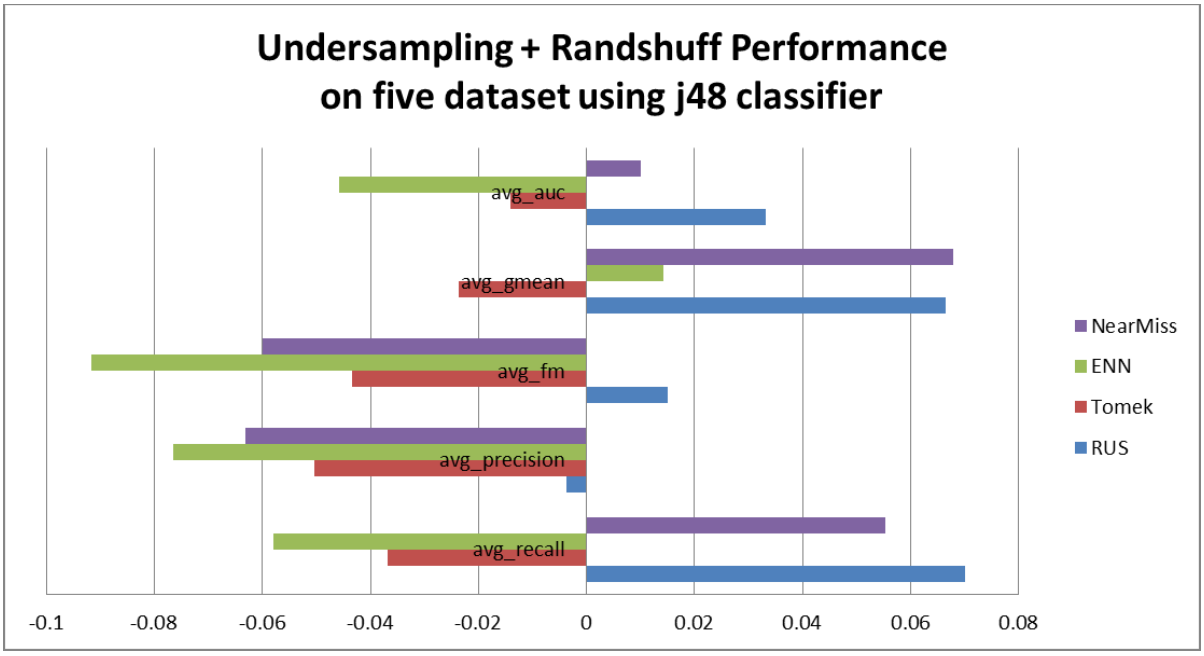

Fig. 3. Undersampling methods performance on five datasets using j48 classifier.

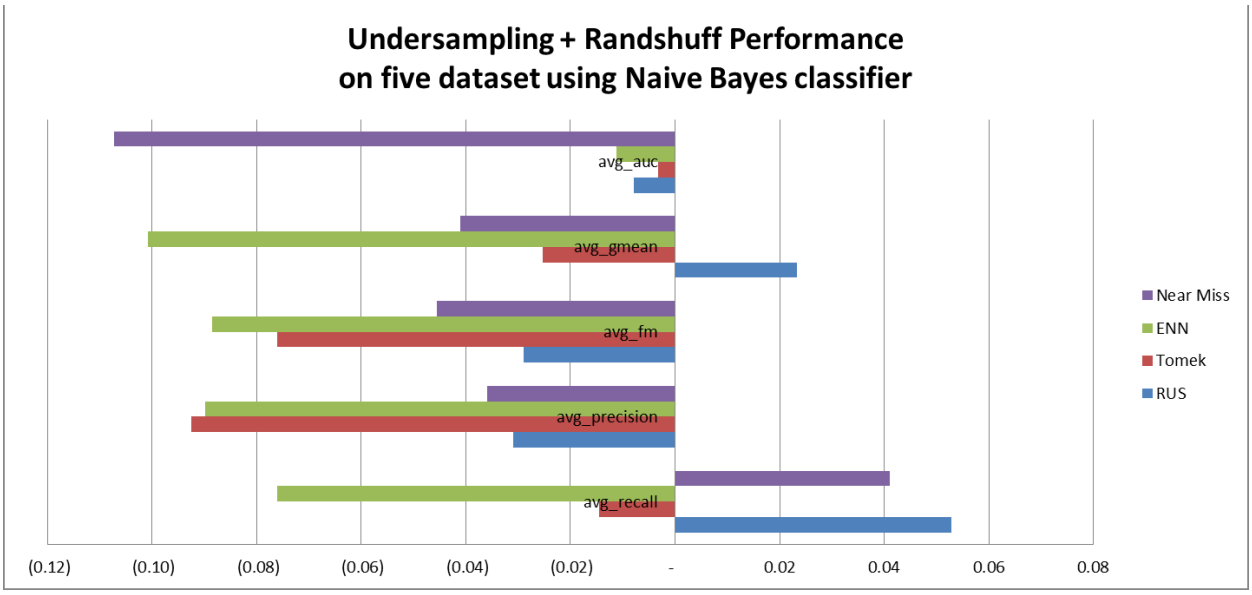

Fig. 4. Undersampling methods performance on five dataset using Naïve Bayes classifier.

Figure 3 shows the average values of five datasets performance for each undersampling methods by using j48 classifier. RUS in Randshuff gives better performance on average recall, f-measure, g-mean and auc values than Randshuff without undersampling for j48 and only gives better performance on average recall and g-mean for Naïve Bayes. Meanwhile, better performance also achieved by Near Miss for average recall, average g-mean and average auc.

Figure 4 shows the average values of five datasets performance for each undersampling methods by using Naïve Bayes classifier. The performance of undersampling methods by using Naïve Bayes classifier was decreased. Only RUS and Near miss achieved better performance on average recall and g-mean.

