Hierarchical Ensemble based Imbalance Classification

Jie Xie[1,2][0000−0002−7897−9584], Mingying Zhu[3,4][0000−0002−3423−074X], and Kai Hu[2][0000−0002−3521−4178]

[1] School of Computer and Electronic Information and the School of Artificial Intelligence, Nanjing Normal University, Nanjing, China
[2] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
[3] Nanjing University - School of Economics, 22 Hankou Road, Nanjing, Jiangsu, China, 210093
[4] Johns Hopkins University - Hopkins-Nanjing Center for Chinese and American Studies 162 Shanghai Road Nanjing University Nanjing 210093, Jiangsu, China 210093
zhumy@nju.edu.cn

Abstract. In this paper, we propose a hierarchical ensemble method for improved imbalance classification. Specifically, we perform the first-level ensemble based on bootstrap sampling with replacement to create an ensemble. Then, the second-level ensemble is generated based on two different weighting strategies, where the strategy having better performance is selected for the subsequent analysis. Next, the third-level ensemble is obtained via the combination of two methods for obtaining mean and covariance of multivariate Gaussian distribution, where the oversampling is then realized via the fitted multivariate Gaussian distribution. Here, different subsets are created by (1) the cluster that the current instance belongs to, and (2) the current instance and its k nearest minority neighbors. Furthermore, Euclidean distance-based sample optimization is developed for improved imbalance classification. Finally, late fusion based on majority voting is utilized to obtain final predictions. Experiment results on 15 KEEL datasets demonstrate the great effectiveness of our proposed method.

Keywords: Imbalance learning · bootstrap sampling · Gaussian-based oversampling · hierarchical ensembel · fraud detection.

1 Introduction

A class imbalance has been a practical problem of data mining and machine learning. The three cases that are specifically listed here are animal sound classification [1], fraud detection [2], and software defect prediction [3]. For the standard algorithm, the disadvantage of imbalanced problems is that the positive class (minority class) is easily misclassified. However, those positive samples

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

are often more important. Take fraud detection for instance, it is prevalent across various domains such as banking, government, and medical and public sectors. However, the number of positive samples are difficult to be obtained. Meanwhile, in a driving maneuver recognition system, lane change is often regarded as one of the most common causes of accidents [4]. However, for a standard driving dataset, there are fewer lane change events than other maneuvers. Therefore, it is necessary to investigate imbalance learning for solving real-world problems.

Numerous methods have been developed to treat imbalanced datasets, which can be divided into three categories: (1) data resampling [5, 6], (2) algorithm modification [7, 8, 9], and (3) ensemble methods [10, 11, 12]. Among those categories, ensemble methods are the important area that prove to improve the classification performance. Given a specific learning task, one single learner often cannot guarantee the result of the imbalance classification. Instead, a fusion including multiple classifiers can generate different outputs. Then, selected models are fused aiming to improve the final classification performance and avoid the choice of those worst classifiers. To create the ensemble, previous techniques such as data resampling and algorithm modification are used. Furthermore, the ensemble of data resampling achieves the state-of-the-art performance [19, 20, 3, 21].

In this study, we propose a hierarchical ensemble oversampling for improved imbalance classification. In particular, our contribution to this work can be summarized as follows: (1) A hierarchical ensemble is proposed for the imbalance classification, which includes bootstrap sampling for creating the first-level fusion, the second-level instance weighting-based fusion. (2) Bootstrap sampling with replacement is used to create the ensemble and avoid the worst classifiers for improving the performance. (3) Two different weighting strategies are compared and discussed. (4) k-means clustering is used to generate a select minority class subset for oversampling based on multivariate Gaussian distribution.

2 Related work

Various methods have been proposed in recent years. [22] proposed a novel evolutionary cluster-based oversampling ensemble. This framework combined a novel cluster-based synthetic data generation method with an evolutionary algorithm to create an ensemble. [3] used different oversampling techniques to build an ensemble classifier that can reduce the effect of low minority samples in the defective data. [23] proposed a three-way decision model by considering the differences in the cost of selecting key samples. Here, the ensemble was created by applying Constructive Covering Algorithm to divide the minority samples. [24] predicted default events by analyzing different ensemble classification methods that empowered the effects of the synthetic minority oversampling technique (SMOTE) used in the preprocessing of the imbalanced microcredit dataset. [25] investigated heterogeneous ensembles for imbalance learning. [21] applied an entropy-based method to minority samples to create an ensemble. Then, minority samples in various views were combined with the majority sample, where SMOTE is further

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

used. [26] integrated ensemble learning with the union of a margin-based undersampling and diversity-enhancing oversampling. Here the ensemble was created by applying bootstrap sampling to all samples. [27] proposed an imbalanced classification ensemble method, where a weighted bootstrap method was introduced to generated sub-datasets containing diverse local information.

3 The proposed method

Our proposed hierarchical ensemble-based imbalance classification system consists of four main steps: (1) bootstrap sampling with replacement, (2) minority instance selection and weighting, (3) oversampling based on multivariate Gaussian distribution, and (4) Euclidean distance-based sample selection

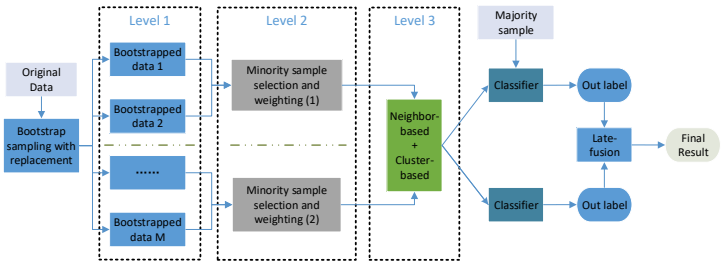


Fig. 1: Flowchart of our proposed work using hierarchical-ensemble based imbalance classification

3.1 Bootstrap sampling

Bootstrap sampling is a sampling method that uses random sampling with replacement. Previous studies have demonstrated that bootstrap sampling with data resampling is an excellent strategy in dealing with class imbalance issues [20, 26]. Specifically, bootstrap sampling with replacement ensures the independent training of each base classifier. Furthermore, bootstrap sampling can change the specific imbalance characters, which can thus reduce the variance of the loss functions of the base classifier. In this study, bootstrap sampling with replacement is used to create the ensemble (first-level fusion).

3.2 Instance selection and weighting strategy

For the oversampling, the selection of minority instances is important for the classification results. Previous studies used various features to select instance for

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

oversampling including classification error [28] and k-disagree neighbors (kDN) [29]. The kDN denotes the number of disagree class within k neighbors of a specific instance. For classification error, each instance of the train set is classified by Decision Tree (DT). The DT is run five times using 10-fold cross-validation to obtain the final classification error. Naturally, the sample carrying *more information* will be oversampled more times. However, the definition of *more information* is confusing based on the analysis of previous studies. [5] oversampled those borderline minority samples. In contrast, [29] selected those safe samples for oversampling. Both methods prove to be effective in oversampling. Here, we believe that *datasets with different characters work well using different sample selection and weighting strategies.*

After obtaining kDN and $error$ of sample X_i, they are first normalized to [0, 1]. Then two different weighting strategies are used to calculate sample hardness (Eq.1 and Eq.2). The first weighting strategy (Eq.1) assumes that those difficult minority samples carry more information, while the second weighting strategy (Eq.2) assumes that those safe minority samples carry more information. The selection of the weighting strategy is based on the average performance of each ensemble. The value of calculated hardness is between 0 and 1.