ENN shows the worst performance on average f-measure, average precision and average recall for j48 classifier. Near miss shows the worst performance on average auc.

V. Conclusion

Some undersampling methods combined with Randshuff algorithm may increase the performance on recall, f-measure and g-mean. RUS, near miss and ENN combined with Randshuff algorithm gives better performance than tomek link. All undersampling methods gives worse effect on precision performance.

References

[1] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, “Measuring the class-imbalance extent of multi-class problems,” Pattern Recognit. Lett., vol. 98, pp. 32–38, 2017.

[2] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.

[3] P. G. Sun, “Imbalance problem in community detection,” Phys. A Stat. Mech. its Appl., vol. 457, no. 2, pp. 364–376, 2016.

[4] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” Prog. Artif. Intell., vol. 5, no. 4, pp. 221–232, 2016.

[5] M. J. Siers and Z. Islam, “Software defect prediction using a cost sensitive decision forest and voting , and a potential solution to the class imbalance problem,” Inf. Syst., vol. 51, pp. 62–71, 2015.

[6] T. Fahrudin, J. L. Buliali, and C. Fatichah, “Predictive Modeling of the First Year Evaluation Based on Demographics Data : Case Study Students of Telkom University , Indonesia,” in 2016 International Conference on Data and Software Engineering (ICoDSE), 2016.

[7] X. Wan, J. Liu, W. K. Cheung, and T. Tong, “Learning to improve medical decision making from imbalanced data without a priori cost,” BMC Med. Inform. Decis. Mak., vol. 14, no. 1, p. 111, 2014.

[8] J. Stefanowski, Dealing with Data Difficulty Factors while Learning from Imbalanced Data, 2016th ed., vol. 605. Springer International Publishing Switzerland, 2016.

[9] M. Koziarski, B. Krawczyk, and M. Woźniak, “Radial-Based oversampling for noisy imbalanced data classification,” Neurocomputing, vol. 343, no. 2019, pp. 19–33, 2019.

[10] R. Guermazi, I. Chaabane, and M. Hammami, “AECID: Asymmetric entropy for classifying imbalanced data,” Inf. Sci. (Ny)., vol. 467, pp. 373–397, 2018.

[11] L. Peng, Y. Xiao-yang, B. Ting-ting, and H. Jiu-ling, “Imbalanced Data SVM Classification Method Based on Cluster Boundary Sampling and DT-KNN Pruning,” vol. 7, no. 2, pp. 61–68, 2014.

[12] T. Fahrudin, J. L. Buliali, and C. Fatichah, “Enhancing the Performance of SMOTE Algorithm by Using Attribute Weighting Scheme and New Selective Sampling Method for Imbalanced Data Set,” Int. J. Innov. Comput. Inf. Control, vol. 15, no. 2, p. -, 2018.

[13] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE : Synthetic Minority Over-sampling Technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[14] T. Fahrudin, J. L. Buliali, and C. Fatichah, “RANDSHUFF: An algorithm to handle imbalance class for qualitative data,” Int. Rev. Comput. Softw., vol. 11, no. 12, pp. 1093–1104, 2016.

[15] V. Ganganwar, “An overview of classification algorithms for imbalanced datasets,” Int. J. Emerg. Technol. Adv. Eng, vol. 2, no. 4, pp. 42–47, 2012.

[16] K. Napierala and J. Stefanowski, “Types of minority class examples and their influence on learning classifiers from imbalanced data,” J. Intell. Inf. Syst., vol. 46, no. 3, pp. 563–597, 2016.

[17] P. J. Huang, “Classification of Imbalanced Data Using Synthetic Over-Sampling Techniques,” University of California, 2015.

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” in Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, 2008, no. 3, pp. 1322–1328.

[19] H. Han, W. Wang, and B. Mao, “Borderline-SMOTE : A New Over-Sampling Method in,” in International Conference on Intelligent

Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I, Springer Berlin Heidelberg, 2005, pp. 878–887.

[20] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," J. Artif. Intell. Res., vol. 6, pp. 1–34, 1997.

[21] A. M. Mahmood, "Class Imbalance Learning in Data Mining – A Survey," Int. J. Commun. Technol. Soc. Netw. Serv., vol. 3, no. 2, pp. 17–36, 2015.

[22] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," Neurocomputing, vol. 135, pp. 32–41, 2014.

[23] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," SIGKDD Explor. Newsl., vol. 6, no. 1, pp. 20–29, 2004.

[24] I. Tomek, "Two Modifications of CNN," IEEE Trans. Syst. Man Cybern. 6, vol. 6, no. 11, pp. 769–772, 1976.

[25] D. L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," IEEE Trans. Syst. Man Cybern., vol. 2, no. 3, pp. 408–421, 1972.

[26] "kNN Approach to Unbalanced Data Distributions: A Case Study involving Information Extraction," in Proc. Int'l Conf. Machine Learning (ICML '2003), Workshop on Learning from Imbalanced Data Sets, 2003, no. c, pp. 2–6.

[27] J. A. F. Fernández, S. G. L. López, and A. F. Hilario, "Imbalanced data sets (Keel data set repository)," KEEL, 2018. [Online]. Available: https://sci2s.ugr.es/keel/imbalanced.php. [Accessed: 13-Aug-2019].

[28] "Weka 3: Data Mining Software in Java," Machine Learning Group at the University of Waikato, 2016. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 01-Sep-2017].

[29] M. Beckmann, N. F. F. Ebecken, B. S. L. Lima, and P. De Lima, "A KNN Undersampling Approach for Data Balancing," J. Intell. Learn. Syst. Appl., vol. 7, no. November, pp. 104–116, 2015.