$$I_1(X_i) = \left(\frac{kDN(X_i)}{max(kDN(X_i))} + \frac{error(X_i)}{max(error(X_i))}\right) * 0.5 \tag{1}$$

$$I_2(X_i) = 1 - I_1(X_i) \tag{2}$$

Finally, the number of instances, which will be over-sampled for each minority instance is calculated as follows:

$$E(X_i) = \frac{e^{I(X_i)}}{\sum_{i=1}^{|N_i^{min}|} e^{I(X_i)}} (|N_i^{maj}| - |N_i^{min}|) \tag{3}$$

where $|N_i^{maj}|$ and $|N_i^{min}|$ denote the number of minority and majority class instances in K neighbor instances, respectively, $I(X_i)$ is $I_1(X_i)$ or $I_2(X_i)$.

When the number of majority instances of K neighbors for a minority instance is K, the minority instance will be regarded as noise. However, if all minority instances are regarded as noise, the number of instances, which will be over-sampled for each minority instance, is calculated as follows:

$$E(X_i) = \frac{1}{|N_i^{min}|} (|N_i^{maj}| - |N_i^{min}|) \tag{4}$$

In addition to minority class instance selection and weighting, we use the kDN value of the majority class instance for noisy instance removal. Here, those majority class instances with a kDN value of K will be removed for the subsequent analysis. The value of K is set to 5. To control the quality of generated instances, we first generate $\lambda * E(X_i)$ new instances, then select $E(X_i)$ instances whose Euclidean distances are the smallest. Here, the value of λ is set to 50.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

3.3 Multivariate Gaussian distribution based oversampling

After obtaining the hardness of minority samples, we use multivariate Gaussian distribution [30] for new instance generation since most real data follows Gaussian distribution. In this study, multivariate Gaussian distribution fits the minority class subset for instance oversampling.

Specifically, multivariate Gaussian distribution $N(\mu_i, \Sigma_i)$ first fits the selected subset of the current instance using two strategies, which are described as follows: In the first strategy, we combine both current and its closest instance for calculating μ_i and Σ_i. In the second strategy, we first apply k-means clustering to minority clusters. Then, μ_i and Σ_i are obtained from the cluster, which the current instance belongs to. After obtaining μ_i and Σ_i, the synthetic instances are generated using the following equation:

$$f_X(x_1, ..., x_k) = \frac{exp(-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu))}{\sqrt{(2\pi)^k |\Sigma|}} \tag{5}$$

For both strategies, the number of synthetic samples of each minority class instance is split as follows:

$$E(X_i) = \theta * E_1(X_i) + (1-\theta) * E_2(X_i) \tag{6}$$

where θ is used to balance the number of synthetic samples by the combination of current and closest samples $(1-\theta) * E_1(X_i)$, and the clusters $\theta * E_2(X_i)$. E_1 and E_2 denote the number of synthetic samples for neighbor- and cluster-based Gaussian oversampling, respectively. When θ is set to 1, it represents cluster-based Gaussian oversampling. In contrast, it stands for neighbor-based Gaussian oversampling when θ is set to 0. The setting of θ will be investigated in Experiment section.

For the Gaussian oversampling, we use the subsets obtained by GMM to calculate μ_i and Σ_i. After generating $\lambda * E(X_i)$ new instances, we further select $E(X_i)$ instances whose Euclidean distances are the smallest.

3.4 Hierarchical ensemble

In the first level, we create an ensemble $(D^M_{m=1})$ by applying bootstrap sampling with replacement to all samples. Then, we compare the averaged performance of two different weighting strategies, and the one with better performance in the second level is selected as follows.

$$\bar{D} = \begin{cases} avg(D^{M/2}_{m=1}) > avg(D^{M}_{m=M/2+1}) \\ avg(D^{M/2}_{m=1}) <= avg(D^{M}_{m=M/2+1}) \end{cases} \tag{7}$$

where $avg(\cdot)$ denotes the averaged performance of selected ensembles, and $\bar{D}$ is the selected ensembles for the subsequent analysis. Next, multivariate Gaussian distribution based oversampling is used to obtain synthetic instances, where the subsets are obtained by Gaussian Mixture Models. Finally, performance-based majority voting is used to make a final prediction for each test sample.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

4 Datasets and Experiment setting

In total, we selected 15 datasets from the KEEL data repository [32] for evaluating our proposed method. Here, those 15 datasets exhibited a considerable variety in sample number, feature number, and IR. Table 1 shows the detailed properties of the 15 selected KEEL datasets including the number of samples for both majority and minority classes, feature dimension, and imbalance ratio (IR). For all datasets, each method is evaluated with a 5-fold cross-validation procedure. For fairness and uniformity, we use an ensemble size of 40 and a Decision Tree (DT) as the base learning model in all experiments. All experiments are implemented using the Python library scikit-learn [33] with default parameters unless stated otherwise.

Table 1: Summary of datasets for evaluating and comparing the proposed method. Here, the data is sorted by the name, IR denotes imbalance ratio.

No.	Dataset	Samples	Majority samples	Minority samples	Features	IR
1	car-good	1728	1659	69	6	24.04
2	dermatology-6	358	338	20	34	16.90
3	ecoli-0-3-4_vs_5	200	180	20	7	9.00
4	glass0	214	144	70	9	2.06
5	haberman	306	225	81	3	2.78
6	iris0	150	100	50	4	2.00
7	kddcup-buffer_overflow_vs_back	2233	2203	30	41	73.43
8	kr-vs-k-zero-one_vs_draw	2901	2796	105	6	26.63
9	page-blocks-1-3_vs_4	472	444	28	10	15.86
10	pima	768	500	268	8	1.87
11	segment0	2308	1979	329	19	6.02
12	shuttle-2_vs_5	3316	3267	49	9	66.67
13	vehicle3	846	634	212	18	2.99
14	vowel0	988	898	90	13	9.98
15	yeast-0-2-5-7-9_vs_3-6-8	1004	905	99	8	9.14

To demonstrate the effectiveness of our proposed method, several ensemble-based techniques are included for the comparison: (1) undersampling-based ensemble: EasyEnsemble, RUSBoost, BalancedBagging, SelfPacedEnsemble. (2) oversampling-based ensemble: OverBoost, SMOTEBoost, OverBagging, and SMOTEBagging.

4.1 Evaluation metrics

For imbalance classification, accuracy often does not well reflect the overall performance of a proposed classification method. Usually, other metrics are adopted for better comparing different classification methods. In our study, G-mean and Area Under the Curve (AUC) are selected as the performance measures, since they are trade-off metrics between the correctly classified positive and negative instances. The definition of G-mean is

$$G - mean = \sqrt{Sens \cdot Spec} \tag{8}$$

where $Sens = \frac{TP}{TP+FN}$ and $Spec = \frac{TN}{TN+FP}$. TP (true positive) is the number of correctly classified positive instances. FN (false negative) is the number of

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

the positive instances that were misclassified as negative. FP (false positive) is the number of negative instances that were misclassified as positive. TN (true negative) is the number of correctly classified negative instances.

For AUC, it can be calculated as follows:

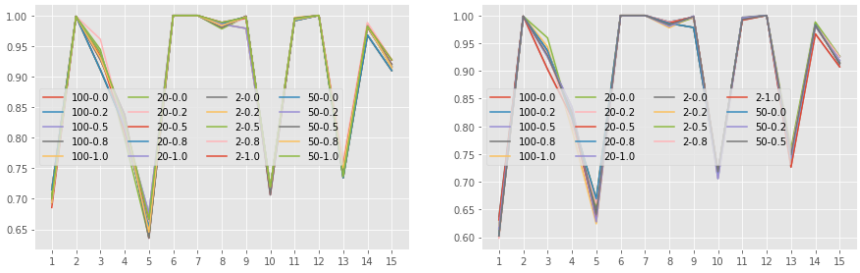
$$AUC = \frac{Sens + Spec}{2} \tag{9}$$

5 Experiments

In this section, we first investigate the effect of hyperparameters of our proposed method. Then, we will compare our proposed method with two types of ensemble-based methods: undersampling-based ensembles and oversampling-based ensembles.

5.1 The effect of hyperparameters

Tables 2 show the averaged AUC and G-mean over 15 datasets. Comparing classification results in terms of cluster size and θ, although the difference between different combinations of k (k-means clustering) and θ is not significant, the performance using a cluster size of 2 and θ of 0.8 is the best which will be selected for the subsequent analysis. Furthermore, we plot the performance for all datasets (Figure 2). We can observe that similar performance is obtained for all datasets over all combinations of k and θ.



(a) AUC (b) Gmean

Fig. 2: Classification AUC and G-mean of different cluster sizes and θ for 15 datasets. Here, x-axis denotes dataset index.

5.2 Our method VS. Undersampling-based ensembles

Tables 3 and 4 show the classification AUC and G-mean between our method and undersampling-based ensembles. The result highlighted in bold indicates better

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

Table 2: Averaged AUC and Gmean of different cluster sizes and θ.

		θ				
	cluster size	0	0.2	0.5	0.8	1
AUC	2	0.8933	0.8972	0.8945	**0.8992**	0.8964
	20	0.8933	0.8968	0.8950	0.8955	0.8978
	50	0.8933	0.8965	0.8951	0.8958	0.8985
	100	0.8933	0.8965	0.8951	0.8958	0.8984
Gmean	2	0.8850	0.8907	0.8871	**0.8922**	0.8895
	20	0.8850	0.8896	0.8882	0.8884	0.8910
	50	0.8850	0.8893	0.8883	0.8888	0.8917
	100	0.8850	0.8893	0.8883	0.8888	0.8915

AUC | G-mean of our method against all other undersampling-based ensemble methods in 9 | 9 datasets out of 15, followed by SelfPacedEnsemble with only 6 | 6 wins.

For BalancedBagging, the ensemble is created based on the undersampling technique using a negative binomial distribution. EasyEnsemble is to create an ensemble set by iteratively applying random under-sampling. Although the complexity of those methods is low, the performance of those methods is worse due to the information loss in the undersampling step. Furthermore, oversampling has been demonstrated to be better than undersampling in terms of the ROC curve [34]. Therefore, our proposed multivariate Gaussian distribution-based ensemble oversampling can obtain better performance.

Table 3: Comparison of AUC between Our method vs. Undersampling-based ensembles.

No.	**BalancedBagging**	**EasyEnsemble**	**RUSBoost**	**SelfPacedEnsemble**	**Ours**
1	0.8219	**0.8264**	0.7142	0.6974	0.6937
2	**0.9985**	0.9735	0.9706	**0.9985**	**0.9985**
3	0.9306	0.9083	0.8639	0.9000	**0.9611**
4	**0.8273**	0.8203	0.7692	0.8128	0.8094
5	0.6078	0.6128	0.5271	0.5867	**0.6629**
6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
7	0.9964	0.9995	0.9982	**1.0000**	**1.0000**
8	**0.9859**	0.9843	0.9846	0.9735	0.9823
9	0.9876	0.9899	0.9721	**0.9978**	0.9955
10	0.7237	0.7213	0.6654	**0.7239**	0.7130
11	0.9901	0.9896	0.9911	0.9939	**0.9942**
12	**1.0000**	0.9989	**1.0000**	**1.0000**	**1.0000**
13	0.7511	**0.7865**	0.6358	0.7558	0.7612
14	0.9605	0.9666	0.9805	0.9683	**0.9883**
15	0.9154	0.9052	0.8785	0.8987	**0.9276**
Rank	5	3	2	6	9

5.3 Our method VS. Oversampling-based ensembles

Tables 5 and 6 show the classification AUC and G-mean between our method and oversampling-based ensembles. The results highlighted in bold indicate better

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

Table 4: Comparison of G-mean between Our method vs. Undersampling-based ensembles.

No.	BalancedBagging	EasyEnsemble	RUSBoost	SelfPacedEnsemble	Ours
1	0.8041	**0.8080**	0.6247	0.4791	0.6022
2	**0.9985**	0.9717	0.9690	**0.9985**	**0.9985**
3	0.9282	0.9050	0.8374	0.8907	**0.9595**
4	**0.8260**	0.8194	0.7539	0.8115	0.8060
5	0.5941	0.6061	0.4495	0.5732	**0.6587**
6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
7	0.9963	0.9995	0.9982	**1.0000**	**1.0000**
8	**0.9858**	0.9841	0.9845	0.9731	0.9822
9	0.9875	0.9898	0.9712	**0.9977**	0.9955
10	0.7200	0.7178	0.6406	**0.7224**	0.7120
11	0.9901	0.9896	0.9911	0.9939	**0.9942**
12	**1.0000**	0.9989	**1.0000**	**1.0000**	**1.0000**
13	0.7478	**0.7837**	0.5681	0.7527	0.7596
14	0.9592	0.9659	0.9802	0.9661	**0.9882**
15	0.9135	0.9035	0.8702	0.8948	**0.9266**
Rank	5	3	2	6	9

AUC | G-mean of our method against all other regular ensemble methods in 11 | 11 datasets out of 15. For SMOTEBagging, it has 7 | 8 wins in terms of both AUC and G-mean. For OverBagging, it has 6 | 5 wins in terms of both AUC and G-mean.

Fig. 3 plots the comparison between our method VS. oversampling-based ensembles in terms of average performance and winning time. Our proposed method achieves the best performance. In terms of those datasets having higher IR than 20, our method hits 3 of 4 highest AUC and G-mean values respectively. For 9 datasets whose IR are less than 10, our method hits the 7 highest AUC and G-mean values. This result indicates that our proposed method can work well for both high and low IR.

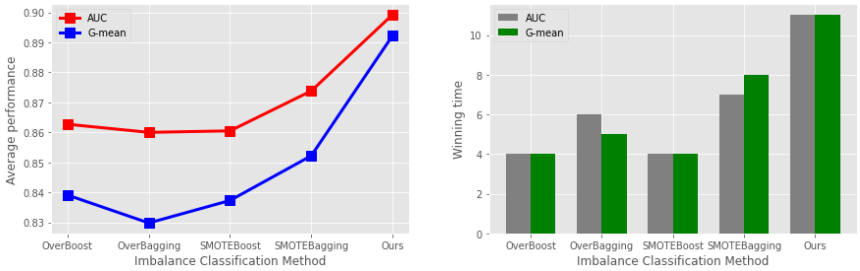
Table 5: Comparison of AUC between Our method VS. Oversampling-based ensembles.

No.	OverBoost	OverBagging	SMOTEBoost	SMOTEBagging	Ours
1	0.6355	**0.6064**	0.5760	0.6203	**0.6937**
2	**0.9985**	**0.9985**	**0.9985**	**0.9985**	**0.9985**
3	0.9139	0.8417	0.9056	0.8639	**0.9611**
4	**0.7842**	**0.8195**	0.7701	0.8195	0.8094
5	0.5820	0.5151	0.5593	0.5414	**0.6629**
6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
7	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8	**0.9647**	0.9691	0.9474	**0.9827**	0.9823
9	0.9811	**0.9978**	0.9811	**0.9978**	0.9955
10	0.6243	0.7024	0.6673	**0.7341**	0.7130
11	0.9889	0.9896	0.9924	0.9899	**0.9942**
12	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
13	0.6695	**0.6415**	0.6761	0.7122	**0.7612**
14	0.9289	0.9405	0.9567	0.9511	**0.9883**
15	0.8685	0.8782	0.8768	0.8965	**0.9276**
Rank	4	6	4	7	11

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

Table 6: Comparison of Gmean between Our method VS. Oversampling-based ensembles.

No.	OverBoost	OverBagging	SMOTEBoost	SMOTEBagging	Ours
1	0.4008	**0.3643**	0.3148	0.4272	**0.6022**
2	**0.9985**	**0.9985**	**0.9985**	**0.9985**	**0.9985**
3	0.9042	0.8216	0.8960	0.8365	**0.9595**
4	**0.7777**	**0.8147**	0.7586	**0.8165**	0.8060
5	0.5357	0.4199	0.5270	0.4760	**0.6587**
6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
7	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8	**0.9635**	0.9680	0.9458	**0.9825**	0.9822
9	0.9803	**0.9977**	0.9803	**0.9977**	0.9955
10	0.6080	0.6826	0.6619	**0.7257**	0.7120
11	0.9888	0.9896	0.9924	0.9898	**0.9942**
12	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
13	0.6427	**0.5812**	0.6596	0.6927	**0.7596**
14	0.9257	0.9377	0.9544	0.9487	**0.9882**
15	0.8600	0.8706	0.8705	0.8908	**0.9266**
Rank	4	5	4	8	11



(a) Average performance (b) Winning time

Fig. 3: Classification AUC and G-mean of different ensemble methods in terms of averaged result and winning time.

6 Conclusion and Future work

In this study, we propose an ensemble oversampling framework, which shows significant improvement over state-of-the-art algorithms using a total of 15 imbalance datasets. The framework consists of a novel hierarchical ensemble utilizing bootstrap sampling, instance selection and weighting, and multivariate Gaussian distribution-based oversampling. For multivariate Gaussian distribution-based oversampling, the parameters, μ, and Σ are calculated using (1) clusters, (2) current, and its k nearest minority neighbors. The final predict label is obtained by combining selected ensemble members to deliver relatively superior results.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

In future work, we are considering including an evolutionary algorithm for adaptively selecting ensemble members. In addition, we will extend the proposed approach to cover multi-class imbalanced datasets.

7 Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No: 61902154 and 72004092). This work is also partially supported by Natural Science Foundation of Jiangsu Province (Grant No: BK2019043526), Jiangsu Province Post Doctoral Fund (Grant No: 2020Z430), and China Postdoctoral Science special Foundation (Grant No. 2021T140281).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

Bibliography

[1] S. Kahl, T. Wilhelm-Stein, H. Hussein, H. Klinck, D. Kowerko, M. Ritter, M. Eibl, Large-scale bird sound classification using convolutional neural networks., in: CLEF (working notes), volume 1866, 2017.

[2] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, Q. Kang, Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection, Neurocomputing 407 (2020) 50–62.

[3] S. Huda, K. Liu, M. Abdelrazek, A. Ibrahim, S. Alyahya, H. Al-Dossari, S. Ahmad, An ensemble oversampling model for class imbalance problem in software defect prediction, IEEE access 6 (2018) 24184–24195.

[4] M. Shawky, Factors affecting lane change crashes, IATSS research 44 (2020) 155–161.

[5] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: International conference on intelligent computing, Springer, pp. 878–887.

[6] Q. Kang, X. Chen, S. Li, M. Zhou, A noise-filtered under-sampling scheme for imbalanced classification, IEEE transactions on cybernetics 47 (2016) 4263–4274.

[7] V. López, A. Fernández, J. G. Moreno-Torres, F. Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics, Expert Systems with Applications 39 (2012) 6585–6608.

[8] Y. Liu, H. Lu, K. Yan, H. Xia, C. An, Applying cost-sensitive extreme learning machine and dissimilarity integration to gene expression data classification, Computational intelligence and neuroscience 2016 (2016).

[9] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE transactions on neural networks and learning systems 29 (2017) 3573–3587.

[10] J. Li, S. Fong, R. K. Wong, V. W. Chu, Adaptive multi-objective swarm fusion for imbalanced data classification, Information Fusion 39 (2018) 1–24.

[11] R. Chen, S.-K. Guo, X.-Z. Wang, T.-L. Zhang, Fusion of multi-rsmote with fuzzy integral to classify bug reports with an imbalanced distribution, IEEE Transactions on Fuzzy Systems 27 (2019) 2406–2420.

[12] J. Yang, G. Xie, Y. Yang, An improved ensemble fusion autoencoder model for fault diagnosis from imbalanced and incomplete data, Control Engineering Practice 98 (2020) 104358.

[13] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2008) 539–550.

[14] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, IEEE Transactions on

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

Systems, Man, and Cybernetics-Part A: Systems and Humans 40 (2009) 185–197.
[15] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: 2009 IEEE symposium on computational intelligence and data mining, IEEE, pp. 324–331.
[16] C. Chen, A. Liaw, L. Breiman, Using random for-est to learn imbalanced data, Technical Report, 2004.
[17] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, Smoteboost: Improving prediction of the minority class in boosting, in: European conference on principles of data mining and knowledge discovery, Springer, pp. 107–119.
[18] R. Maclin, D. Opitz, An empirical evaluation of bagging and boosting, AAAI/IAAI 1997 (1997) 546–551.
[19] T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Comparing boosting and bagging techniques with noisy and imbalanced data, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 41 (2010) 552–568.
[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2011) 463–484.
[21] L. Dongdong, C. Ziqiu, W. Bolu, W. Zhe, Y. Hai, D. Wenli, Entropy-based hybrid sampling ensemble learning for imbalanced data, International Journal of Intelligent Systems 36 (2021) 3039–3067.
[22] P. Lim, C. K. Goh, K. C. Tan, Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning, IEEE transactions on cybernetics 47 (2016) 2850–2861.
[23] Y. T. Yan, Z. B. Wu, X. Q. Du, J. Chen, S. Zhao, Y. P. Zhang, A three-way decision ensemble method for imbalanced data oversampling, International Journal of Approximate Reasoning 107 (2019) 1–16.
[24] A. Gicić, A. Subasi, Credit scoring for a microcredit data set using the synthetic minority oversampling technique and ensemble classifiers, Expert Systems 36 (2019) e12363.
[25] H. G. Zefrehi, H. Altınçay, Imbalance learning using heterogeneous ensembles, Expert Systems with Applications 142 (2020) 113005.
[26] Z. Chen, J. Duan, L. Kang, G. Qiu, A hybrid data-level ensemble to enable learning from highly imbalanced dataset, Information Sciences 554 (2021) 157–176.
[27] B.-W. Yuan, Z.-L. Zhang, X.-G. Luo, Y. Yu, X.-H. Zou, X.-D. Zou, Oisrf: A novel overlap and imbalance sensitive random forest, Engineering Applications of Artificial Intelligence 104 (2021) 104355.
[28] H. Chongomweru, A. Kasem, A novel ensemble method for classification in imbalanced datasets using split balancing technique based on instance hardness (sbal_ih), Neural Computing and Applications (2021) 1–22.
[29] Y. Xie, M. Qiu, H. Zhang, L. Peng, Z. Chen, Gaussian distribution based oversampling for imbalanced data classification, IEEE Transactions on Knowledge and Data Engineering (2020).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14

[30] C. Biernacki, G. Celeux, G. Govaert, Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models, Computational Statistics & Data Analysis 41 (2003) 561–575.
[31] M. J. Crump, D. Navarro, J. Suzuki, Answering questions with data: Introductory statistics for psychology students, 2019.
[32] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework., Journal of Multiple-Valued Logic & Soft Computing 17 (2011).
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.
[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_14